Article

Not peer-reviewed version

Adding Link Points to Improve Nearest Neighbour Classification

Kieran Greer *

Posted Date: 21 October 2025

doi: 10.20944/preprints202510.1657.v1

Keywords: Nearest Neighbour; new data points

 doi:10.20944/preprints202510.1657.v1

Article

Adding Link Points to Improve Nearest Neighbour Classification

Kieran Greer

Distributed Computing Systems, Belfast, UK; kgreer@distributedcomputingsystems.co.uk
http://distributedcomputingsystems.co.uk
Version 1.0

Abstract

One solution to nearest neighbour outliers is to add new points between the outlier and the main cluster, so that the new points are now closer to the outlier point. This idea is not very prevalent, maybe because of a fear that it will result in unbalanced data, overfitting, or noise. In the age of Generative AI however, new data points are now legitimate and if they are added equally, or only when needed, they might even help to re-balance the dataset. This paper describes a nearest neighbour algorithm called Step Nearest Neighbour (Step NN). It is so-called because points that are incorrectly identified in a supervised setting, can be linked-to, by adding new points between them and the main cluster, where the setting is for a single nearest neighbour only. The step points can be also noted, so that a system which retrieves similar cases can navigate around them, for example. Other adaptive versions may have to fine-tune parameters, while this method is very basic and works automatically. Test results show that the new algorithm outperforms a version of k-NN.

Keywords: nearest neighbour; new data points

1. Introduction

Classification algorithms can be used in many fields involving pattern recognition or data processing. The k-nearest neighbours algorithm (k-NN) [4] is one of the first to be invented and is used in recommendation systems, finance, healthcare and Internet of Things (IoT), to name a few. One problem with nearest neighbour algorithms is how to deal with outliers, which are points of one category that lie closer to other category points. One solution would be to add a new point between the outlier and its main cluster, so that the new point is now closer to the outlier. This idea is not very prevalent, maybe because of a fear that it will result in unbalanced data, overfitting, or noise. In the age of Generative AI [5] however, new data points are now legitimate and if they are added equally, or only when needed, they might even help to re-balance the dataset. This paper describes a new nearest neighbour algorithm called Step Nearest Neighbour (Step NN). It is so-called because points that are incorrectly identified in a supervised setting, can be linked-to by adding new points between them and the main cluster, which will then lie closer to the point and classify it correctly. These new points can be noted however, so that a system which retrieves similar cases could still navigate from them to the original data. Another idea would be to physically move the erroneous points, apart by a small amount. This would, however, permanently delete the original position from the dataset and small increments would require more iterations, because the distance in the opposite direction is more difficult to measure. Moving or adding from the main cluster to the outlier can be done in a single operation, for example. Because the new points may also need to be resolved, the error reduces over multiple runs, using the updated dataset each time. Basically, any row that lies inside a category boundary is valid for that category. The question is therefore, where to place a new point so that it best links with the outlier. A biological solution would maybe half the distance each time and while a computer program can measure the distance more accurately, this still seems reasonable. For one thing, it removes the bias in either direction. The changed dataset therefore contains the original data

 doi:10.20944/preprints202510.1657.v1

plus new data points and it needs to be kept and used for evaluating new datasets. It should be a simple matter to add more points as well. The algorithm is currently expected to work with only 1 nearest neighbour.

The rest of the paper is organised as follows: Section 2 gives some related work. Section 3 describes the new Step NN algorithm. Section 4 gives some test results, while section 5 gives some conclusions on the work.

