A New Version of the Dendritic Cell Immune Algorithm Based on the K-Nearest Neighbors

Kaouther Ben Ali(✉), Zeineb Chelly, and Zied Elouedi

LARODEC, Institut Supérieur de Gestion de Tunis, Tunis, Tunisia
kaoutherbenali17@gmail.com, zeinebchelly@yahoo.fr, zied.elouedi@gmx.fr

Abstract. In this paper, we propose a new approach of classification based on the artificial immune Dendritic Cell Algorithm (DCA). Many researches have demonstrated the promising DCA classification results in many real world applications. Despite of that, it was shown that the DCA has a main limitation while performing its classification task. To classify a new data item, the expert knowledge is required to calculate a set of signal values. Indeed, to achieve this, the expert has to provide some specific formula capable of generating these values. Yet, the expert mandatory presence has received criticism from researchers. Therefore, in order to overcome this restriction, we have proposed a new version of the DCA combined with the K-Nearest Neighbors (KNN). KNN is used to provide a new way to calculate the signal values independently from the expert knowledge. Experimental results demonstrate the significant performance of our proposed solution in terms of classification accuracy, in comparison to several state-of-the-art classifiers, while avoiding the mandatory presence of the expert.

Keywords: Artificial immune systems · K-Nearest Neighbors · Classification

1 Introduction

The Dendritic Cell Algorithm (DCA) is an immune inspired classification algorithm based on the abstract model of immune Dendritic Cells (DCs) [1]. It was applied to a wide range of applications, precisely in data classification such as in [2,3]. The DCA performance relies on its data-preprocessing phase where a signal dataset is generated for classification and which is based on three input signals pre-categorized as Pathogenic Associated Molecular Patterns (PAMPs), Danger Signals (DS) and Safe Signals (SS). All input signals cooperate with each other to give a final decision; i.e., the class of the data item. To achieve this classification task, the expert knowledge is required to calculate the signal values of the new data item to classify. More precisely, the expert has to give specific formula to calculate the signal values and specifically the DS values that play the leading role in assigning the class of each data instance.

Our aim, in this paper, is to propose a new DCA version capable of overcoming the mentioned DCA restriction which is based on the need of the expert

S. Arik et al. (Eds.): ICONIP 2015, Part I, LNCS 9489, pp. 688–695, 2015.
DOI: 10.1007/978-3-319-26532-2_76

knowledge to provide a complete signal data set for classification. Our proposed method, named KNN-DCA, is a new version of the DCA hybridized with the K-Nearest Neighbors (KNN) machine learning technique [4]. KNN is used as a technique to automatically calculate signals, precisely, the danger signal values that strongly depend on expert knowledge. We will show that our KNN-DCA is capable of processing the danger signal values and finding out the class of a new item without the need of an expert knowledge.

To guarantee the effectiveness and the efficiency of our proposed method, the material in this paper is organized as follows: In the remainder of this introduction, we specify our issue. In Sect. 2, the problem statement is highlighted. In Sect. 3, a detailed description of our proposed approach KNN-DCA is presented. This is followed by Sect. 4, where the results obtained from a set of experiments are discussed. Finally, Sect. 5 concludes the paper and presents some future directions.

2 Problem Statement

In this section, we will mainly clarify the main DCA limitation while performing its classification task. Yet, first, we have to elucidate one important characteristic of the DCA. Actually the algorithm, in literature, was applied in two different manners to machine learning datasets depending on the presence or absence of the expert. The first application manner of the DCA is its application as an unsupervised algorithm. In this case, no information about the previous data item classes are needed while classifying a new data instance. Technically and in this case, the presence of the expert is mandatory where he/she will provide a specific formula showing how to calculate the signal values of that new instance. Once the signal values are calculated, the DCA will generate the MCAVs and classify the new antigen. The second case is where the DCA is applied as a semi-supervised algorithm and in this case some information is required to acquire from the initial training dataset that includes the classes of all antigens. Here, the expert knowledge is not needed. Meanwhile, the classes of the data items have to be known and based on that, the algorithm applies a formula to classify the new data item. More precisely, the needed information from the initial training data set is the number of data items belonging to the normal class (class 1). Based on this information, a formula is applied to calculate only the danger signal values of the new data instance. However, to calculate the values of PAMPs and SS of the same new instance, a second formula is applied which does not depend on the class type; either class 1 or class 2. As discussed, the manner of how to apply the DCA strongly depends on the presence or absence of the expert knowledge. Yet, in most cases no information is afforded about the classes of the data items belonging to the training dataset and at the same time we want to avoid the expert knowledge. In this case, DCA is not capable of performing its classification task nor able it is to classify a new data instance. Thus, it would be very interesting to propose a new DCA version capable of performing its classification task in an autonomous way; i.e., independently from the need to the expert knowledge.

