A.Z
Machine Learning
Nov, 2023

lec 4 Model selection

Abstract

"Pluralitas non est ponenda sine neccesitate"

William of Ockham (1285–1347)

1 Stochastic scenario

So far we assumed that the labels are deterimistic function of the input, the stochastic scenario relaxs this assumption by assuming the output is a probabilistic function of the input.
The input and output is generated by a joint probability distribution $\mathcal{X} \times \mathcal{Y}$, this setup covers cases where the same input x can have different labels y
In the stochastic scenario, there may not always exista target concept f that has zero generalization error $R(f) = 0$
So in practive, we need to balance the complexity of the hypothesis and the empirical error carefully

Two general approaches to control the complexity:
- Selecting a hypothesis class, e.g. the maximum degree of polynomial to fit the regression model
- Regularization: penalizing the use of too many parameters, e.g. bybounding the norm of the weights (used in SVMs and neuralnetworks)
Some measures for complexity
- Number of distinct hypothesises
- VC-dim
- Rademacher complexity

2 Bayes error

Bayes error: a minimal non-zero error for any hypothesis in the stochastic scenario. It represents the inherent "noise" or uncertainty in the data distribution.

$$R^* = \inf_{\{h\}} R(h)$$

Bayes error serves us a a theoretical measure of best possible performance
Bayes classifier: A hypothesis with $R(h) = R^*$

The bayes classfier can be defined in terms of conditional probabilities as

$$h_{\text{Bayes}}(x) = \text{argmax}_{y \in \{0,1\}} \Pr(y \mid x)$$

This means for any input x, the Bayes classifier predicts the class y that maximizes the probability $\Pr(y \mid x)$

Definition (Noise): Given a distribution $\mathcal{D}$ over $X \times y$, the noise at point $x \in X$ is defined by

$$\text{noise}(x) = \min\{\Pr(1 \mid x), \Pr(0 \mid x)\}$$

Bayes error:

$$E(\text{noise}(x)) = R^*$$

The exceed error $R(h) - R^*$, which can be thought of as the additional error introduced by using a specific hypothesis h instead of an ideal, optimal classifier.

3 Bias and Variance

The excess error $R(h) - R^*$ is the difference between the error rate of a given hypothesis h and the Bayes error R^*, which is the lowest possible error can be achieved.
It can be decomposition as

$$R(h) - R^* = \epsilon_{\text{estimation}} + \epsilon_{\text{approximation}}$$

Estimation error (Variance) $\epsilon_{\text{estimation}} = R(h) - R(h^*)$, reflects how much worse our chosen hypothesis h performs compared to the best possible hypothesis in our current hypothesis class.
h^* is the best hypothesis in $\mathcal{H}$, meaning it has the lowest error rate $R(h^*)$ within that class.
Approximation error (Bias) $\epsilon_{\text{approximation}} = R(h^*) - R^*$, reflects the inherent limitations of the chosen hypothesis class. It is determined by how well the class of models we are considering can possibly represent the true underlying relationship in the data.

Even the best hypothesis h^* in $\mathcal{H}$ may not be as good as the theoretically optimal hypothesis (which would achieve the Bayes error R^*) because $\mathcal{H}$ may not include this optimal hypothesis.
Bias (approximation error) refers to the error due to simplistic assumptions in the learning algorithm. High bias can cause underfitting
Variance (estimation error) refers to the error due to too much complexity in the learning algorithm. High variance can cause overfitting

We can bound the estimation error $\epsilon_{\text{estimation}} = R(h) - R(h^*)$ by generalization bounds arising from the PAC theory
However, we cannot do the same for the approximation error since $\epsilon_{\text{approximation}} = R(h^*) - R^*$ remains unknown to us
In other words, we do not know how good the hypothesis class is for approximating the label distribution

3.1 Trade-off

As we increasing the complexity of the hypothesis chass
- decreases the approximation error as the class is more likely to contain a hypothesis with error close to the Bayes error
- increases the estimation error as finding the good hypothesis becomes more hard and the generalization bounds become looser(due to increasing $\log |\mathcal{H}|$ or the VC dimension)

