



Research article

Adding link points to improve Nearest Neighbor classification

Kieran Greer*

Distributed Computing Systems, Belfast, UK

* **Correspondence:** Email: kgreer@distributedcomputingsystems.co.uk.

Academic Editor: Pasi Fränti

Abstract: One solution to nearest neighbor outliers is to link them with the class cluster by adding new points between the outlier and the cluster. In the age of Generative AI, new data points are legitimate, and if they are added equally and only when needed, they might help re-balance the dataset. In this paper, we describe a nearest neighbor algorithm called Step Nearest Neighbor (Step NN), where incorrectly identified points can be re-linked by adding new points between them and the main cluster. The step points can also be noted, so that they can, for instance, be ignored as real cases. As new points were added, it worked only with distance-based metrics, however. While other adaptive versions may have to fine-tune parameters, this method was easier to use and as it could be pre-trained, this offers options for big data and distributed environments. Test results showed that accuracy comparisons with the classic version of k-NN is almost equal, but Step NN can generalise. In the cases where k-NN did well, k-NN should be equal to or better than Step NN. In the cases where k-NN did not do well, Step NN is likely to be better.

Keywords: nearest neighbor; step points; train phase; modelling; generalise

1. Introduction

Classification algorithms can be used in many fields involving pattern recognition or data processing. The k-nearest neighbors algorithm (k-NN) [1] is one of the first invented and is used in recommendation systems, finance, healthcare, and Internet of Things (IoT), to name a few. One problem with nearest neighbor algorithms is how to deal with outliers, which are points of one class that lie closer to other class points. One solution would be to add a new point between the outlier and its main cluster, so that the new point is 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 [2] however, new data points are legitimate and if they are added equally, and only when needed, they might even help re-balance the dataset.

In this paper, we describe a new nearest neighbor algorithm called Step Nearest Neighbor (Step NN). It is so-called because points that are incorrectly identified in a supervised setting can be re-classified by adding new points between them and the main class cluster, so that the path re-links the point with its correct class cluster. These new points can also be noted, so that a system whose purpose is to retrieve similar cases could still navigate from them to the original data. Another idea would be to physically move the points causing incorrect classification apart, by a small amount. This would, however, permanently delete the original position from the dataset, and small increments would require more iterations, because a distance in the opposite direction is more difficult to measure.

Moving or adding from the main cluster to the outlier can be done in fewer steps, where the new points may also need to be resolved. The error however, reduces over multiple runs, using the updated dataset each time. Basically, any point that lies inside a cluster boundary is valid for that class. The question is therefore where to place a new point so that it best links with the outlier. A biological solution¹ would maybe half the distance each time, and while a computer program can measure the distance more accurately, this still seems reasonable. The changed dataset therefore contains the original data plus new data points and needs to be kept and used for evaluating new datasets. The algorithm is currently considered with only 1 nearest neighbor.

The rest of the paper is organized as follows: Section 2 describes the classic k-NN algorithm and gives some related work. Section 3 describes the new Step NN algorithm. Section 4 gives some test results, while section 5 provides some conclusions.

2. Classic k-NN and related work

Almost all nearest neighbor algorithms are a variation of k-NN. A survey on k-Nearest Neighbors is found in [4], where the researchers explain that ‘the K-Nearest Neighbors (k-NN) algorithm operates as a non-parametric, instance-based learning method, commonly employed in supervised learning tasks, including classification and regression.’ The researchers also write that in contrast to model-based approaches that deduce a function to make predictions, k-NN is categorized as a lazy learning algorithm. It formulates predictions by analyzing the data structure in real-time, without necessitating a preceding training phase.

