agriculture

MDPI

Article

An Efficient Case Retrieval Algorithm for Agricultural Case-Based Reasoning Systems, with Consideration of Case Base Maintenance

Zhaoyu Zhai [1,*], José-Fernán Martínez Ortega [1], Néstor Lucas Martínez [1] and Huanliang Xu [2]

[1] Departamento de Ingeniería Telemática y Electrónica (DTE), Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETSIST), Universidad Politécnica de Madrid (UPM), C/Nikola Tesla, s/n, 28031 Madrid, Spain; jf.martinez@upm.es (J.-F.M.O.); nestor.lucas@upm.es (N.L.M.)

[2] College of Information Science and Technology, Nanjing Agricultural University, Nanjing 210095, China; huanliangxu@njau.edu.cn

* Correspondence: zhaoyu.zhai@upm.es

Received: 14 July 2020; Accepted: 1 September 2020; Published: 3 September 2020

Abstract: Case-based reasoning has considerable potential to model decision support systems for smart agriculture, assisting farmers in managing farming operations. However, with the explosive amount of sensing data, these systems may achieve poor performance in knowledge management like case retrieval and case base maintenance. Typical approaches of case retrieval have to traverse all past cases for matching similar ones, leading to low efficiency. Thus, a new case retrieval algorithm for agricultural case-based reasoning systems is proposed in this paper. At the initial stage, an association table is constructed, containing the relationships between all past cases. Afterwards, attributes of a new case are compared with an entry case. According to the similarity measurement, associated similar or dissimilar cases are then compared preferentially, instead of traversing the whole case base. The association of the new case is generated through case retrieval and added in the association table at the step of case retention. The association table is also updated when a closer relationship is detected. The experiment result demonstrates that our proposal enables rapid case retrieval with promising accuracy by comparing a fewer number of past cases. Thus, the retrieval efficiency of our proposal outperforms typical approaches.

Keywords: case-based reasoning; case retrieval; case base maintenance; knowledge management; smart agriculture

1. Introduction

Managing farming operations is a challenging task due to its complexity and unpredictability [1]. It includes the activities like irrigation scheduling [2,3], pest management [4,5], nutrient management [6,7], investment of agricultural machinery [8,9], harvesting [10,11], logistics [12,13] and so forth. Farmers and stakeholders not only need to deal with short-term (daily and weekly) scheduling problems but also have to consider long-term (yearly) management. Typically, farmers are used to make these decisions according to their own observations and experiences [14,15]. However, an inappropriate decision may usually cause serious issues like decreasing the productivity, damaging the soil fields, increasing the costs and so forth. Owing to the latest advance of Internet of Things (IoT) and sensor techniques, data collected by climate sensors, ground sensors, radiation sensors and weather stations (made of sensors) enable researchers to build an IoT-based platform and therefore to execute tasks like monitoring, knowledge mining, reasoning and control [16,17]. However, with the growing amount of data collected by various sensors, farmers sometimes have great difficulties in making proper judgments, since they are not data scientists. As a consequence, farmers are now gradually employing decision support systems (DSSs) [18,19] for obtaining advice, because DSSs are able to transfer unstructured raw data into useful knowledge, therefore assisting farmers in managing agricultural activities efficiently and profitably.

As one of the most popular techniques in artificial intelligence, case-based reasoning (CBR) has been gradually employed for modelling DSSs in the domain of smart agriculture [20,21]. In general, an agricultural decision support system (ADSS) is a platform that gathers and analyses data collected from a variety of sources (meteorological, plant/crop-related, economic data). The purpose of an ADSS aims at assisting farmers in smoothening the decision-making processes for agricultural management by providing a list of feasible solutions [18]. With strong reasoning capability, CBR can be used to generate these solutions. Once farmers encounter a new agricultural problem, the description of this problem is treated as a new case to a CBR enabled ADSS. Afterwards, the ADSS uses similarity measures to retrieve the most similar past cases from the case base, along with corresponding solutions. It is acknowledged that if the new case and retrieved past cases have great commonalities, then the solutions of retrieved cases can be used to solve the new case as well [22]. Therefore, farmers can obtain the decision supports from the ADSS for managing the agricultural tasks.

Though applying case-based reasoning has promising advantages like ease of use and precise response, some critical issues in case retrieval have been pointed out by researchers [23]. For example, each past case is typically considered as an independent individual in the case base and assigned with a sequential number as its unique identifier. However, a case in the case base could share similar (or dissimilar) feature values with others, leading to the fact that these cases can be interconnected by a similar association. Under such circumstance, the retrieval task may skip unnecessary comparisons between new and past cases and therefore to accelerate the retrieval process. Unfortunately, few researches pay attention to use the internal associations between cases. The negligence on these relations may lead to poor performance at the stage of case retrieval, because the case retrieval algorithms would sequentially traverse all the past cases for matching the most similar ones, even though a large volume of cases is stored in the case base.

To improve retrieval efficiency, some methods like the rough set theory [24,25] and filtering techniques [26,27] were adopted. On the one hand, the rough set theory could reduce the number of compared cases by defining lower and upper approximations. However, all past cases have to get involved when generating a set of qualified cases that meets the approximations. On the other hand, some researchers defined a rule set manually for filtering past cases. The rules were specified based on the observation of cases and researchers' own interests for case retrieval tasks. Unfortunately, both rough set theory and filtering techniques failed to address any associations between cases and they were task specific. Once a new task is put forward, the filtering process has to be executed over and over.

Case retrieval plays an essential role in CBR systems because the rest of steps (reuse, revise and retain) cannot further proceed without successfully retrieving the most similar past cases in the first place. Current studies on the case retrieval algorithms mainly concern the following two aspects—(i) proposing new similarity measures and (ii) proposing new indexing methods.

In case-based reasoning, similarity measures are used to quantify the similarity between two objects [28]. Usually, a smaller distance value means that the compared two objects have more commonalities. For retrieving the most similar past cases, researchers have contributed a lot towards proposing new similarity measures.

Wang et al. [29] proposed a novel hybrid similarity measure for case retrieval in case-based reasoning systems, with considerations for five formats of attributes values—crisp symbols, crisp numbers, fuzzy numbers, fuzzy linguistic variables and fuzzy intervals. The calculation formula of the global similarity was established by combining the hybrid similarity measure and the synthesis weight measure for retrieving the proper historical case. Yoon et al. [30] presented C-Rank, a link-based similarity measure for identifying similar scientific literatures in databases. This similarity measure used both in-link and out-link references, disregarding the direction of the references. The experimental result demonstrated that C-Rank achieved higher accuracy than existing approaches. Yazid et al. [31] designed a new similarity measure based on Bayesian network for brain tumors cases retrieval. Their proposal was based on graph correspondences and signature nodes comparison from the Bayesian classifiers. The promising experimental results indicated that the proposed similarity measure outperformed classical methods. Zhai et al. [32] proposed a novel triangular similarity measure, overcoming the shortcomings of cosine similarity and Euclidean distance similarity. The experimental result showed that their proposal had strong robustness and great accuracy. Jiang et al. [33] introduced a novel semantic similarity measure for formal concept analysis by taking advantages of linked data and WordNet. The proposed method was not only used for data analysis and knowledge representation but also for concept formation and learning.

Though newly-proposed similarity measures indeed enable case-based reasoning systems to retrieve more accurate past cases, these measures do not improve the efficiency of case retrieval. During the step of case retrieval, the algorithms have to traverse all past cases in the case base, leading to low efficiency when a large volume of cases is stored. Therefore, proposing new similarity measures is not enough for improving the performance of case-based reasoning systems.

As a computational data structure, an index enables a case to be stored and searched in memory. Case indexing assigns indexes to cases for facilitating their retrieval [34] and it plays a key role in case base maintenance. Many literatures have concerned the indexing issues.

Honigl and Kung [35] proposed a data quality index method for maintaining the case base and avoiding redundant cases. Three indices (average solutions per case, count of similar retained queries and missing values) were used to build an index for the quality of the case base. Wiltgen et al. [36] presented two indexing methods, named functional indexing and structural indexing. Both indexing methods generated separate discrimination networks and had mechanisms for preventing the network from having duplicate nodes. Similar past cases could be retrieved by adopting the indexing methods and similarity measurements. Ahmad et al. [37] adopted the locality sensitive hashing (LSH) technique for obtaining short binary codes to represent medical radiographs. These hashing codes enabled indexing and efficient retrieval in large scale image collections. Durmaz and Bilge [38] proposed an approach named randomized distributed hashing (RDH), which used LSH in a distributed scheme. RDH randomly distributed data to different nodes on a cluster and used hash function for indexing. Then the query sample was locally searched in different nodes during the query stage. The experimental result showed that the proposed distributed scheme had great potential to search images in large datasets with multiple nodes. Ahmed and Sarma [39] detected that the accuracy of a system degraded with the increase in the size of the database, therefore an indexing approach was designed to deal with the feature deviation under noise. Considering the retrieval task, the proposed indexing approach gave higher hit rate than existing approaches, even at low penetration rate.

From above review on current literatures, it is concluded that indexing methods have great influence on case retrieval and case base maintenance. LSH is especially popular in case indexing. LSH refers to use a family of functions to map hash data points into buckets [40]. As a result, data points that near to each other are located in the same buckets with high probability, while data points that are far from each other are likely to be placed in different buckets. This makes it easier and more efficient to identify past cases that are similar to the new one. However, LSH does not guarantee accuracy of classified cases. For example, two similar data points may be separated into different buckets due to the design of hashing functions. Thus, improvements on new indexing methods for case retrieval and case base maintenance are expected. It is worth noticing that none of above literatures mentioned mining and using the internal associations between past cases. In other words, each case is still individually stored and searched.

Therefore, in this paper, a new case retrieval algorithm for agricultural case-based reasoning systems is proposed. Before executing the algorithm, an association table is constructed, containing the associations between past cases. At first, the new case is compared with an entry-point case. Based on the similarity measurement, the similar or dissimilar association is then selected for comparison in the next iteration until the most similar past cases are detected. Under such circumstance, potential similar past cases can be evaluated preferentially and the number of compared cases is therefore reduced, because the proposed algorithm is able to skip unnecessary comparisons. Meanwhile, our proposal takes case base maintenance into account. The association table is updated during runtime. After resolving the problem, the new case is retained in the case base, as well as its similar and dissimilar associations.

The rest of this paper is organized as follows. Section 2 presents the materials and methods of the proposed case retrieval algorithm. The results and discussions are presented in Section 3. Finally, conclusions are drawn in Section 4.

