

FLEXIBLE K NEAREST NEIGHBORS CLASSIFIER: DERIVATION AND APPLICATION FOR ION-MOBILITY SPECTROMETRY-BASED INDOOR LOCALIZATION

PHILIPP MÜLLER*

ABSTRACT. The K Nearest Neighbors (KNN) classifier is widely used in many fields such as fingerprint-based localization or medicine. It determines the class membership of unlabelled sample based on the class memberships of the K labelled samples, the so-called nearest neighbors, that are closest to the unlabelled sample. The choice of K has been the topic of various studies and proposed KNN-variants. Yet no variant has been proven to outperform all other variants. In this paper a new KNN-variant is proposed which ensures that the K nearest neighbors are indeed close to the unlabelled sample and finds K along the way. The proposed algorithm is tested and compared to the standard KNN in theoretical scenarios and for indoor localization based on ion-mobility spectrometry fingerprints. It achieves a higher classification accuracy than the KNN in the tests, while requiring having the same computational demand.

1. IMPORTANT NOTE BY THE AUTHOR

As of October 2023 the author has been aware that **the concept behind the introduced Flexible K Nearest Neighbors classifier has been proposed and used before under the name Fixed Radius Near Neighbor search or Radius Neighbor classifier**. According to a survey by Bentley [1], the method was first used in 1966 by Levinthal in an "interactive computer graphics study of protein molecules", but from Levinthal's paper [2] neither a clear mentioning nor a derivation of the Fixed Radius Near Neighbor technique could be found. Hence it is unclear to the author where the underlying idea of the Flexible K Nearest Neighbors classifier was proposed for the first time.

To the authors knowledge, **the application of the concept for ion-mobility spectrometry fingerprint-based indoor localization is novel**. In addition, the **paper provides an insight into the strengths and the weaknesses of using only training samples within a predefined distance to the test sample for inferring the latter one's label**, which might be of interest for readers new to fixed radius-based nearest neighbors classifiers. Recent articles using the concept include, for example, [3], [4], and [5].

*Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland,
Email: philipp.muller@tuni.fi

Version: 5 October 2023

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This work was supported in part by the Academy of Finland, grant 323472 (under consortium "GaitMaven: Machine learning for gait analysis and performance prediction").

2. INTRODUCTION

The K Nearest Neighbors (K NN) classifier is a widely used and thoroughly studied machine learning algorithm due to its simplicity, ease of implementation and being parameter-free [6]. It is an extension of the nearest neighbor rule, which is a suboptimal classifier whose error rate is at most twice the Bayes rate [7, p. 177].

In classification the label of so-called test samples is derived from the labels of so-called training samples. If the training and test samples follow the same distribution then support vector machines, Fuzzy and Bayesian classifiers, as well as neural-network-based classifier perform well. However, if the distributions of training and test samples vary even slightly then the performance of these methods can degrade significantly and K NN and its variants are commonly used [8]. The main idea of the K NN is to find the K training samples that are closest, with respect to a predefined distance measure, to the test sample and infer the class membership from class memberships of these K training samples.

One application where differences between distributions of training and test samples can be observed is fingerprint-based indoor localization. For example, in [9] the weighted K NN outperformed several parametric classifiers for indoor localization based on Wireless Local Area Network (WLAN) measurements if the WLAN access point density was high. However, if the density was low then the accuracy level of the weighted K NN degraded considerably. One reason for the lowered performance was the lack of training sample locations in which the same access points were observed as in the test sample (see [9] for details). In such a case choosing a different K might help to improve the accuracy and numerous K NN-variants have been proposed that focus on optimizing K .

However, in some situations even optimizing K is of little use. If there are no training samples within the close neighborhood of the test sample than any K NN-variant will yield a wrong class membership label for the test sample. Examples of such situations can be found, for example, in indoor localization (test sample from a room or area of a building for which no training data has been collected) or medicine (classification of a disease for which no training data is available). In order, to handle also such situations well and avoid time-consuming optimizations this paper proposes a K NN-variant in which the allowed distance between training and test samples is limited from above. Only training samples that are within the limit are used for inferring the label of the test sample. This way, a different K will be found for each test sample, and therefore the proposed classifier is named Flexible K NN (Flex K NN). If $K = 0$ then the Flex K NN will simply yield information that no training sample is close enough to the test sample to provide a reliable estimate of class membership for the test samples. This is, in the authors opinion, more useful than an untrustworthy label provided by K NN and its existing variants. To the authors knowledge no such K NN-variant has been published in earlier papers.