A survey of nearest neighbor applied to the medical domain is found in [5]. They describe that the k nearest data points from the query point are found according to some metric. The algorithm then performs a majority vote to find which class appeared the most in those points and assigns that class to the query point. It is described that many variants try to optimize the k value because it can have a profound effect on the result. If it is too low, then overfitting can occur, where the comparison is too specific. If the value is too large, then underfitting can occur, and the model is too general to match accurately. Extra calculations are thus carried out to try to select the best k -value, and so an algorithm that does not need to do this would have an advantage. While the best k value can be estimated, typical values include 3 or 5. Other variants try to improve the accuracy, either by improving the evaluation metric, or by weighting the association of a data point to a class. These are more accurate, but at the cost of a much more elaborate algorithm. The new Step NN algorithm to be described does a small amount of modelling and requires a specific training phase.

Some problems with clustering algorithms are summarized in [6], who also describe their own Shared NN method. They state that: ‘K-means [7] is one of the most commonly used clustering algorithms, but it does not perform well on data with outliers or with clusters of different sizes or non-

¹ It was decided that, in the human brain, adding a new neuron half-way between two other neurons is the most economic energy-wise [3]. If the best position is not known, then it will also remove any bias.

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 neighbors of each data point and then, as in the Jarvis-Patrick approach [8], redefines the similarity between pairs of points in terms of how many nearest neighbors the two points share. This helps address those problems and is also an idea that author has used ([9], 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 further multiple links. This can also address irregular shaped clusters to some extent. For instance, adding step points can possibly even-out the density over the region, or make the shape more arbitrary.

SMOTE [10] is another method that adds new data points to correct an unbalanced dataset. In [10], it was shown that a combination of over-sampling the minority class using SMOTE and under-sampling the majority class can achieve better classifier performance. Moreover, 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 neighbors. Then to create the synthetic sample, the difference between the feature vector (sample) under consideration and its nearest neighbor 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.

Machine Learning classifiers are also used on Big Data [4], but this produces new challenges due to the size of the datasets. This is particularly true with k -NN, which is non-parametric. This means that there are no assumptions about the data, and it must be used in full when performing any classification task. One option is to parallelize the process; another option is to create a separate train dataset and manage it, so that only the essential features remain [11]. Thus, the separate train dataset used by Step NN would be beneficial for Big Data as well.

3. The Step nearest neighbor algorithm

The theory for Step NN is very simple. If a data point belongs to one class but is positioned closer to other class points, then the algorithm should try to position a point from the same class closer to classify it correctly. This involves adding a new data point, at a position that is half way between the closest point from the class cluster and the outlier, as illustrated in Figure 1. This positioning is less likely to interfere with other comparisons. If the new point is still not the closest, then a second point is added halfway between it and the outlier, and so on. The curse of dimensionality, however, means that points tend to get arranged very close together when the number of dimensions increases. The newly added points could possibly be closer to other class points as well. Thus, there should be an iterative process of adding new points for all classes, including for any new points already added, until the error reduces or a maximum number of iterations has been executed. The error appears to reduce to some minimum every time and can be a count of how many points are incorrectly classified. This likely increases the density of the points in the confused region, but in an even manner, for any class equally. This therefore means that new points 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, which may help it to generalize.

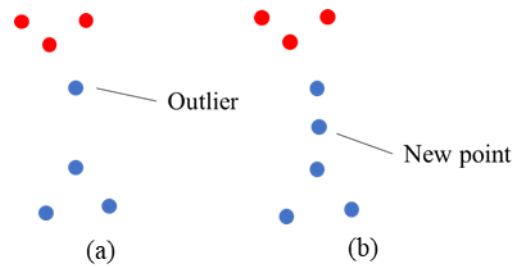


Figure 1. A point from class A can be correctly classified if new points from class A are added to lie closer to it. (a) The top blue point from class A is closer to red class B points; and (b) a new blue point can be added between the outlier and the main blue cluster, with minimal disturbance to the rest of the dataset. The outlier is now classified correctly.

