Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

A Multilayered Clustering Framework to build a Service Portfolio using Swarm-based algorithms

I. R. Praveen Joe & P. Varalakshmi

To cite this article: I. R. Praveen Joe & P. Varalakshmi (2019) A Multilayered Clustering Framework to build a Service Portfolio using Swarm-based algorithms, Automatika, 60:3, 294-304, DOI: 10.1080/00051144.2019.1590951

To link to this article: https://doi.org/10.1080/00051144.2019.1590951

Published online: 13 Jun 2019.

AUTOMATIKA
2019, VOL. 60, NO. 3, 294–304
https://doi.org/10.1080/00051144.2019.1590951

SPECIAL ISSUE: COMPUTATIONAL INTELLIGENCE AND CAPSULE NETWORKS

OPEN ACCESS

A Multilayered Clustering Framework to build a Service Portfolio using Swarm-based algorithms

I. R. Praveen Joe[a] and P. Varalakshmi[b]

[a]Department of CSE, KCG College of Technology, Chennai, India; [b]Department of CT, MIT, Anna University, Chennai, India

ABSTRACT

In this paper, a multilayered clustering framework is proposed to build a service portfolio to select web services of choice. It is important for every service provider to create a service portfolio in order to facilitate the service selection process for someone to obtain the desired service in the absence of public UDDI registries. To address this problem, a multilayered clustering approach applied on a variety of data pertaining to web services in order to filter and group the services of a similar kind which in turn will improve the leniency in the process of service selection is used. The advantages of the layer approach are reduced search space, combination of incremental learning and competitive learning strategies, reduced computational labour, scalability, robustness and fault tolerance. The results are subjected to cluster analysis to verify their degree of compactness and isolation and appropriate evaluation indices are used. The results were found passable with an improved degree of similarity.

ARTICLE HISTORY
Received 18 December 2018
Accepted 25 February 2019

KEYWORDS
Web services; Multilayered clustering; service portfolio; service selection

1. Introduction

A Service provider in general has to do the following to put a service available for use in a conventional approach. Firstly, the provider has to decide on the service he needs to provide, then choose a registry/registries for uploading information about the service. Next, decide on how to list the service at the registry and finally provide an explicit definition on how an user can connect to the service.

However, as public registries are closed it becomes essential for service providers to make all the web service descriptions that are published available in order to choose or search a web service preferably in a proprietary portal. The service requester initiates a query by specifying his/her requirements. So, there must exist a service match maker, a broker that matches the request up on comparison with the published services and a recommendation is provided which contains a set of web services that match the needs by identifying the degree of similarity. Thus the discovery process is made successful defending on the maturity and capability of the matching process Therefore in order to make the search process easier an arrangement and organization of services grouping them based on certain relevancy factors is an indispensable factor.

In this context two issues are of primary concern, One, a procedure for categorization of services through efficient clustering techniques in order to facilitate any match making process to fetch the right service based on the requirements. The other is data pertaining to the services to be considered for categorizing the services.

The present research summarizes three approaches addressing the first issue and four categories of data in order to address the second issue.

2. Literature survey

2.1. Common approaches for web services clustering

Five different approaches for web services clustering are presented in this survey.

(a) Syntactic vs. Semantic Clustering

One of the popular strategies that are used for web services analysis is though syntactic structure [1]. Selected research has sought to the intensification of the discovery of web services with search engines by recommending proven approaches to clustering of WSDL descriptions into functionally-alike sets before responding to discovery requests [2]. This methodology mines the WSDL to extract attributes that define the semantics and performance of the web service, which reveal the functionality of the service and after which suitable text mining strategies are applied [3].

CONTACT I. R. Praveen Joe praveenjoeir@yahoo.com

(b) Functional vs. Non-Functional Clustering

There may be several service providers who may be dealing with the similar functionalities defined in a service interface [4]. Identifying and choosing the best service is an important task for service requesters. The particulars in WSDL descriptions are not sufficient for ranking the best services [5]. Non-functional properties together with description of cost, performance, security, and trustiness of a service are presented for computing the Quality of Services (QoS) [6]. There are numerous attributes of QoS that can be equipped into categories with a set of measurable parameters. The "best" service may have diverse implications for diverse requesters. One may prefer security to cost, while the other may prefer lower cost to performance [7]. Measurements of these non-functional properties can be attained using statistical analysis, data mining, and text mining technologies [8]. It is generally prepared by a third-party through the assembly of subjective evaluations from requesters. This data vigorously changes over time [9].

(c) Biologically Inspired Clustering

In addition to the classical and conventional approaches, researchers contemplate the use of biologically inspired methods for clustering [10]. It was stated that the clustering method based on Particle Swarm Optimization is better than partitioning clustering as it avoids the problem of local optima stagnation [11]. Ant colony optimization yields better results than classical clustering methods. ACO based clustering does not require to know in advance the number of clusters and the obtained clusters have a higher quality [12]. A hybrid technique, Tree Traversing Ant (TTA), combines features of ant based clustering with features of classical clustering techniques [13]. In the case of service clustering, a method based on TTA is applied that considers the services' syntactic descriptions as clustering norms.

(d) Taxonomic Clustering

Certain research works recommend the use of taxonomic clustering algorithm that groups web services based on their functional similarity [14]. This clustering scheme proceeds into attention not only discrete factors such as input or output of service operations, but also the latent inter-relationships among the discrete factors [15]. When a set of services are given, that may or may not have been categorized, individual methods to handle the issue and mark out their classification labels in terms of a common (given) taxonomy, such as UNSPSC is adopted [16]. When a new service description is published, the unclassified service is compared with the classified ones and measures of the likelihood that the new service description belonging to each cluster are calculated [17]. Pertaining to this calculation, the service will be assigned to a suitable category.