3 The Proposed Approach: A New Dendritic Cell Algorithm Based on the K-Nearest Neighbors

In this section, we will give a detailed description of our proposed new DCA version; the Dendritic Cell Algorithm based on the K-Nearest Neighbors (KNN-DCA). First, we will highlight the KNN-DCA architecture and then we will explain how our KNN-DCA is capable of performing its classification task without the mandatory presence of the expert nor the need of his/her guidelines on how to generate the signal values of the new data item to classify.

3.1 The KNN-DCA Architecture

The contribution of our work is to present a new DCA version capable of surmounting the mentioned DCA restriction which is based on the need of the expert knowledge to provide a signal dataset for classification. Our proposed method is a new version of the DCA hybridized with the K-Nearest Neighbors (KNN) machine learning technique. KNN is used as a technique to automatically calculate signals, precisely, the danger signal values that strongly depend on expert knowledge. The algorithmic steps of our KNN-DCA are as follows:

1. Preprocessing and Initialization phase.
2. Detection phase.
3. Context Assessment phase.
4. Classification phase based on KNN.

As presented in the itemized list, our proposed KNN-DCA is based on the same DCA steps [1,5,6] except for the classification phase which is based on the KNN concept. That is why in this section, we will focus mainly on the KNN-DCA classification step where we will explain in details how our KNN-DCA is capable of classifying a new data item via a new calculation process showing how to generate the signal values, without calling the expert knowledge.

3.2 The KNN-DCA Classification Phase

To classify a new data instance, KNN-DCA has to calculate a set of signals which are the PAMP signals, the safe signals and the danger signals without referring to the expert guidelines to do so. In what follows, we will give the algorithmic steps showing how to perform the signals calculation process.

Calculating the SS and the PAMP Signal Values. Based on immunological concepts, both PAMP and SS are considered as positive indicators of an anomalous and normal signal. This is because the PAMP signals are essential molecules produced by microbes but not produced by the host. They are definite indicators of abnormality indicating the presence of a non-host entity. However, the SSs are released as a result of a normal programmed cell death. They are indicators of normality which means that the antigen collected by the DC was found in a normal context. Hence, tolerance is generated to that antigen.

Mapping the immunological semantics of these two signals to the algorithmic KNN-DCA signal calculation process, one attribute is used to form both PAMP and SS values. The selected attribute is the one having the highest standard deviation among the feature set presented in the input training dataset. Using one attribute to derive the signal values of the new data item X to classify requires a threshold level to be set: values greater than this can be classified as a safe signal with a specific value, while values under this level would be used as a PAMP signal with another specific value. The process of calculating PAMP and SS values is itemized as follows:

1. Recall the most interesting feature which was selected to represent both SS and PAMPs, i.e., the selected one having the highest standard deviation among the feature set presented in the input training dataset; during the data pre-processing phase.
2. Calculate the median (M) of that attribute for all data instances in the training data set.
3. For the data item to classify determine its PAMP and safe signal values based on its attribute value (Val_X) which is the same used to calculate the median in Step 2. If the attribute value is greater than the median then this value is used to form the safe signal of the new data item. The absolute distance from the median is calculated and attached to the safe signal value and the PAMP signal value takes 0 (and vise versa). This can be seen as follows:

$$\text{If } (Val_X > M) \text{ then } SS_X = |M - Val_X| \text{ and } PAMP_X = 0; \qquad (1)$$

As noticed, the process of calculating SS_X and $PAMP_X$ is independent from the need to the expert knowledge. This process is the same one performed by the standard DCA to calculate the values of PAMP and SS for all data items in the training dataset [1].

Calculating the DS Values. Let us remind that the standard DCA version, to calculate the DS value of a new data item X, has either to call the expert or to use the information related to the number of data items having the label "normal" in the training dataset. Yet, we aim to avoid the expert mandatory presence and to avoid using the mentioned needed information from the training dataset. This is because this information, in most cases, is not accessible. Thus, we propose to apply the KNN machine learning technique in order to calculate automatically the DS value of X without referring to the two main restrictions; i.e., expert knowledge and the number of class 1 data items.

Our proposed KNN-DCA is based on the idea of applying KNN in order to select the nearest neighbors to the new data item X that will be classified and to find out an adequate formula mapping the nearest neighbors danger signal values to the DS_X value. This is how the DS_X value will be automatically calculated. Yet, while applying the KNN machine learning technique we may face two possible cases; either we may select the first nearest neighbor to X (k=1) and define the DS_X value based on that or to select a set of the nearest neighbors to X (k>1) and define the DS_X value. Based on these two possibilities, in what follows, we will propose two KNN-DCA methodologies to calculate the DS_X value.

