Machine Learning

Ali Bakhshesh - 400222014

Spring 1403

Question 1:

Write the exact formulation of Bias-Variance tradeoff.

Answer:

proof:

$$MSE = E[(Y-\hat{f}(x))^2] = E[(f(x)+\epsilon-\hat{f}(x))^2] = E[f(x)^2+\hat{f}(x)^2-2f(x)2\hat{f}(x)+\epsilon^2-2\epsilon(f(x)-\hat{f}(x))]$$

$$= E[f(x)^2]+E[\hat{f}(x)^2]-E[2f(x)\hat{f}(x)]+E[\epsilon^2]-2E[\epsilon]E[f(x)-\hat{f}(x)], \quad E[\epsilon] = 0 \quad then$$

$$= E[f(x)^2] + E[\hat{f}(x)^2] - E[2f(x)\hat{f}(x)] + E[\epsilon^2]$$

$$= E[f(x)^2] + E[\hat{f}(x)]^2 - E[\hat{f}(x)]^2 + E[\hat{f}(x)^2] - E[2f(x)\hat{f}(x)] + E[\epsilon^2]$$

$$= f(x)^2 - E[\hat{f}(x)]^2 + E[\hat{f}(x)^2] + E[\hat{f}(x)^2] - 2f(x)E[\hat{f}(x)] + \sigma_\epsilon^2$$

$$= E[\hat{f}(x)^2] - E[\hat{f}(x)]^2 + (f(x) - E[\hat{f}(x)])^2 + \sigma_\epsilon^2$$

$$MSE = Variance + Bias^2 + \sigma_\epsilon^2$$

Question 2:

Under what conditions might logistic regression outperform linear discriminant analysis, and vice versa?

Answer:

1. When classes are well distinguished using logistic regression may be unstable.
2. When we have a few data and distribution in each class is normal then using LDA would perform better.
3. LDA will be better when there are more than two classes.
4. When the covariance matrices of the classes are significantly different Logistic regression would be better to use.
5. On imbalanced data, Logistic regression would perform better.

Question 3:

Discuss the impact of the curse of dimensionality on the K-Nearest Neighbor algorithm.

Answer:

The curse of dimensionality refers to the exponential growth of the size of the feature space as the number of dimensions increases. This phenomenon has several consequences for the K-Nearest Neighbor (KNN) algorithm:

- The curse of dimensionality significantly affects the computational complexity of KNN. As the number of dimensions increases, the number of data points required to adequately cover the space increases exponentially. As a result, processing each query point becomes computationally expensive, as it involves considering a larger number of neighbors.
- In high-dimensional spaces, data points are scattered. This scatter means that in regions without training samples, the nearest neighbors of a query point may not be representative of the underlying data distribution. This can lead to poorer performance as the algorithm tries to generalize effectively.
- In high-dimensional spaces, data points are scattered. This scatter means that in regions without training samples, the nearest neighbors of a query point may not be representative of the underlying data distribution. This can lead to poorer performance as the algorithm tries to generalize effectively.
- Traditional distance measures, such as Euclidean distance, become meaningless in high-dimensional spaces. Differences in distances between points are less pronounced, reducing the discriminating power of the distance measure used by KNN.
- The risk of overfitting increases with the number of dimensions. KNN is sensitive to noisy or irrelevant features, and in high-dimensional spaces, the presence of such features can lead to poor generalization performance.

Question 4:

Explain the concept of nested cross-validation and its benefits.

Answer:

The key idea behind nested cross-validation is to have two levels of cross-validation: an inner loop and an outer loop. The inner loop is used to set the metaparameters of the model, while the outer loop is used to estimate the generalization performance of the model. The key benefits of nested cross-validation are unbiased performance estimation, robustness of data availability,

and prevention of data leakage. Nested cross-validation is particularly useful in situations where the data set is limited, the model has many metaparameters to adjust, or the problem is particularly challenging. This is a standard technique in machine learning to obtain a reliable and unbiased estimate of a model's performance. Nested cross-validation (CV) is often used to train a model in which hyperparameters also need to be optimized. Nested CV estimates the generalization error of the underlying model and its (hyper)parameter search. Choosing the parameters that maximize nonnested CV biases the model to the dataset, yielding an overly-optimistic score. Model selection without nested CV uses the same data to tune model parameters and evaluate model performance. Information may thus "leak" into the model and overfit the data. The magnitude of this effect is primarily dependent on the size of the dataset and the stability of the model.