2. Related Work

A recent survey on k-Nearest Neighbours is [7], where they explain that 'the K-Nearest Neighbours (k-NN) algorithm operates as a non-parametric, instance-based learning method, commonly employed in supervised learning tasks, including classification and regression. Contrasting with model-based learning approaches that deduce a function from training data to make predictions, k-NN is categorized as a lazy learning algorithm. It formulates predictions by analysing the data structure in real-time upon the introduction of new instances, without necessitating a preceding explicit training phase.' The new Step NN algorithm to be described in fact does a small amount of modelling and does require a specific training phase. Some problems with clustering algorithms are summarised in [3], who also describe their own Shared NN method. They state that: 'K-means [8] is one of the most commonly used clustering algorithm, but it does not perform well on data with outliers or with clusters of different sizes or non-globular shapes. The single link agglomerative clustering method is the most suitable for capturing clusters with non-globular shapes, but this approach is very sensitive to noise and cannot handle clusters of varying density.' Their Shared NN algorithm first finds the nearest neighbours of each data point and then, as in the Jarvis-Patrick approach [9], redefines the similarity between pairs of points, in terms of how many nearest neighbours the two points share. This helps to address those problems and it is an idea that the author has also used ([6], section 4.2), where the Step NN algorithm would be a supervised version. In that case it was fully-linked structures, whereas in this case it is single links, but they might induce multiple links. It can also address the clustering problems to some extent. For example, adding new points can possibly even-out the density over the region, or make the shape more arbitrary. SMOTE [2] is another method, and it adds new data points to correct an unbalanced dataset. They show that a combination of over-sampling the minority class using SMOTE and under-sampling the majority class can achieve better classifier performance. The minority class was over-sampled by taking each sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours. Then to create the synthetic sample, the difference between the feature vector (sample) under consideration and its nearest neighbour was multiplied by a random number between 0 and 1, and added to the feature vector under consideration. This is therefore almost the exact same method, except that SMOTE creates more examples of a class, whereas Step NN wants to link a specific instance. They note that overfitting is still a danger.

3. The Step Nearest Neighbour Algorithm

The theory for Step NN is very simple. If an outlier data row belongs to one category but is positioned closer to other category rows, then the algorithm should try to position a point from the same category closer, to classify it correctly. This involves adding a new data point to the category, at a position that is half way between the closest point from the category and the outlier. This positioning is less likely to interfere with other comparisons. The curse of dimensionality however, means that points tend to get arranged very close together, when the number of dimensions increases. Adding a new point to some region therefore, could possibly be closer to other category points as well. There should thus be an iterative process of adding new points for all categories, including for any points already added, until the error reduces and the correct classifications are obtained. The error appears to reduce to some minimum every time. This is likely to increase the density of the points in the confused region, but in an even manner and for all of the categories. This therefore

 doi:10.20944/preprints202510.1657.v1

means that new rows get added to the dataset and the train dataset must be saved and re-used when classifying new data. Other adaptive versions may have to fine-tune parameters, while this method can work automatically. The new data points are a small amount of modelling that may help it to generalise.

3.1. Pseudo-Code

Some pseudo-code for the algorithm is as follows:

1. For each data row *dr* in the dataset:
   1.1. Calculate the distance to every other row *dr2*.
   1.2. Record the row with the minimum distance *md* and the row from the same category with the minimum distance *mdc*.
2. Determine the row *dr* is correctly classified:
   1.1. If *md* and *mdc* are the same row, then row *dr* is correctly classified.
   1.2. If *md* and *mdc* are different rows, then row *dr* is not correctly classified.
3. If row *dr* is not correctly classified:
   1.1. Create a new data row, half-way between *dr* and *mdc*.
   1.2. Add it to the end of the list of data rows.
   1.3. Any new data rows are also evaluated during this iteration.
4. When completed, generate closest row clusters for each data row and check if the categories match.
   1.1. This gives an accuracy score.
   1.2. If not accurate enough, then repeat the process.

4. Testing

A computer program was written in the Java programming language and runs on a standard laptop. The algorithm is based on a very basic k-nearest neighbour, with a Euclidean distance measurement between the points. The new algorithm then, is a single nearest neighbour that also adds step points. The number of new data rows could almost double the size of the dataset, but the algorithm is still reasonably quick to run and does not require any fine tuning. Step NN took longer to train, where maybe 10 – 15 iterations were required to reduce the error on the train dataset to nothing. Step NN would then use the changed train dataset to evaluate the test set as well, which required only 1 iteration. The k-NN algorithm was trained and tested on the test set only, with a k-size of 3. If it generalised on a different train dataset, the performance would be much worse. The two evaluations therefore differed slightly, where Step NN could adjust to achieve 100% accuracy, but k-NN is not designed to do this. The test set was generated by removing 20% of the dataset and leaving it for the test phase. Each result was averaged over 50 test runs, where the data was re-split for each run.