The remainder of this paper is organized as follow. Section 3 discusses the standard K NN in detail and provides an overview on proposed K NN-variants. The Flexible K NN is introduced and explained in detail in Section 4. In Section 5 the performances of Flex K NN and standard K NN are compared for indoor localization based on ion-mobility spectrometry (IMS) measurements. Section 6 contains concluding remarks and an outlook on further research.

3. RELATED WORK

The standard K Nearest Neighbors classifier is a model-free classifier that returns a label for an n -dimensional vector $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$, which indicates that the so-called test sample \mathbf{x} belongs to a class m ($m = \{1, \dots, M\}$). The label is derived by finding the K closest samples

from a database \mathbf{Y} containing N training samples $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N] \in \mathbb{R}^{n \times N}$ for which their class affiliations (i.e. their labels) are known. The label that occurs most frequently amongst the labels of the K closest training samples (i.e. the K nearest neighbors) is then chosen as label for the test sample.

Closeness between test sample \mathbf{x} and a training sample \mathbf{y}_i is often measured by the Euclidean distance d_E , which is defined as

$$(1) \quad d_E(\mathbf{x}, \mathbf{y}_i) = \sqrt{\sum_{j=1}^n (x_j - y_{i,j})^2}.$$

For $\mathbf{x}, \mathbf{Y} \in \mathbb{R}^n$ Euclidean distance is a reasonable choice because \mathbb{R}^n is a Euclidean space. For non-Euclidean spaces a different distance measure should be used. Furthermore, alternative distance measures have been proposed and tested also for use in Euclidean space. For example, [10] proposed the Hassanat distance, which uses maximum and minimum vector points. It outperformed the standard KNN and eight KNN -variants over eight machine learning benchmark datasets from the field of medicine in [11]. One of the KNN -variants it outperformed was the Generalised Mean Distance KNN , which finds the K closest training samples from every class m , converts lists of these samples to local mean values, and then computes multiple mean distances to obtain distances between \mathbf{x} and all M classes [11]. In [12] the Euclidean distance was compared to 66 alternative distance measures, such as Minkowski and Manhattan distances, inside a KNN for indoor localization based on ion-mobility spectrometry fingerprints.

The choice of K has been target of numerous studies. For fixed K all neighbors of the test sample would converge towards \mathbf{x} if N would converge to infinity [7, p. 183] and any value of K would be acceptable. However, in the real world $N \ll \infty$ and choosing a suitable K is not straightforward.

For $K \rightarrow \infty$ the K nearest neighbors rule would be optimal [7, p. 183] as it eliminates any measurement noise, so a large K is desirable. The drawback is that for large K classes with large numbers of samples would be preferred over classes with small numbers of samples [6], which could weaken the classification accuracy. Therefore, a small K would be desirable to ensure that all nearest neighbors are close to \mathbf{x} , but here class outliers might cause false labelling [6]. Furthermore, having the same or at least similarly number of samples for each class is desirable, but still a compromise needs to be found. One option would be to calculate the error rates for a large range of feasible K values and then choose the one with the minimal error rate.

In general, an odd value is chosen for K as it limits the large-sample two-class error rate from above [7, p. 183] and helps to avoid ties in the majority vote in classification problems where the K closest training samples are from two classes. This does not ensure, however, that the KNN always finds a label. Consider the case where $K = 9$ and $M > 3$. Then it would be possible that the nine nearest training samples consist of three times three samples from three different classes, which would result in a tie. One approach to avoid such ties is using weights w_k ($k = \{1, 2, \dots, K\}$) for the nearest neighbors such that $\sum_{k=1}^K w_k = 1$. A common approach is to use the inverses of distances between test sample and nearest neighbors and normalize them to sum up to one.

Besides finding optimal K values that would be used for classification of any test sample independent on its location and the training samples in its neighborhood, attempts have been

For a two-category classification problem the error rate converges to the lower bound, known as the Bayes rate if $K \rightarrow \infty$ [7, p. 184.].