2. Materials and Methods

The proposed algorithm relies on a pre-constructed association table. Within this table, each past case is interconnected to several similar and dissimilar past cases. Once a new case is reported, it firstly compares with an entry point (as a starting case for comparison in the first iteration). If the similarity measurement between the new case and the entry point indicates these two cases have great commonalities, the similar association of the entry point is then selected for comparison in the next iteration, otherwise, the new case is compared with the dissimilar past cases which are associated with the entry point. The retrieval process keeps going until the termination of the algorithm is reached. On the contrary to traversing all past cases in typical case retrieval algorithms, our proposal measures the similarity of associated cases preferentially. Under this circumstance, the number of compared cases can be greatly reduced, therefore efficiency of case retrieval can be improved. For case retention, features of the new case, its similar and dissimilar associations, as well as its solutions, are stored in the case base. Meanwhile, the association table is updated if the new case shows closer relations than the old ones (associations).

2.1. Case Representational Formulism

The proposed algorithm focuses on retrieving agricultural cases which are formularized by the feature vector representation [41]. As the simplest formulism of case representation, it represents cases by a set of features that describes the problems and corresponding solutions. In this manuscript, the agricultural case-based reasoning systems tries to manage pest problems, therefore, the agricultural cases are defined and shown in Table 1.

Table 1. Agricultural cases defined by the feature vector representation.

Feature Category	Feature Name	Feature Type	Unit	Content
Pest	Pest name	Text	/	Name of pests
	Pest quantity	Integer	Unit	Number of pests
	Pest stage	Text	/	Life cycle of pests
	Infected area	Integer	m^2	Area of infected districts
Crop	Crop name	Text	/	Name of crops
	Growth stage	Text	/	Life cycle of crops
	Planting density	Integer	Seeds/m^2	Seeds density in an area
Environment	Temperature min	Integer	Celsius degree	The minimum temperature
	Temperature max	Integer	Celsius degree	The maximum temperature
	Humidity	Double	/	Water vapor in air
	Rainfall	Double	/	Possibility of rainfall
	Sunlight	Integer	Lux	Daylight during daytime
	Wind speed	Integer	m/s	Wind flow velocity

In Table 1, pest, crop and environment data are considered in agricultural cases [42]. Each case has the same type and number of features. For implementation, past cases are stored in the CSV format. Since our agricultural case-based reasoning system is coded by the programming language

Python, libraries like "Numpy" and "Pandas" can be used to manipulate the stored past cases easily. Furthermore, data in CSV format are understandable and readable for farmers, even though they do not have any expertise in knowledge management and computer science.

Contents of some features in Table 1 are given by texts, like "pest name," "pest stage," "crop name" and "growth stage." To deal with these textual features, we encode them into integers. Transforming a linguistic feature into a real number is a common approach in case-based reasoning systems [43,44]. For example, the life cycle of pest includes "egg," "pupae," "larvae" and "adult." An integer is assigned to each stage respectively. Thus, integer "1" represents "egg," "2" represents "pupae," "3" represents "larvae" and "4" represents "adult." The same transformational process works for the rest of textual features as well. For normalization both numeric and textual features, we adopted the Min-Max feature scaling method, mapping the original features into the range from 0 to 1. Although, there are some interrelations between a single feature, such as the life cycle of pest follows a time sequence. The process of data normalization does not eliminate these interrelations, since the original values (1,2,3,4) would be normalized as (0,0.3333,0.6667,1), which also reflects the interrelations.

In the feature vector representation, it is worth mentioning that there are no relationships built between cases. In other words, each case is individually stored in memory. This is the reason why an association table is constructed in the next section for case retrieval.

2.2. Construction of an Association Table

The association table contains the interconnection between past cases. Within the case base, a case could be similar or dissimilar to several other cases. An example of a case base is given in Table 2, including six past cases. Each case has four features. After data normalization [45], all the cases are visualized in Figure 1.

Table 2. An example of a case base.

Case ID	Feature 1	Feature 2	Feature 3	Feature 4
case 1	10	103	55	21
case 2	15	99	79	98
case 3	11	105	52	22
case 4	18	60	89	105
case 5	9	112	60	25
case 6	16	153	95	44

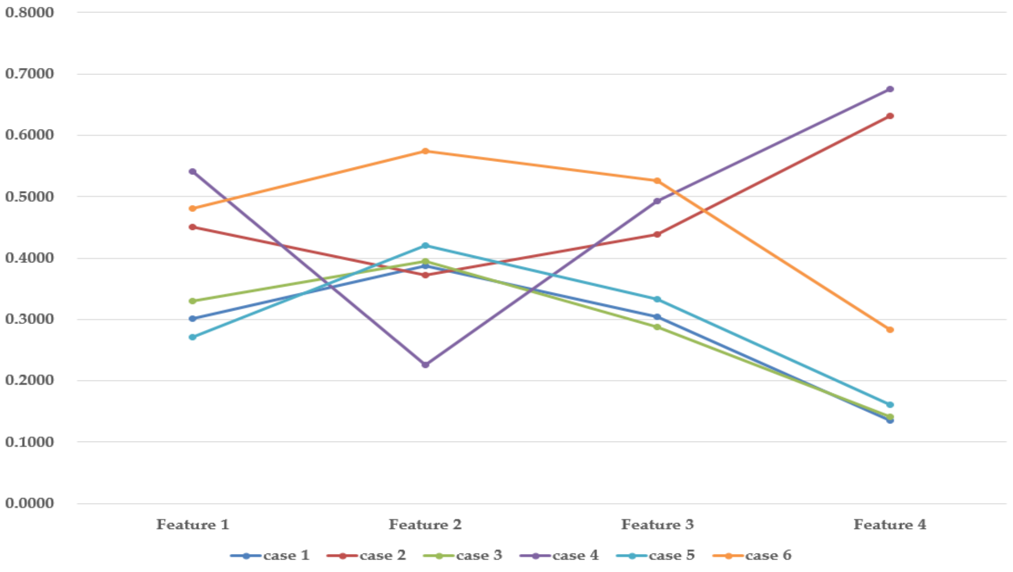


Figure 1. Visualization of cases from Table 2.

According to the visualization result in Figure 1, it is obvious that case 1 is similar to cases 3 and 5 because their data deviation is small. Meanwhile case 1 is dissimilar to cases 2 and 4 because their data distribution has major differences. Similarly, it is observed that case 2 is similar to cases 4 and 6, while case 2 is dissimilar to cases 3 and 5. Thus, the following association table can be constructed, shown in Table 3. Each past case is associated with two similar and two dissimilar associations. The similarity measurements between two associated cases are stored in the association table as well. The similarity and the dissimilarity measurements are both calculated according to Reference [32]. In Table 3, for filling in the similar association, the cases that achieve the top two highest measurements will be chosen. Meanwhile, the cases that achieve the last two lowest measurements will be selected as the dissimilar association.

Table 3. Association table for cases from Table 2.

Case ID	Similar Association				Dissimilar Association			
	1st Similar	Sim	2nd Similar	Sim	1st Dissimilar	Sim	2nd Dissimilar	Sim
case 1	case 3	96.49%	case 5	94.27%	case 4	52.62%	case 2	58.56%
case 2	case 4	83.01%	case 6	66.15%	case 3	59.08%	case 5	59.65%
case 3	case 1	96.49%	case 5	92.18%	case 4	53.10%	case 2	59.08%

In Table 3, each case has two types of associations:

- Similar association—This type of association indicates that the features of concerned two cases have great commonalities. Consequently, the IDs of these similar cases are stored in the similar association, building interconnections to the source case. For example, case 1 is associated with cases 3 and 5. Once a new case is reported and case 1 is treated as the entry point, the cases 3 and 5 are selected for comparison if the new case is considered similar to case 1. Because other potential similar cases might exist among the similar association, the similar association offers the chance of evaluating the past cases within a smaller range, instead of searching the whole case base. As a result, the number of compared cases can be reduced and retrieval efficiency can be improved.
- Dissimilar association—This type of association specifies that there are significant differences between the features of concerned two cases. The IDs of these dissimilar cases are stored in the association table as well. For example, case 2 is associated with cases 5 and 3. The dissimilar association aims at assisting the new case in identifying a relative similar case at the very beginning of case retrieval. Meanwhile, this association is also helpful when the retrieval process is trapped in a local optimal solution. In other words, the dissimilar association can adjust the searching trajectory in order to detect the global optimal solution.

For constructing such association table, it is necessary to measure the similarity between each past case. For instance, in Table 2, case 1 has to be compared with cases 2, 3, 4, 5 and 6 respectively, case 2 has to be compared with cases 1, 3, 4, 5 and 6 respectively and so forth. After obtaining all the similarity measurements, each case can be associated with several similar and dissimilar ones. For instance, the number of similar associations is two, then two cases with the top similarity measurements are selected and stored in the association table. As shown in Table 3, cases 3 and 5 achieves the top 2 similarity measurements when being compared with case 1, therefore, cases 3 and 5 are selected as the similar associations of case 1. The number of associated cases depends on the size of the case base. More cases are stored in the case base, more associations should be built.

Though constructing the association table is a time-consuming process when a large volume of cases is stored, it is still essential to explore the relations between past cases, because these relations could be useful for case retrieval. Besides, this construction is a one-time task. The association table is constructed at the initial stage, just before the system receives new inquires. After completing the construction, the association table is ready for use. For maintaining the association table, on the one hand, if a new retrieval task is completed and the CBR system decides to retain this new case, the association of this new case will be added to the association table. Meanwhile, the detected closer relations will update the old ones in the association table. On the other hand, if the CBR system determines not to retain the new case, the association table would remain unchanged.

2.3. Case Retrieval Algorithm

The workflow of the proposed case retrieval algorithm is presented in Figure 2.

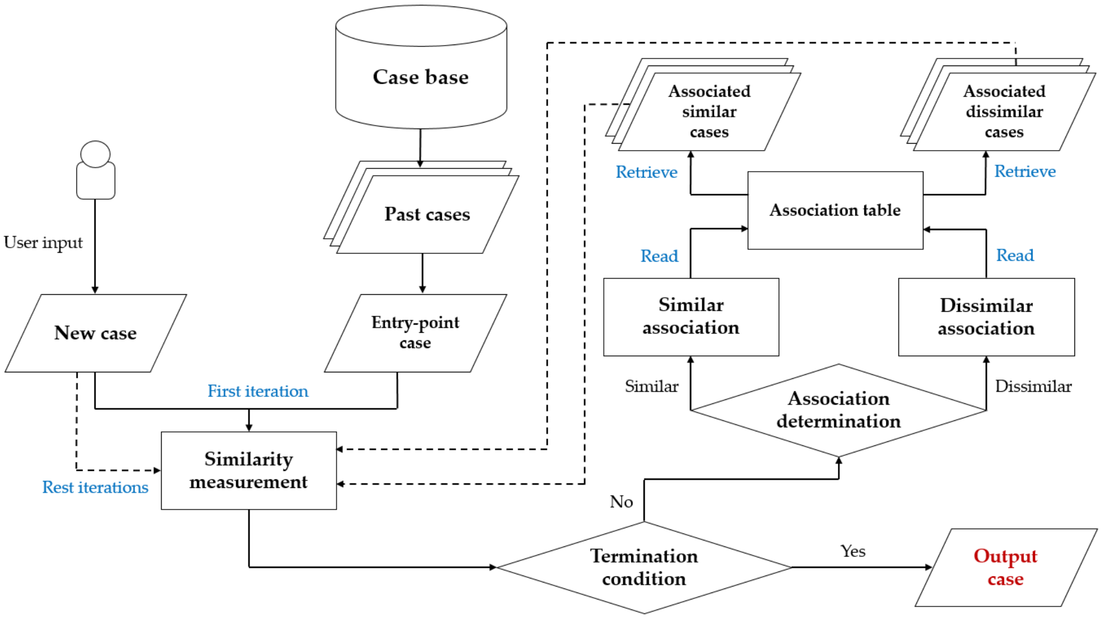


Figure 2. Workflow of the proposed case retrieval algorithm.

