Universidad Politécnica de Yucatán

Machine Learning

IRC9B

Unit 1

Professor:

Ortiz Victor

Student:

Brandon Isaac Pacheco Chan

Brandon Isaac Pacheco Chan ML.

Solution to most common problems in ML

Define the concepts of: Overfitting & Underfitting:

When we talk about the Machine Learning model, we talk about how well it performs and its accuracy which is known as prediction errors. Let us consider that we are designing a machine learning model. A model is said to be a good machine learning model if it generalizes any new input data from the problem domain in a proper way. This helps us to make predictions about future data, that the data model has never seen. Now, suppose we want to check how well our machine learning model learns and generalizes to the new data. For that, we have overfitting and underfitting, which are majorly responsible for the poor performances of the machine learning algorithms.

A statistical model or a machine learning algorithm is said to have underfitting when a model is too simple to capture data complexities. It represents the inability of the model to learn the training data effectively result in poor performance both on the training and testing data. In simple terms, an underfit model's are inaccurate, especially when applied to new, unseen examples. It mainly happens when we uses very simple model with overly simplified assumptions. To address underfitting problem of the model, we need to use more complex models, with enhanced feature representation, and less regularization.

Note: The underfitting model has High bias and low variance.

Reasons for Underfitting:

The model is too simple, So it may be not capable to represent the complexities in the data.

The input features which is used to train the model is not the adequate representations of underlying factors influencing the target variable.

The size of the training dataset used is not enough.

Excessive regularizations are used to prevent the overfitting, which constraint the model to capture the data well.

Features are not scaled.

Techniques to Reduce Underfitting:

Increase model complexity.

Increase the number of features, performing feature engineering.

Remove noise from the data.

Increase the number of epochs or increase the duration of training to get better results.

Brandon Isaac Pacheco Chan ML.

A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore, they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

Reasons for Overfitting:

High variance and low bias.

The model is too complex.

The size of the training data.

Techniques to Reduce Overfitting:

Increase training data.

Reduce model complexity.

Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).

Ridge Regularization and Lasso Regularization.

Use dropout for neural networks to tackle overfitting.

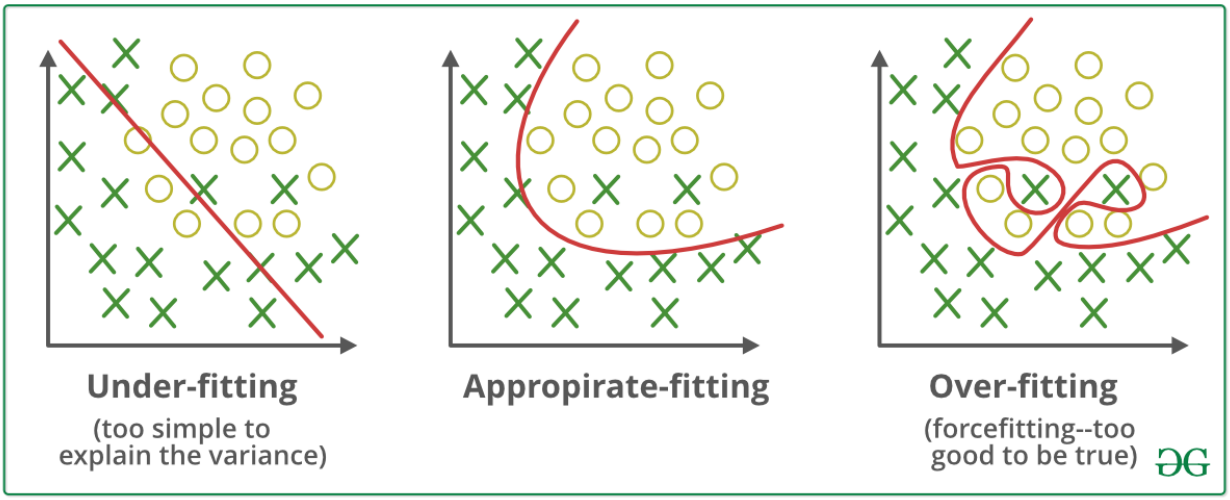

Fig 1. Under-fitting and Over-fitting

Brandon Isaac Pacheco Chan ML.

Define and distinguish the characteristics of outliers.

Outliers are those data points that are significantly different from the rest of the dataset. They are often abnormal observations that skew the data distribution, and arise due to inconsistent data entry, or erroneous observations.

In the machine learning pipeline, data cleaning and preprocessing is an important step as it helps you better understand the data. During this step, you deal with missing values, detect outliers, and more.