What values are feasible depends, for example, on the total number of training samples N and classes M .

made to use flexible K values. For example, Wettschereck and Dietterich [13] proposed four KNN variants that determine optimal K values for each training sample. Their first approach stores for each training sample a list of K values that would correctly classify the training sample using leave-one-out cross-validation. For classifying a test sample \mathbf{x} its \tilde{K} nearest neighbors are searched. Next K_{opt} , which is the K value that would correctly classify most of the \tilde{K} neighbors, is determined. K_{opt} is then used to determine the label of \mathbf{x} based on majority vote as in the standard KNN . Thus, the K_{opt} differs locally, and hence is called by the authors a locally adaptive KNN method. The three remaining KNN variants in [13] are modifications of this adaptive KNN . All four variants are suitable for applications in which patterns differ considerably for different regions. For example, in [?] K values estimated to be optimal ranged from 1 to 10 for regions with low training sample density, while for regions with high intensity optimal K values ranged from 30 to 50. Numerous adaptive KNN variants have been proposed since the publication of [13]. One example is the locally adaptive KNN based on discrimination class (DC-LAKNN), which first determines the discrimination classes of the first and second majority class for various K values. It then uses quantity and distribution information in the discrimination classes to find the optimal K to be used in classification [14].

In [15] authors proposed a KNN -variant based on an ensemble approach. It classifies the test sample \sqrt{N} times with KNN classifiers for which $K = \{1, 2, \dots, \sqrt{N}\}$. The overall label is obtained by using the weighted sum of the results from the \sqrt{N} KNN s. In [16] a KNN variant that demands the user to define a confidence level p ($p \in [0, 1]$) rather than K is proposed. The algorithm adjusts K such that the probability for the label of \mathbf{x} to be correct is at least p . In [17] sparse learning is used. The proposed KNN -variant accounts for correlation between samples and reconstructs \mathbf{x} as a linear combination of training samples before determining the label by majority vote. The drawback of most methods that search for optimal K values is that the optimization process is often time-consuming.

To avoid the need for optimization Fuzzy KNN -variants could be used, which provide probabilities of \mathbf{x} belonging to any of the classes present amongst the K nearest neighbors. Thus, it is less likely to get ties and similarly to the approach in [16] a confidence level on the overall label is available. However, even for Fuzzy KNN identifying a suitable K is nontrivial and if training samples are uncertain or ambiguous then using a fixed K is unreliable [8]. Therefore, [8] proposes to derive an optimal K for any test sample, increasing the computational demand considerably. In [18] the standard KNN is extended by a training stage in which a decision tree is build for predicting an optimal K for any test sample. In the classification stage first the optimal K is derived from the decision tree for a test sample, before the standard KNN with K set to the optimal value is used to obtain the label for the test sample. In [19] the N-Tuple Bandit Evolutionary Algorithm [20] is used to obtain the optimal combination of K and distance metric or a combination close to the optimum.

Various further KNN -variants have been proposed. The reader is referred to surveys (e.g., [11, 21]) that provide a thorough overview on the topic. The algorithm proposed in the next section does not require any optimization of K . Instead it works exactly like the standard KNN except for requiring a different number as input.

4. FLEXIBLE K NEAREST NEIGHBORS

One limitation of the KNN -variants mentioned above is that they assume that a test sample belongs to one of the classes for which training data is available. However, this assumption

In case $\sqrt{N} \notin \mathbb{N}$ then \sqrt{N} rounded to the nearest integer towards $-\infty$ is used as upper value for K .