In Figure 2, the case retrieval algorithm firstly starts with the new case input by users. An entry point is randomly selected from the case base for comparison in the first iteration. Based on the similarity measurement, the algorithm decides whether the new case is similar (or dissimilar) to the entry-point case. Afterwards, the corresponding association is determined and associated similar (or dissimilar) cases are read and retrieved from the association table. Then, the similarity between the new case and associated similar (or dissimilar) ones are measured in the next iteration, until the termination condition is reached. The termination condition of the algorithm is defined as—(i) the maximum iteration number is reached or (ii) a satisfied similar past case is found.

For determining whether the compared two cases are similar or dissimilar, Table 4 is presented, indicating the correspondence between similarity level and measurements.

Table 4. Correspondence between similarity level and measurements.

Level	Condition
Identical	The compared two cases are exactly the same, achieving a similarity measurement at 100.00%
Highly similar	The compared two cases achieve a similarity measurement, ranging from 75.00% to 99.99%
Similar	The compared two cases achieve a similarity measurement, ranging from 50.00% to 74.99%
Dissimilar	The compared two cases achieve a similarity measurement, ranging from 25.00% to 49.99%
Highly dissimilar	The compared two cases achieve a similarity measurement, ranging from 0.00% to 24.99%

In regard to the association determination (for selecting the associated similar or dissimilar cases), a set of policies is defined in the case retrieval algorithm as follows.

- Policy 1—Detection of identical cases—If a past case is detected identical to the new case, the case retrieval algorithm terminates immediately. The output is the retrieved past case.
- Policy 2—Token assignments—Once a past case is considered highly similar to the new case, three positive tokens will be assigned to this past case. Once a past case is considered similar to the new case, one positive token will be assigned to this past case. Once a past case is considered highly dissimilar to the new case, three negative tokens will be assigned to this past case. Lastly, once a past case is considered dissimilar to the new case, one negative token will be assigned to this past case.
- Policy 3—Association selection—In general, the association with more tokens will be selected for comparison. When the number of positive tokens is greater than negative ones, the past case with the highest similarity measurement will be selected. The associated similar cases of this chosen one will be evaluated in the next iteration. While the comparative result of the current iteration suggests that the number of negative tokens is more, then the past case with the lowest similarity measurement will be selected. Consequently, the associated dissimilar cases of this selected one are retrieved from the association table for comparison in the next iteration.
- Policy 4—Selection of previous cases—It happens that all associated cases in a single iteration have been compared previously due to the reason that a past case can be associated with a 1-to-N relation. For instance, in Table 3, case 5 has the similar association with cases 1 and 3. It makes no sense to repeatedly evaluate cases that have been already compared, resulting in an endless loop for the algorithm. Under this circumstance, the cases to be evaluated in the next iteration are selected from previous iterations. Based on the number of tokens, corresponding association is determined and the past case with the second highest (or lowest) similarity measurement from the previous iteration will be chosen for comparison. If the past cases in the previous iteration have all been selected, then the algorithm will repeat Policy 4 one more time.

In Table 4, apart from the identical, similar and dissimilar levels, we also defined highly similar and highly dissimilar levels. Assume that the retrieval algorithm meets the following situation—the similarity measurements between the new case 1 and past cases 1, 2 and 3 are 90.00%, 30.00% and 40.00% respectively. Without the definition of highly similar and highly dissimilar levels, according to the pre-defined Policy 3, the dissimilar association of past case 2 will be selected for comparison in the next iteration. However, since the past case 1 is so similar to the new case 1, the similar association of the past case 1 has a great chance of being similar to the new case 1. Therefore, it would be a better choice to search in the similar association of the past case 1. Therefore, for avoiding this situation from happening, we decided to define the highly similar and highly dissimilar levels. Under such circumstance, the proposed retrieval algorithm would be forced to follow the potential optimal searching path. In summary, we equally divide the measurement into four intervals, denoting the highly similar, similar, dissimilar and highly dissimilar respectively.

The pseudo code of the proposed case retrieval algorithm is displayed in Table 5.

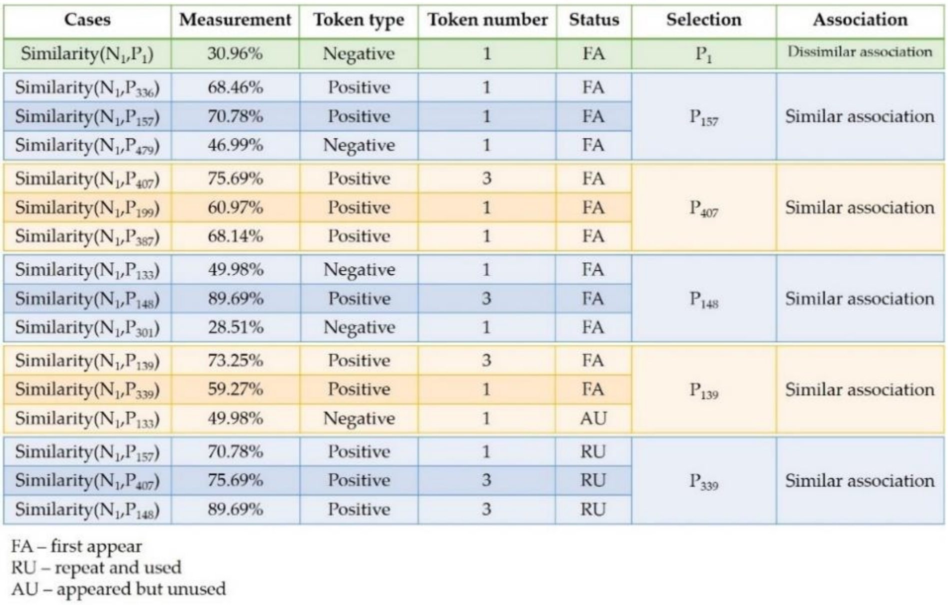
For better demonstrating the proposed case retrieval algorithm, an example is presented in Figure 3.

In Figure 3, P_i represents the ith past case in the case base, while N_1 is the first new case. Initially, P_1 is selected as the entry point for comparing with N_1 in the first iteration. The comparative result suggests that the dissimilar association of P_1 should be chosen for comparison (Policies 2 and 3). Thus, P_{336}, P_{157} and P_{479} are compared with N_1. In the second iteration, the number of positive tokens is greater than negative ones. Consequently, the associated similar cases of $P_{157,}$ which has the greatest similarity measurement, are selected for comparison in the next iteration (Policies 2 and 3). The case retrieval algorithm keeps running until the 6th iteration, all past cases have been repeated and used previously. According to Policy 4, P_{339} which has the second highest similarity measurement from the 5th iteration is chosen as a substitution (Policy 4). The output of this algorithm is a past case which has the greatest commonalities with the new case. The termination condition of the proposed algorithm is defined as—(i) the maximum iteration number is reached; or (ii) an identical case is detected. The travelling sequence of the proposed case retrieval algorithm in above scenario is presented in Figure 4.

Table 5. The pseudo code of the proposed case retrieval algorithm.

Pseudo code: Case retrieval

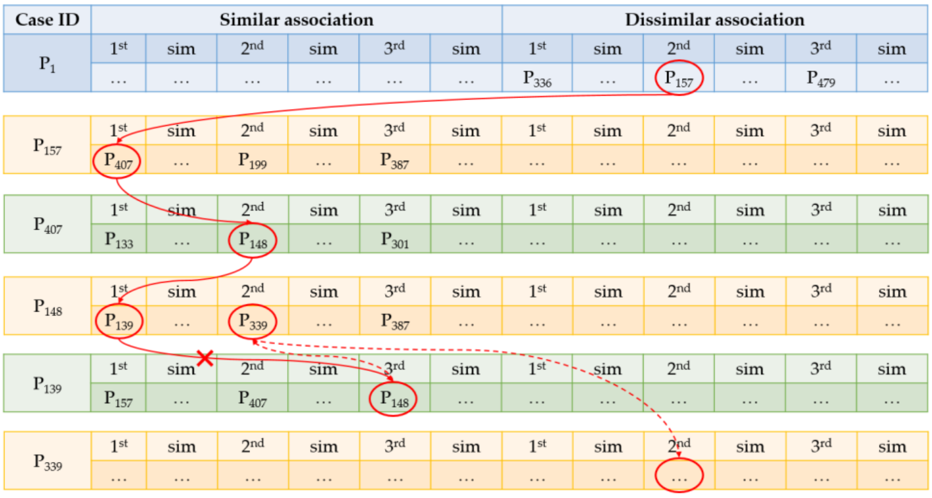
```
newCase = getUserInput(attributes)
entryCase = getCase(caseBase, entryPointID)
sim = simMeasure(newCase, entryCase)
token = assignTokens(entryCase, sim) // Policy 2
association = associationDetermine(token) // Policy 3
associatedCase_k = readAssociationTable(caseID, association) // k -> number of associated cases
for i in range (terminationCondition)
    for j in range (0,k)
        caseAvailability = checkRepeatedCase(associatedCase_j) // Policy 4
    if (caseAvailability != null)
        sim = simMeasure(newCase, associatedCase)
        token = assignTokens(associatedCase, sim)
        association = associationDetermine(token)
        associatedCase_k = readAssociationTable(caseID, association)
        output = findSimCase(sim, associatedCase_k)
    else
        re-select associatedCase_k from caseBase
    if (new case = associatedCase_k) // Policy 1
        output = associatedCase_k
        return output
return output
```



Cases	Measurement	Token type	Token number	Status	Selection	Association
Similarity(N_1,P_1)	30.96%	Negative	1	FA	P_1	Dissimilar association
Similarity(N_1,P_{336})	68.46%	Positive	1	FA	P_{157}	Similar association
Similarity(N_1,P_{157})	70.78%	Positive	1	FA		
Similarity(N_1,P_{479})	46.99%	Negative	1	FA		
Similarity(N_1,P_{407})	75.69%	Positive	3	FA	P_{407}	Similar association
Similarity(N_1,P_{199})	60.97%	Positive	1	FA		
Similarity(N_1,P_{387})	68.14%	Positive	1	FA		
Similarity(N_1,P_{133})	49.98%	Negative	1	FA	P_{148}	Similar association
Similarity(N_1,P_{148})	89.69%	Positive	3	FA		
Similarity(N_1,P_{301})	28.51%	Negative	1	FA		
Similarity(N_1,P_{139})	73.25%	Positive	3	FA	P_{139}	Similar association
Similarity(N_1,P_{339})	59.27%	Positive	1	FA		
Similarity(N_1,P_{133})	49.98%	Negative	1	AU		
Similarity(N_1,P_{157})	70.78%	Positive	1	RU	P_{339}	Similar association
Similarity(N_1,P_{407})	75.69%	Positive	3	RU		
Similarity(N_1,P_{148})	89.69%	Positive	3	RU		

FA – first appear
RU – repeat and used
AU – appeared but unused

Figure 3. An example of the proposed case retrieval algorithm.