As outliers are very different values—abnormally low or abnormally high—their presence can often skew the results of statistical analyses on the dataset. This could lead to less effective and less useful models.

But dealing with outliers often requires domain expertise, and none of the outlier detection techniques should be applied without understanding the data distribution and the use case.

For example, in a dataset of house prices, if you find a few houses priced at around $1.5 million—much higher than the median house price, they're likely outliers. However, if the dataset contains a significantly large number of houses priced at $1 million and above—they may be indicative of an increasing trend in house prices. So, it would be incorrect to label them all as outliers. In this case, you need some knowledge of the real estate domain.

To ensure that the trained model generalizes well to the valid range of test inputs, it's important to detect and remove outliers.

If the data, or feature of interest is normally distributed, you may use standard deviation and z-score to label points that are farther than three standard deviations away from the mean as outliers.

If the data is not normally distributed, you can use the interquartile range or percentage methods to detect outliers.

Discuss the most common solutions for overfitting and underfitting and presence of outliers in datasets.

1. Overfitting:

Overfitting occurs when a model learns to perform exceptionally well on the training data but fails to generalize to new, unseen data. To address overfitting, you can consider the following solutions:

Regularization: Use techniques like L1 and L2 regularization to penalize large model weights, discouraging the model from fitting noise in the data.

Cross-Validation: Employ techniques like k-fold cross-validation to assess the model's performance on different subsets of the data. This helps in selecting models that generalize better.

Simplifying the Model: Reduce the complexity of your model by reducing the number of features, layers, or neurons. Use feature selection techniques to choose the most relevant features.

Data Augmentation: Increase the size of your training dataset by generating new examples through techniques like data augmentation (for image data) or synthetic data generation.

Ensemble Methods: Combine multiple models (e.g., Random Forests, Gradient Boosting) to reduce overfitting. Ensemble methods tend to generalize better than individual models.

Early Stopping: Monitor the model's performance on a validation set during training and stop when performance starts to degrade, indicating overfitting.

2. Underfitting:

Underfitting occurs when a model is too simple to capture the underlying patterns in the data. To address underfitting, you can consider the following solutions:

Increase Model Complexity: Add more layers, neurons, or features to your model to increase its capacity to learn from the data.

Feature Engineering: Create new features or transform existing ones to make the data more amenable to modeling.

Choose a More Complex Model: If you're using a simple model, consider switching to a more complex one, such as using a deep neural network instead of a linear model.

Hyperparameter Tuning: Experiment with different hyperparameters (e.g., learning rate, batch size) to find the settings that improve model performance.

3. Presence of Outliers:

Outliers are data points that deviate significantly from the majority of the data. They can negatively impact model performance. Here are some solutions to handle outliers:

Outlier Detection: Identify and remove or down-weight outliers in the dataset using statistical methods like Z-score, IQR (Interquartile Range), or visual inspection with scatter plots.

Data Transformation: Apply data transformations like log-transform or box-cox to make the data more robust to outliers.

Robust Models: Use models that are less sensitive to outliers, such as robust regression techniques like Huber Regression or robust versions of decision trees and random forests.

Imputation: For missing values that might be considered outliers, impute them using appropriate techniques rather than removing the entire data point.

Feature Engineering: Create new features that capture the presence of outliers or their effects, which can help the model learn to handle them.

Brandon Isaac Pacheco Chan ML.

Describe the dimensionality problem.

Is the problem caused by the exponential increase in volume associated with adding extra dimensions to Euclidean space.

The curse of dimensionality basically means that the error increases with the increase in the number of features. It refers to the fact that algorithms are harder to design in high dimensions and often have a running time exponential in the dimensions. A higher number of dimensions theoretically allow more information to be stored, but practically it rarely helps due to the higher possibility of noise and redundancy in the real-world data.

Gathering a huge number of data may lead to the dimensionality problem where highly noisy dimensions with fewer pieces of information and without significant benefit can be obtained due to the large data. The exploding nature of spatial volume is at the forefront is the reason for the curse of dimensionality.

The difficulty in analyzing high-dimensional data results from the conjunction of two effects.

High-dimensional spaces have geometrical properties which are counter-intuitive, and far from the properties which can be observed in two-or three-dimensional spaces.

Data analysis tools are most often designed to have in mind intuitive properties and examples in low-dimensional spaces and usually, data analysis tools are best illustrated in 2-or 3-dimensional spaces.