Pseudo-code for the algorithm is as follows:

```

while (points are incorrectly classified, or up to a maximum number of iterations)
  for (each data point  $dr$  in the dataset)
    Calculate the distance to every other point  $dr2$ .
    Record the point with the minimum distance  $md$ .
    Record the point from the same class with the minimum distance  $mdc$ .
    if ( $md$  equals  $mdc$ )
      Point  $dr$  is correctly classified.
    endif
    if ( $dr$  is not correctly classified)
      Create a new data point, half-way between  $dr$  and  $mdc$ .
      Add it to the end of the list of data points.
      Thus, any new data points are also evaluated during the iteration.
    endif
  endfor
  Generate closest point pairs for each data point in the updated dataset.
  Test if any points are incorrectly classified.
endwhile

```

4. Testing

The classic k-nearest neighbor algorithm was implemented in the Java programming language. The new algorithm then was a single nearest neighbor that added step points. The source code for this can be found at [12] and tests were run on a standard laptop with a 3.3 GHz processor. The increase in the dataset size was typically less than a factor of 2. Step NN took longer to train, where 10–15 iterations were required to reduce the error on the train dataset to a minimum. 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 tested on the whole dataset, with a k-size of 5. If it was tested on the smaller Step NN test dataset, the performance would be much worse and so this is one advantage of the Step NN algorithm. As it does a small amount of modelling, it can adjust its parameters and accuracy, whereas k-NN is not designed to do this. The test therefore entails comparing the new points added in Step NN versus the k-point averaging of k-NN. Each result was averaged over 50 test runs, where the data was re-split for each test run.

4.1. Possible algorithm improvement

There are cases where this algorithm will not work well by default. One example would be to try the Iris dataset [13] but use only 2 features for either the petal width or length, not both. In this case, the scenario shown in Figure 2 occurs. A point from one class is blocked from connecting with the other points by a point from a different class. Because of the alignment, adding new points in this direction will not solve the problem, as it would require a much more complicated geometry of finding a circular path that is outside the radius of the blocking point. In this case, however, the incorrectly classified point is a genuine outlier, which is a peculiarity of the dataset and it may not be worth the more complicated calculations to try to re-join it with the other points from the same class. It may also affect the integrity of the dataset too much. One possibility, however, is to select any point from the same class, not just the closest one, so long as the new point added is not then closer to points from other classes. This does not improve the accuracy, but it can reduce the number of new points that get added, making it a possibility for some datasets.

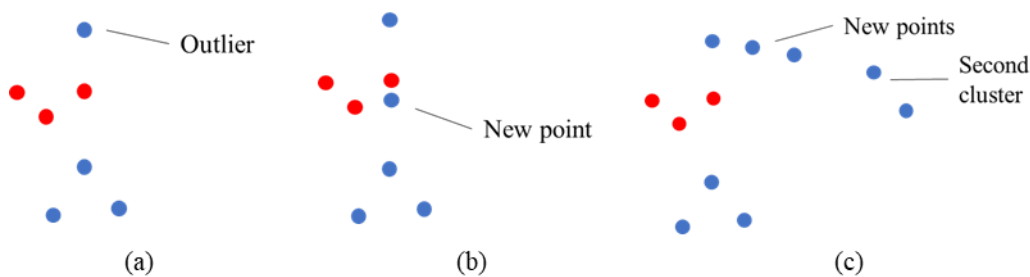


Figure 2. If a point is blocked and becomes a genuine outlier, then the algorithm will not be able to re-connect it directly, where a build-up of points around the blocking one may occur. Thus, it is more economic to select a second cluster that is slightly further away but requires fewer points to link. (a) The top point is from class A, but is blocked by a point from class B, to the other class A points; (b) the new point is also closer to the class B point; and (c) if a second cluster that is slightly further away is used, then the linking can be OK.

4.2. Evaluation metric

K-NN can be used with any type of metric, but the Step NN algorithm may only work with ones that strictly measure a distance. This would include Euclidean and City Block (Manhattan), for example, but would exclude Cosine or Jaccard, for example. This is because the nature of the algorithm requires new points to be positioned, which does not work with a similarity measure.