First Case: DS Calculation Process based on the 1-Nearest Neighbor (K=1). At this stage the $PAMP_X$ and the SS_X values are calculated and what is missing is the DS_X value. Based on the KNN-$DCA_{k=1}$, we will search the input signal dataset which includes all the signal values of the antigens of the training dataset (PAMP, SS and DS) and from there we will select the nearest antigen (K=1). We will apply the Euclidian distance to calculate the similarities between all antigens and the new data item X. The similarity is calculated between both $PAMP_X$ and SS_X and between $PAMP_{y_i}$ and the SS_{y_i}; where y is referring to a data instance belonging to the training data set and $i \in \{1, n\}$ where n refers to the length of the input signal dataset; i.e., the number of all antigens in the training data base. The calculation of the DS_X value is given by Eq. 2.

$$DS_X = \frac{DS_Y * Signal_X}{Signal_Y} \tag{2}$$

In Eq. 2, Y refers to the nearest object to the X data item to be classified and DS_Y is the value of the Y danger signal. Let us remind that while calculating the PAMP and SS values, if the PAMP has a value different than zero then the SS value equals zero and vise versa. Thus, if both SS_Y and SS_X are null then $Signal_X$ equals the value of $PAMP_X$ and $Signal_Y$ equals the value of $PAMP_Y$. In the opposite case, where both SS_Y and SS_X are different from null and where $PAMP_X$ and $PAMP_Y$ are null then $Signal_X$ equals the value of SS_X and $Signal_Y$ equals the value of SS_Y.

Second Case: DS Calculation Process based on the K-Nearest Neighbors (K>1). The second case is focused on calculating the DS_X value based on the KNN-$DCA_{k>1}$. In this case, we will search the same input signal dataset used before and from there we will select the K nearest antigens (K>1). Just like the first case, we will apply the Euclidian distance to calculate the similarities between all antigens and the new data item X but instead of selecting one nearest neighbor we will select a set of nearest neighbors. The calculation of the DS_X value is given by Eq. 3.

$$DS_X = \frac{mean(DS_{Y_k}) * mean(Signal_{X_k})}{mean(Signal_{Y_k})} \tag{3}$$

In Eq. 3, Y_k refers to the set of the k nearest objects to X and DS_{Y_k} is the set of the Y_k danger signal values. The semantics of both $Signal_{X_k}$ and $Signal_{Y_k}$ hold as in Eq. 2. So, we will calculate the mean of all DS_{Y_k} values and the mean of both $Signal_{X_k}$ and $Signal_{Y_k}$ values and apply Eq. 3 to calculate the DS_X value.

Based on the set of these three calculated signal values, $PAMP_X$, SS_X and DS_X, and just like the standard DCA process, KNN-DCA can generate the

MCAV of the new data item and then compare the later value to the anomaly threshold which is generated automatically from the data at hand. So, if the MCAV of the new instance is greater than the anomaly threshold; the data item will be classified as a dangerous one (class 2) else it will be tolerated and assigned a normal label (class 1).

4 Experimental Setup and Results

In this section, we try to show the effectiveness of our KNN-DCA as well as its performance. The aim of our method is to show that our KNN-DCA is capable of performing well its classification task, in an autonomous way, in comparison to a set of well known state-of-the-art classifiers. We will, also, test the performance of our proposed KNN-DCA under a variation of the k parameter which is referring to the number of the nearest neighbors to the object to be classified. We have developed our program in Eclipse V 4.2.2 for the evaluation of our KNN-DCA. Different experiments are performed using two-class data sets from the UCI Machine Learning Repository [7]. The used datasets are described in Table 1.

Table 1. Details about the used Datasets

Data set	Ref	Instances	Attributes
Wisconsin Breast Cancer	WBC	699	10
SPECTF Heart	SH	267	45
Pima Indians	PI	768	6
Blood Transfusion	BT	748	5
Haberman's Survival	HS	306	4

In all experiments, each data item is mapped as an antigen, with the value of the antigen equals to the data ID of the item. A population of 100 cells is used. The DC migration threshold is set to 10. To perform anomaly detection, a threshold is applied to the MCAVs. The threshold is calculated by dividing the number of anomalous data items presented in the used data set by the total number of data items. So, if the MCAV is greater than the anomaly threshold then the antigen is classified as anomalous else it is classified as normal. For each experiment, the results presented are based on mean MCAVs generated across a 10-folds cross validation. We evaluate the performance of our KNN-DCA in terms of classification accuracy where we compare it with a set of well known classifiers, namely, the Decision Tree (C4.5), the Support Vector Machine (LibSVM), BayesNet, NaiveBayes, Hoeffding Tree (HT), Wrapper Classifier and the K star classifier. All the parameters used for these classifiers are set to the most adequate values to perform the classification results based on the Weka Software. We will divide our comparison methodology into two main phases.