4.1. Test Results

The classifiers were tested on several datasets from the UCI Machine Learning Repository [10], or from this site [1]. The Nearest Neighbour algorithm is not supposed to cluster some datasets well and so the comparison between the two methods is as important as the actual result. UCI [10] lists the Iris Plants and Wine Recognition, which have 3 categories each. Other datasets included the Abalone shellfish dataset, the Hayes-Roth concept learning dataset, the BUPA Liver dataset, the Cleveland Heart Disease and the Breast Cancer datasets. With the Heart Disease dataset, this test matched with the output category exactly. Another web site [1] lists the Sonar, Wheat Seeds, Car and Wine Quality datasets. It was not possible to cluster the Car dataset on buying price, but it clustered perfectly on the safety column. Then, User Modelling, Bank Notes, SPECT images heart classification, Letter recognition, the first Monks dataset, Solar flares, Pima Indians Diabetes and Ionosphere

 doi:10.20944/preprints202510.1657.v1

datasets, from UCI, were also tested. The results are shown in Table 1, where Step NN produces 100% accuracy on almost every train dataset and outperforms k-NN on almost every test dataset. For the Iris dataset, only about 5 or 6 additional points were added.

Table 1. Classifier Test results. Step NN was trained on the 80% dataset and then tested on the 20% dataset. K-NN was trained and tested on the 20% dataset, with a k-size of 3. The distance metric was Euclidean.

Dataset	Percentage Correct		
	Step NN Train 80%	Step NN Test 20%	K-NN Train/Test 20%
Iris	100	95.7	94.2
Wine	100	95.6	94.4
Abalone	100	49.8	47.9
Hayes-Roth	100	65.5	36.7
Liver	100	61.9	56.1
Cleveland	100	53.1	48.6
Breast	100	95.4	95.6
Sonar	100	86.8	67
Wheat	100	93.3	90.8
Car	100	97.5	89.5
Wine Quality	100	63.1	48.6
UM	100	85	69.7
Bank	100	99.9	99.3
SPECT	99.3	93.8	82.5
Letters	100	96.2	85.9
Monks-1	100	83.5	64
Solar	100	100	100
Diabetes	100	69.9	69.8
Ionosphere	100	87.3	81.4

5. Conclusions

Step NN is clearly not a universal classifier, but nearest neighbour algorithms are not supposed to be. So, in comparison with other nearest neighbour algorithms, such as k-NN, there is a clear improvement, including some ability to generalise. This may be because it does a small amount of modelling through adding new data rows. For the classic Iris and Wine datasets, it performs at the level that is expected, or even for the Abalone dataset. The new algorithm tackles, at least partly, some of the problems identified with clustering, where it can help to re-balance a dataset, or through local linking, it could produce clusters of more arbitrary shape. While they may or may not be principal information points themselves, they can still link the principal data together, so that it can be coherently traversed.

References

1. Brownlee, J. (2019). 10 Standard Datasets for Practicing Applied Machine Learning, https://machinelearningmastery.com/standard-machine-learning-datasets/.
2. Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W. P. (2011), SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research, 16:321–357.
3. Ertoz, L., Steinbach, M. and Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In Proceedings of the 2003 SIAM international conference on data mining (pp. 47-58). Society for Industrial and Applied Mathematics.
4. Fix, E. and Hodges, J.L. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (PDF) (Report). USAF School of Aviation Medicine, Randolph Field, Texas.

 doi:10.20944/preprints202510.1657.v1

5. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. Generative adversarial nets. Advances in neural information processing systems, 2014, 27..
6. Greer, K. (2020). A Pattern-Hierarchy Classifier for Reduced Teaching, WSEAS Transactions on Computers, ISSN / E-ISSN: 1109-2750 / 2224-2872, Volume 19, 2020, Art. #23, pp. 183-193.
7. Halder, R.K., Uddin, M.N., Uddin, M.A., Aryal, S. and Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. Journal of Big Data, 11(1), p.113.
8. Jain A.K. and Dubes, R.C. (1988). Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, New Jersey, March 1988.
9. Jarvis, R.A. and E. A. Patrick, E.A. (1973). Clustering using a similarity measure based on shared nearest neighbors, IEEE Transactions on Computers, C-22(11).
10. UCI Machine Learning Repository (2019). http://archive.ics.uci.edu/ml/.