Lab 7 Occupancy Detection Algorithm Comparison

ETGG 4803\_02 Artificial Intelligence

By. Thomas Gilman

Shawnee State University

May 6th, 2020

Given the five classification algorithms to compare, Simple Statistical Classifier(SSC), The Naïve Bayesian Classifier (BC), The Nearest-Neighbour Classifier (NN), The k-Nearest-Neighbour Classifier (kNN), and The Instance-Based Algorithm (IB1) along with some of their sub methods; all methods required an initial setup of reading in and organizing the training data. The initial setup is generally the same for each classification, involving reading in the training data, classifying, seperating, and organizing the data, and retrieving statistical information from the training data. The initial difference between each classification algorithm, lies with the initial statistical data used to then classify any incoming test data. Depending on how much precomputation needs to be done, and the complexity of the classification, can both affect the overall time and accuracy of the classification. Sometimes it’s better to perform more computations and gather more statistical information before hand; while it may prove to cost a little more time in the beginning, it makes accessing that information later trivial, improves speed in the long run, and improves performance of algorithms requiring more statistical information.

In the case of SSC, the initial overhead cost to setup involves reading in all the training data, splitting the training data up into classifications, adding every attribute up individually in both classifications, then dividing those individual classified values by the number of training instances in that classification to get that classifications mean values. The mean values for both classifications are then used to compute the distance any incoming test data is from that classification, tallying up the likelihood for one or the other classification, depending on the distance the tests’ attributes are from either of the means. Depending on which classification is the closest, is what the test will end up being classified as. SSC is a rather simple classification, and because of its simplicity it sacrifices accuracy for time, as it classifies tests very quickly.

The BC classification is the most complex out of all the other classifications. In this case the majority of the work is done prior to classifying any test data. BC requires the use of an interval map, which is a table containing a range an attribute value could fall into as well as the true percentage classifications in that interval. In order to build this table, you must first read in all the training data, sort the data so that there is no overlap, and gather the minimum attribute values, maximum attribute values, and also calculate the range of the attribute values. After sorting the data, you can then build the interval map; this is done by separating each attribute into interval sections from a minimum value ranging to a high value in its designated interval space. The percentage of classifications, in this case being occupied and unoccupied, are then determined given the interval range. Once the interval map has been created, every test can then use this map to determine the interval range that each of its attributes falls into. Each attribute of the test data will then have its probability of either of the two classifications calculated using the data from the interval table along with the bayesian theorem. To calculate the probability of a test being in either of the classifications; for each attribute that falls into an interval range, the value is then normalized using the min values, and ranges precalculated, and used alongside the normalized low range of the interval and the true percentage. This same process is done for the high range value of the interval and multiplied together with all the other attribute probabilities to give the probability of being a part of the particular classification calculated for. The same process is also done for the opposite classification using all except the same true percentage for the Bayesian Theorem formula. The two probabilities are then compared, and whichever classifications probability is higher, is what the test will be classified as. While this is a lot of overhead, and understanding the process can be rather confusing at first; the process is also rather simplistic. Because of the simplicity of the formula, and the pre computations done prior to classification, the test is decently accurate, and fast.

The NN classification and kNN classification are exactly the same in setup, but differ in the number data points used to determine the classification of a test. Both classifications use a distance formula to determine which classification a test belongs to. The benefits to using NN or kNN is that there is no overhead to setting up the classification other than reading in and setting the training data to be used to classify the incoming test data. The only difference between the two classifications is that kNN can use k amount of closest data points to then vote on the classification of the data entry. This means as the test is being performed, once a list has k entries, you must then determine for every new node entry being tested if it is closer than one of the entries in the list, and if so, insert the data point and move the further entries back in the list, inevitably pushing out the furthest data point. Taking multiple samples improves the accuracy of the function overall, as it tallies the classifications between the k closest data entries and picks the classification with the highest tally. The downside compared to NN is that it is slightly slower due to maintaining the closest entries list rather than picking the singler closest entry. The other downside is that having too large of a sample size can decrease accuracy depending on the range training samples, or it can increase the accuracy; it all depends on the training data and the variance in data along with the way the distance is calculated. The distance formula chosen for both NN and kNN also affects the accuracy, as using a formula such as the one-norm or infinity-norm is faster than finding the euclidean distance(two-norm), they are also less accurate. While the one-norm is a less accurate of the two-norm; the infinity-norm simply takes the largest attribute distance as the overall distance between the training data and the test data. This makes it less accurate, and could even mean a larger likelihood of false classifications if the data varied dramatically and chaotically given a different training set. Overall the downside to using NN and kNN is that every time a test point is being classified, it has to be compared with every training data entry leaving the formula to be very close to being On^2, which can be seen in the time it takes to complete depending on the number of tests.

Lastly IB1 is exactly the same as NN using the euclidean distance, except with added complexity. For IB1 all the training data and test data coming in, must be normalized, which means retrieving the min attribute values, max attribute values, and calculating the range. In order to reduce some of the time taken, it is better to precompute the normals for all of the training data to reduce time later, and only have to normalize incoming test data. The other thing IB1 does differently, when compared to the euclidean distance formula, is that it takes in account of missing data entries. Any missing data entry will either be set to the maximum distance of the other existing data entry in the comparison, or will be set to a maximum distance of one if both entries are missing in the comparison. Lastly IB1 takes the negative square root, rather than the positive one, but still picks the value closest to zero. In the case of the training data given and the test points given, the outputs for both NN two-norm and IB1 are the same.

Given the different algorithms used to classify the incoming data, The Naive Bayesian Classification seems to be the best approach to classifying any incoming test data entry. While there is more overhead in terms of setup compared to the other methods, it proved to be both decently accurate with a 90% correct classification rate, and quick being the second quickest algorithm in terms of classification speed after setup, with SSC being the first.