(e) Fuzzy Clustering

With a different perspective on web services clustering, a proposal of fuzzy clustering of web services grounded on quality of service is presented [18]. It delivers a description of how web services' quality of service data can be clustered fuzzily using unsupervised methods. The fuzzy clustering of web services could help requestors who have limited technical knowledge in order to understand the realistic quality of a service [19]. They could subscribe to services that give the best value for their money. It could further be used as a reference for requestors in the process of negotiating and specifying service requirements. This could also provide an alternative approach to those that depend on expert knowledge and it requires only less time, has better accuracy and wider availability [20].

2.2. Criteria for choosing a clustering algorithm

Some of the concerns to be accounted

(1) Ability of the algorithm to handle non-linear data
(2) The algorithm should be able to handle voluminous data
(3) Algorithms using distance calculations for separating clusters are very receptive to ranges of variables. For example, "age" in general ranges 0 $\sim$ 100 and "salary" can extend from 0 to 100,000. When both variables are used jointly, distance from salary can overwhelm the other
(4) Formation of outliers should be in control
(5) Handling categorized (non-numeric data, non-numeric variables, categorical data, nominal data, or nominal variables) is a great deal
(6) An effectual clustering practice should hold up the mechanical discovery of clusters in a variety of subspaces of a higher dimensional space
(7) According to the situations, it is necessary to toggle between supervised or unsupervised approaches
(8) While handling time variant data, capturing hidden patterns becomes a challenge
(9) Need for a vigilance value to decide upon the significance and threshold of match making

A detailed study on the following two major categories of algorithms is made as an outcome of the survey.

(1) Neural Networks based algorithms
(2) Swarm-based algorithms

2.2.1. *Artificial neural networks*

ANN models for learning may be sorted out as supervised learning, unsupervised learning and reinforcement learning. Supervised learning system accepts the accessibility of a supervisor who classifies the training instances into groups and uses the data on the class membership of each training example, whereas, unsupervised learning scheme categorize the pattern class data heuristically and enables reinforcement learning studies by means of trial and error connections with its atmosphere. Some of the primary advantages of choosing ANNs are listed below:

(1) ANNs have the capability to study and model non-linear and complex relationships, which are certainly significant since in real-life, several relationships amongst inputs and outputs are non-linear and also complex.
(2) ANNs can generalize – After learning from the initial inputs and their relationships, ANNs can infer unseen relationships on unseen data also, thereby making the model generalize and predict on unseen data.
(3) Unlike several other forecasting techniques, ANN does not impose any limitations on the input variables (like in what way they should be distributed). Moreover, many studies have shown that ANNs have improved heteroskedasticity i.e. data with high volatility and non-constant variance, given its ability to learn hidden relationships in the data without imposing any fixed relationships in the data.

2.2.2. *Swarm-based algorithms*

Swarm Intelligence is a fairly new interdisciplinary field of research, which has gained huge acceptance these days. Algorithms fitting to the domain pull inspiration from the collective intelligence developed from the behaviour of a group of social insects like bees, termites and wasps. When acting as a community, these insects with limited individual ability can cooperatively perform many complex tasks essential for their existence. Problems like finding and storing foods, selecting and picking up materials for future usage require a detailed planning, and are solved by insect colonies without any kind of supervisor or controller. An illustration of predominantly successful research direction in swarm intelligence is Ant Colony Optimization (ACO) which emphases on discrete optimization problems, and has been applied effectively to a large number of NP hard discrete optimization problems which include the travelling salesman, the quadratic assignment, scheduling, vehicle routing, etc., and also to routing in telecommunication networks. Particle Swarm Optimization (PSO) is another very prevalent SI algorithm for global optimization over continuous search spaces. Since its beginning in 1995, PSO has fascinated the attention of numerous researchers all over the world ensuing into a vast number of variants of the elementary algorithm as well as various parameter automation strategies.

A detailed study has been made on the applications and merits of the following algorithms.

(1) Ant Colony optimization algorithm
(2) Artificial Bee Colony algorithm
(3) Birds flocking algorithm
(4) Fishes Schooling algorithm
(5) Bacterial Foraging algorithm
(6) PSO algorithm

3. Proposed Work

The Proposed work suggests the application of using ART algorithm for primary clustering with functional data and sub-clustering through swarm-based algorithms employing non-fictional data like metadata, QoS data and Doman log information. The cluster results are then compared for quality.

3.1. *Algorithms used*

Upon studying the two broad categories of algorithms ART (Adaptive Resonance Theory) Network which is a very apt learning algorithm that suitably addresses incremental leaning is chosen for the first iteration of clustering. Three specialties of the ART algorithm relevant to the context are

(1) It handles the problem of 'Stability and Plasticity Dilemma' in an effective manner. Plasticity is required for the upgradation of new knowledge and at the same time stability has to be preserved in order to retain the previously earned knowledge. This is taken care in the ART Network.
(2) It works on binary input which could facilitate in accommodating voluminous data.
(3) A Vigilance parameter which is a threshold of recognition is pre-set. Vigilance value of 'zero' puts data sets as independent items and when vigilance is set to 'one', two data items will fall in the same cluster if and only if there is a 100% match. In the current experiment, average values between 0.5 and 0.7 are set.

In the second iteration of clustering which is applied over the results of the first iteration through ART, swarm-based algorithms are experimented. This is because all biologically inspired algorithms draw their inspiration from animals or birds or insects belonging to the same kind forming clusters. No two different kinds of birds, fishes, animals or insects cluster together. Hence it is appropriate to apply swarm-based algorithms upon the results of the first cycle where a certain amount of homogeneity exists.