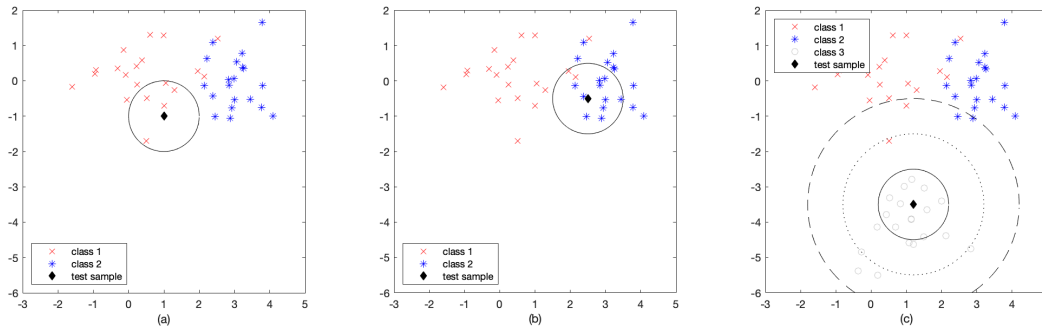


FIGURE 1. Illustration of the working principle of the Flexible KNN for two-dimensional data. Training samples from two classes are visualised by red crosses (class 1) and blue asterisks (class 2) in all three subfigures. The maximum allowed distance d_{\max} between test sample (black diamond symbol) is set to one and is visualized by the black circle around the test sample. The Flexible KNN classifies the test sample belonging to class 1 in (a) or class 2 in (b). For the KNN the yielded label depends on the choice of K . In subfigure (c) samples from a third class are visualized by grey circles. These samples are unknown to the classifier. The test sample clearly belongs to this unknown class. The Flexible KNN returns information that no label can be provided because no training sample lies within d_{\max} . Contrary, the KNN would classify the test sample wrongly as belonging to either class 1 or 2 depending on the choice of K .

does not hold in the real world all the time. Furthermore, many variants aim at optimizing the value of K without considering the distances between nearest neighbors and the test sample for anything other than weighting the impact of the nearest neighbors on the final label for the test sample.

Figure 1 illustrates why this approach is not appropriate all time. Here all samples are assumed, for simplicity, to be $n = 2$. However, it is straightforward to extend the example to higher dimensional spaces. In Figure 1(a) and (b) training samples from two classes are visualised by red crosses (class 1) and blue asterisks (class 2), and each class consists of 20 training samples. The test sample is visualized by the black diamond. The circle around it has a radius of one. The standard KNN would classify the test sample in Figure 1(a) as belonging to class 1 for any $K = \{1, 2, \dots, 5\}$. For the test sample in Figure 1(b) KNN would classify it as belonging to class 2 for, at least, any $K = \{1, 2, \dots, 12\}$ (there are twelve training samples inside the black circle, with ten of them belonging to class 2). Figure 1(c) now contains also samples from class 3 illustrated by 20 grey circles. Clearly the test sample would now belong to class 3 (eleven training samples from class 3 are inside the circle with radius one). However, if these training samples were not available to the classifier then the standard KNN and the variants presented before would label the test sample as being a member of either class 1 or class 2, depending on the choice of K and whether distance-based weights are used.

In order to prevent such a misclassification, this paper introduces a KNN variant that does not use K as a parameter chosen by either the user or an optimization algorithm. Instead the maximum distance d_{\max} between test sample and training samples is used as the only input parameter. Thus, the value of K is here an output rather than input parameter as in the KNN .

It varies for each test sample and depends on the locations of the test sample and the training samples in the d -dimensional space. Hence, the proposed K NN-variant is called Flexible K NN (Flex K NN).

Let us revisit the examples in Figure 1 and assume that $d_{\max} = 1$ based on prior knowledge on the training data. In the example Figure 1(a) five training samples from class 1 are inside the circle with radius d_{\max} . Thus, $K = 5$ and Flex K NN would classify the sample as being a member of class 1. In the example in Figure 1(b) ten samples from class 2 and two from class 1 are inside the circle, hence $K = 12$ and Flex K NN would classify the sample as being a member of class 2. For the example in Figure 1(c) no training samples are inside the circle with radius d_{\max} . Hence, Flex K NN would return information that the test sample is either an outlier or does not belong to either of the two classes for which training samples are available to the classifier. Only by increasing d_{\max} to 2 (dotted circle) one training sample from class 1 would be inside the maximum allowed distance. However, based on the distances of this red cross to other red crosses it might be an outlier itself. Further increasing d_{\max} to 3 (dashed circle) would result in two samples from each class 1 and class 2 inside the circle. In such a situation the normalized, inverse distance between test sample and training samples could be used as weights.