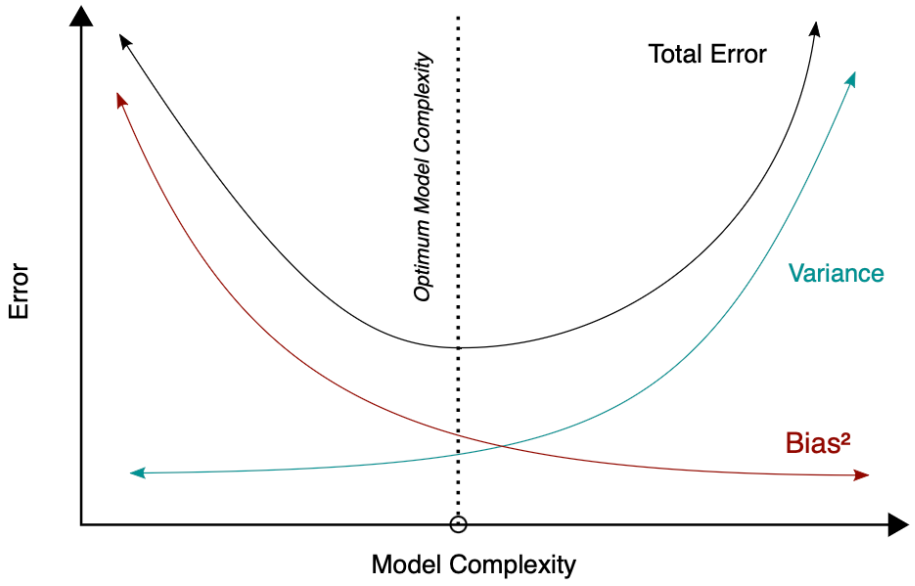

Figure 1: trade-off

3.2 Model selection

One strategy for model selection to initially choose a very complex hypothesis class with zero or very low empirical risk (error on the training data) $\mathcal{H}$.
Assume in addition the class can be decomposed into a union of increasingly complex hypothesis classes $\mathcal{H} = \bigcup_{\gamma \in \Gamma} \mathcal{H}_\gamma$, parametrized by γ, e.g.
- γ = number of variables in a boolean monomial
- γ = degree of a polynomial function
- γ = size of a neural network
- γ = regularization parameter penalizing large weights

We expect the approximation error to go down and the estimation error to up when γ increases
Model selection entails choosing γ^* that gives the best trade-off

4 Regularization-based algorithms

Regularization is technique that **penalizes large weights** in a model based on weighting input features
The classes as parametrized by the norm $\|\mathbf{w}\|$ of the weight vector bounded by $\gamma : \mathcal{H}_\gamma = \left\{\mathbf{x} \mapsto \mathbf{w}^T\mathbf{x} : \|\mathbf{w}\| \leq \gamma\right\}$

Typical norm:
L^2 norm (Also called Euclidean norm): $\|\mathbf{w}\|_2 = \sqrt{\sum_{j=1}^n w_j^2}$: used e.g. in support vector machines and ridge regression
L^1 norm (Also called Manhattan norm): $\|\mathbf{w}\|_1 = \sum_{j=1}^n |w_j|$: used e.g. in LASSO regression

> Recall: For a given sample S and a hypothesis class $\mathcal{H}_\gamma$, the empirical Rademacher complexity $\hat{\mathcal{R}}_S(\mathcal{H}_\gamma)$ is a measure of the class's ability to fit random labels assigned to the sample S.

Let $S \subset \{\mathbf{x}\|\mathbf{x}\| \leq r\}$ be a sample of size m and let $\mathcal{H}_\gamma = \left\{\mathbf{x} \mapsto \mathbf{w}^T\mathbf{x} : \|\mathbf{w}\|_2 \leq \gamma\right\}$.
For the L^2-norm case, there is a computational shortcut that allows us to bound the empirical Rademacher complexity analytically. The bound is given by:

$$\hat{\mathcal{R}}_S(\mathcal{H}_\gamma) \leq \sqrt{\frac{r^2\gamma^2}{m}} = \frac{r\gamma}{\sqrt{m}}$$

Here, r is the upper bound on the norm of the input vectors, γ is the upper bound on the norm of the weight vector, and m is the size of the sample.

Since r and m are constants for a fixed training set, the key variable affecting the Rademacher complexity in this scenario is γ.
In regularization-based algorithms, controlling the L^2-norm of the weight vector $\mathbf{w}$ (i.e., controlling γ) is a way to control the capacity of the model to fit to noise. This is crucial for preventing overfitting.
So a regularized learning problem is to minimize

$$\text{argmin}_{h\in\mathcal{H}}\, \hat{R}_S(h) + \lambda\Omega(h)$$

where
$\hat{R}_S(h)$ is the empirical error
$\Omega(h)$ is the regularization term which increases when the complexity of the hypothesis class increases

λ is a regularization parameter, which is usually set by cross-validation

For the linear functions $h : \mathbf{x} \mapsto \mathbf{w}^T\mathbf{x}$, usually $\Omega(h) = \|\mathbf{w}\|_2^2$ or $\Omega(h) = \|\mathbf{w}\|_1$

5 Model selection using a validation set

Split the Data into Training, Validation, and Test Sets
Validation Set : Used to tune hyperparameters and make decisions about which model or hyperparameters are working best. It acts as a proxy for the test set during the model development phase.
The larger the training set, the better the generalization error will be (e.g. by PAC theory)
The larger the validation set, the less variance there is in the test error estimate.

5.1 Grid search

Grid search is an technique frequently used to optimize hyperparameters, including those that define the complexity of the models
In its basic form it goes through all combinations of parameter values, given a set of candidate values for each parameter
For two parameters, taking of value combinations $(v, u) \in V \times U$, where V and U are the sets of values for the two parameters, defines a two-dimensional grid to be searched
As you can imagine the exhaustive search becomes computationally hard due to exponentially exploding search space

5.2 Stratification

Example:
Suppose you have a dataset with 1000 examples: 800 from Class A and 200 from Class B. A random split without stratification might end up with a training set having a very different proportion of Class A and B examples compared to the validation set.
With stratification, you would first split the examples from Class A into training and validation sets, and do the same for Class B, ensuring that the ratio of Class A to Class B is similar in both the training and validation sets.

5.3 Cross-validation

5.3.1 n-Fold Cross-Validation

In n-fold cross-validation, the dataset S is divided into n equal-sized parts, known as "folds". The idea is to use each fold as a validation set once, and the remaining $n - 1$ folds as the training set.

5.3.2 Leave-one-out cross-validation (LOO)

Extreme case of cross-validation is leave-one-out (LOO) : given a dataset of m examples, only one example is left out as the validation set and training uses the $m - 1$ examples.

5.3.3 Nested cross-validation

n-fold cross-validation gives us a well-founded way for model selection
However, only using a single test set may result in unwanted variation
Nested cross-validation solves this problem by using two cross-validation loops