Case ID	Similar association						Dissimilar association					
P_1	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
	…	…	…	…	…	…	P_{336}	…	P_{157}	…	P_{479}	…
P_{157}	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
	P_{407}	…	P_{199}	…	P_{387}	…	…	…	…	…	…	…
P_{407}	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
	P_{133}	…	P_{148}	…	P_{301}	…	…	…	…	…	…	…
P_{148}	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
	P_{139}	…	P_{339}	…	P_{387}	…	…	…	…	…	…	…
P_{139}	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
	P_{157}	…	P_{407}	…	P_{148}	…	…	…	…	…	…	…
P_{339}	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
	…	…	…	…	…	…	…	…	…	…	…	…

Figure 4. Travelling sequence of the proposed algorithm for the scenario in Figure 3.

2.4. Case Base Maintenance

After retrieving the most similar past case, the solution of this past case can be reused and revised for resolving the new problem. However, the solution reuse and revision are not the main concern of this manuscript. Thus, this issue is not going to be further discussed. Our main objective focuses on case retrieval and case base maintenance.

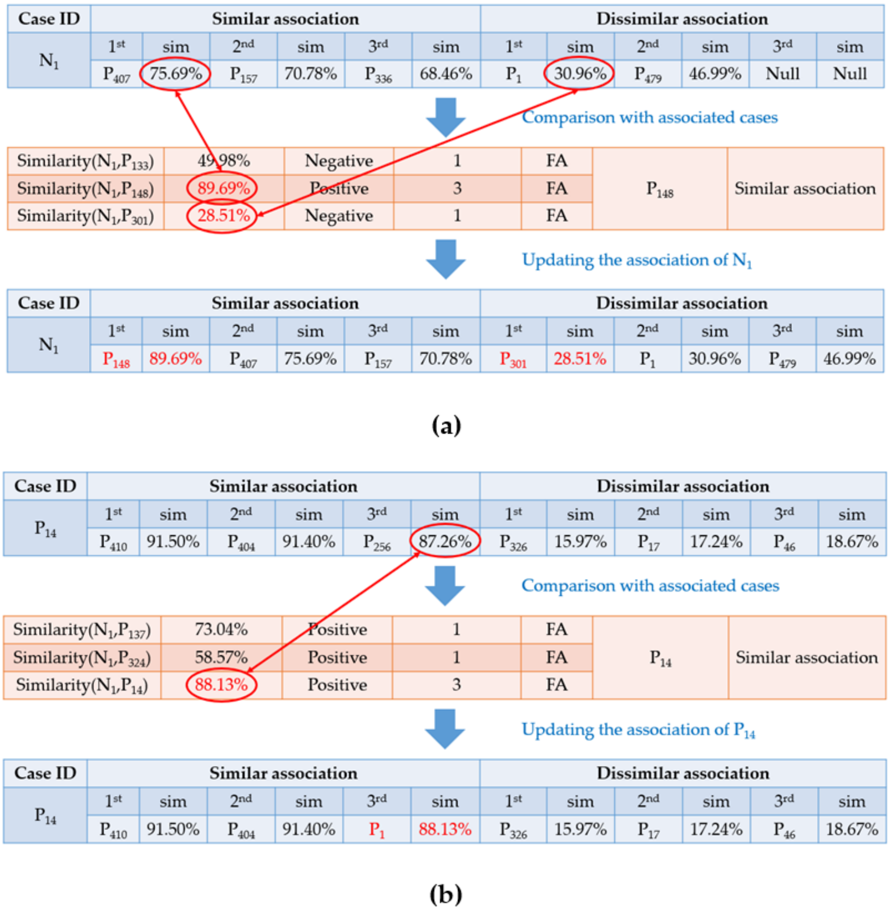
In terms of case retention and case base maintenance, the typical approach is to directly add the newly-solved case into the case base, along with its solution [46]. Under the circumstance when the new case is extremely similar to a past case that has been already stored in the case base, a case forgetting strategy could be applied after evaluating the quality of both cases [47]. In our case, we have to pay attention to the association table as well, because the performance of our algorithm depends on this table.

The proposed case retrieval algorithm takes care of case base maintenance in the following two aspects—(i) storing the learned case and (ii) updating the existing association of past cases.

Firstly, the case base should retain the learned case which is composed by the problem description of the new case, the corresponding solution and its association. In general, the learned case is assigned with a sequential number, as its unique identifier and then stored in the case base. The similar and dissimilar associations of this learned case are added at the end of the association table. The addition of new cases certainly ensures the possibility of retrieving cases that are similar to the target problems, however, this continuous addition also enlarges the size of the case base, leading to the complexity and low efficiency of case retrieval tasks [48]. As a consequence, if the new case is extremely similar to a certain past case in the case base, its retention should be dropped for avoiding redundancy. This solution is acknowledged as forgetting strategy [49]. By calculating the goodness of the learned case, the case-based reasoning system decides whether the learned case should be remembered or forgotten. For simplifying the process of case retention, a threshold is defined at 98.00%, suggesting that if the similarity measurement between the learned case and the retrieved most similar past case achieves beyond 98.00%, then the learned case will be forgotten and it will be not stored in the case base. Otherwise, case retention follows the general circumstance mentioned at the beginning of this paragraph.

Secondly, once the CBR system decides to retain the learned case, the existing association of past cases should be updated as well. The following two scenarios are considered in the manuscript.

- Scenario 1—Updating the association of the new case—If a closer similar or dissimilar association with the new case is detected during runtime, this past case should replace the old ones (Figure 5a).
- Scenario 2—Updating the association of the past cases—If the new case shows a closer association with the compared past case, the old association of this past case should be replaced by the new case (Figure 5b).



Case ID	Similar association						Dissimilar association					
	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
N_1	P_{407}	75.69%	P_{157}	70.78%	P_{336}	68.46%	P_1	30.96%	P_{479}	46.99%	Null	Null

Comparison with associated cases

Similarity(N_1,P_{133})	49.98%	Negative	1	FA	P_{148}	Similar association
Similarity(N_1,P_{148})	89.69%	Positive	3	FA		
Similarity(N_1,P_{301})	28.51%	Negative	1	FA		

Updating the association of N_1

Case ID	Similar association						Dissimilar association					
	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
N_1	P_{148}	89.69%	P_{407}	75.69%	P_{157}	70.78%	P_{301}	28.51%	P_1	30.96%	P_{479}	46.99%

(a)

Case ID	Similar association						Dissimilar association					
	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
P_{14}	P_{410}	91.50%	P_{404}	91.40%	P_{256}	87.26%	P_{326}	15.97%	P_{17}	17.24%	P_{46}	18.67%

Comparison with associated cases

Similarity(N_1,P_{137})	73.04%	Positive	1	FA	P_{14}	Similar association
Similarity(N_1,P_{324})	58.57%	Positive	1	FA		
Similarity(N_1,P_{14})	88.13%	Positive	3	FA		

Updating the association of P_{14}

Case ID	Similar association						Dissimilar association					
	1st	sim	2nd	sim	3rd	sim	1st	sim	2nd	sim	3rd	sim
P_{14}	P_{410}	91.50%	P_{404}	91.40%	P_1	88.13%	P_{326}	15.97%	P_{17}	17.24%	P_{46}	18.67%

(b)

Figure 5. Examples of updating the association table: (**a**) scenario 1 and (**b**) scenario 2.

In Figure 5a, when N_1 is compared with P_{133}, P_{148} and P_{301}, it is detected that the similarity measurement between N_1 and P_{148} achieves the highest value. As a result, P_{148} takes the first position in the similar association of N_1. Meanwhile, N1's existing association with P_{407} and P_{157} has a minor adjustment by moving backward their positions in the association table. It works the same for updating the association with P_{301}.

In Figure 5b, the similar and dissimilar association of P_{14} is presented. During the iteration, P_{14} is compared with N_1. The comparative result indicates that N_1 has a closer association than P_{256} with P_{14}. Consequently, N_1 updates the third position in the similar association of P_{14} and P_{256} is therefore removed from the association table.

2.5. *Scalability of the Retrieval Algorithm and the Case Base*

It is necessary to consider the scalability issues since the number of cases in the case base may keep growing. On the one hand, for the retrieval algorithm, the number of positive and negative tokens can be increased as the size of the case base enlarges. The retrieval process remains the same presented in Section 3.3. On the other hand, the size of the case base would not increase infinitely. It is noted that we do not record the value of each variable for every day. The CBR system only stores useful past experiences. In other words, the system only remembers those variables in executed tasks with a complete pair of problem and solution features and the system does not record daily measurements. Therefore, the case base in a CBR system is quite different from those databases for social networks and weather stations. For instance, the weather station would store all the measured data. For an agricultural task like spraying the pesticide for rice, the average times are around 5 to 6 during the complete growing circle. Even when the farmland is divided into grids consisting of 50 blocks. The useful past experiences can be stored in the case base are maximum 300 pieces for a single farmland. If we have 50 farmlands in total, the maximum number stored in the case base is 15,000. In conclusion, the scalability of the retrieval algorithm and the case base would not be an obstacle for the CBR system.

3. Results and Discussions

In this section, extensive experiments are performed for verifying the effectiveness and efficiency of the proposed case retrieval algorithm from the following perspectives—(i) generation of the association table, (ii) result of case retrieval and iii) update of the association table.

3.1. Experimental Settings

The agricultural case-based reasoning system that adopts the proposed algorithm tries to retrieve the most similar past cases from a case base. As introduced in Section 3.1, our proposal is employed to manage pest problems. The pest considered in the experiment is Chilo suppressalis (CS), while the target crop is rice. Totally, 3000 past cases are stored in the case base and 500 new cases are prepared for testing purpose. The entry point is sequentially selected from the past cases for each retrieval task. Though this case retrieval algorithm is developed within a European research project, named Aggregate Farming in the Cloud (AFarCloud) (link to the project—http://www.afarcloud.eu/), the deployment of sensors and vehicles has not been fully completed yet. Therefore, simulated data are used currently and they are generated within a given range. For instance, judging from the current literature [50], the planting density of rice is generated from 180 to 525 seeds/m^2. The life cycle of rice can be categorized by “embryogenesis,” “vegetative,” “ripening” and “reproductive” stages [51], encoded by integers “1,” “2,” “3” and “4.” We are expecting to receive data from real farming fields in the near future once the devices are fully deployed. The simulation data we used can be found in the following link—https://github.com/ZhaoyuZHAI/Case-base. It is assumed that all the information is complete and there are no missing data of crop, pest and environmental features within all new and past cases.

