Learning Vector Quantization with Local Mean Based to Determine K Value in the K-Nearest Neighbor Method

M A Munir[1], E B Nababan[2], Tulus[3]
{oreym29@gmail.com[1], ernabrn@usu.ac.id[2], tulus_ip@yahoo.com[3]}

Graduate School of Computer Science, Faculty of Computer Science and Information Technology Universitas Sumatera Utara, Medan, Indonesia [1]
Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia[2]
Department of Mathematics, Universitas Sumatera Utara, Medan, Indonesia [3]

Abstract: classification is a process that explains and functions to distinguish data classes or concepts that aim to be able to predictions in classes of objects unknown to the label class. Many popular classification techniques, one of which is K-Nearest Neighbor (KNN). The K-NN algorithm functions to find the closest k neighbors and use the majority class. This study aims to determine the best k value by using Learning Vector Quantization as weight weights. Determination of the Local Mean Based test data class K-Nearest Neighbor uses the measurement of the closest distance to each local model of each data class. In processing Learning Vector Quantization, Cross-Validation and Local K-Fold in the K-Nearest Neighbor classification the lowest k = 4 was 72%, while the highest k value was 9 = 80%. And the highest k value is a good K value that is k = 9 for Iris Data.

Keywords: *Data Iris, Learning Vector Quantization, K-Nearest Neighbour, Local Mean K-Nearest Neighbour, K-Fold Cross-Validation*

1 Introduction

Classification is one of the tasks of data mining that aims to predict the labels of categories of objects that were not previously known, in distinguishing between objects with one another. based on attributes or features [1]. Saputra [2] say that the classification method is a process that explains and functions to distinguish data classes or concepts that aim to be able to predict in classes of objects unknown to the label class. Classification is part of data mining, where data mining is a term used to explain the discovery of knowledge in data. Data mining is also a process that uses statistical techniques, mathematics, artificial intelligence and machine learning to extract and recognize useful information and relevant knowledge from various large databases.

K-Nearset Neigbhour (KNN) is a method that uses a supervised algorithm. The difference between supervised learning and unsupervised learning is that supervised learning aims to find new patterns in data by connecting existing data patterns with new data. Whereas in unsupervised learning, data does not yet have any pattern, and the purpose of unsupervised learning is to find patterns in data. The purpose of the k-NN algorithm is to classify new objects based on attributes and training samples [1]. Where the results of the new test samples are classified based on the majority of the categories on k-NN. In the classification process,

ICMASES 2019, August 03-04, Samarinda, Indonesia

DOI 10.4108/eai.3-8-2019.2290750

this algorithm does not use any model to match and is only based on memory. The k-NN algorithm uses neighboring classification as the predictive value of the new test sample. The distance used is the Euclidean Distance. Euclidean distance is the most commonly used distance for numerical data. Euclidean distance is defined as follows:

$$d(x_i,x_j) = \sqrt{\sum_{r=1}^{n} (ar(xi) - ar(xj))^2} \quad (1)$$

Information :
d(xi,xj) : (Euclidean Distance).
(xi) : record ke- i
(xj) : record ke- j
(a_r) : data ke-r
i,j :1,2,3,…n

To classify a new class k-NN looks for k neighbors that are closest and use the majority class. To do this, first, the closest neighbor is identified first [3]. Where x1, x2, ... xn are predictors for instances 1 and u1, u2, ..., un are predictors for instance 2. The K-nearest neighbor (KNN) algorithm is a supervised learning algorithm in which the results of calcification of new data are based on the category of the closest K neighbor. The purpose of this algorithm is to classify new objects based on attributes and training data. Classification is done without using a model but only based on memory. Suppose that given a query, it will get a number of K training data objects closest to the query.

Classification is done by using the majority of votes (such as in elections) between classifications of K objects. The KNN algorithm uses the classification of security as a prediction of new data. This algorithm works based on the minimum distance from the new data to the nearest K neighbor that has been determined. After obtaining the nearest K neighbor, the prediction of the class from the new data will be determined based on the majority of the nearest K neighbors. The steps in classification in the KNN Algorithm:
1. Determine the parameter K = the number of closest neighbors.
2. Calculate the distance between the new data and all the training data.
3. Sort the distance and set the nearest neighbor based on the minimum distance K.
4. Check the class from the nearest neighbor.
5. Use a simple majority of the closest neighbor class as the new data predictive value.

The use of Local Mean is proven to improve performance and also reduce the influence of outliers on the K-NN method, especially for small amounts of data [4].

The K value on the LMKNN is very different from that of K-NN, on the LMKNN the value of K is the number of closest neighbors of each data class, while in K-NN the value of K is the number of closest neighbors of all data. LMKNN is equal to 1-NN if the K value is 1 [4].

Learning Vector Quantization (LVQ) is a method for classifying (grouping) patterns and having output representing a particular class. LVQ neural network architecture is basically the same as Kohonen's Self Organizing Map (without a topological structure assumed for output). Its architecture consists of input layers, competitive layers (hidden layers), and output layers. The competitive layer will learn automatically to classify the input vectors given. If several input vectors have very close distances, then the input vectors will be grouped in the same class [5].

The steps in the Lvq algorithm are as follows:
Step 0: Initialization Weight
Step 1: If the stop condition fails, do steps 2-8
Step 2: For each input vector Xi, do steps 3 to 6
Step 3: For each j, count:

$$\sqrt{\sum_{j=1}^{m}(W_{\text{ji}} - \text{X}_{\text{i}})^2} \qquad (2)$$

Step 4: Find the index j so that D (j) is minimum
Step 5: Check j index and compare it with class information
Step 6: For each j

- Update its weight if index = Class information
  Wji (New) = Old Wji + (Xi - Wji (old)) (3)
- Update its weight if the index ≠ class information
  Wji (New) = old Wji - (Xi - Wji (lama)) (4)
- By is the rate of understanding / learning rate (used 0.1)

Step 7: Modify the comprehension rate (used 0.5)
Step 8: Check the stop condition [5].

2 Method

This study aims to optimize good k values, can improve classification performance and also reduce outliers, especially in smaller size data in the k-Nearest Neighbor classification process. By using the Learning Vector Quantization to manage data measurement processes based on dataset criteria from the data mining UCI Repository as training data and tests in classification.[6]

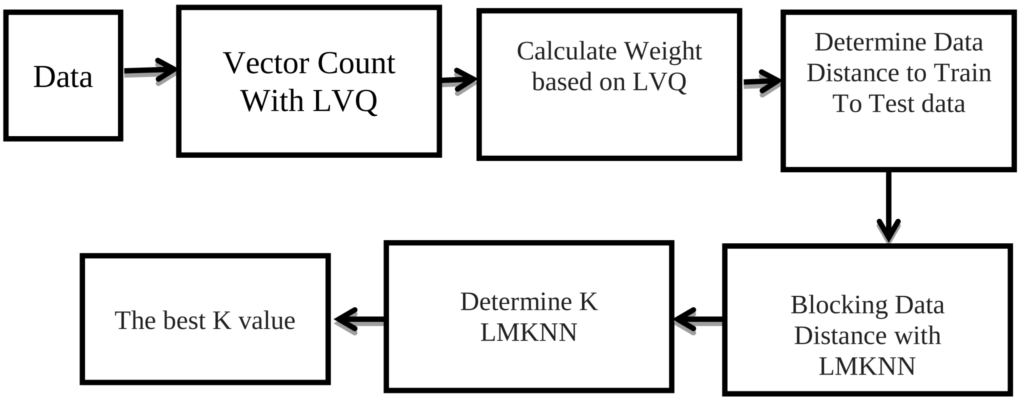

Data
Vector Count With LVQ
Calculate Weight based on LVQ
Determine Data Distance to Train To Test data
Blocking Data Distance with LMKNN
Determine K LMKNN
The best K value

Figure 1. Work System Design

3 Result and Discussion

From the process described in the previous sub-chapter at this stage it goes into implementation. The output of the test from k value = 1 - 50 on UCI data uses Learning Vector Quantization, K-fold Cross-Validation and Local Mean base K-Nearset Neighbor.[7]

Testing is done using some data obtained from the UCI Repository, Iris data. At the time of testing, as much as 80% of the data is used as training data and as much as 20% of the data is used as test data carried out randomly. This study aims to determine the appropriate k value in the K-Nearest Neighbor value classification process to get good accuracy or high accuracy.[8]

No	Category	class			Total Data
		Setosa	Versicolor	Virginica	
1	Training Data	40	40	40	120
2	Test Data	10	10	10	30
Total		50	50	50	150

Tabel 1 Distribusi Data *Iris*

Vector count results using Learning Vector Quantization (LVQ) as a measurement tool for correlations from data sets, where the Learning Vector Quantization will be used as a weighting basis.[9]

3	2	3	2	3	3	2	2	3	3	3	3	3	2	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

3	3	3	2	3	3	3	2	3	3	2	3	3	3	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Table 2 Results of new weights generated from LVQ

Then the distance between training data and test data is calculated using the Euclaudien distance model. The distance generated can be seen in table 3[10]. Furthermore, sorting the distance between the data is done ascending, while the order of the closest distance between the data can be seen in table 4[11].

Data Test	Training Data						
	L1	L2	L3	L4	L5	…	L281
U1	3.990	3.023	4.607	2.860	4.117	…	0.173
U2	1.924	2.775	1.304	3.295	1.857	…	5.599
U3	0.794	0.332	1.338	1.100	0.837	…	3.314
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
U30	4.052	3.130	4.650	3.040	4.204	…	0.520

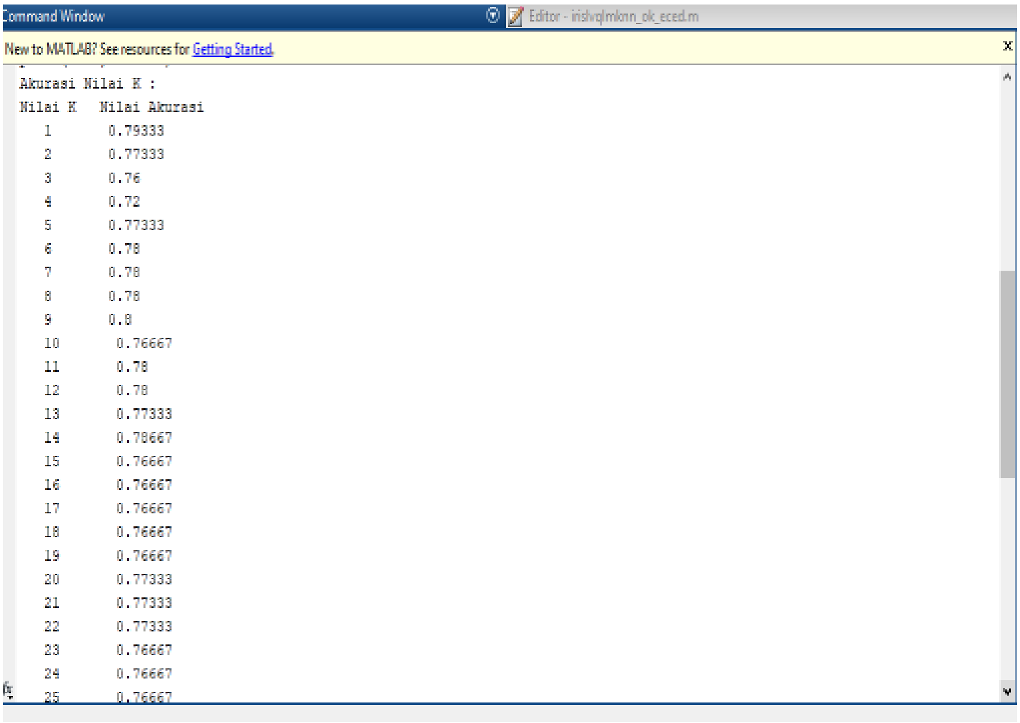
Table 3 Distance Between Data on the Iris data

Data Test	The closest distance order						
	1st	2nd	3rd	4th	5th	…	120th
U1	L75	L120	L69	L81	L37	…	L97
U2	L57	L47	L45	L82	L98	…	L10
U3	L2	L83	L39	L118	L58	…	L10
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
U30	L46	L117	L71	L34	L51	…	L97

Table 4 Sequence of the Nearest Distance on the Iris dataset

The process for the first stage is to determine the parameters and then calculate the vector using LVQ as a measure for the correlation of the data set, where the LQV will be used as the basis for weighting. [12]

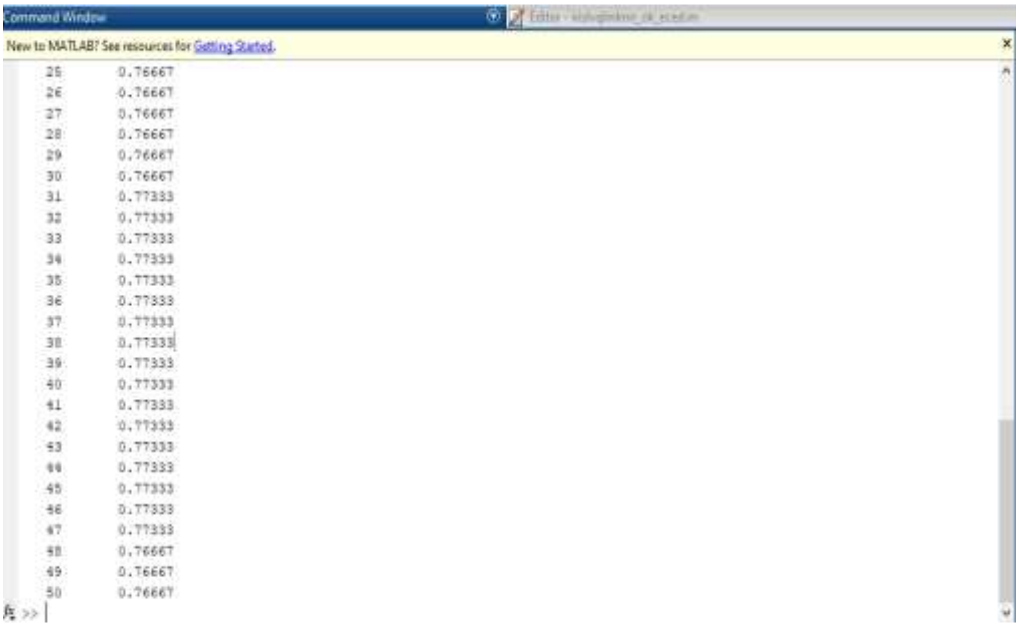
Seto Aji [13]The Local Mean and LVQ processes aim to calculate the average value of weights from the distance of the nearest neighbor to each data class and make the highest weighting average as the k value of the test data class and from k = 1-50. After obtaining a new weighting, it will be calculated by using LVQ and LMKNN. The results of LVQ + LMKNN with the Euclidean distance model from the test on the Iris dataset can be seen in Figure 2 and 3.



Command Window
Editor - irislvqlmknn_ok_eced.m
New to MATLAB? See resources for Getting Started.

```
Akurasi Nilai K :
Nilai K   Nilai Akurasi
   1       0.79333
   2       0.77333
   3       0.76
   4       0.72
   5       0.77333
   6       0.78
   7       0.78
   8       0.78
   9       0.8
   10       0.76667
   11       0.78
   12       0.78
   13       0.77333
   14       0.78667
   15       0.76667
   16       0.76667
   17       0.76667
   18       0.76667
   19       0.76667
   20       0.77333
   21       0.77333
   22       0.77333
   23       0.76667
   24       0.76667
   25       0.76667
```

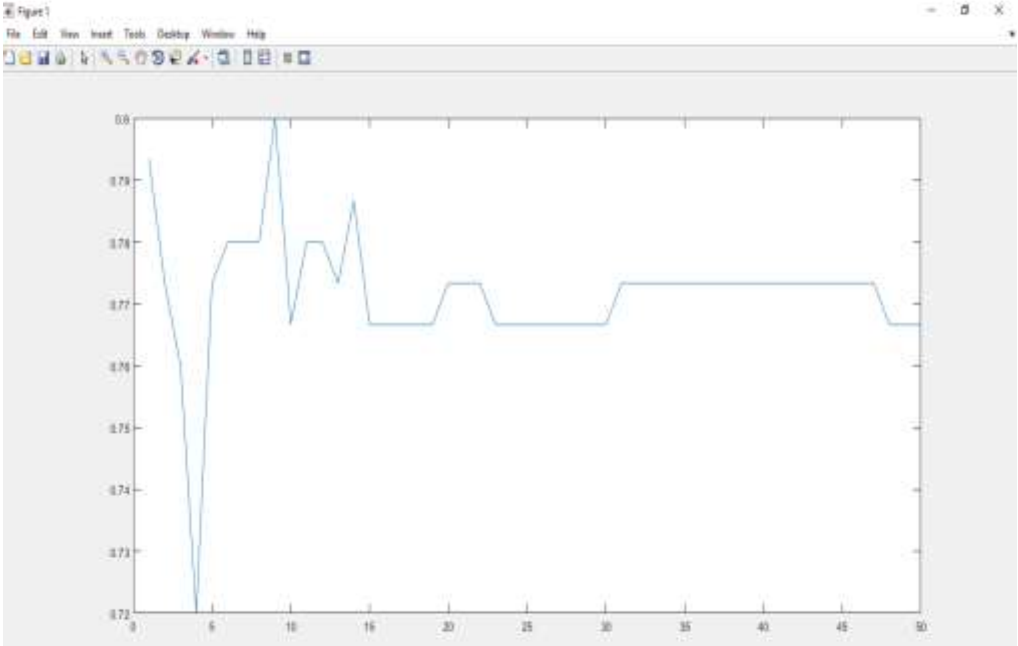
Figure 2 Output of LVQ Testing on Iris dataset



```
    25        0.76667
    26        0.76667
    27        0.76667
    28        0.76667
    29        0.76667
    30        0.76667
    31        0.77333
    32        0.77333
    33        0.77333
    34        0.77333
    35        0.77333
    36        0.77333
    37        0.77333
    38        0.77333
    39        0.77333
    40        0.77333
    41        0.77333
    42        0.77333
    43        0.77333
    44        0.77333
    45        0.77333
    46        0.77333
    47        0.77333
    48        0.76667
    49        0.76667
    50        0.76667
fx >>
```

Figure 3 Output of LVQ Testing on Iris dataset

The results of Plot k = 1-50 from the testing of the Iris dataset can be seen in Figure 4



Can be obtained from the results of Figure 2 and Figure 3 and Results Graph Figure 4 in the K-Nearest Niegbor classification uses the parameter k for 1 to 50 using Learning Vector Quantization with Local Mean K-Nearest Neighbor to determine a good k value. Then can be obtained in this process the most accurate value of the combined LVQ and LMKNN is very consistent the highest value is only one, namely the value at k = 9 is 0.8. The lowest accuracy value is located at k = 4 of 0.72 resulting from a combination of LVQ and LMKNN.

4 Conclusion

After discussing in the previous chapters, conclusions can be obtained by looking at Figure 2, Figure 3 and graphical images 4 by processing Learning Vector Quantization with Local Maen to determine the value of K in K-Nearest Neighbor. The lowest k value in this study is at k = 5 by 72%, while the highest k value for k = 9 is 80% and a good k value for iris data is k = 9.

References

[1] S. Mutrofin, A. Izzah, A. Kurniawardhani, and M. Masrur, “Optimasi Teknik Klasifikasi Modified K Nearest Neighbor Menggunakan Algoritma Genetika,” *J. Gamma*, vol. 10, no. 1, 2015.

[2] M. E. Saputra, H. Mawengkang, and E. B. Nababan, “Gini Index With Local Mean Based For Determining K Value In K-Nearest Neighbor Classification,” *J. Phys. Conf. Ser.*, vol. 1235, p. 12006, Jun. 2019.

[3] B. M. Susanto, “K-Nearst Neigbour (Knn) Untuk Mendeteksi Gangguan Jaringan Komputer Pada Intrusion Detection Dataset,” *J. Khatulistiwa Inform.*, vol. 2, no. 1, 2014.

[4] J. T. Hardinata, M. Zarlis, E. B. Nababan, D. Hartama, and R. W. Sembiring, “Modification Of Learning Rate With Lvq Model Improvement In Learning Backpropagation,” *J. Phys. Conf. Ser.*, vol. 930, p. 12025, Dec. 2017.

[5] E. Prasetyo, “Fuzzy K-Nearest Neighbor In Every Class Untuk Klasifikasi Data,” in *Seminar Nasional Teknik Informatika*, 2012, pp. 57–60.

[6] A. Danades, D. Pratama, D. Anggraini, and D. Anggriani, “Comparison of accuracy level K-Nearest Neighbor algorithm and Support Vector Machine algorithm in classification water quality status,” in *2016 6th International Conference on System Engineering and Technology (ICSET)*, 2016, vol. 137–144, pp. 137–141.

[7] I. Afrianto, “Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah,” *KOMPUTA J. Komput. dan Inform.*, vol. 1, no. 1, 2012.

[8] F. S. Ni’mah, T. Sutojo, and D. R. I. M. Setiadi, “Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-occurence Matrix dan K-Nearest Neighbor,” *J. Teknol. dan Sist. Komput.*, vol. 6, no. 2, p. 51, Mar. 2018.

[9] M. Nasir and M. Syahroni, “Pengujian Kualitas Citra Sidik Jari Kotor Menggunakan Learning Vector Quantization (Lvq),” *J. Litek*, vol. 9, no. 1, pp. 65–69, 2012.

[10] S. K. Lidya, O. S. Sitompul, and S. Efendi, “Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine (SVM) Dan K-Nearest Neighbor (K-NN),” in *Seminar Nasional Teknologi Informasi dan Komunikasi*, 2015.

[11] W. E. Nurjanah, R. S. Perdana, and M. A. Fauzi, “Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode K-Nearest Neighbor dan Pembobotan Jumlah Retweet,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1750–1757, 2017.

[12] M. Azmi, “Komparasi Metode Jaringan Syaraf Tiruan Som Dan Lvq Untuk Mengidentifikasi Data Bunga Iris,” *J. TeknoIf*, vol. 2, no. 1, 2014.

[13] A. S. Arifianto, M. Sarosa, and O. Setyawati, “Klasifikasi Stroke Berdasarkan Kelainan Patologis dengan Learning Vector Quantization,” *J. EECCIS*, vol. 8, no. 2, pp. 117–122, 2014.