Table 2. Comparison of Classifiers in terms of Classification Accuracy (%)

DataSets	KNN-DCA			DT	HT	Wrapper	BayesNet	NaiveBayes	SVM	K Star
	k=1	k=3	k=5							
WBC	99.71	99.57	99.86	94.70	96.20	42.90	96.20	97.20	96.70	95.60
SH	99.63	99.25	98.87	72.10	69.30	30.30	69.30	69.30	66.80	65.10
PI	99.61	99.35	99.22	74.40	75.70	42.40	79.80	79.80	77.60	70.90
BT	99.73	99.47	99.33	58.10	58.10	58.10	71.60	70.80	87.17	82.10
HS	87.26	86.60	84.64	68.60	71.50	54.10	69.10	73.80	63.80	71.00

First, we will test the influence of the variation of the k parameter on the KNN-DCA classification performance. We have chosen three different values of k; $k \in \{1, 3, 5\}$. The objective is to select the most convenient k value where KNN-DCA gives the most interesting results. Second, we will compare the KNN-DCA classification results with the already mentioned state-of-the-art classifiers.

Studying the influence of the k parameter on our KNN-DCA classification results and from Table 2, we notice that the three KNN-DCA PCCs are close to each other. For example, applying the KNN-DCA to the PI dataset, the classification accuracies are set to 99.61 %, 99.35 % and 99.22 % for KNN-$DCA_{k=1}$, KNN-$DCA_{k=3}$, KNN-$DCA_{k=5}$, respectively. The same remark is noted for the rest of the used datasets. Yet, we can notice that even if the classification accuracies are roughly the same, the PCC of KNN-$DCA_{k=1}$ is better than KNN-$DCA_{k=3}$ and KNN-$DCA_{k=5}$; in most datasets. Thus, we can conclude that the variation of k keeps the good performance of our proposed KNN-DCA and, as a consequence, this shows that our proposed danger signal calculation methodologies are valid as good classification results are obtained. Now, from Table 2, we can notice that in most cases our KNN-DCA is outperforming the used state-of-the-art classifiers in terms of classification accuracy. For instance, while applying the algorithms to the HS database and for the different values of k, the lowest PCC value of our KNN-DCA is set to 84.64 % with k=5. Yet, this value is greater than 68.60 %, 71.50 %, 54.10 %, 69.10 %, 73.80 %, 63.80 % and 71 % which are given by C4.5, HT, Wrapper, BayesNet, NaiveBayes, LibSVM and K Star; respectively.

To summarize, with the variation of the parameter k on the used datasets, our KNN-DCA gives better classification results when compared to other classifiers. Moreover, we have shown that despite of varying the k value, the KNN-DCAs are producing close PCCs which endorses the validity of our mathematical DS calculation processes which were detailed in Sect. 4.

5 Conclusion and Future Directions

This work extends the original DCA to be more applicable to the problem of interest without the need of expert knowledge. KNN-DCA has provided good classification results on a number of datasets. Indeed, our proposed algorithm is based on robust mathematical formula based on the KNN machine learning technique allowing the DCA classification task in an autonomous way. As future work, we intend to further explore the new instantiation of our KNN-DCA by extending the applicability of the KNN algorithm within a fuzzy context.

References

1. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
2. Greensmith, J., Aickelin, U.: Dendritic cells for syn scan detection. In: GECCO, pp. 49–56 (2007)
3. Chelly, Z., Elouedi, Z.: Hybridization schemes of the fuzzy dendritic cell immune binary classifier based on different fuzzy clustering techniques. In: New Generation Compution, vol. 33(1), pp. 1–31. Ohmsha, Chiyoda-ku (2015)
4. Ghosh, A.: On optimum choice of k in nearest neighbor classification. Comput. Stat. Data Anal. **50**(11), 3113–3123 (2006)
5. Chelly, Z., Elouedi, Z.: Supporting fuzzy-rough sets in the dendritic cell algorithm data pre-processing phase. In: Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M. (eds.) ICONIP 2013, Part II. LNCS, vol. 8227, pp. 164–171. Springer, Heidelberg (2013)
6. Chelly, Z., Elouedi, Z.: RST-DCA: a dendritic cell algorithm based on rough set theory. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 480–487. Springer, Heidelberg (2012)
7. Asuncion, A., Newman, D.J.: UCI machine learning repository, (2007). http://mlearn.ics.uci.edu/mlrepository.html