3.2. Result of Generated Association Table

The association table is generated by comparing all past cases with each other sequentially. In this experiment, each past case is associated with three similar ones and three dissimilar ones, along with their similarity measurements. A part of this association table is given in Table 6, where ‘ . . . ’ hides the associations of past cases 6 to 2995. The full association table for all 3000 past cases in the case base can be found in the following link—https://github.com/ZhaoyuZHAI/Case-base/blob/master/associationTableWith3.

Table 6. Part of the generated association table.

Past Case	Similar Association						Dissimilar Association					
	1st	Sim	2nd	Sim	3rd	Sim	1st	Sim	2nd	Sim	3rd	Sim
1	541	94.53%	588	94.18%	2799	93.17%	1356	6.31%	1465	8.07%	2764	8.15%
2	981	93.20%	1707	92.68%	1931	91.32%	2264	7.50%	1446	9.50%	2027	9.89%
3	2107	93.95%	187	89.71%	1946	89.11%	2134	6.08%	184	6.67%	1911	7.11%
4	2687	93.85%	1193	93.81%	241	92.15%	272	6.89%	2095	7.51%	2990	8.14%
5	2454	97.14%	672	95.65%	2438	94.97%	964	17.58%	2470	18.09%	204	18.11%
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
2996	2104	92.91%	405	91.34%	2111	90.64%	1668	10.39%	3000	11.06%	2569	11.79%
2997	644	96.98%	563	95.31%	634	93.70%	328	12.18%	114	12.30%	1898	12.84%
2998	2600	99.50%	311	95.79%	1265	93.52%	328	7.10%	2458	7.76%	1316	8.12%
2999	1263	95.41%	246	94.67%	373	94.45%	1140	13.44%	1845	13.84%	2134	14.70%
3000	78	87.55%	2258	87.15%	553	86.52%	2863	5.28%	2430	5.49%	548	5.67%

Owing to the adequate coverage, each past case is associated for at least one time. It is worth noting that one past case can be associated with others several times, depending on the similarity measurements. For instance, in Table 6, the past case 328 is associated with both past cases 2997 and

2998, achieving at 12.18% and 7.10% respectively. However, these associations do not guarantee that past cases 2997 and 2998 are similar. For better demonstration, Figure 6 displays the data visualization of past cases 328, 2997 and 2998 after normalization.

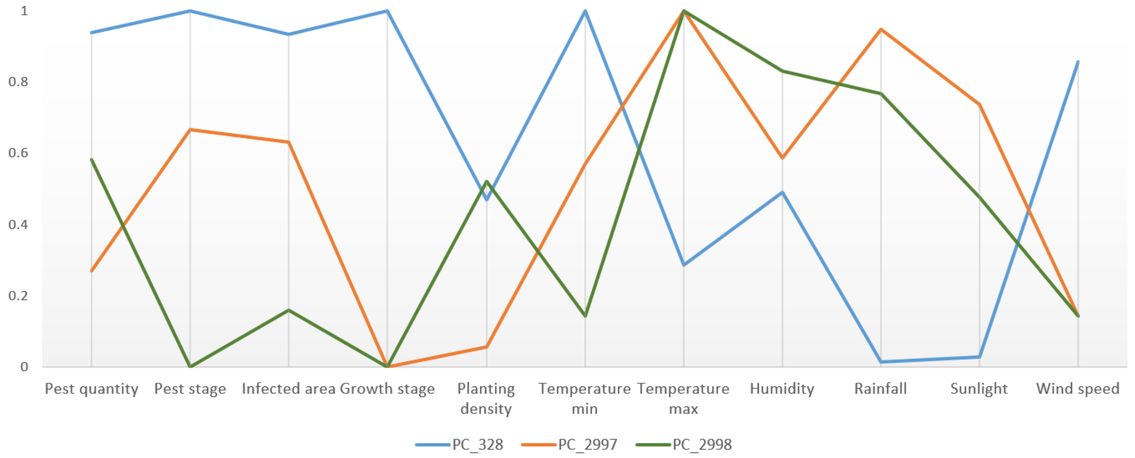


Figure 6. Data visualization of past cases 328, 2997 and 2998 after normalization.

Figure 6 demonstrates that though both past cases 2997 and 2998 are dissimilar to the past case 328, they still have a major difference among data like pest stage, infected area and planting density. Actually, the similarity measurement between past cases 2997 and 2998 only achieves at 67.08%.

Another interesting fact we noticed is that one past case may appear in the similar association of others more than one time. For example, in the association table, the past case 984 appeared in the similar association of past cases 748 and 749 respectively. The data visualization of these three past cases is displayed in Figure 7. For evaluating the commonalities of these three cases, the data covariation [52] is used for analyses. The result is presented in Table 7.

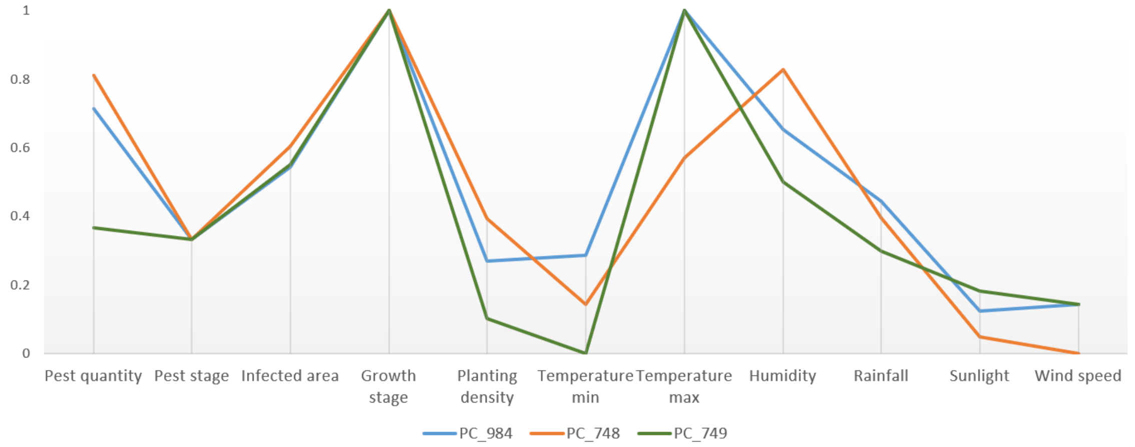


Figure 7. Data visualization of past cases 984, 748 and 749 after normalization.

Table 7. Statistical analysis of past cases 984, 748 and 749.

Cases	Similarity Measurement	Data Covariation
(P_{984}, P_{748})	95.76%	0.0890
(P_{984}, P_{749})	95.79%	0.0919
(P_{748}, P_{749})	86.78%	0.0966

In Figure 7 and Table 7, the result shows that past cases 984, 748 and 749 all have great commonalities and their data inflections match with each other. According to the data covariation specification, if the covariation value is positive, it means that the data distribution of compared cases is the same. Meanwhile, a smaller covariation value indicates a closer correlation between cases. From the result in Table 7, the data covariation is positive and the values have a tiny difference.

As a consequence, two conclusions can be drawn according to the result of above examples.

- If the past case P_x is stored in the dissimilar association of past cases P_y and P_z at the same time, it is not guaranteed that past cases P_y and P_z are similar with each other.
- If the past case P_x is stored in the similar association of past cases P_y and P_z at the same time, then past cases P_y and P_z are potentially similar with each other.

This is the reason why the proposed case retrieval algorithm tries to compare the associated past cases preferentially, instead of traversing all past cases in the case base. Under general circumstances, the potential target case usually exists among the association.

3.3. Result of Case Retrieval

The proposed case retrieval algorithm is compared with the typical algorithm which traverses all the past cases in the case base. For the result of case retrieval, we mainly evaluate it through two aspects—retrieval accuracy and efficiency. On the one hand, retrieval accuracy specifies that the retrieved past case should be as similar as possible to the target. On the other hand, retrieval efficiency specifies that the number of compared past cases should be as fewer as possible. Please note that the new cases are not retained in the case base after case retrieval and the association table is not updated during the experiments in this section. For each new case, we tried to use each past case as the entry-point case for testing. Thus, the total times of tests are 1.5 million (3000 × 500). By this design, we are able to verify whether the selection of the entry-point case has any effects on the performance of the proposed retrieval algorithm.

Firstly, retrieval accuracy concerns the average precision of retrieved top three similar cases. The formula of the average precision is given in Equation (1).

$$\text{Average precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}\ (\%) \qquad (1)$$

where TP means true positive and FP stands for false positive.

The result of the average precision is displayed in Figure 8.

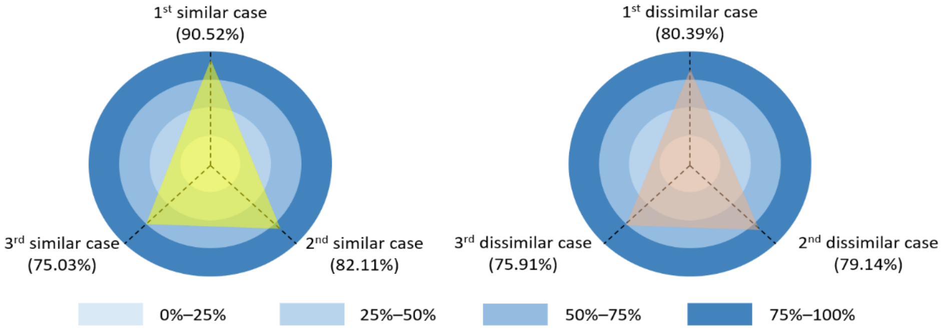


Figure 8. The average precision of retrieved top three similar and dissimilar past cases (without retaining new cases).

In Figure 8, the average precision of retrieved top three similar cases achieves at 90.52% (1,357,804/1,500,000), 82.11% (1,231,654/1,500,000) and 75.03% (1,125,449/1,500,000). The average precision of retrieved top three dissimilar cases achieves at 80.39% (1,205,858/1,500,000), 79.14% (1,187,119/1,500,000) and 75.91% (1,138,655/1,500,000). The result of the average precision demonstrates that the proposed case retrieval algorithm achieves promising retrieval accuracy. Meanwhile, the selection of the entry-point case has minor influence on the performance of the retrieval algorithm.