3.2. Nature of data retrieved from web services

3.2.1. Functional data

Functional data are the basic query data used by the client to make a search for a service. It may be keyword search or a collection of functional attributes that characterize the service.

3.2.2. Meta data

The following are the sources of the metadata of a web service

- XML Schema – For defining data types and structures.
- WSDL – For defining messages, message exchange patterns, interfaces and endpoints.
- WS-Policy – For declaring assertions for various qualities of service requirements, such as reliability, security, and transactions.
- WS-Addressing – For defining Web service endpoint references and associated message patterns.
- WS-MetadataExchange – For dynamically accessing XML, WSDL, and WS-Policy metadata when required.

3.2.3. Non-Functional data (QoS parameters)

QoS (Quality of Service) attributes of a web service can be understood as the capacity of a web service to act in response to expected invocations. In this attempt, numeric QoS values are single-handedly taken in to consideration for experimentation which can either be positive (or) negative. A positive QoS attribute is one whose value when increases is a sign of a better quality, for instance throughput, reliability and availability are positive QoS attributes. Negative QoS elements, when taking higher values point toward poor quality, for example response time and price are negative QoS elements.

3.2.4. Service generated data

These are data recorded over a period of time after the service becomes completely functional.

Reason for choosing Service Generated Data are

- Core Functional Data is not self-sufficient for the analysis.
- Availability of Metadata documents with all attributes filled in cannot be guaranteed
- Qi's parameters observed in one system differ from the Qi's observed in another system for the same service due to multiple network conditions.

Hence a service is allowed to function for a period of time and then the generated historical values are observed and taken for analysis. In this perspective, for web service recommendation several service generated Big Data (in a relative sense) can be considered which include Trace Logs, QoS and Service Relationships. Here QoS refers to personalized QoS data.

3.3. Clustering schemes

The layered clustering framework is presented in Figure 1. In this layered architecture, functional data or the core information used for the search of a web service is converted in to binary matrices representing the service features and are given as input to the ART network and the vigilance or threshold for clustering is set and the services are clustered. This attributes to the first phase of clustering. The clustered services are then subjected to subclustering through three different swarm algorithms. During the second phase, while using swarm algorithms, non-functional data of the already shortlisted services are considered. Thus sub-clustering happens in the second phase. The non-functional data used are meta-data extracted from the documents associated with the web services, QoS data and domain logs registered over a period of time. With these three types of data, three different swarm-based clustering approaches namely BOIDS(Birds Flocking) algorithm, ABC (Artificial Bee Colony) algorithm and PSO (Particle Swarm Optimization) algorithm are used and then cluster analysis is done considering two metrics namely intra- and inter-cluster distances. Certain evaluation indices like dunn index are also used to check the quality of the clusters.

3.3.1. Clustering using ART and birds flocking algorithm (Scheme 1)

Core Functional Data are taken as the input to the ART Clustering algorithm and the feature set given as input to ART is a binary sequence. The threshold value for clustering i.e., the vigilance parameter is set between 0.5 and 0.8. In the second iteration, the output of ART is taken as the input and the feature sets are updated with values parsed from metadata documents. Now birds flocking algorithm does the clustering work. The output clusters undergo a cluster analysis. The metrics used are inter-cluster distance and intra-cluster distances. The results are found drivable.

3.3.2. Clustering using ART and ABC algorithm (Scheme 2)

Core Functional Data are again taken as the input to the ART Clustering algorithm and the feature set given as input to ART is a binary sequence as in the previous scheme. The threshold value for clustering i.e., the vigilance parameter is set between 0.5 and 0.8. In the second iteration, the output of ART is taken as the input and the feature sets are updated with QoS information. For our

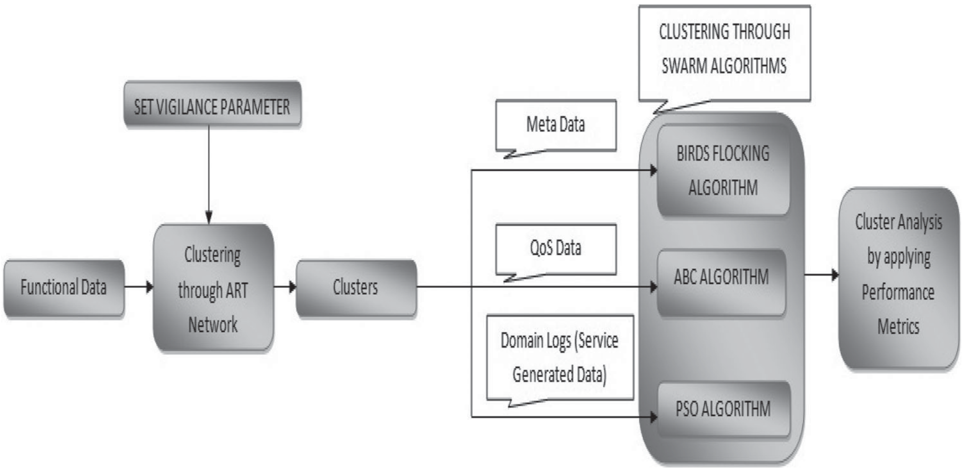

CLUSTERING THROUGH SWARM ALGORITHMS
SET VIGILANCE PARAMETER
Meta Data
BIRDS FLOCKING ALGORITHM
QoS Data
Functional Data
Clustering through ART Network
Clusters
ABC ALGORITHM
Cluster Analysis by applying Performance Metrics
Domain Logs (Service Generated Data)
PSO ALGORITHM

Figure 1. General clustering framework applied on different classes of data.

study, only one positive QoS value which is throughput and one negative QoS value which is response time are taken into account. Now instead of birds flocking algorithm, Artificial Bee Colony (ABC) algorithm does the clustering work. The output clusters undergo a cluster analysis. The metrics used are inter-cluster distance and intra-cluster distances. Mean and Standard deviations are also computed. The results are still found passable with improved closeness within clusters.