4.3. Test results

The classifiers were tested on several datasets from the UCI Machine Learning Repository [14], or from [15]. The Nearest Neighbor algorithm is not supposed to cluster some datasets well, and so the comparison between the two methods is as important as the accuracy score. The possible algorithm improvement from Section 4.1 was included but is not significantly better. UCI [14] lists the Iris Plants and Wine Recognition datasets. Also, Hayes-Roth concept learning, BUPA Liver disease, Cleveland Heart disease, and the Breast Cancer datasets. With the heart disease dataset, this test matched with the output class exactly. Another web site [15] lists the Sonar, Wheat Seeds, Car, and Wine Quality

datasets. It was not possible to classify the Car dataset on buying price, but it classified 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 datasets, from UCI, were also tested. The results are shown in Table 1, where the datasets were first normalized.

Table 1. Classifier Test results. Step NN was trained on 75% of the dataset and then tested on the remaining 25%. Step NN V2 includes the possible improvement. K-NN was tested on the 100% dataset only, with a k-size of 5. The distance metric was Euclidean for Step NN and either City Block or Cosine for k-NN. Numbers are the percentage of points correctly classified. Total is the number of wins by either Step NN or k-NN.

Dataset		Percentage Correct				
		Step NN V1		Step NN V2	Classic k-NN	
		Train 75%	Test 25%	Test 25%	Test only 100%	
Name	Cols x Rows	Euclidean Distance		City Block	Cosine	
Iris	4 x 150	100	95.8	95.7	94.7	72.7
Wine	13 x 178	100	95.1	97.4	96.6	89.3
Hayes-Roth	5 x 132	100	66	67.4	44.7	35.6
Liver	7 x 345	100	62.3	60.3	66.7	56.2
Heart	14 x 303	100	53.5	54	50.8	54.5
Breast	32 x 569	100	95.5	95.6	97	76.5
Sonar	61 x 208	100	86.4	86.4	84.1	62
Wheat	7 x 210	100	93.6	93.5	91.5	76.7
Car	6 x 1728	100	89.3	88.8	98.4	70
Wine Quality	11 x 1599	100	62.2	62.1	53.8	45
UM	5 x 403	100	84.5	83.5	88.6	29.2
Bank	4 x 1372	100	99.9	99.9	99.9	60.3
SPECT	23 x 187	99.3	63.8	62.8	63.1	55.1
Letters	17 x 20000	100	95.6	95.6	94.2	23.5
Monks-1	6 x 432	100	72.3	72.4	91.7	60.9
Solar	61 x 208	100	99.4	99.3	99.5	99.5
Diabetes	8 x 768	100	70.2	69.3	73.7	64.3
Ionosphere	35 x 351	100	87.6	87	88.6	88.9
Total			8.5		9.5	

Step NN produced 100% accuracy on almost every train dataset and was almost equal on its unseen 25% test set versus k-NN's 100% test set. As such, using the 100% test set is much more advantageous for k-NN. For the Iris dataset, only about 5 or 6 additional points were added to Step NN. K-NN was tested with 2 distance metrics to show that they can affect the result. City Block was shown instead of Euclidean because it was slightly better, and then Cosine was tried because k-NN can also use similarity measures. The results showed that the 2 algorithms scored almost the same but with different mentioned setups. The Step NN variations were better in 8.5 cases, while k-NN was better in 9.5 cases.

4.4. Relative times

The potential improvement to the algorithm in Step NN V2 did not affect the timings in these datasets, suggesting that the type of outlier is not very common in categorising data. A timing result

for Step NN in general can therefore be presented. To make the timings equivalent, the dataset was split in half, so that 50% would be in the train dataset and 50% would be in the test dataset, and all tests then used the same test dataset. The accuracy was not a matter here, and the relative times for some datasets are listed in Table 2. The Step NN algorithm required a much longer training phase, but the testing phase also took longer. While both performed a quadratic comparison of each point with every other point, there were a few more calculations in Step NN.

Table 2. Time in milliseconds to execute either the train or the test phase. Step NN requires a train and a test phase, while k-NN uses only a test phase. The dataset size for either phase was the same.

