Worksheet 4 – Similarity-based Learning

Theory

Review the videos from week 4. Answer the following questions based on those lectures

1. What is meant by an N-dimensional feature space?

Answer: Although we can’t graph dimensions higher than three, conceptually, an instance of N features can be represented as coordinates in an N-dimensional feature space.

1. What is a distance metric and what does it measure? Give an example.

Answer: As distance metric is some measure of the spatial distance between two points in space, in our case an N-dimensional feature space. We can use different distance metrics for model training depending on the problem domain in question.

1. How does the Nearest Neighbor algorithm work? What is there no training step?

Answer: We can make a prediction using Nearest Neighbor by considering an unlabelled feature we have not previously seen. To do this we calculate a distance metric between our unlabelled feature values and each of the features in our training data set. The class of the training set feature which is closest in distance to our unlabelled feature will be our class prediction by this algorithm.

1. How does the K Nearest Neighbor (kNN) variant of this algorithm work and how is it an improvement?

Answer: When making a prediction for an unlabelled feature, we find the k nearest labelled features using the similarity metric and use their labels to determine the predicted class using a majority vote. For this reason, it is typical to choose k to be an odd number such as three, five or seven neighbors. In extreme cases where each k prediction is the same, something which should not happen very often, then we can just choose the most frequently occurring class of the target feature.

1. What is the difference in the way we make predictions between categorical and continuous target features?

Answer: Instead of using the k nearest neighbors classes to determine a categorical variable, when can take an average value of the k nearest continuous target feature values. This represents a point in the feature space equidistant from the nearest neighbors representing an average value between them.

1. Why is it recommended to normalise continuous feature values before using kNN?

Answer: This is because distance metrics can be sensitive to very high or very low values on one axis with respect to another access. Normalisation eliminates these magnitude differences by forcing all of the feature ranges into a common, constrained range. This will improve the overall performance of the predictions on average.

1. The kNN algorithm performs poorly over large datasets. Explain one way we can improve its O(N) performance.

Answer: A K-dimensional tree (called a K-D Tree) can be used to used to improve the performance from linear in N to logarithmic in N where N is the length of the dataset.

1. Explain why the choice of distance metric can be important when using similarity-based learning.

Answer: That choice will be a trade-off between accuracy and computational complexity based on trial-and-error experimentation. When starting out with similarity-based modelling, it is common to choose a simple distance metric such as Euclidean distance or Manhattan distance, both of which are relatively easy to compute.

Practice

Follow the tutorial videos from week 4 and carry out the following steps

1. Download the code archive and extract the file from the week 4 learning materials. Make sure that you can run the examples code as provided.
2. Load the breast cancer data set from sklearn and run the model again. You will notice that it takes a relatively long time to classify and measure the accuracy. Extend this implementation by adding K-D-Tree optimization to your model search. Retest with the breast cancer data set and compare your new performance with the old implementation.

Answer: I wrote a KD-Tree implementation to check the result, it is OK and has no salient difference.



1. The model as provided in this week’s source code is a classifier. Modify the code to build a regressor, this is a model capable of predicting continuous target values. Test your implementation on the diabetes dataset from sklearn.

Answer: The accuracy\_score to judge the performance is mean square error loss. And based on the regressor, we compute the mse\_loss is approximately 3450.

1. Compare your regressor implementation with the sklearn built-in implementation.

Answer: If we use the built-in linearRegreesion regressor, we can get the result with the mean\_square\_error as 3608.