3.3.3. *Clustering using ART and PSO optimization (Scheme 3)*

Core Functional Data are again taken as the input to the ART Clustering algorithm and the feature set given as input to ART is a binary sequence as in the previous two schemes. The threshold value for clustering i.e., the vigilance parameter is set between 0.5 and 0.7. In the second iteration, the output of ART is taken as the input and the feature sets are updated with domain log information. This information is retrieved from the services over a specific period of time, preprocessed and then quantified appropriately to fit in to the feature set. In the third scheme, Particle Swarm Optimization algorithm does the clustering work. The output clusters undergo a cluster analysis. The metrics used are inter-cluster distance and intra-cluster distances. Mean and Standard deviations are also computed. The results are still found passable with improved closeness within clusters and inter-cluster distances are also passable.

The above three schemes are illustrated considering the following data set. Web services that deal with online purchase of domain names through the web portal www.databasereseller.com in India are considered for the examination. Consequently the functional and non-functional characteristics of the web services are to be transformed in to bit patterns duly in order to give as input to the ART algorithm as explained in all the three schemes initially and later in the second phase three kinds of data associated with the web services namely metadata, QoS data and service generated data are used and iterated with birds flocking, ABC and PSO algorithms respectively. Though the first two schemes produced quality outputs it is suggested to use domain logs which is a kind of service generated data observed over a period of time. They are promising because QoS values keep varying from one machine to another due to network conditions and again availability of metadata attributes cannot be guaranteed all the time for all services.

Dataset is taken from www.databasereseller.com which consists of the domain logs of different web services recorded over a period of time pertaining to services that offer domain name purchase in India.

4. Art algorithm and its features

The main features and functioning of the ART algorithm are presented in this section.

It usually comprises a comparison and recognition fields made-up of neurons, a vigilance limitation, and a reset module. Higher vigilance yields highly comprehensive memories (many fine-grained categories), while lower vigilance ends up in more general memories. The comparison field takes an input vector (a one-dimensional array of values) and transfers it to its best match in the recognition field. Its best match is the single neuron whose set of weights (weight vector) most closely matches the input vector. Each recognition field neuron outputs a negative signal (proportional to that neuron's quality of match to the input vector) to each of the other recognition field neurons and inhibits their output accordingly.

It supports self stabilized learning and incremental learning, also handles Stability – Plasticity dilemma effectively. This means it can learn new things at the same time retain old information.

Vigilance Parameter is a threshold value that helps to set the degree of similarity expected. It assigns bottom up weights (activation) and top down weights

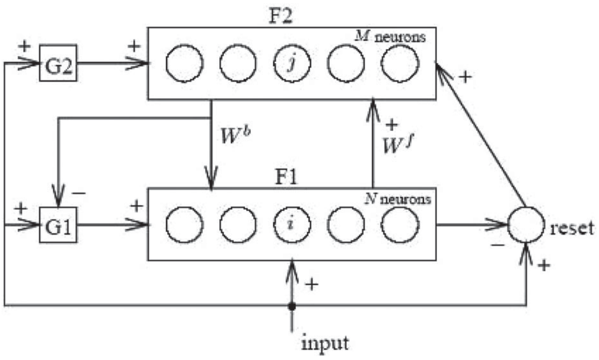
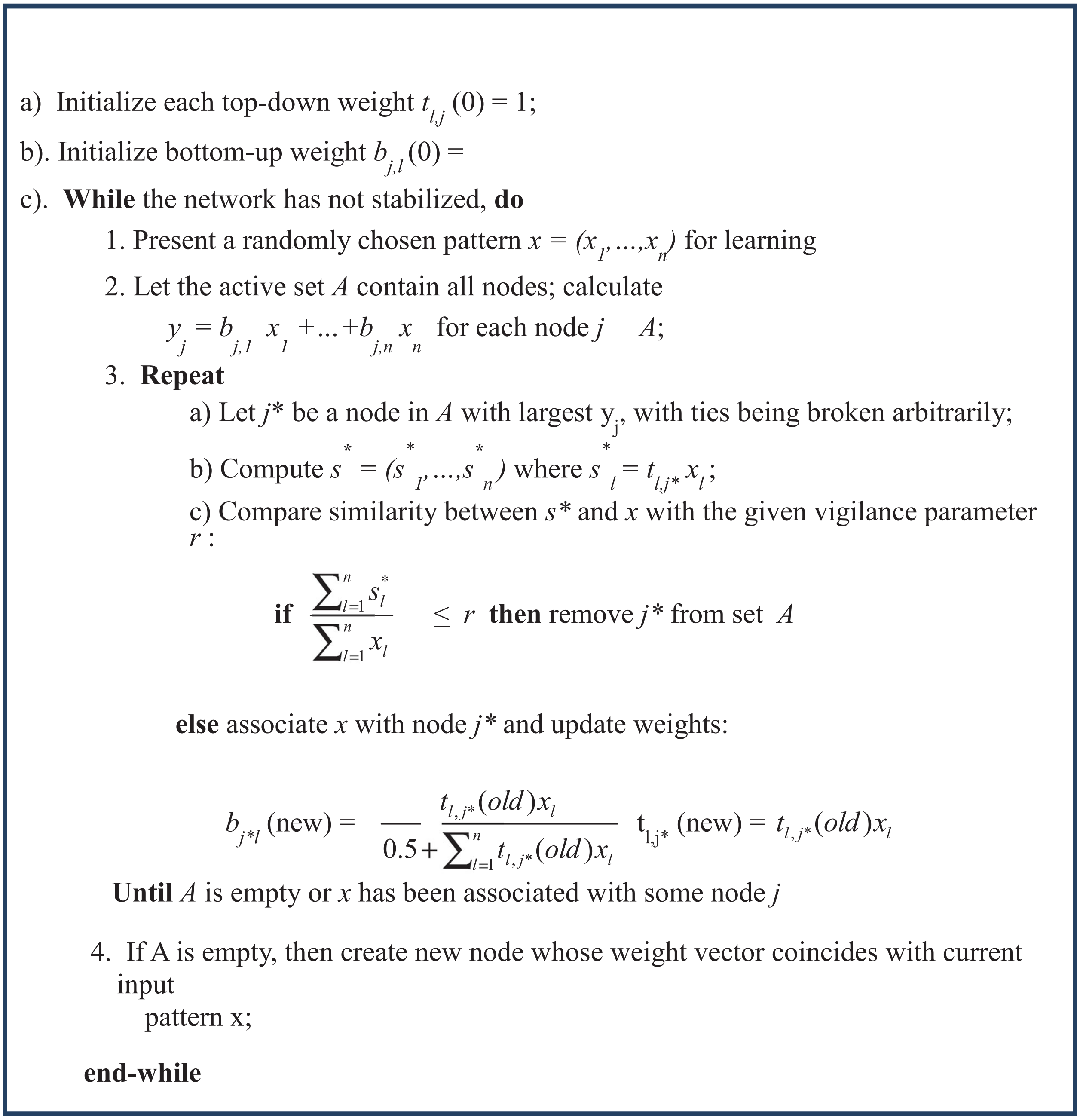


