COMP30027 Machine Learning
2017 Semester 1
Project 1: K-Nearest Neighbour Classification
Report: Key Insights

Student Name : Ivan Ken Weng Chee
Student ID : 736901
Student Email : ichee@student.unimelb.edu.au

A. Similarity Metrics

The K Nearest Neighbours algorithm is simple, incremental, and does not learn anything about the model other than storing it. It makes predictions at runtime and needs a useful distance function to perform well. Although it can perform well, it is expensive to compute and may become infeasible for larger datasets.

I have chosen to use distance metrics such as Euclidean distance, Hamming distance, and Manhattan distance in my classifier. There are a few advantages and disadvantages of each metric. Euclidean distance works well since the input variables are similar in type, but can be influenced by unusual values. Manhattan distance is similar but faster than Euclidean distance and generally works better for high dimensional vectors. However, Manhattan and Euclidean measures are only valid for continuous variables. Hamming distance can be used when categorical variables are present, although it brings up the issue of standardization of numerical variables between 0 and 1 when the data set contains both numerical and categorical values.

Both Euclidean and Manhattan distances perform equally well with the dataset, as its dimensionality is both not too high and not too low. Hamming distance however, performs notably poorer compared to the prior two.

B. Validation Framework

In terms of evaluation strategies, I decided to go with Leave-One-Out using holdout, as there is no sampling bias in evaluating the system and results will be unique and repeatable. Disadvantages include the training being very expensive and a tradeoff between variance vs. bias in deciding on the train-test split ratio, as it is difficult to find a suitable value without many trials. Cross-Validation would be quicker to execute, and it measures the stability of the system across different training and test combinations, but the distribution of the data set may lead to bias and results might give lower accuracy values as less data is used for training.

The Curse of Dimensionality also affects the performance of the classifier. KNN works well with smaller inputs, but may struggle when inputs get larger. As the number of dimensions increases, the volume of input space increases exponentially as well. Similar points in high dimensions may have very large distances. Using feature selection to reduce the dimensionality will improve the performance of KNN.

C. Recommendations

a. Effective Representation

There are no missing values in the data set, which helps in data preprocessing. Almost the entire data set consists of float-valued data, however the column “Sex” is a three-class nominal feature, and it would be reasonable to discretise these values into 0, 1, and 2 as a way of accounting for this in measuring similarity.

I have chosen Accuracy and Precision as means to evaluate the classifier's performance. Accuracy is the proportion of instances for which the classifier correctly predicts the class of abalone. Precision on the other hand is the classifiers performance relative to a single class, in this case "young" or "old" abalones.

I do not see any significant difference in results when using different voting methods such as majority class vs. inverse distance with epsilon = 0.5, but this could be due to my implementation. Accuracy and precision both average around 70%.

b. Good Similarity Metric

Both Euclidean distance and Manhattan distance perform equally well for this dataset, but I would recommend Manhattan over Euclidean for its additional runtime speed due to not using a square root function.

c. Good Value of "k"

It is reasonable to select k as an odd value, to avoid breaking ties when the number of nearest neighbours are equal. In general, a large k value is more precise as it reduces the overall noise. Smaller values of k tend to lead to overfitting, hence lower classifier performance due to noise. Larger values of k on the other hand tend to drive classifier performance towards that of Zero-R. My results generally shows a higher score for when k = 7 as compared to 3 and 11, for all distance measures, so my recommendation would be 7.

Results

Voting Strategy	Distance Measure	k	Classifier Accuracy (%)	Classifier Precision (%)
Majority Class	Euclidean Distance	3	75.277	74.526
		7	77.949	76.496
		11	77.474	77.852
	Manhattan Distance	3	75.537	74.476
		7	78.282	75.144
		11	77.127	76.709
	Hamming Distance	3	67.510	68.093
		7	72.195	69.209
		11	71.647	68.485
Inverse Distance	Euclidean Distance	3	75.018	74.184
		7	78.332	76.496
		11	77.989	76.638
	Manhattan Distance	3	75.069	74.088
		7	77.243	75.983
		11	77.043	76.640
	Hamming Distance	3	68.637	67.970
		7	69.737	70.042
		11	70.464	68.319