Under certain circumstances, the proposed case retrieval algorithm fails to retrieve the correct top three similar cases. For instance, the proposed algorithm is unable to identify the most similar past case. This most similar one was missed during runtime due to the limitation of retrieving time (iteration number). The second most similar past case usually takes the first position instead and is therefore treated as the output. This is also the reason why the second and the third similar past cases are not 100% correctly retrieved. However, this is acceptable because case-based reasoning does not necessarily require the successful retrieval of the most similar cases. All retrieved top three similar past cases are not exactly the same as the target one [53]. It is worth noting that apart from case retrieval, case-based reasoning also adopts the processes of solution reuse and revision. These processes are responsible to update the solutions of retrieved past cases for adapting to the current situation. Therefore, successfully retrieving the second or the third similar past cases is enough for the CBR-based systems. For supporting this point of view, an example is presented in Figures 9–11, displaying the data visualization of the new case 7 and retrieved top three similar past cases 52, 267 and 646. The statistical analysis of these cases is given in Table 7. The data covariance function, the root mean square error (RMSE) and the mean absolute error (MAE) are adopted here.

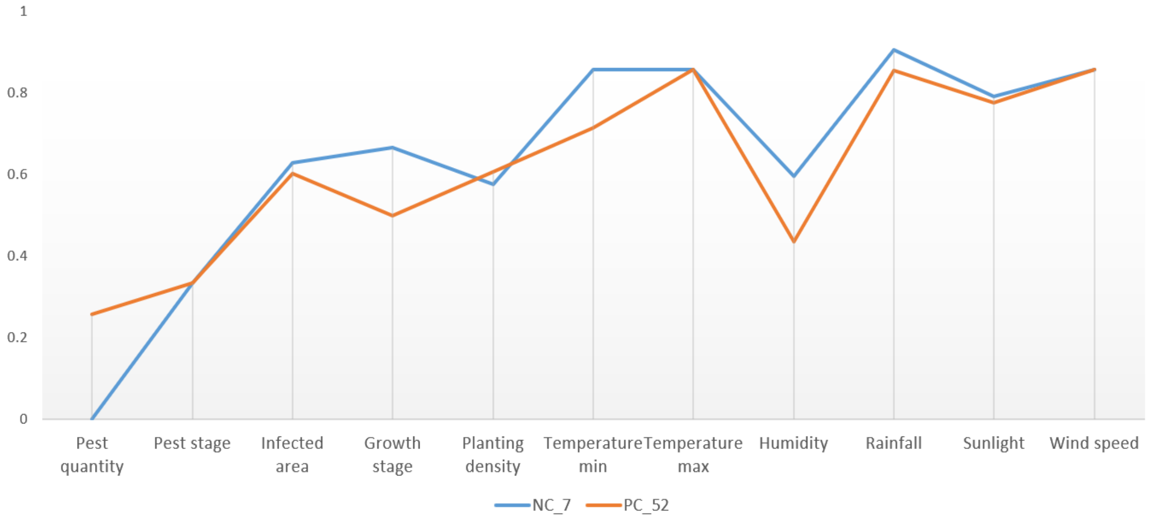


Figure 9. Data visualization of the new case 7 and the retrieved most similar case (past case 52) after normalization.

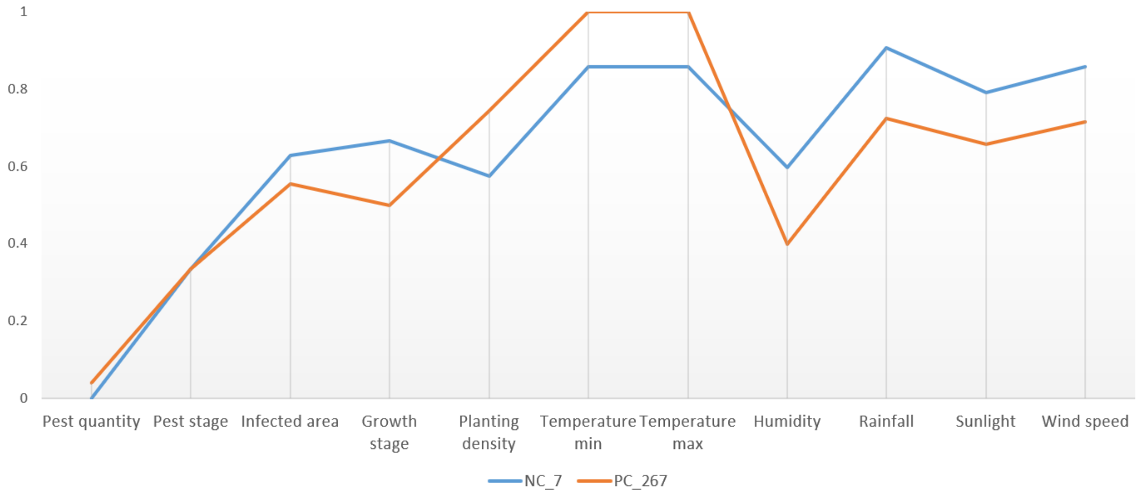


Figure 10. Data visualization of the new case 7 and the retrieved most similar case (past case 267) after normalization.

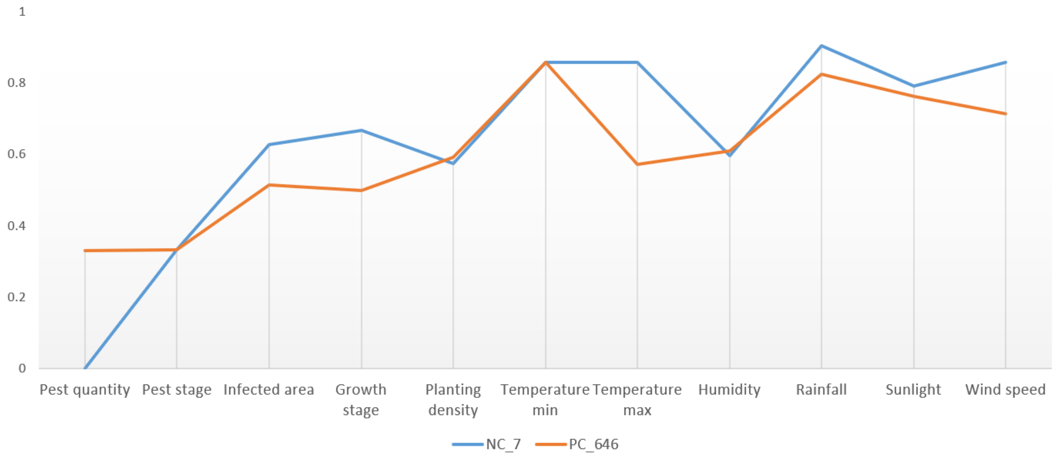


Figure 11. Data visualization of the new case 7 and the retrieved most similar case (past case 646) after normalization.

From the result in Figures 9–11 and Table 8, it is concluded that the new case 7 has great commonalities with retrieved past cases 52, 267 and 646. The similarity measurements achieve at 95.28%, 95.14% and 95.09% respectively. Meanwhile, the values of data covariation are positive, showing a closer correlation between the new case and past cases. In other words, there are minor differences between the retrieved top three similar past cases. The solutions of all these three cases can be reused and revised. Therefore, retrieval accuracy of the proposed algorithm is proved.

Table 8. Statistical analysis of the new case 7 and retrieved top three similar past cases.

Cases	Similarity Measurement	Data Covariation	RMSE	MAE
(P_7, P_{52})	95.28%	0.0535	0.0215	0.0194
(P_7, P_{267})	95.14%	0.0677	0.0248	0.0237
(P_7, P_{646})	95.09%	0.0682	0.0293	0.0251

Secondly, we evaluate retrieval efficiency of the proposed algorithm by being compared with the typical case retrieval algorithms. Traditionally, the typical case retrieval algorithms try to identify the most similar past cases by traversing the whole data base. As a consequence, the number of compared cases in this experiment reaches 3000 in total for a single search. Differing from typical approaches, the proposed case retrieval algorithm takes advantage of the association table and therefore measures the similarity between the new case and associated cases preferentially. From the evaluation perspective, a fewer number of compared cases indicates greater efficiency of case retrieval. The result of the number of travelled cases for 1.5 million tests is summarized in Table 9.

Table 9. The number of travelled cases for 1.5 million tests.

Number of Travelled Cases	(800,899)	(900,999)	(1000,1099)	(1100,1199)
Times	320,359	479,221	493,057	207,363
Proportion	21.36%	31.95%	32.87%	13.82%

In Table 9, the result demonstrates that the number of compared past cases ranges from 800 to 1197. The least number of compared cases is 834 while the largest number is 1197. The average number of compared cases is around 1047 (1046.95). As a consequence, compared with the result of typical case retrieval algorithms (3000 compared cases), it is proved that the proposed algorithm is able to retrieve similar past cases by fewer comparison. In other words, it has greater retrieval efficiency.

Overall, the proposed algorithm enables case retrieval with great accuracy and efficiency. After successful case retrieval, the solutions of retrieved past cases can be reused and revised to resolve the new problems.

3.4. Result of Updated Association Table

During the experiments in this section, new cases will be retained in the case base after successful retrieval and the association table will be updated accordingly. Since there are 500 new cases for testing, the experiment in this section is performed for 500 times. In each test, the past case 1 is selected as the entry point (entry case) for comparison at the initial iteration. The average precision of these 500 retrieval tasks is shown in Figure 12.

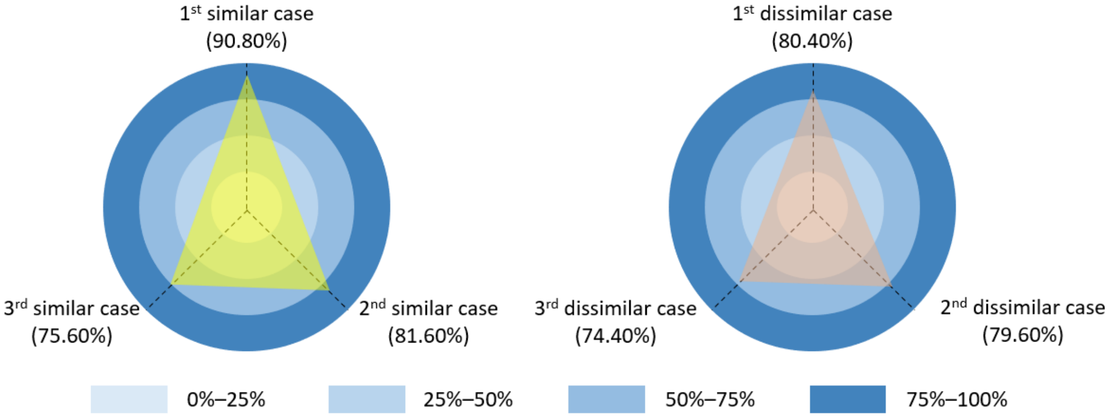


Figure 12. The average precision of retrieved top three similar and dissimilar past cases (new cases are retained).