Question 5:

If we use a combination of the penalty term in Lasso and Ridge regression, what will happen in classification. Describe that. Also compare norm-one and norm-two and write about the intuition. If we replace Norm-p or norm-infinity instead of norm-1 or norm-2, what is the result?

Answer:

a. Combination of logistic and lasso regression is formulated as below:

$$loss = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda(\alpha\sum_{i=1}^{n}|\Theta_i| + (1-\alpha)\sum_{i=1}^{n}\Theta_i^2)$$

In logistic regression we can also add this penalty term to the loss function to perform feature selection and model regularization.

b. Norm-two regularization also known as ridge regression decreases RSS and shrinks the coefficients at the same time. Since ridge regression uses norm-two, it rarely omit features and usually all the features exist in the final model but norm-one regularization also known as lasso regression can completely omit some of the features.

c. Substituting a different penalty term, denoted by Lp , changes the properties and behavior of the resulting regularization technique. The Lp penalty term introduces a different type of constraint on the coefficients of a regression model that has distinct effects on variable selection, coefficient shrinkage, and model performance. If we substitute the Lp regularization (where p is some other value), the resulting normalization method emphasizes a different type of dispersion penalty than the Lasso absolute value penalty. Similarly, replacing the L2 (bump) regularization with the Lp regularization introduces a different form of quadratic penalty on the coefficients, with potentially different effects on multicollinearity and coefficient

shrinkage. L-infinity-norm normalization, also known as maximum-norm normalization, penalizes the largest absolute value of a vector element. This causes scarcity. That is, it encourages most elements of the vector to be zero. This can be useful in situations where you want to select a small subset of important features from a large set of potential features.

Question 6:

How can the bootstrap method be used to estimate the standard error of a sample mean?

Answer:

Assume we have an n-sample dataset called (Z). Then we choose n sample randomly with replacement and make another dataset called Z^1 and repeat this and make $Z^2, Z^3, \cdots, Z^\beta$. now for each Z we estimate α and call them $\alpha_1, \alpha_2, \cdots, \alpha_\beta$. This is the standard error estimation:

$$SE_\beta(\alpha) = \sqrt{\frac{1}{\beta - 1}\sum_{r=1}^{\beta}(\alpha_r - \frac{1}{\beta}\sum_{k=1}^{\beta}\alpha_k)^2}$$

Question 7:

Discuss how cross-validation can be used to assess the bias-variance trade-off of a model.

Answer:

We mentioned in Section 5.1.3 that k-fold CV with k¡nhas a computational advantage to LOOCV. But putting computational issues aside, a less obvious but potentially more important advantage of k-fold CV is that it often gives more accurate estimates of the test error rate than does LOOCV. This has to do with a bias-variance trade-off. It was mentioned in Section 5.1.1 that the validation set approach can lead to overestimates of the test error rate, since in this approach the training set used to fit the statistical learning method contains only half the observations of the entire data set. Using this logic, it is not hard to see that LOOCV will give approximately unbiased estimates of the test error, since each training set contains $n - 1$ observations, which is almost as many as the number of observations in the full data set. And performing k-fold CV for, say, k=5 or k=10 will lead to an intermediate level of bias, since each training set contains $(k - 1)n/k$ observations-fewer than in the LOOCV approach, but substantially more than in the validation set approach. Therefore, from the perspective of bias reduction, it is clear that LOOCV is to be preferred to k-fold CV. However, we know that bias is not the only source for concern in an estimating procedure; we must also consider the procedure's variance. It turns

out that LOOCVhashigher variance than does k-fold CV with k¡n.Why is this the case? When we perform LOOCV, we are in effect averaging the outputs of n fitted models, each of which is trained on an almost identical set of observations; therefore, these outputs are highly (positively) correlated with each other. In contrast, when we perform k-fold CV with k¡n, we are averaging the outputs of k fitted models that are somewhat less correlated with each other, since the overlap between the training sets in each model is smaller. Since the mean of many highly correlated quantities has higher variance than does the mean of many quantities that are not as highly correlated, the test error estimate resulting from LOOCV tends to have higher variance than does the test error estimate resulting from k-fold CV. To summarize, there is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically, given these considerations, one performs k-fold cross-validation using k=5 or k=10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance