Computer and Information Science; Vol. 7, No. 1; 2014
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

3N-Q: Natural Nearest Neighbor with Quality

Shu Zhang[1], Malek Mouhoub[1] & Samira Sadaoui[1]

[1] Department of Computer Science, University of Regina, Regina, Canada

Correspondence: Malek Mouhoub, Department of Computer Science, University of Regina, Regina, Canada. E-mail: mouhoubm@cs.uregina.ca

Received: September 25, 2013 Accepted: November 17, 2013 Online Published: January 13, 2014

doi:10.5539/cis.v7n1p94 URL: http://dx.doi.org/10.5539/cis.v7n1p94

Abstract

In this paper, a novel algorithm for enhancing the performance of classification is proposed. This new method provides rich information for clustering and outlier detection. We call it Natural Nearest Neighbor with Quality (3N-Q). Comparing to K-nearest neighbor and E-nearest neighbor, 3N-Q employs a completely different concept to find the nearest neighbors passively, which can adaptively and automatically get the K value. This value as well as distribution of neighbors and frequency of being neighbors of others offer precious foundation not only in classification but also in clustering and outlier detection. Subsequently, we propose a fitness function that reflects the quality of each training sample, retaining the good ones while eliminating the bad ones according to the quality threshold. From the experiment results we report in this paper, it is observed that 3N-Q is efficient and accurate for solving data mining problems.

Keywords: KNN, 3N-Q, nearest neighbor, classification, clustering, outlier, quality

1. Introduction

K-Nearest Neighbor (KNN) (Han & Kamber, 2006) is one of the most popular algorithms in knowledge discovery. It is a supervised learning method where a new instance is classified based on majority of K-nearest neighbor category. The purpose of the algorithm is to classify the new object based on attributes and training samples. However, due to its limitations in large data sets, such as low recognition rate, high computation complexity and no valuation of training samples, an increasing number of modifications have been proposed in recent years (Shi, Li, Liu, He, Zhang, & Song, 2011) to improve the efficiency of KNN, including the best well-known weighted-KNN (Liu & Chaw, 2011). However, several major problems are still unsolved, including the randomness of K value, outlier elimination, excessive computing time and data size reduction. Thus the need for a new method to solve these deficiencies is obvious and our 3N-Q serves this purpose very well.

Our proposed method is divided into two parts. The first one is the proposed concept of 3N and its implementation, which is inspired by the idea of obtaining K dynamically (Gong & Liu, 2011). Our algorithm can get the value of K automatically for different problems when each point in space becomes at least one neighbor of another point. The K value reflects the distribution of data set. The smaller, the sparser, the bigger, the denser. We conducted several experiments to evaluate the stability and accuracy of this proposed algorithm. The first experiment shows the stability of K. Two other variables record the first to Kth neighbors of each training sample and the frequency of being other neighbors. It provides the density and outlier information for us, which is also the evidence of clustering and outlier detection. The second experiment uses the fitness function to show the quality of training samples. It is deemed that a point owns a same class with its most neighbors possessing a high energy and visa versa. We report that the result of outliers and boundaries will have low quality value here, which would be eliminated.

This paper is organized as follows. Section 2 introduces the background for the regular KNN algorithm. Section 3 discuses the related work to this study. Section 4 describes the proposed method in detail. Three experiments conducted to show the stability of the new algorithm, clustering ability and classification accuracy are reported in Section 5. Finally conclusion and future work are listed in Section 6.

2. Background

In this section, we introduce the background as well as principle of the regular KNN. KNN has gained much popularity due to its non-parametric and easy-to-implement properties. Basically, it classifies an unknown instance by its closest neighbors in the training sets. K represents the number of the closet neighbors we select. Figure 1 from (Commons, 2007) illustrates the core idea of this algorithm. The green circle is called unclassified sample and our task is to determine whether it belongs to the blue squares class or the red triangles one. The first step is to calculate the distance between the unclassified instance and each training set. Obviously, if we take its three neighbors into account, the unclassified instance should belong to the class of red triangles. Indeed, among these three neighbors, two are red triangles, which leads to a majority votes. On the contrary, it would be classified into the blue squares class if we take more neighbors into consideration, such as 5. In this latter situation, the three squares hold a majority vote among the total 5. Hence, we can see that the classification accuracy is highly and directly dependent on the number K of the neighbors we consider.

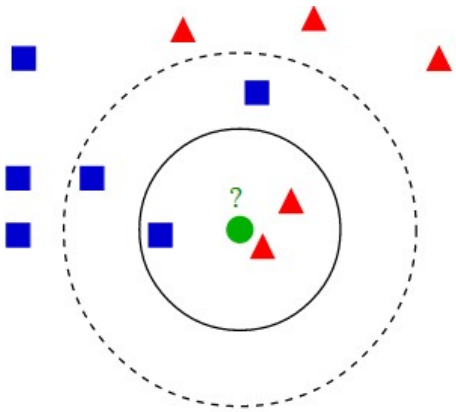

Figure 1. Basic K-nearest neighbor algorithm

3. Related Work

Data mining aims at discovering valuable information behind large data sets in a wide variety of areas including artificial intelligence, machine learning, statistics, and database systems (Fayyad, Piatetsky-shapiro, & Smyth, 1996; Hastie, Tibshirani, & Friedman, 2009). Recently, data mining gains much popularity, resulting from the enormous market and research demand in science and engineering areas, such as bioinformatics, genetics, medicine, education and electrical power engineering (Witten, Frank, & Hall, 2011). Classification, clustering and outlier detection are three major tasks in data mining because of their various applications, including detecting credit card fraud, analyzing the structures of protein, categorizing news as finance, politics, education or sports. Apparently, K-Nearest Neighbor algorithm (Mitchell, 1997) is amongst the simplest of all machine learning algorithms: an object is voted by its K neighbors and classified by a majority votes of these K neighbors. The combination of artificial intelligence algorithms and basic KNN has shown remarkable success in wide areas. In 1991, Kelly and Davis (1991) proposed a hybrid method employing genetic algorithm and KNN to evaluate the weight by individual attributes in the data set. In 2001, Li and Weinberg (2001) proposed a GA/KNN method, which utilizes a subset of predictive genes jointly for classification with a consideration of the expression data. In 2005, Peterson and Doom (2001) provided a GA-facilitated KNN classifier to assign the weight for each attribute and apply different measurements, Euclidian distance, cosine and person similarity which takes k-neighbors straightaway instead of considering all the training samples. In 2010, Suguna and Thanushkodi (2010) also put forward an improved KNN classification using genetic algorithms. In 2012, Yang (2012) presented an integrated genetic algorithm and conditional probability to enhance the performance of the standard KNN. A drawback of the basic 'majority voting' classification occurs when the class distribution is irregular and complex (Mitchell, 1997). One way to overcome this issue is to weigh the samples, taking the distance between the test points and each of its k nearest neighbors into account, we call it weighted-KNN. In 2011, Liu and Chawla (2011) came up with a class confidence weighted-KNN to tackle imbalance data set, which in one of the most complex distribution. However, the inherent bias of majority class can be correct in existing KNN algorithms with any distance measurements. In 2012, Lican and Liu (2012) proposed a parallel GPU-based KNN algorithm to solve the computational complexity in practical applications. In 2013,

Amato and Falchi (2013) addressed the image recognition issue by combining the KNN and local feature search method with similarity functions.

4. Proposed Method

This section introduces the proposed method in detail. The first purpose of this method is to solve some puzzling problems in classification, such as the uncertainty of neighbors' number, interference of noise points, irregular distribution of data sets, etc. Besides, providing precious information for clustering and outlier detection is another purpose. In the following we will first introduce our proposed algorithm and provide the details of the quality function used to pre process the training data. We will then present the threshold setting to eliminate low quality data set and provide the description of the classification we have used.

4.1 3N Algorithm

The main idea of the algorithm is described as follows. Our purpose is to determine the number of neighbors that should be considered in the classification step. We find the first to rth neighbors of each training sample until the terminal condition is reached, that is when every training sample has been other neighbors at least one time. The algorithm can be noted with the following mathematical expression.

$$A(r, x, y) = ((\exists r)(\forall x \in X)(\forall y \in X) \\ (x \neq y \rightarrow x \in Nr(y)) \tag{1}$$

Here, $K = r$ and $Nr(y)$ is the set of r-nearest neighbors of y. r is the considered number of neighbors, x and y are one of the training sample from the set X. Nr(y) records the first to rth neighbors of y. The value of r begins from 1 and ends when $A(r, x, y)$ is true. It is apparent that K can be obtained adaptively instead of subjectivity or randomly. An example is as follows. We randomly choose 9 data from the famous iris data set on UCI (Blake & Merz, 1998), three for each class. Each sample has four attributes and there are three categories (classes) of plants. Table 1 lists these 9 data.

Table 1. 9 random data from iris

Sample No	1	2	3	4	5
1	4.9	3.0	1.4	0.2	1
2	4.6	3.4	1.4	0.3	1
3	5.7	4.4	1.5	0.4	1
4	6.1	2.9	4.7	1.4	2
5	6.2	2.2	4.5	1.5	2
6	5.7	2.6	3.5	1	2
7	7.7	2.6	6.9	2.3	3
8	6.3	2.7	4.9	1.8	3
9	7.2	3	5.8	1.6	3

We simply use N dimensions Euclidean distance (M. Deza & E. Deza, 2013) to calculate the distance between the training samples. The results are shown in Table 2.

Table 2. Distance between each training sample

Sample No	1	2	3	4	5	6	7	8	9
1	0	5.1	16.2	37.1	36.9	24.1	65.3	41.0	51.5
2	5.1	0	14.9	38.2	38.7	25.9	66.7	42.2	52.8
3	16.2	14.9	0	36.9	39.1	27.5	63.2	40.9	49.1
4	37.1	38.2	36.9	0	7.4	13.6	28.8	5.2	15.7
5	36.9	38.7	39.1	7.4	0	12.8	29.6	7.1	18.2
6	24.1	25.9	27.5	13.6	12.8	0	41.5	17.2	28.3
7	65.3	66.7	63.2	28.8	29.6	41.5	0	24.9	14.5
8	41.0	42.2	40.9	5.2	7.1	17.2	24.9	0	13.2
9	51.5	52.8	49.1	15.7	18.2	28.3	14.5	13.2	0

According to Table 2, we can find each one's r neighbors. Neighbors are determined here by the distance. For instance, samples 2 and 3 are sample first and second neighbors respectively.

Table 3. r-nearest neighbor

Sample No.	1	2	3
1	2	3	6
2	1	3	6
3	2	1	6
4	8	5	6
5	8	4	6
6	5	4	8
7	9	8	4
8	4	5	9
9	8	7	4

From Table 3, we can see that after finding each sample's two neighbors, sample 6 is still not a neighbor of others. Hence, we continue to find everyone's third neighbor until sample 6 is one of other sample's neighbor.

Using formula (1) we propose our 3N algorithm as shown in Algorithm 1.

Algorithm 1: *NatrualNearestNeighborAlgorithm*(3*N*)

nb: %records the frequency of *i* being the neighbor of others.
nn:%records the 1*st* neighbor to *rth* neighbor of *i*
$DataSize = n \times m$;
$r = 0, flag = 0$; % r is the number of neighbors
$\forall i \in X, nb(i) = 0, nn = zeros(n, n)$; % X: data set of training sample
% termination condition is when every sample i being others' neighbor at least once
while $flag == 0$ **do**
  $r = r + 1$;
  for $\forall i \in X$ **do**
    computing the r nearest neighbor of sample i,
    recorded in $nn(i, r)$;
  end
  if *all* $(nb(i) \neq 0)$ **then**
    $flag == 1$
  end
end
$K = r$; % Because of other possible uses of r, here we assign the value of r to K Output: r,$nn(n, n)$,$nb(1, n)$

In this algorithm we start looking for each one's first neighbor. If the condition in the if statement is not true, we increment r by one and continue to find each one's second neighbor. The process is repeated until the condition in the if statement is true. The worst situation is when a point deviates from all the other data points and is not the $(n-2)th$ nearest neighbor. In this case, r reaches its peak point $n-1$ (since every point has $n-1$ neighbors). In this case the time complexity is $O(n \times r)$. Additionally, if r is very large, we may consider to deal with the outliers or noise points in the training sample firstly.

4.2 Quality Evaluation

Inspired by the neighbor concept (Parvin, Alizadeh, & Minaei-Bidoli, 2008), we suppose that the quality of a sample is determined by its neighbors. Hence, we provide the following formula to compute the quality. In 3N-Q algorithm, every training sample must first be qualified according to its r neighbors.

$$Quality(X) = \frac{1}{K} \sum_{i=1}^{K} S\left(C(x), Ni(x)\right) \tag{2}$$

K: Number of nearest neighbors ($K = r$).

$C(x)$: Class value of x.

$N_i(x)$: Class value of x' *ith* nearest neighbor.
The function S is defined as follows.

$$S(a,b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \tag{3}$$

This means that if samples a and b belong to the same class, $S = 1$, otherwise $S = 0$.

4.3 Threshold Setting and Classification

Here we take "the major vote" to set the threshold. If more than half of the neighbors belong to the same class with this point, then we consider it a positive energy in classification. In contrast, points with low energy will most likely have a negative impact on the the final result. This is similar to "*The fittest survives the natural selection*" (Oliveira, 2012), the one with low energy is considered isolated from peers and cannot be well adapted to the environment. Consequently, it will be eliminated. In general, we will keep the samples whose quality is greater than 0.5 and eliminate the samples whose quality is equal or lower than 0.5. After the data reduction, the output of the new training sample consists of high quality samples.

The class of new point is determined by its r retaining neighbors. There is no weight valuation here as we assume that the points retained are with high quality, each of them has the ability to vote correctly. If various classes have same votes, then the class is determined by its first neighbor. Just like in our real life, the one who knows us best is our best friend.

Compared to the weight method, Q is likely to predict without being influenced by noise and outliers, since they have a lower Q value according the formula above. As a result, they would be removed by the threshold setting. For example, if there is a red triangle amongst red circles in Figure 2, this noise will be removed by the threshold since the vast majority of its neighbors are from the other class. While the red circle at the bottom of the figure is called outlier, deviating from other instances will also be removed with the low quality value. Also, Q can speed up the time to get the classification result as it removes the instances from training sets, thus decreasing the scale of data.

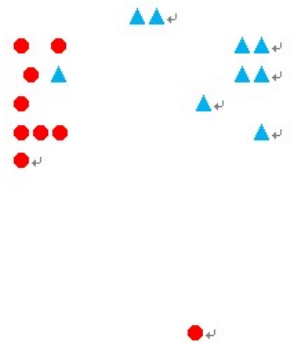

Figure 2. Example of removing noisy points and outliers

5. Experimentation

In this section we report the results of the experiments showing the stability, accuracy and potential ability of the proposed algorithm. These results are divided in three parts. The first part shows the stability of the proposed algorithm. The second part shows the concept and potential ability for clustering and the last part indicates the high efficiency in classification of 3N-Q. All the experiments were conducted on Matlab 2012a, Windows XP system with Intel Core 2 Duo CPU T6400 2.00 GHz.

5.1 Stability of r

Datasets are produced by the $rand(n, 2)$ matlab function generating two dimensional data (where n is ranging from 500 to 4500). The purpose is to prove the stability of 3N algorithm for different data size. 5 tests are taken in for each group, and the average r is obtained for different sample size. Table 4 reporting the results, indicates that r

is very stable in different groups (500, 1000, . . . , 4500), mostly ranging from 5 to 7. The average r grows slightly with the increase of data points.

Table 4. Value of r in different sample size

1	1	2	3	4	5	Average
500	7	5	5	5	5	5.4
1000	6	8	7	5	7	6.6
1500	8	7	6	6	6	6.6
2000	7	6	6	6	6	6.2
2500	6	7	7	7	6	6.6
3000	6	6	7	8	6	6.6
3500	7	6	7	7	6	6.6
4000	8	6	7	7	7	7
4500	7	7	6	7	7	6.8

5.2 Natural Neighbor Nearest Graph: 3NG

Algorithm 1 shows it is a natural process to get the 3N as there is a termination condition corresponding to the convergence of the algorithm. We have used matlab 2012a to randomly produce 500 and 1000 points respectively. When connecting r neighbors of every point, we can obtain an adaptive neighbor graph as shown in Figure 3. We assume that r is equal to 6.

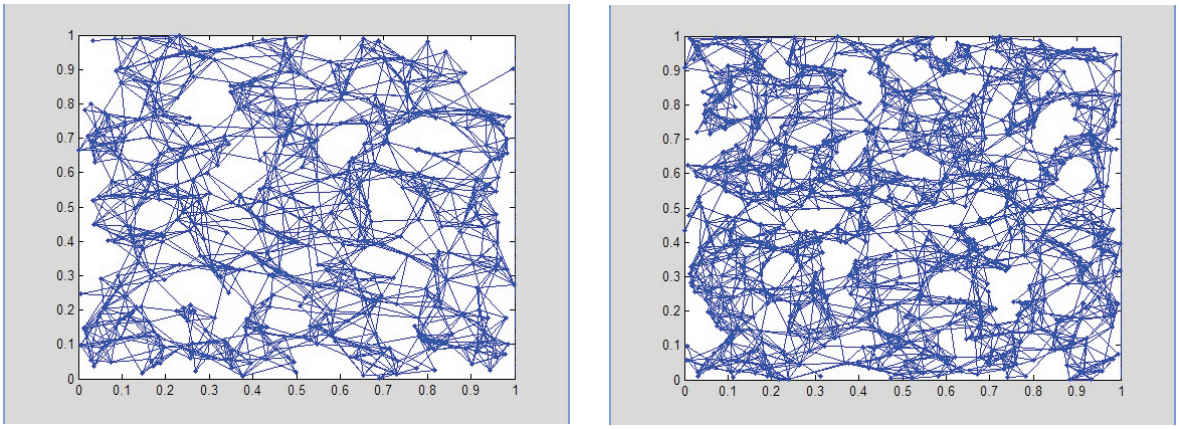

Figure 3. 3N graph respectively for 500 and 1000 random points

The effectiveness of the proposed algorithm is experimented with both artificial (Shi, 2013) and real datasets. First, the artificial dataset demonstrates that there is a potential clustering ability of 3N. The original dataset (left picture in Figure 3) shows tthere is one cluster outside and one inside in addition to two clusters on the right side which totals to four clusters. The corresponding 3N graph (right picture in Figure 4) shows exactly the same distribution with the original dataset. In other words, we can get the data set information, including clusters numbers, distribution, etc. This is important for us in the pre-processing step as it allows us to choose the appropriate type of method in the next step.

5.3 Classification Result

This section discusses the experimental methods, results and comparison between the ordinary KNN, 3N, 3N-W and 3NQ. The Weight (W) is defined as follows (Parvin, Alizadeh, & Minaei-Bidoli, 2008).

$$W = 1/(0.5 + d) \tag{4}$$

where d is the distance between the training and the test samples.

This proposed 3N-Q algorithm is evaluated on three data sets (Blake & Merz, 1998), namely, Iris, Seeds and Breast. All these sets are obtained from UCI repository. The results are respectively reported in Tables 5, 6 and

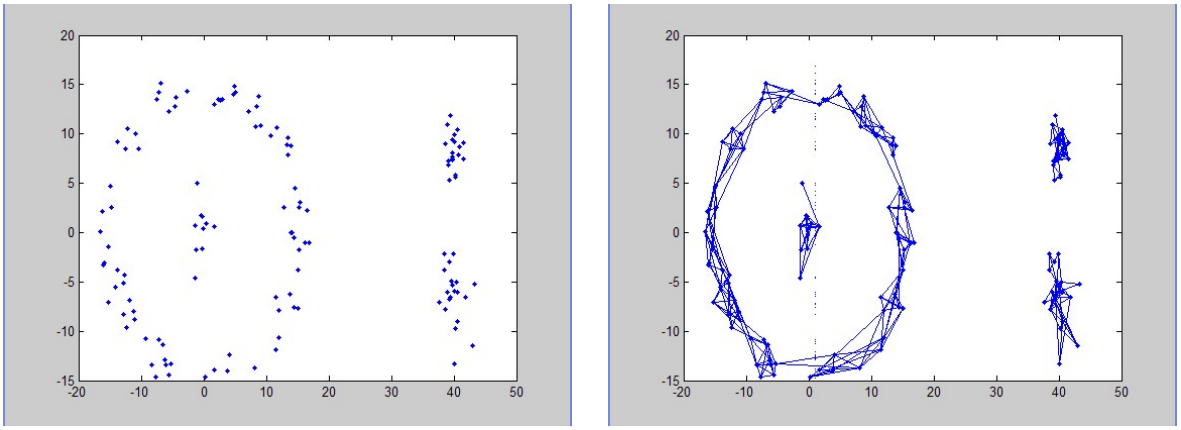

Figure 4. Original artificial points and the corresponding 3N graph

7. We randomly divide each data set into two parts: 90 percent for training set and 10 percent for test set. All the experiments are evaluated over 100 independent runs and the average results are reported. In Table 5, the total data set of Iris is 150 and 4 attributes are used for each data. In Table 6, the total data set of Seeds is 210 and 7 attributes are used for each data. In Table 7, the total data set of Breast is 699 (16 missing values) and 10 attributes are used for each data.

Table 5. Comparison on Iris

Algorithms	K value	Accuracy(%)
KNN	K=5	46.67
KNN	K=r	66.67
KNN	K=30	60.00
3N	K=r	66.67
3NW	K=r	80.00
3NQ	K=r	86.67

Table 6. Comparison on Seeds

Algorithms	K value	Accuracy(%)
KNN	K=r	71.42
KNN	K=15	61.90
KNN	K=25	52.38
3N	K=r	71.42
3NW	K=r	76.19
3NQ	K=r	85.71

Table 7. Comparison on Breast

Algorithms	K value	Accuracy(%)
KNN	K=10	61.43
KNN	K=r	78.57
KNN	K=40	68.57
3N	K=r	78.57
3NW	K=r	92.86
3NQ	K=r	94.28

From Tables 5, 6 and 7, we first notice that the KNN algorithm has a higher accuracy when $k = r$ than other k values. Second, when comparing the accuracy of 3N, 3NW and 3NQ, we can see that both W and Q considerably

improve the accuracy since both 3NW and 3NQ have a higher accuracy than 3N. Moreover, *Q* outperforms *W* since 3NQ has a higher accuracy than 3NW.

6. Conclusion and Future Work

In this paper, a novel algorithm, 3N-Q, is proposed for enhancing the performance of classification. Experiments show its stability, feasibility, potential and efficiency in knowledge discovery areas, classification, clustering and anomaly detection. In the near future, using the fitness function in genetic algorithm (Suguna & Thanushkodi, 2010), it is possible to add a constraint condition before finding r, thus finding a more accurate r which corresponds to making data the most saturated instead of unsaturation or over-saturation. 3N graph can reflect the distribution of data set more realistically and provide a more accurate evidence for clustering. Also, according to the 3N graph we can determine which similarity measure should be better used for classification.