Figure 2. Functioning of ART1 network.

(expectations). ART networks consist of an input layer and an output layer. Bottom-up weights are used to determine output-layer candidates that may best match the current input. Top-down weights represent the "prototype" for the cluster defined by each output neuron. A close match between input and prototype is necessary for categorizing the input. Finding this match can require multiple signal exchanges between the two layers in both directions until "resonance" is established or a new neuron is added.

ART networks tackle the stability-plasticity dilemma: **Plasticity:** They can always adapt to unknown inputs (by creating a new cluster with a new weight vector) if the given input cannot be classified by existing clusters. **Stability:** Existing clusters are not deleted by the introduction of new inputs (new clusters will just be created in addition to the old ones). Clusters are of fixed size, depending on ρ (Vigilance Parameter). Also, the algorithm gets only binary input. The

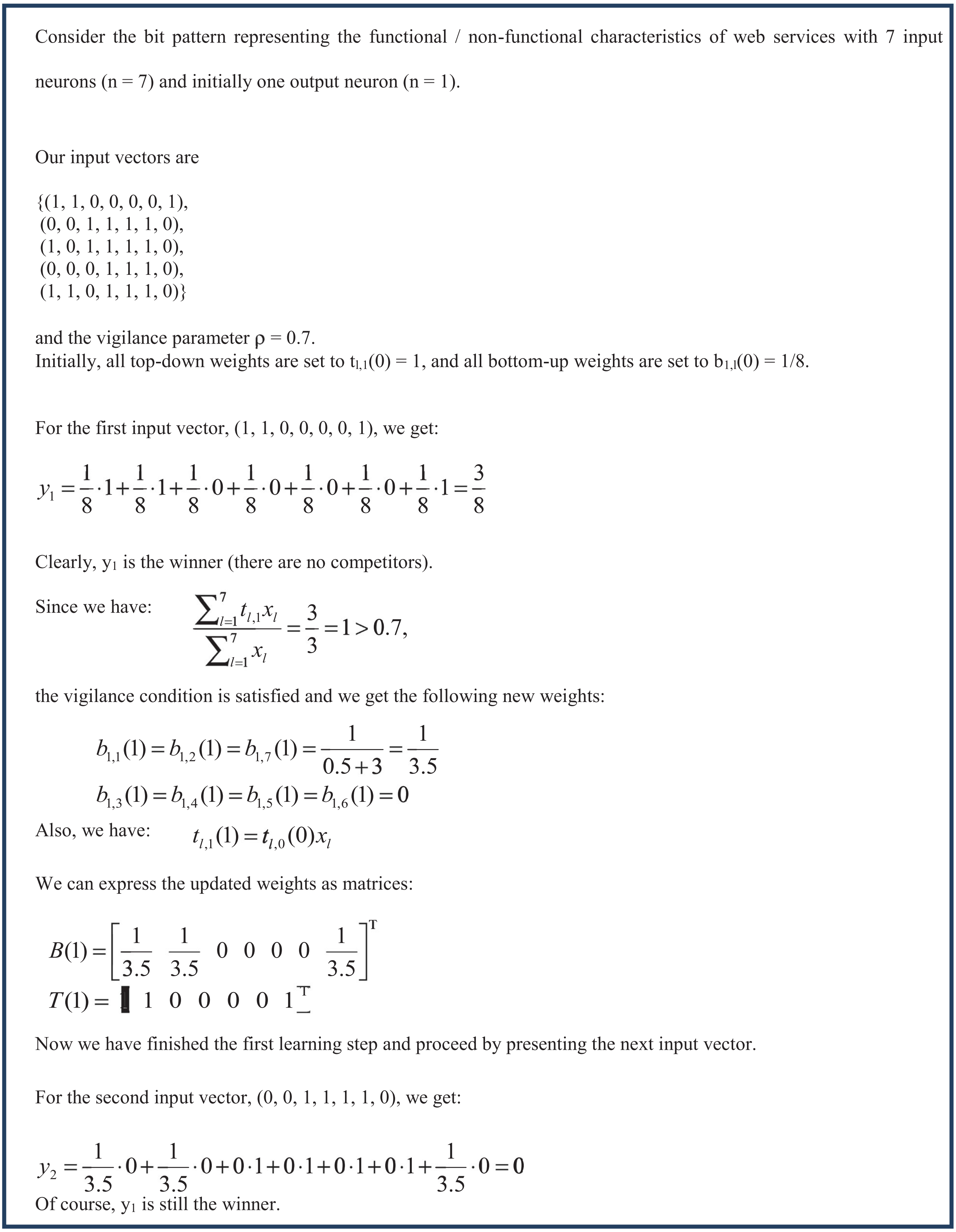


```
a) Initialize each top-down weight $t_{l,j}(0) = 1$;
b). Initialize bottom-up weight $b_{j,l}(0) =$
c). While the network has not stabilized, do
        1. Present a randomly chosen pattern $x = (x_1,\ldots,x_n)$ for learning
        2. Let the active set $A$ contain all nodes; calculate
                $y_j = b_{j,1}\, x_1 + \ldots + b_{j,n}\, x_n$ for each node $j$   $A$;
        3. Repeat
                a) Let $j^*$ be a node in $A$ with largest $y_j$, with ties being broken arbitrarily;
                b) Compute $s^* = (s^*_1,\ldots,s^*_n)$ where $s^*_l = t_{l,j^*} x_l$;
                c) Compare similarity between $s^*$ and $x$ with the given vigilance parameter
                $r$ :
                        if $\frac{\sum_{l=1}^{n} s_l^*}{\sum_{l=1}^{n} x_l} \leq r$ then remove $j^*$ from set $A$

                else associate $x$ with node $j^*$ and update weights:

                        $b_{j^*l}(\text{new}) = \frac{t_{l,j^*}(old)x_l}{0.5+\sum_{l=1}^{n} t_{l,j^*}(old)x_l}$   $\text{t}_{\text{l,j*}}(\text{new}) = t_{l,j^*}(old)x_l$
            Until $A$ is empty or $x$ has been associated with some node $j$
        4. If A is empty, then create new node whose weight vector coincides with current
        input
            pattern x;
    end-while
```