Dataset	Step NN		K-NN
	Train	Test	Test
Iris	30	1	1
Sonar	89	4	2
Wine Quality	5674	279	34
Letters	182150	11064	6008

5. Conclusions

Step NN is not a universal classifier, but nearest neighbor algorithms are not supposed to be. The classic k-NN algorithm appears to be just as accurate in general, but under favorable conditions. As it is non-parametric, it uses only a test phase and was evaluated on the whole dataset. If it was evaluated on the smaller test set that Step NN used, it would be appreciably worse. The algorithms appear to perform at the level that is expected, such as the classic Iris and Wine datasets. Because a distance value is critical to Step NN, there is some restriction on what evaluation metric can be used. Moreover, while a separate training phase will take more time, it also offers advantages. With regards to big data, comparisons to a separate train dataset may enable a test set to be split or parallelized, for example. This would be done in conjunction with selectively adding to and managing the train dataset, which is a focus of recent work (for example, [11]). The new algorithm tackles, at least partly, some of the problems identified with classification, where it can help to re-balance a dataset or, through local linking, it could produce clusters of more arbitrary shapes. In the context of this paper, re-balancing would mean to re-join outliers with their main clusters. To summarize: In the cases where k-NN does well, k-NN is as accurate or better than Step NN. In the cases where k-NN does not do well, Step NN is likely to do better. Regarding timing, k-NN is quicker, but it should be relative to the number of new points that get added to Step NN. Moreover, if the number of new points becomes negligible, then Step NN has a comparable complexity to k-NN. An advantage of Step NN however is that it can generalize a bit.

Use of Generative-AI tools declaration

The author declares he has not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The author declares no conflicts of interest.

References

1. E. Fix, J. L. Hodges, Discriminatory analysis: nonparametric discrimination, consistency properties, Project Report, 1951, 4.
2. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial nets, *Commun. ACM*, **63** (2020), 139–144. <https://doi.org/10.1145/3422622>
3. K. Greer, New ideas for brain modelling 2, In: *Intelligent systems in science and information 2014*, Cham: Springer, 2014, 23–39. https://doi.org/10.1007/978-3-319-14654-6_2
4. R. Halder, M. Uddin, M. Uddin, S. Aryal, A. Khraisat, Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications, *J. Big Data*, **11** (2024), 113. <https://doi.org/10.1186/s40537-024-00973-y>
5. S. Uddin, I. Haque, H. Lu, M. Moni, E. Gide, Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction, *Sci. Rep.*, **12** (2022), 6256. <https://doi.org/10.1038/s41598-022-10358-x>
6. L. Ertöz, M. Steinbach, V. Kumar, Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, *Proceedings of the 2003 SIAM International Conference on Data Mining*, 2023, 47–58. <https://doi.org/10.1137/1.9781611972733.5>
7. A. Jain, R. Dubes, *Algorithms for clustering data*, Englewood Cliffs: Prentice Hall, 1988.
8. R. Jarvis, E. Patrick, Clustering using a similarity measure based on shared nearest neighbors, *IEEE Trans. Comput.*, **C-22** (1973), 1025–1034. <https://doi.org/10.1109/T-C.1973.223640>
9. K. Greer, A Pattern-Hierarchy classifier for reduced teaching, *WSEAS Transactions on Computers*, **19** (2020), 183–193. <https://doi.org/10.37394/23205.2020.19.23>
10. N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, **16** (2002), 321–357. <https://doi.org/10.1613/jair.953>
11. H. Saadatfar, S. Khosravi, J. Joloudari, A. Mosavi, S. Shamshirband, A new K-nearest neighbors classifier for big data based on efficient data pruning, *Mathematics*, **8** (2020), 286. <https://doi.org/10.3390/math8020286>
12. K. Greer, *Step NN source code*, GitHub, Inc., 2026. Available from: <https://github.com/discompsys/Step-NN>.
13. R. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, **7** (1936), 179–188. <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>
14. UCI, *UCI machine learning repository*, UC Irvine, 2026. Available from: <http://archive.ics.uci.edu/ml/>.
15. J. Brownlee, *10 standard datasets for practicing applied machine learning*, Guiding Tech Media, 2026. Available from: <https://machinelearningmastery.com/standard-machine-learning-datasets/>.



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)