Machine Learning: Practice 1

Pavel Ghazaryan

March 12 2023

1 Experiment 1: Explain in your report which distance measure gives better performance and analyse the reason.

When cosine distance is utilised for nearest-neighbor calculation rather than Euclidean distance, the kNN classification algorithm performs more accurately. To be more specific, I got an accuracy rate of about 87 percent using Euclidean distance, but I got a rate of about 96 percent using Cosine distance. This is so because, in contrast to Euclidean distance, Cosine distance is less impacted by variations in the magnitude of the feature values. To put it another way, Cosine distance will result in a smaller number compared to Euclidean distance if two data points have comparable directions but different magnitudes. We determine the angle between the vectors that represent each article using the kNN algorithm and Cosine distance as the distance metric. This approach means that we are assessing the similarity between articles based on the direction of their vectors rather than their magnitudes.

2 Experiment 2: Analyse in your report the effect of k based on this experiment. What do you think is a reasonable value for k? Comment specifically on the bias and variance of your model at small and large values of k.

In general, it can be seen by comparing the two graphs that an increase in the value of k causes an increase in both training and testing errors. The training error is initially 0, which makes sense, and the test error is initially roughly 4%. As the training error is smaller than the test error and the model does not generalise well to new data, this suggests that the model is overfitting. Low bias and high variance may occur in this situation. On the other hand, when the value of k is high, both the training and testing errors are high, suggesting that the model is suffering from underfitting, which means that the model is too simple and does not capture the underlying patterns in the data. In such cases, high bias and low variance may be observed. To find a reasonable value of k, we need to identify the sweet spot where the model is neither overfitting nor underfitting, while minimizing the training error. Based on this analysis and the graphs, I believe that a reasonable value of k would be around 1 and 3 or 4, as these values correspond to the lowest training errors while maintaining reasonable testing errors.

3 Experiment 3: What classes do you think these 5 articles should belong to, based on your own judgement of their content?

Regarding the first article, I believe it should be classified as "interest" because it centers around a professional tennis player's personal experiences and thoughts about her mother's health and its impact on her tennis career. As for the second article, it's difficult to categorize it using the given categories, but if I had to choose one, I would say "earn" is the most appropriate because it discusses the concept of scoring goals, winning games, and earning points and victories. The third article is also challenging to classify, but I think "interest" would be the best fit as it focuses on the plans of Manchester United's manager to build a culture and develop players over the long term. Moving on to the fourth article, I believe it should also be classified as "interest" because it talks about an upcoming football game between the Philadelphia Eagles and the Kansas City Chiefs and highlights the skills of the players involved, as well as the interest in the match. Finally, for the last article, I think "earn" is the appropriate category because it discusses winning, earning records, and victories.

However, the classifications provided by the classifier - "crude" for article 1, "earn" for article 2, "trade" for article 3, "trade" for article 4, and "earn" for article 5 - are not accurate in my opinion. The categories given are very specific and related more to the economic and business spheres, and they do not accurately capture the content of the articles.

4 Comment on your classifier's performance in your report. What are the consequences of having no training data and limited training data for the 'sports' class?

As expected, the model has demonstrated strong performance on the prior four classes, achieving high levels of accuracy between 95-96%. However, the model has exhibited poor performance in classifying the newly added data, with an accuracy of only 33%. This highlights the crucial role that sufficient training data plays in the efficacy of the model. Notably, the model was only trained on a limited dataset of three articles from the 'sports' class, and was evaluated on just two additional examples. To improve the model's accuracy in this new class, it is recommended that additional training data be obtained and incorporated into the model's training process.

5 In your report, link these concepts to the experiments you've just performed. Is your model performing zero- or few-shot learning? Explain your reasoning.

Zero-shot learning is a type of machine learning in which a model is trained to recognize objects or concepts that it has never seen before. Few-shot learning, on the other hand, is a type of machine learning in which a model is trained to recognize objects or concepts with only a small amount of data. In the context of our scenario, the limited availability of only three articles in the "sports" class indicates that the model has been trained on a small dataset, therefore suggesting a few-shot learning approach.

6 Result analysis

In order to estimate the interval where its true error lies with 90% probability, I have utilized the following formula: $error \pm const * \sqrt{\frac{(error*(1-error))}{n}}$

In order to have 90% probability, *const* should be equal to 1.64 as suggested in the table in our lecture materials. In the formula n is the number of samples tested which in our case is equal to 480(as 320 articles were used to train the model). The variable error is the test error. Answer obtained: 0.035 ± 0.014

45-NN has higher testing error than 1-NN. To compute the probability that 45-NN also has a high true error I have used the following formulas: $z = \frac{d}{\sigma}$; $C = 1 - \frac{1-p}{2}$. Where d is the difference between errors and $\sigma = \sqrt{\frac{(error_1 \cdot (1-error_1))+(error_2 \cdot (1-error_2))}{n}}$. Finding the value for d, I was able to compute the value of p using the provided function Get_p_value and afterwards find the final probability C. I have obtained a value of around 98%.

7 Hyperparameter selection: Explain in the report your strategy for splitting the data, and the design of your chosen hyperparameter selection method. Why is it important to split the data into train, test, and validation sets in machine learning experiments?

In the present experiment, the dataset has been partitioned into two separate subsets, namely, the training set and the testing set, with a ratio of 80:20 respectively. Such a split is commonly used in machine learning to train a model on a subset of the data and test it on unseen data. Specifically, the training data is employed to fit the model and optimize its hyperparameters, while the testing data is utilized to evaluate the model's performance on previously unseen data. In order to conduct an effective training, I have chosen 160 articles for each class using the provided sample_indices function. This means that I will use 4 * 160 = 640 i.e. 80% of my dataset for training.

To further fine-tune the model, the hyperparameter k (number of neighbors) is selected using k-fold cross-validation. The range of values of k to be tested is defined as 1 to 50, and the model's performance is evaluated using 10-fold cross-validation for each value of k. The use of cross-validation is important as it provides a more reliable estimate of the model's performance than a single train-test split. Moreover, by evaluating the model's performance across multiple folds, it ensures that the model's generalization is not dependent on the particular partitioning of the dataset. I have chosen 10-fold cv as my dataset size is moderate and my research has shown me that values from 5-10 are common choice for such kind of dataset size.

The results of the cross-validation process, namely the accuracy scores for each fold, are stored in an array named cv_scores. Subsequently, the k value with the highest cross-validation score is selected as the optimal hyperparameter for the model, which is printed as best_k. In this specific case, the best hyperparameter value was determined to be 18, as it produces the highest average accuracy amongst all the other k-values across every fold. Evaluating my model on the testing dataset I have obtained accuracy of 98.75%.

It is important to emphasize that splitting the dataset into training, testing, and validation sets is a crucial step in machine learning experiments. It helps to avoid the problem of overfitting the model to the training data, which occurs when the model learns to memorize the training data instead of generalizing to new data. The validation set is used to fine-tune the model's hyperparameters, while the testing set is reserved for evaluating the model's performance on previously unseen data. This approach helps to ensure that the model generalizes well to new data and is not biased towards the training data. Without such a partitioning of the data, it would be difficult to know whether the model is overfitting or underfitting the data, which can lead to poor performance on new data.