Figure 3. ART algorithm.



Consider the bit pattern representing the functional / non-functional characteristics of web services with 7 input neurons (n = 7) and initially one output neuron (n = 1).

Our input vectors are

{(1, 1, 0, 0, 0, 0, 1),
(0, 0, 1, 1, 1, 1, 0),
(1, 0, 1, 1, 1, 1, 0),
(0, 0, 0, 1, 1, 1, 0),
(1, 1, 0, 1, 1, 1, 0)}

and the vigilance parameter ρ = 0.7.
Initially, all top-down weights are set to $t_{l,1}(0) = 1$, and all bottom-up weights are set to $b_{1,l}(0) = 1/8$.

For the first input vector, (1, 1, 0, 0, 0, 0, 1), we get:

$$y_1 = \frac{1}{8}\cdot 1 + \frac{1}{8}\cdot 1 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 1 = \frac{3}{8}$$

Clearly, y_1 is the winner (there are no competitors).

Since we have: $$\frac{\sum_{l=1}^{7} t_{l,1}x_l}{\sum_{l=1}^{7} x_l} = \frac{3}{3} = 1 > 0.7,$$

the vigilance condition is satisfied and we get the following new weights:

$$b_{1,1}(1) = b_{1,2}(1) = b_{1,7}(1) = \frac{1}{0.5+3} = \frac{1}{3.5}$$
$$b_{1,3}(1) = b_{1,4}(1) = b_{1,5}(1) = b_{1,6}(1) = 0$$

Also, we have: $$t_{l,1}(1) = t_{l,0}(0)x_l$$

We can express the updated weights as matrices:

$$B(1) = \left[\frac{1}{3.5} \quad \frac{1}{3.5} \quad 0 \quad 0 \quad 0 \quad 0 \quad \frac{1}{3.5}\right]^{\mathrm{T}}$$
$$T(1) = \blacksquare \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1\rfloor^{\mathrm{T}}$$

Now we have finished the first learning step and proceed by presenting the next input vector.

For the second input vector, (0, 0, 1, 1, 1, 1, 0), we get:

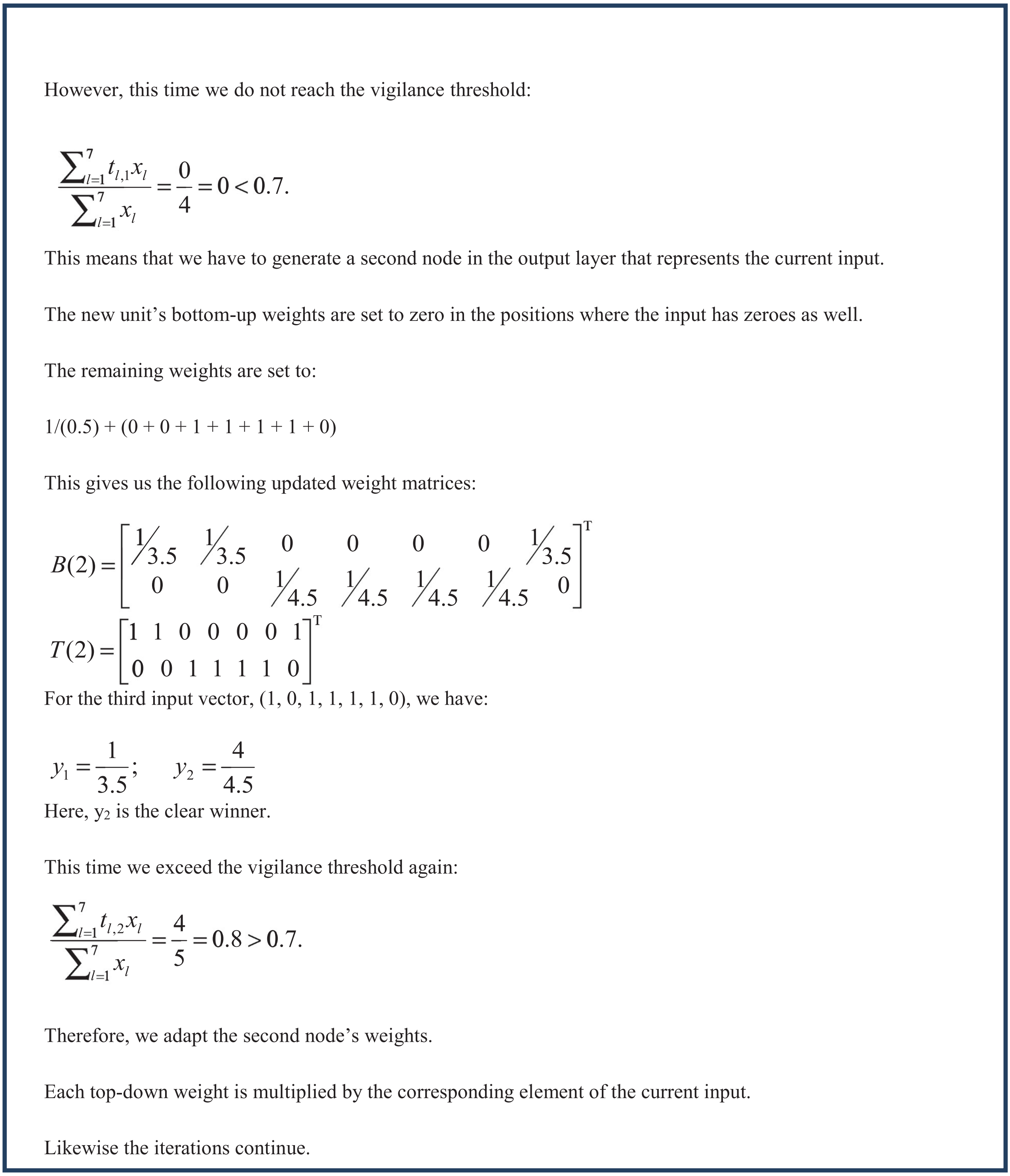
$$y_2 = \frac{1}{3.5}\cdot 0 + \frac{1}{3.5}\cdot 0 + 0\cdot 1 + 0\cdot 1 + 0\cdot 1 + 0\cdot 1 + \frac{1}{3.5}\cdot 0 = 0$$

Of course, y_1 is still the winner.

Figure 4. Weight computations of ART.

basic functionality of ART1 algorithm is presented in Figure 2.

Training: There are two elementary approaches of training ART-based neural networks: slow and fast. In the slow learning technique, the degree of training of the recognition neuron's weights towards the input vector is intended to continuous values with differential equations and is consequently dependent on the length of time the input vector is presented. In fast learning approach, algebraic equations are used to compute the degree of weight adjustments to be made, and binary values are considered. While fast learning is operative



However, this time we do not reach the vigilance threshold:

$$\frac{\sum_{l=1}^{7} t_{l,1} x_l}{\sum_{l=1}^{7} x_l} = \frac{0}{4} = 0 < 0.7.$$

This means that we have to generate a second node in the output layer that represents the current input.

The new unit’s bottom-up weights are set to zero in the positions where the input has zeroes as well.

The remaining weights are set to:

1/(0.5) + (0 + 0 + 1 + 1 + 1 + 1 + 0)

This gives us the following updated weight matrices:

$$B(2) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 1/4.5 & 1/4.5 & 1/4.5 & 1/4.5 & 0 \end{bmatrix}^{\mathrm{T}}$$

$$T(2) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

For the third input vector, (1, 0, 1, 1, 1, 1, 0), we have:

$$y_1 = \frac{1}{3.5}; \qquad y_2 = \frac{4}{4.5}$$

Here, y_2 is the clear winner.

This time we exceed the vigilance threshold again:

$$\frac{\sum_{l=1}^{7} t_{l,2} x_l}{\sum_{l=1}^{7} x_l} = \frac{4}{5} = 0.8 > 0.7.$$

Therefore, we adapt the second node’s weights.

Each top-down weight is multiplied by the corresponding element of the current input.

Likewise the iterations continue.

Figure 4. Continued.

and competent for a variety of tasks, the slow learning method is more biologically reasonable and can be used with continuous-time networks (i.e. when the input vector can vary continuously).

4.1. Algorithm

4.1.1. Parameters used

Following parameters are used.

- n – Number of components in the input vector
- m – Maximum number of clusters that can be formed
- b_{ig} – Weight from F_1(b) to F_2 layer, i.e. bottom-up weights
- t_{his} – Weight from F_2 to F_1(b) layer, i.e. top-down weights
- ρ – Vigilance parameter
- ||x|| – Norm of vector x and |s|| – Norm of Vector S

The algorithm and sample weight computations are furnished in Figures 3 and 4 respectively.

Table 1. Comparision of relavancy percentage.