Algorithm 1 provides the pseudo-code for the Flexible K NN. The only difference compared to the standard K NN is that instead of K maximum distance d_{\max} is required as input parameter. As mentioned before the maximum distance could be chosen randomly, but it is advisable to use prior knowledge. For example, one could calculate for each class the average distance over all training samples from class l using leave-one-out cross-validation. This means, for each sample x_i ($i = \{1, 2, \dots, N_l\}$) in the class distances to samples x_j ($j = \{1, 2, \dots, N_l\}$, $j \neq i$) are calculated and the average distance d_i for sample x_i is stored. Finally, the average over all d_i ($i = \{1, 2, \dots, n\}$) is calculated to obtain d_l . Assuming that there are M classes $d_{\max} = \frac{1}{m} \sum_{l=1}^M d_l$. Alternative definitions for d_{\max} , such as the median over all d_l , are also possible.

Algorithm 1 Pseudo-code of Flexible K NN

Input: training samples $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N] \in \mathbb{R}^{n \times N}$ from M classes, test sample $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$, maximum distance between test and training samples d_{\max} .

Output: label for test sample \mathbf{x} , number of training samples K within distance d_{\max} of test sample \mathbf{x} .

Step 1: Calculate Euclidean distances d_i between \mathbf{x} and \mathbf{y}_i ($i = \{1, \dots, N\}$) using equation (1).

Step 2: Search all training samples for which $d_i \leq d_{\max}$. Training samples fulfilling the condition form the set of K nearest neighbors.

Step 3: Determine label of \mathbf{x} based on the labels of the K nearest neighbors using majority vote. If $K = 0$, return information that no training samples are within d_{\max} from \mathbf{x} and that no label can be provided.

Although Euclidean distance yields in general good performance, alternative distance measures could be employed in Algorithm 1 to potentially improve accuracy levels (for a thorough overview on distance measures see, e.g. [22]). For example, in [12] 66 alternative distance measures were compared to the Euclidean distance for indoor localization based on IMS fingerprints. Ruzicka, Canberra, and Vicis Symmetric χ^2 were Pareto optimal in the tests and

Grey dots illustrate training samples that are not available to the classifier.

achieved higher localization accuracy than the Euclidean distance while requiring less computation time. However, other metrics can require additional parameters to be defined by the user.

5. APPLICATION FOR IMS-BASED LOCALIZATION

5.1. Test with full training data. This section compares the performances of the proposed FlexKNN and standard KNN for positioning based on IMS fingerprints. For evaluation the dataset from [23] is used, as the KNN performed in that paper poorly for part of the test samples due to training and test samples being too dissimilar. The FlexKNN is believed to mitigate this problem.

Ion-mobility spectrometry is a technique for measuring volatile organic compounds (VOCs). To the authors knowledge, the possibility to localize based on VOC fingerprints was studied for the first time in [23]. The dataset contained 8,736 IMS fingerprints from seven different rooms on the campus of Tampere University of Technology, Finland. For each room data was collected once during weekend when the university buildings were (almost) empty and once during the week when staff and students were present. This was done to investigate the temporal stability of IMS fingerprints. It is known that especially humidity and temperature, but also air currents and barometric pressure influence the mobility of molecules [24, p. 250 ff.] and thus have an impact on the IMS readings.

The data from [23] were collected using a ChemPro100i from Environics Oy (Mikkeli, Finland), which yielded IMS fingerprints of dimension $n = 14$. For each room approximately 600 fingerprints were collected for both empty (ie., during weekend) and crowded (i.e., on a weekday) conditions.

In [23] the KNN performed well (classification accuracies close to 100%) for $K = \{1, 3, 5, 7\}$ when training and test samples were collected on the same day but performed poorly (classification accuracies between 28.21% and 37.38%) when training and test data were collected on different days, in different conditions. The analysis confirmed that IMS fingerprints depend strongly on the environmental conditions and normalizing the data was insufficient to mitigate the impact.