References

Amato, G., & Falchi, F. (2013). On knn classification and local feature based similarity functions. *Agents and Artificial Intelligence Communications in Computer and Information Science, 271*, 224-239.

Blake, C., & Merz, C. (1998). *Uci repository of machine learning databases*. Retrieved from http://archive.ics.uci.edu/ml/index.html

Commons, W. (2007). Retrieved from http://www.cis.upenn.edu/ jshi/software/demo1.html

Deza, M. M., & Deza, E. (2013). *Encyclopedia of Distances*. Springer. http://dx.doi.org/10.1007/978-3-642-30958-8

Fayyad, U., Piatetsky-shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine, 17*, 37-54.

Gong, A., & Liu, Y. (2011). Improved KNN Classification Algorithm by Dynamic Obtaining K. In *Advanced Research on Electronic Commerce, Web Application, and Communication* (pp. 320-324). Berlin Heidelberg: Springer . http://dx.doi.org/10.1007/978-3-642-20367-1_51

Han, J. M., & Kamber, J. P. (2006). *Data Mining* (2nd ed.). Concepts and Techniques. Diane Cerra.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction* (2nd ed.). Springer. http://dx.doi.org/10.1007/978-0-387-84858-7

Huang, L., Liu, Z., Yan, Z., Liu, P., & Cai, Q. (2012). An implementation of high performance parallel knn algorithm based on gpu. In *Networking and Distributed Computing (ICNDC), 2012 Third International Conference on*. http://dx.doi.org/10.1109/ICNDC.2012.15

JD Kelly, L. D. (1991). A hybrid algorithm for classfication. In *Proceedings of Twelfth International Joint Conference on Artificail Intelligence*.

Li, L., Weinberg, C. R., & Darden, A. T. (2001). *Gene selection for sample classfication based on gene expression data: Study of sensitivity to chioce of parameters of the ga/knn method*. Oxford University Press.

Liu, W., & Chaw, S. (2011). Class confidence weighted kNN algorithms for imbalanced data sets. In *Advances in Knowledge Discovery and Data Mining* (pp. 345-356). Berlin Heidelberg: Springer. http://dx.doi.org/10.1007/978-3-642-20847-8_29

Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill.

Oliveira, L. (2012). Clonal selection classifier with data reduction: Classification as an optimization task. In *Evolutionary Computation (CEC),IEEE Congress on*.

Parvin, H., Alizadeh, H., & Minaei-Bidoli, B. (2008). Mknn: Modified k-nearest neighbor. In *World Congress on Engineering and Computer Science*.

Peterson, M. R., Doom, T. E., & Raymer, M. L. (2005). Ga-facilitated knn classifier optimization with varying similarity measures. In *Evolutionary Computation, 2005, The 2005 IEEE Congress on*. http://dx.doi.org/10.1109/CEC.2005.1555009

Shi, K., Li, L., Liu, H., He, J., Zhang, N., & Song, W. (2011). An improved KNN text classification algorithm based on density. In: *IEEE International Conference on Cloud Computing and Intelligence Systems*.

Shi, J. (2013). Retrieved from http://www.cis.upenn.edu/ jshi/software/demo1.html

Suguna, N., & Thanushkodi, K. (2010). An improved k-nearest classfication using genetic algorithm. *IJCSI International journal of computer science issue, 7*(4), No. 2.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Morgan Kaufmann, Amsterdam.

Yang, D. Z., & Liang, C.Y. (2012.) An intelligent knowledge discovery method based on genetic algorithms and conditional probability. *Network Computing and Information Security Communications in Computer and Information Science, 345*, 281-286. http://dx.doi.org/10.1007/978-3-642-35211-9_36