The difficulty is that those tools are also used when data are high-dimensional and more complex and hence, there is a probability to lose the intuition of the behavior of the tool and thus drawing incorrect conclusions.

Describe the dimensionality reduction process.

Before we give a clear definition of dimensionality reduction, we first need to understand dimensionality. If you have too many input variables, machine learning algorithm performance may degrade. Suppose you use rows and columns, like those commonly found on a spreadsheet, to represent your ML data. In that case, the columns become input variables (also called features) fed to a model predicting the target variable.

Additionally, we can treat the data columns as dimensions on an n-dimensional feature space, while the data rows are points located in the space. This process is known as interpreting a data set geometrically.

Unfortunately, if many dimensions reside in the feature space, that results in a large volume of space. Consequently, the points in the space and rows of data may represent only a tiny, non-representative sample. This imbalance can negatively affect machine learning algorithm performance. This condition is known as “the curse of dimensionality.” The bottom line, a data set with vast input features complicates the predictive modeling task, putting performance and accuracy at risk.

Here’s an example to help visualize the problem. Assume you walked in a straight line for 50 yards, and somewhere along that line, you dropped a quarter. You will probably find it fast. But now, let’s say your

search area covers a square 50 yards by 50 yards. Now your search will take days! But we're not done yet. Now, make that search area a cube that's 50 by 50 by 50 yards. You may want to say "goodbye" to that quarter! The more dimensions involved, the more complex and longer it is to search.

How do we lift the curse of dimensionality? By reducing the number of input features, thereby reducing the number of dimensions in the feature space. Hence, "dimensionality reduction."

To make a long story short, dimensionality reduction means reducing your feature set's dimension.

Explain the bias-variance trade-off.

The bias-variance trade-off is a fundamental concept in machine learning and statistical modeling. It refers to the balance between two types of errors that a predictive model can make: bias and variance. Understanding this trade-off is crucial for building models that generalize well to unseen data.

Bias:

Bias refers to the error introduced by approximating a real-world problem (which may be complex) with a simplified model. It is essentially the model's tendency to make assumptions about the data, resulting in systematic errors.

High bias models are overly simplistic and fail to capture the underlying patterns in the data. They tend to underfit, performing poorly on both the training and test data.

Examples of high bias models include linear regression on a non-linear dataset or a shallow decision tree on a complex dataset.

Variance:

Variance refers to the model's sensitivity to small fluctuations or noise in the training data. Models with high variance are complex and can capture noise in the data.

High variance models fit the training data very closely but generalize poorly to new, unseen data. They tend to overfit.

Examples of high variance models include deep neural networks with many layers and too many features for the amount of training data.

The bias-variance trade-off can be visualized as follows:

Low Bias, High Variance (Overfitting): Models that fit the training data closely but generalize poorly have low bias and high variance. They are highly flexible and can capture noise, leading to overfitting.

High Bias, Low Variance (Underfitting): Models that are too simple and make strong assumptions about the data have high bias and low variance. They are inflexible and fail to capture the underlying patterns, leading to underfitting.

Balanced Trade-Off: The goal in machine learning is to find a model that strikes a balance between bias and variance. This is where the model generalizes well to new data without overfitting or underfitting. Achieving this balance depends on factors like model complexity, the size of the training dataset, and the inherent noise in the data.

Practical considerations related to the bias-variance trade-off:

More Data: Increasing the size of the training dataset often helps reduce variance, allowing more complex models to generalize better.

Model Complexity: Adjusting the complexity of the model, such as reducing the number of features or layers, can reduce variance and prevent overfitting.

Regularization: Applying regularization techniques like L1 or L2 regularization can help control overfitting by adding a penalty to overly complex models.

Cross-Validation: Using techniques like cross-validation helps assess a model's performance on unseen data and can guide the selection of an appropriate trade-off.

References

Follow, D. (2017, noviembre 23). *ML*. GeeksforGeeks. https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

Bala, P. C. (2022, julio 5). *How to detect outliers in machine learning – 4 methods for outlier detection*. Freecodecamp.org. https://www.freecodecamp.org/news/how-to-detect-outliers-in-machine-learning/

Choudhury, A. (2019, mayo 22). *Curse of dimensionality and what beginners should do to overcome it*. Analytics India Magazine. https://analyticsindiamag.com/curse-of-dimensionality-and-what-beginners-should-do-to-overcome-it/

Simplilearn. (2021, junio 25). *What is Dimensionality Reduction? Overview, and Popular Techniques*. Simplilearn.com; Simplilearn. https://www.simplilearn.com/what-is-dimensionality-reduction-article