5.2. Test with full training data. For evaluating FlexKNN and standard KNN in this section normalized data from crowded (4,375 samples) and empty conditions (4,361 samples) were used for training and testing respectively. Distances between samples were measured by the Euclidean distance (1). The KNN with $K = 3$ classified 37.38% of the training samples correctly. For the FlexKNN the classification performance was checked for $d_{\max} = \{0.1, 0.2, \dots, 8\}$ and the accuracies are shown in Figure 2. The blue line illustrates the ratio of test samples for which the FlexKNN yielded a correct label. Between $d_{\max} = 1.3$ and $d_{\max} = 4.2$ (vertical dotted lines) the accuracy level is higher or approximately the same than that of the KNN (dash-dotted black line), with the highest accuracy at $d_{\max} = 1.5$ (vertical solid line). However, the more important line is the red, dashed line in Figure 2. It illustrates the ratio of test sample for which the FlexKNN yielded either the correct label or returned information that no training samples within d_{\max} were found and no label could be provided. In the latter case the KNN yielded a label, but its trustworthiness was low and this resulted, in general, in a misclassification.

Data was normalized in [23] by subtracting the mean and dividing by the standard deviation of all measurements for each of the 14 dimensions.

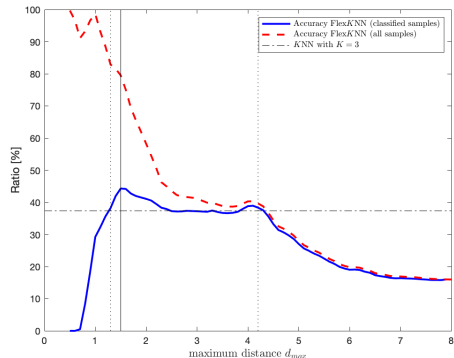


FIGURE 2. Classification accuracy of FlexKNN for varying d_{\max} . The blue, solid line shows ratio of accurately classified test samples for which FlexKNN returned a label. The red, dashed line shows the ratio of test samples that were either correctly classified or for which FlexKNN did not yield a label because no training sample was within d_{\max} . For comparison the classification accuracy of the KNN with $K = 3$ is shown (black, dashed-dotted line).

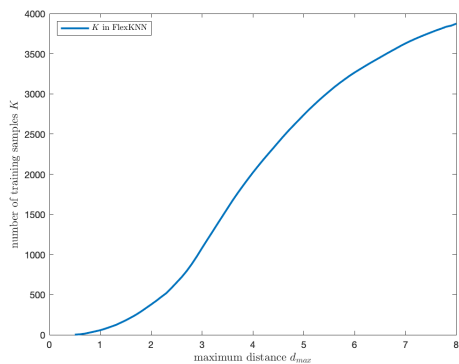


FIGURE 3. Number of training samples K inside d_{\max} for FlexKNN. Label of test sample is derived from labels of these K training samples.

As d_{\max} increases the red and blue lines converge. This indicates that the ratio of test samples that cannot be classified due to missing training samples within d_{\max} decreases. For $d_{\max} = 8$ all test samples were classified, but the classification accuracy of the FlexKNN already dropped at $d_{\max} = 2.5$ below that of the KNN. This can be, partly, explained by the fact that the number of training samples within d_{\max} steadily increase as d_{\max} increases and that it is considerably higher than the K value usually used inside the KNN. For example, at $d_{\max} = 1.5$ the average K over all 4,361 test samples was already 181.24 and at $d_{\max} = 4.2$ it was 2,176.83. Considering that for each room there were only roughly 600 training samples it is not surprising that the classification accuracy decreases once $d_{\max} > 4.2$. For $d_{\max} = 4.2$ at most $\approx 28\%$ of the training samples inside d_{\max} could be from the correct class, making a misclassification likely. To conclude, for the presented dataset choosing $d_{\max} \approx 1.5$ provides the best compromise between high accuracy of provided labels and low number of non-classified test samples.

5.3. Test with missing room data. For the data from [23] it was noted that IMS measurements differed most noticeably for training and test samples collected in rooms 6 and 7. Therefore, in this section only training samples from rooms 1 to 5 were used for determining the labels of the same 4,361 test samples as in Section 5-A, of which 1,234 were collected in rooms 6 and 7. For the KNN $K = 3$ and for the Flex KNN $d_{\max} = 1.5$.