In Figure 12, the average precision of retrieved top three similar cases achieves at 90.80% (454/500), 82.11% (408/500) and 75.60% (378/500). The average precision of retrieved top three dissimilar cases achieves at 80.40% (402/500), 79.60% (398/500) and 74.40% (372/500). The result of average precision indicates that the proposed retrieval algorithm is able to guarantee the retrieval accuracy after new cases are retained in the case base.

After performing the retrieval tasks for all 500 new cases, the result shows that 30 new cases are not retained in the case base because the similarity measurement between these cases and existed ones achieves beyond 98.00%. These 30 new cases are listed in Table 10.

Table 10. List of unretained new cases.

Unretained Cases	Cause	Unretained Cases	Cause
N_{1}	Sim (N_{1}, P_{158}) = 98.35%	N_{183}	Sim (N_{183}, P_{542}) = 98.96%
N_{4}	Sim (N_{4}, P_{2210}) = 98.65%	N_{190}	Sim (N_{190}, P_{2468}) = 98.73%
N_{27}	Sim (N_{27}, P_{1380}) = 98.14%	N_{307}	Sim (N_{307}, P_{460}) = 98.25%
N_{53}	Sim (N_{53}, P_{2361}) = 98.02%	N_{324}	Sim (N_{324}, P_{547}) = 98.27%
N_{81}	Sim (N_{81}, P_{2943}) = 98.45%	N_{365}	Sim (N_{365}, P_{1440}) = 98.18%
N_{82}	Sim (N_{82}, P_{1944}) = 99.56%	N_{367}	Sim (N_{367}, P_{84}) = 98.44%
N_{84}	Sim (N_{84}, P_{1326}) = 98.16%	N_{376}	Sim (N_{376}, P_{1353}) = 98.41%
N_{88}	Sim (N_{88}, P_{411}) = 98.14%	N_{399}	Sim (N_{399}, P_{3185}) = 98.58%
N_{91}	Sim (N_{91}, P_{299}) = 98.80%	N_{411}	Sim (N_{411}, P_{2237}) = 98.51%
N_{99}	Sim (N_{99}, P_{196}) = 98.29%	N_{437}	Sim (N_{437}, P_{921}) = 99.30%
N_{102}	Sim (N_{102}, P_{2921}) = 99.28%	N_{447}	Sim (N_{447}, P_{1095}) = 99.07%
N_{125}	Sim (N_{125}, P_{1551}) = 98.35%	N_{459}	Sim (N_{459}, P_{388}) = 98.33%
N_{138}	Sim (N_{138}, P_{3057}) = 98.06%	N_{476}	Sim (N_{476}, P_{1358}) = 98.13%
N_{150}	Sim (N_{150}, P_{324}) = 98.49%	N_{486}	Sim (N_{486}, P_{2069}) = 98.68%
N_{162}	Sim (N_{162}, P_{1883}) = 98.77%	N_{496}	Sim (N_{496}, P_{372}) = 99.07%

In Table 10, it is worth mentioning that the forgetting strategy is applied to new cases 138 and 399 due to the reason that these two cases have great commonalities with newly retained cases 3057 and 3185 respectively. The rest of unretained cases is all similar to the past cases which have been already stored in the case base.

For verifying the updated association table, the times of updates in both similar and dissimilar associations are counted, shown in Figure 13.

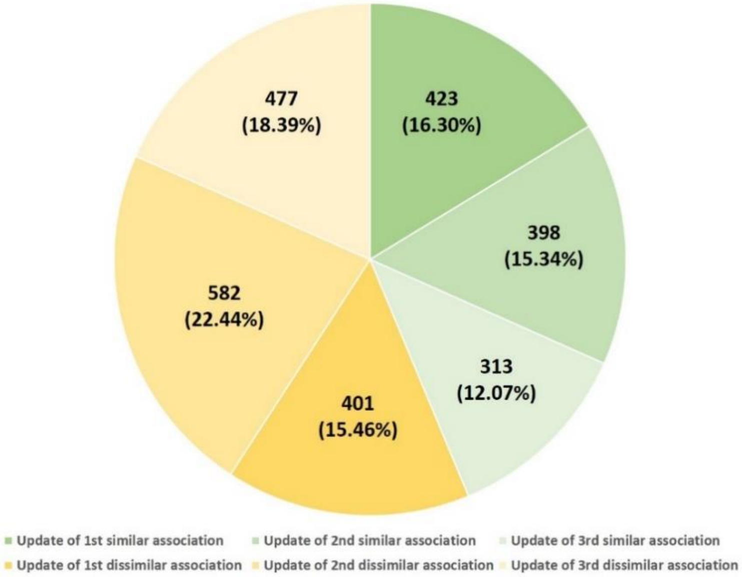


Figure 13. Times of updates in the association table.

In Figure 13, updates in the association table are counted 2594 in total. 43.71% happens in updating the similar association while 56.29% in the dissimilar association.

Lastly, the association table also concerns the similar and dissimilar associations of newly retained cases, which are presented in Table 11, where '... ' hides the association of past cases 3006 to 3465.

Table 11. Part of the generated association table for newly retained cases.

Past Case	Similar Association						Dissimilar Association					
	1st	Sim	2nd	Sim	3rd	Sim	1st	Sim	2nd	Sim	3rd	Sim
3001	679	94.91%	555	94.19%	2371	94.11%	1845	12.07%	2095	12.75%	685	13.03%
3002	1398	96.78%	3423	96.03%	1956	95.82%	2807	18.75%	2095	20.83%	490	21.70%
3003	539	94.56%	1266	92.82%	2799	91.31%	2396	11.95%	2745	13.27%	1132	13.36%
3004	1119	95.17%	3400	94.83%	365	94.40%	1217	14.72%	1557	14.93%	2863	15.11%
3005	1135	94.69%	386	94.51%	53	93.29%	38	14.48%	3478	14.69%	1663	14.82%
...	...	...	...	...	...	...	...	...	...	...	...	...
3466	33	96.31%	3253	93.46%	3252	93.26%	1278	16.07%	3308	16.48%	2990	18.00%
3467	542	94.78%	1846	94.76%	3183	94.36%	2774	10.81%	1079	13.97%	2027	14.17%
3468	2199	97.38%	2167	94.12%	782	93.45%	3472	11.35%	3309	11.44%	674	12.75%
3469	2523	96.24%	2860	95.84%	2749	93.84%	2095	7.21%	564	9.39%	396	10.57%
3470	2995	95.91%	2882	94.79%	2042	94.14%	3218	13.61%	3434	13.77%	1239	13.92%

In total, 470 newly retained cases have their own similar and dissimilar associations, which means the updates in the association table are successful.

4. Conclusions

Typical approaches of case retrieval try to match the most similar past cases by traversing the whole case base, leading to low efficiency when a large volume of cases is stored. Therefore, this paper focuses on proposing a case retrieval algorithm for agricultural case-based reasoning systems. Before performing the retrieval tasks, an association table is constructed, consisting of both the similar and dissimilar relationships between all past cases. The novelty of our proposal lies on selecting associated cases from the table and evaluating their similarity between the new case preferentially. Under this circumstance, the proposed case retrieval algorithm is able to retrieve similar past cases by comparing fewer cases in the case base. Our proposal also concerns the retention part in the loop of case-based reasoning. The association table is updated during runtime. After successful retrieval, the new case is retained in the case base, along with its similar and dissimilar associations when the similarity measurement between the new case and the retrieved most similar past case is smaller than 98.00%. Meanwhile, the associations of past cases are updated as well. The experimental result demonstrates that our proposal is able to retrieve similar past cases with great efficiency and accuracy. The case base is successfully maintained with newly retained cases and their associations.

It is acknowledged that case retrieval is one of the most significant parts in case-based reasoning. Because the rest of processes like reuse and revision cannot proceed further without successful case retrieval. Thus, the proposed case retrieval algorithm is not only useful in CBR enabled agricultural systems but also has great potential in CBR systems for other domains. With the efficient retrieval capability, a CBR based ADSS enables to provide farmers with quick decision supports about agricultural management.

Since this work was developed within the AFarCloud project, we are expecting to receive real data from the farms to verify the proposed case retrieval algorithm. For further improving the performance of the proposed retrieval algorithm, it is also worth looking into the selection of preferable entry-point case. By classifying similar cases in a single cluster and selecting the most representative case as the entry-point to be compared with the new case, it might potentially improve the performance of the algorithm. Furthermore, it might be helpful for improving the algorithm performance if we could set the range of similarity levels (presented in Table 4) more precisely. Lastly, it would be interesting to investigate the performance of the proposed algorithm when the size of the case base increases to a larger magnitude. Under such circumstances, the number of similar and dissimilar associations are supposed to increase as well.

Author Contributions: Conceptualization, Z.Z.; methodology, Z.Z.; validation, Z.Z. and N.L.M.; formal analysis, Z.Z.; investigation, Z.Z.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z., J.-F.M.O., N.L.M and H.X.; visualization, Z.Z.; supervision, J.-F.M.O.; project administration, J.-F.M.O.; funding acquisition, J.-F.M.O. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to the presented results has been undertaken within the AFARCLOUD European Project (Aggregate Farming in the Cloud), under Grant Agreement No. 783221-AFarCloud-H2020- ECSEL-2017-2 and supported in part by the ECSEL JU and in part by the Spanish Ministry of Science, Innovation and Universities under Grant PCI2018-092965. This work has been also partially supported by the China Scholarship Council (CSC).