The classification accuracy of the KNN was 71.70%, which is a clear improvement over the result in Section 5-A and supports the hypothesis that larger differences in the samples from rooms 6 and 7 collected for training and testing caused a large portion of the misclassifications. The Flex KNN yielded for 3,648 test samples a label, of which 76.75% were correct; for the remaining 713 test samples no training sample was within d_{\max} (i.e., $K=0$) and hence the Flex KNN did not return a label. The overall accuracy, which accounts for test samples being correctly classified as belonging to rooms 1 to 5 or having no label due to $K=0$, was 80.55%.

Test samples without label stem mostly from rooms 6 and 7. However, 42.22% of the samples from these two rooms were also misclassified by the Flex KNN , which indicates that either a smaller d_{\max} should be used or that some IMS fingerprints between rooms 6 and/or 7 and one of the remaining five rooms were too similar to be distinguished reliably. It also shows that choosing a suitable d_{\max} is non trivial. A thorough analysis of the training data investigating the density of training samples, the closeness of training samples from the same class as well as the closeness of training samples from different classes could help find the d_{\max} that yields the highest accuracy.

6. CONCLUDING REMARKS

This paper introduced a modification of the widely used K Nearest Neighbors classifier for which the maximum allowed distance between the sample to be classified (test sample) and training samples is used as input parameter. Hence, K is flexible and can differ considerably between different training samples. Consequently, the algorithm was named Flexible KNN (abbreviated as Flex KNN). The reasoning behind the Flex KNN is that the standard KNN and its variants will always yield a label for a test sample even if the K closest training samples are far away from the test sample. This might occur, for example, if the test sample is from a class for which no training data is available. In such scenario existing KNN -variants will yield a wrong label while the Flex KNN will provide information that no label could be determined due to all training samples being too dissimilar and that it is reasonable to assume that the test sample belongs to a yet unknown class.

In Section 5, the Flex KNN was compared to the standard KNN for localization based on ion-mobility spectrometry fingerprints. The dataset from [23] was chosen because it highlighted the limitation of the KNN , in situations where training samples were very dissimilar to the test sample, and the capability of the Flex KNN to solve or at least mitigate this limitation. The test showed that the Flex KNN can outperform the KNN for reasonable choices of d_{\max} , even when training data are available for all classes observed in the test data.

The maximum allowed distance d_{\max} between the test sample and training samples can be determined, for example, from the training data using leave-one-out cross-validation or by simply testing the Flex KNN 's performance for various values of d_{\max} to find the one yielding the highest accuracy. In future work also techniques for systematically or dynamically determining the optimal d_{\max} will be studied. Alternatively, other prior information on the data could be used. An advantage of the Flex KNN is that the number of training samples K inside d_{\max}

The KNN misclassified all samples from rooms 6 and 7 due to missing training data from these two rooms.

provides information how well the test sample fits into the existing clusters of training data. Large K and/or K closest training samples mostly from one class indicate a good fit and high confidence in the yielded label; low K and/or K closest training samples from various classes indicate a poor fit and low confidence in the yielded label. A reasonable requirement, to avoid the latter case, would be to demand that K is above a certain threshold to avoid misclassification based on some outlier data.

Several methods that have been proposed for improving existing KNN -variants have been discussed in Section 3. Part of their ideas could also be used within the $FlexKNN$. For uneven class distribution it would be advisable to calculate the ratio of training samples from a certain class inside d_{\max} [6] and either choose the label based on the class with the highest ratio or use normalized ratios of all classes to assign the test samples fuzzy memberships for all classes. In order to avoid ties the concept of the weighted KNN could be used, meaning that each of the samples inside the maximum allowed distance is given a weight that is inverse proportional to its distance to the test sample.

The usefulness of all these potential modifications to the $FlexKNN$ will be tested in the future with various datasets. Also, a thorough comparison with other KNN -variants will be carried out. This comparison will provide an insight into the pros and cons of each KNN -variant, their computational efficiency, and in which scenarios to use which variant.

ACKNOWLEDGMENT

The author thanks Anton Rauhameri for proofreading and commenting the manuscript as well as Dr. Simo Ali-Löytty for discussing K Nearest Neighbors and the proposed approach.

REFERENCES

- [1] J.L. Bentley, “A survey of techniques for fixed radius near neighbor searching,” Technical Report SLAC-186 and STAN-CS-75-513, Stanford Linear Accelerator Center, August 1975
- [2] C. Levinthal, “Molecular model-building by computer,” *Scientific American* 214, pp. 42–52. June 1966.
- [3] A. Król, W. Rzasa, and P. Grochowalski, “Aggregation of Fuzzy Equivalences in Data Exploration by kNN Classifier,” *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2020)*, July 2020.
- [4] Z. Wang, Y. L. D. Li, Z. Zhu, and W. Du, “Entropy and gravitation based dynamic radius nearest neighbor classification for imbalanced problem,” *Knowledge-Based Systems* 193 Article 105474, January 2020.
- [5] P. Grochowalski, A. Król, and W. Rzasa, “Radius kNN classifier using aggregation of fuzzy equivalences,” *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2021)*, July 2021.
- [6] A. Onyewe, A. F. Kana, F. B. Abdullahi, and A. O. Abdulsalami, “An Enhanced Adaptive k -Nearest Neighbor Classifier Using Simulated Annealing,” *I. J. Intelligent Systems and Applications*, vol. 1, pp. 34–44, February 2021.
- [7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., Wiley-Interscience, 2001.
- [8] Z. Bian, C. M. Vong, P. K. Wong, and S. Wang, “Fuzzy KNN method with adaptive nearest neighbors,” *IEEE Transactions on Cybernetics*, vol. 52 (6), pp. 5380–5393, June 2022.
- [9] P. Müller, M. Raitoharju, and R. Piché, “A field test of parametric WLAN-fingerprint-positioning methods,” *17th International Conference on Information Fusion (FUSION)*, July 2014.
- [10] M. Alkasassbeh, G. Altarawneh, and A. Hassanat, “On enhancing the performance of nearest neighbour classifiers using hassanat distance metric,” *Can. J. Pure Appl. Sci.*, vol. 9 (1), pp. 1–6, February 2015.
- [11] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, “Comparative performance analysis of K -nearest neighbour (KNN) algorithm and its different variants for disease prediction,” *Scientific Reports*, vol. 12, 6256, April 2022.
- [12] G. Minaev, P. Müller, A. Visa, and R. Piché, “Indoor localisation using aroma fingerprints: Comparing nearest neighbour classification accuracy using different distance measures,” *7th International Conference on Systems and Control (ICSC’18)*, October 2018.

- [13] D. Wettschereck and T. G. Dietterich, "Locally adaptive nearest neighbor algorithms," *Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS'93)*, pp. 184–191, November 1993.
- [14] Z. Pan, Y. Wang, and Y. Pan, "A new locally adaptive k-nearest neighbor algorithm based on discrimination class," *Knowledge-based Systems*, vol. 204, 106185, September 2020.
- [15] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat, "Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach," *International Journal of Computer Science and Information Security*, Vol. 12, No. 8, August 2014.
- [16] J. Wang, P. Neskovic, and L. N. Cooper, "Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence," *Pattern Recognition*, vol. 39 (3), pp. 417–423, March 2006.
- [17] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, 2014. "kNN algorithm with data-driven k value," *Proceedings of the 10th International Conference on Advanced Data Mining and Applications (ADMA 2014)*, vol. 8933, pp. 499–512, 2014.
- [18] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29 (5), pp. 1774–1785, February 2017.
- [19] R. Montoliu, A. Pérez-Navarro, and J. Torres-Sospedra, "Efficient tuning of kNN hyperparameters for indoor positioning with N-TBEA," *2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, October 2022.
- [20] S. M. Lucas, J. Liu, and D. Perez-Liebana, "The N-Tuple Bandit Evolutionary Algorithm for Game Agent Optimisation", *2018 IEEE Congress on Evolutionary Computation (CEC'18)*, July 2018.
- [21] L. Jiang, Z. Chi, D. Wang, and S. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, pp. 679–683, August 2007.
- [22] M. M. Deza and E. Deza, *Encyclopaedia of Distances*, Springer, 2009.
- [23] P. Müller, S. Ali-Löytty, J. Lekkala, R. Piché, "Indoor Localisation using Aroma Fingerprints: A First Sniff," *14th Workshop in Positioning, Navigation and Communication (WPNC'17)*, October 2017.
- [24] G. A. Eiceman, Z. Karpas, and H. H. Hill Jr., *Ion Mobility Spectrometry*, 3rd ed., CRC Press, 2014.