Acknowledgments: We would like to thank the editors and anonymous reviewers for their suggestions and comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tummers, J.; Kassahun, A.; Tekinerdogan, B. Obstacles and features of farm management information systems: A systematic literature review. *Comput. Electron. Agric.* **2019**, *157*, 189–204. [CrossRef]
2. Mkhaliphi, P.V.; Dlamini, N.M.; Sifundza, J.T. The impact of improving irrigation scheduling for smallholder growers in Swaziland. *Int. Sugar J.* **2016**, *118*, 284–292.
3. Bernier, M.H.; Madramootoo, C.A.; Mehdi, B.B.; Gollamudi, A. Assessing on-farm irrigation water use efficiency in Southern Ontario. *Can. Water Resour. J.* **2010**, *35*, 115–130. [CrossRef]
4. Damos, P. Modular structure of web-based decision support systems for integrated pest management. A review. *Agron. Sustain. Dev.* **2015**, *35*, 1347–1372. [CrossRef]
5. Hughes, G. The application of decision theory in pest and disease management. In Proceedings of the BCPC International Conference on Pests and Diseases, Brighton, UK, 18–21 November 2002.
6. Vagstad, N. Nutrient management for integrating productivity and environmental concerns–Framework of a joint China-Norway research initiative. *ACTA Agric. Scand. B Soil Plant Sci.* **2014**, *63*, 105–108. [CrossRef]

7. Serrano, J.; da Silva, J.M.; Shahidian, S.; Silva, L.L.; Sousa, A.; Baptista, F. Differential vineyard fertilizer management based on nutrient's spatio-temporal variability. *J. Soil Sci. Plant Nutr.* **2017**, *17*, 46–61.
8. Ma, W.L.; Renwick, A.; Grafton, Q. Farm machinery use, off-farm employment and farm performance in China. *Aust. J. Agric. Resour. Econ.* **2018**, *62*, 279–298. [CrossRef]
9. Faggion, F.; Neto, P.A.R.; Correia, T.P.D.; Martin, S. Feasibility of machinery transfer for growing grain in distant areas. *Appl. Res. Agron.* **2016**, *9*, 17–26.
10. Pannakkong, W.; Buddhakulsomsiri, J.; Parthanadee, P. Simulation modeling analysis to support decision making of Cassava harvesting in Thailand. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, 10–13 December 2013.
11. Poldaru, R.; Roots, J. Using a nonlinear stochastic model to schedule silage maize harvesting on Estonian farms. *Comput. Electron. Agric.* **2014**, *107*, 89–96. [CrossRef]
12. Hong, H.X.; Wang, C. An empirical research on the development of level of the agricultural products logistics in the region of Sichuan Province. In Proceedings of the 4th International Conference on Education Technology, Management and Humanities Science, Taiyuan, Shanxi, China, 27–28 February 2018.
13. Wang, L.P. Study on agricultural products logistics mode in Henan Province of China. In Proceedings of the Pacific-Asia Conference on Knowledge Engineering and Software Engineering, Shenzhen, China, 19–20 December 2009.
14. Deepa, N.; Ganesan, K. Decision-making tool for crop selection for agriculture development. *Neural Comput. Appl.* **2019**, *31*, 1215–1225. [CrossRef]
15. Zhang, C.; Hu, R.F.; Shi, G.M.; Jin, Y.H.; Robson, M.G.; Huang, X.S. Overuse or underuse? An observation of pesticide use in China. *Sci. Total Environ.* **2015**, *538*, 1–6. [CrossRef] [PubMed]
16. Tzounis, A.; Katsoulas, N.; Bartzanas, T.; Kittas, C. Internet of Things in agriculture, recent advances and future challenges. *Biosyst. Eng.* **2017**, *164*, 31–48. [CrossRef]
17. Jayaraman, P.P.; Yavari, A.; Georgakopoulos, D.; Morshed, A.; Zaslavsky, A. Internet of Things platform for smart farming: Experiences and lessons learnt. *Sensors* **2016**, *16*, 1884. [CrossRef] [PubMed]
18. Zhai, Z.Y.; Martinez, J.F.; Beltran, V.; Martinez, N.L. Decision support systems for agriculture 4.0: Survey and challenges. *Comput. Electron. Agric.* **2020**, *170*, 105256. [CrossRef]
19. Lindblom, J.; Lundstrom, C.; Ljung, M.; Jonsson, A. Promoting sustainable intensification in precision agriculture: Review of decision support systems development and strategies. *Precis. Agric.* **2017**, *18*, 309–331. [CrossRef]
20. Le Ber, F.; Napoli, A.; Metzger, J.L.; Lardon, S. Modeling and comparing farm maps using graphs and case-based reasoning. *J. Univers. Comput. Sci.* **2003**, *9*, 1073–1095.
21. Evans, J.; Terhorst, A.; Kang, B.H. From data to decisions: Helping crop producers build their actionable knowledge. *Crit. Rev. Plant Sci.* **2017**, *36*, 71–88. [CrossRef]
22. Shih, M.L.; Huang, B.W.; Chiu, N.H.; Chiu, C.; Hu, W.Y. Farm price prediction using case-based reasoning approach - A case of broiler industry in Taiwan. *Comput. Electron. Agric.* **2009**, *66*, 70–75. [CrossRef]
23. Du, Y.; Liang, F.; Sun, Y. Integrating spatial relations into case-based reasoning to solve geographic problems. *Knowl. Based Syst.* **2012**, *33*, 111–123. [CrossRef]
24. Li, H.; Song, Y.; Li, X.P.; Liu, Q.X.; Zhu, Y.F. Research of CBR retrieval method based on rough set theory. In Proceedings of the 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 September 2015.
25. Su, W.B.; Lei, Z.F. Rough case-based reasoning system for continues casting. In Proceedings of the 10th International Conference on Machine Vision (ICMV), Vienna, Austria, 13–15 November 2017.
26. Singh, K.; Kansal, A.; Singh, G. An improved median filtering anti-forensics with better image quality and forensic undetectability. Multidimens. *Syst. Signal Process.* **2019**, *30*, 1951–1974.
27. Rahman, M.M.; Antani, S.K.; Thoma, G.R. A learning-based similarity fusion and filtering approach for biomedical image retrieval using SVM classification and relevance feedback. *IEEE Trans. Inf. Technol. Biomed.* **2011**, *15*, 640–646. [CrossRef]
28. Coletti, G.; Bouhchon-Meunier, B. A study of similarity measures through the paradigm of measurement theory: The classic case. *Soft Comput.* **2019**, *23*, 6827–6845. [CrossRef]
29. Wang, H.Q.; Sun, B.B.; Shen, X.F. Hybrid similarity measure for retrieval in case-based reasoning systems and its applications for computer numerical control turret design. *Proc. Inst. Mech. Eng. B J. Eng. Manuf.* **2018**, *232*, 918–927. [CrossRef]

30. Yoon, S.H.; Kim, S.W.; Park, S. C-Rank: A link-based similarity measure for scientific literature databases. *Inf. Sci.* **2016**, *326*, 25–40. [CrossRef]
31. Yazid, H.; Kalti, K.; Benamara, N.E. A new similarity measure based on Bayesian Network signature correspondence for brain tumours cases retrieval. *Int. J. Comput. Intell. Syst.* **2014**, *7*, 1123–1136. [CrossRef]
32. Zhai, Z.Y.; Ortega, J.F.M.; Castillejo, P.; Beltran, V. A triangular similarity measure for case retrieval in CBR and its application to an agricultural decision support system. *Sensors* **2019**, *19*, 4605. [CrossRef]
33. Jiang, Y.C.; Yang, M.X.; Qu, R. Semantic similarity measures for formal concept analysis using linked data and WordNet. *Multimed. Tools Appl.* **2019**, *78*, 19807–19837. [CrossRef]
34. Farhan, U.; Tolouei-Rad, M.; Osseiran, A. Indexing and retrieval using case-based reasoning in special purpose machine designs. *Int. J. Adv. Manuf. Technol.* **2017**, *92*, 2689–2703. [CrossRef]
35. Honigl, J.; Kung, J. A data quality index with respect to case bases within case-based reasoning. In Proceedings of the 6th Asian Conference on Intelligence Information and Database Systems (ACIIDS), Bangkok, Thailand, 7–9 April 2014.
36. Wiltgen, B.; Goel, A.K.; Vattam, S. Representation, indexing, and retrieval of biological cases for biologically inspired design. In Proceedings of the 19th International Conference on Case-Based Reasoning, London, UK, 11–14 September 2011.
37. Ahmad, J.; Sajjad, M.; Mehmood, I.; Baik, S.W. SiNC: Saliency-injected neural codes for representation and efficient retrieval of medical radiographs. *PLoS ONE* **2017**, *12*, e0181707. [CrossRef]
38. Durmaz, O.; Bilge, H.S. Fast image similarity search by distributed locality sensitive hashing. *Pattern Recognit. Lett.* **2019**, *128*, 361–369. [CrossRef]
39. Ahmed, T.; Sarma, M. Hash-based space partitioning approach to iris biometric data indexing. *Expert Syst. Appl.* **2019**, *134*, 1–13. [CrossRef]
40. Aydar, M.; Ayvaz, S. An improved method of locality-sensitive hashing for scalable instance matching. *Knowl. Inf. Syst.* **2019**, *58*, 275–294. [CrossRef]
41. Bergmann, R.; Kolodner, J.; Plaza, E. Representation in case-based reasoning. *Knowl. Eng. Rev.* **2005**, *20*, 209–213. [CrossRef]
42. Decision Support System Data for Farmer Decision Making. Available online: https://pdfs.semanticscholar.org/6304/fb7c0884183d00629af93bf1e05de5e3d0da.pdf (accessed on 10 December 2019).
43. Zhang, J.; Wang, Y.M.; Lin, Y.; Zhang, K. Hybrid multi-attribute case retrieval method based on intuitionistic fuzzy and evidence reasoning. *J. Intell. Fuzzy Syst.* **2019**, *36*, 271–282. [CrossRef]
44. Fan, Z.P.; Li, Y.H.; Wang, X.H.; Liu, Y. Hybrid similarity measure for case retrieval in CBR and its application to emergency response towards gas explosion. *Expert Syst. Appl.* **2014**, *41*, 2526–2534. [CrossRef]
45. Standardize or Normalize?—Examples in Python. Available online: https://medium.com/@rrfd/standardize-or-normalize-examples-in-python-e3f174b65dfc (accessed on 10 December 2019).
46. Huang, Z.H.; Fan, H.Q.; Shen, L.Y. Case-based reasoning for selection of the best practices in low-carbon city development. *Front. Eng. Manag.* **2019**, *6*, 416–432. [CrossRef]
47. Khan, M.J.; Hayat, H.; Awan, I. Hybrid case-base maintenance approach for modeling large scale case-based reasoning systems. *Hum. Cent. Comput. Inf.* **2019**, *9*, 9. [CrossRef]
48. Yan, A.J.; Qian, L.M.; Zhang, C.X. Memory and forgetting: An improved dynamic maintenance method for case-based reasoning. *Inf. Sci.* **2014**, *287*, 50–60. [CrossRef]
49. Salamo, M.; Lopez-Sanchez, M. Adaptive case-based reasoning using retention and forgetting strategies. *Knowl. Based Syst.* **2011**, *24*, 230–247. [CrossRef]
50. Sonderskov, M.; Fritzsche, R.; del Mol, F.; Gerowitt, B.; Goltermann, S.; Kierzek, R.; Krawczyk, R.; Bojer, O.M.; Rydahl, P. DSSHerbicide: Weed control in winter wheat with a decision support system in three South Baltic regions–Field experimental results. *Crop Prot.* **2015**, *76*, 15–23. [CrossRef]
51. Itoh, J.; Nonomura, K.; Ikeda, K.; Yamaki, S.; Inukai, Y.; Yamagishi, H.; Kitano, H.; Nagato, Y. Rice plant development: From zygote to spikelet. *Plant Cell Physiol.* **2005**, *46*, 23–47. [CrossRef] [PubMed]
52. Cobb, P.; McClain, K.; Gravemeijer, K. Learning about statistical covariation. *Cogn. Instr.* **2003**, *21*, 1–78. [CrossRef]
53. Lu, J.; Zhang, X.K.; Li, P.R.; Zhu, Y. Case-based FCFT reasoning system. *Appl. Sci.* **2015**, *5*, 825–839. [CrossRef]