		Number of clusters			% of relevancy		
Number of web services	Vigilance parameter	ART and boids with metadata	ART and ABC with QoS	ART and PSO with domain logs	ART and boids with metadata	ART and ABC with QoS	ART and PSO with domain logs
951	0.5	10	11	12	70	72	80
	0.6	11	12	14			
	0.7	12	12	15			
	0.8	12	14	16			
1426	0.5	11	13	14	71	76	81
	0.6	10	12	15			
	0.7	10	14	15			
	0.8	11	15	16			
1903	0.5	12	13	14	73	78	82
	0.6	12	16	18			
	0.7	13	16	18			
	0.8	13	17	19			
2378	0.5	14	16	18	74	82	83
	0.6	13	17	19			
	0.7	14	18	19			
	0.8	15	18	20			
2854	0.5	15	17	19	76	84	84
	0.6	16	18	20			
	0.7	16	19	20			
	0.8	17	18	21			
3329	0.5	17	18	22	77	85	86
	0.6	16	17	22			
	0.7	15	18	23			
	0.8	18	19	23			
3807	0.5	18	20	24	78	86	87
	0.6	19	21	24			
	0.7	19	22	25			
	0.8	19	21	25			

5. Results and discusslons

5.1. Relevancy factor

The percentage of relevancy indicates that a particular service fits into the same group irrespective of the change in vigilance value. The average relevancy percentage is 83.28% for domain log based clustering, 80.42% for the QoS based clustering and for metadata based clustering it is 74.14%. The details are tabulated in Table 1.

$$\% \text{ of relevancy} = \frac{\text{Computed Mean}}{p \times 100}.$$

Relevancy factor ensures the fact that even when the vigilance values are varied, the service falls in the same cluster. It means that the cluster has a high level of cohesion amidst the web services contained in it. Vigilance parameter could be set between 0 and 1. If it is 0, every single webservice is considered as a unique distinct cluster by itself. On the otherhand, if the vigilance value is 1, the members (web services) within the services are identical by 100%. In the experiment conducted middle values of vigilance ranging between 0.5 and 0.8 are considered and the observations are recorded. Higher or improved relevancy percentage in the table indicates that the particular approach has given better clusters that are cohesive and unique. Thus as tabulated in Table 1, ART algorithm and PSO algorithm experimented with domain logs observed over a period of time gave passable results forming more cohesive and unique clusters. Though a phenomenal improvement is not seen in comparison with ART and ABC algorithm there is a steady increase in the relevancy percentage.

5.2. Intercluster and intracluster distances

The compactness of the data items with in the cluster and the intercluster distances are observed and compared for all the three schemes. ART with PSO (Domain Logs) gave passable results with a high degree of similarity and the clusters are well separated from each other. The Comparison of intercluster and intracluster distances of all the three approaches is tabulated in Table 2. It tabulates the readings observe red in 4 consecutive experiments where the number of services is increased steadily. In the three verticals of varied combinations of clustering algorithms, the number clusters, average inter and intercluster distances of the clusters formed are noted. The vigilance parameter for the ART algorithm is set to 0.8 in all the cases. The values indicate that intercluster distances are higher and intracluster distances are lower in the 'domain logs' scheme which employed the PSO algorithm for subclustering. In all the three approaches ART algorithm is used for phase I. ART with PSO, that employs periodically recorded domain logs data has produced more cohesive and distinctive clusters.

5.3. Key merits of the approaches

Initially, ART algorithm gave better results being an unsupervised algorithm than K-means where k value has to be fixed in advance.

Table 2. Comparision of all three approaches.

Number of services	ART with PSO (domain logs)			ART with ABC (QoS)			ART with BOIDS (Meta Data)		
	No. of clusters	Ave. intracluster distance	Ave. intercluster distance	No. of clusters	Ave. intracluster distance	Ave. intercluster distance	No. of clusters	Ave. intracluster distance	Ave. intercluster distance
951	22	07.342–27.654	50.176–73.234	21	08.122–28.432	49.543–71.345	18	09.911–30.251	48.618–69.451
1903	25	11.245–27.564	52.543–74.321	23	12.275–30.521	50.114–66.432	20	14.145–32.043	49.564–65.513
2854	29	17.453–30.234	54.244–75.345	27	19.453–33.176	52.312–69.123	25	17.535–34.123	51.873–68.136
3807	35	22.345–34.213	56.356–76.345	33	26.734–35.353	55.123–71.234	31	24.812–37.541	53.618–69.451

When non-functional data i.e. data extracted from metadata documents were used in addition to functional data, and birds flocking algorithm was employed, better results were produced upon sub-clustering.

In birds flocking algorithm, different flocks represent different service clusters. Similar to another bio-inspired clustering algorithm, the ant colony algorithm, flocking algorithm does not need initial partitions or the prior knowledge about the class number for each dataset. The advantage of the flocking clustering algorithm is the heuristic principle of the flock's searching mechanism. This heuristic searching mechanism helps them quickly form a flock. Thus the flocking clustering algorithm can generate a better clustering result with fewer iterations than that of the ant clustering algorithm which is widely used. The clustering results generated by the flocking algorithm can be easily visualized and recognized by an untrained human user.

Though the above method was efficient, data extracted through metadata documents were inconsistent and there were many missing data in many instances. Thereby an attempt to use numeric QoS attributes was made by employing ART and ABC algorithms. But the problems with artificial bee colony algorithm are (1) No centralized processor to guide the ABC towards good solutions (2) Slower convergence than other heuristics. Moreover, QoS values were different from one client machine to another due to the fluctuating network conditions in different instances.

So in the third approach, service generated data like domain logs which are historical in nature, i.e. observed over a period of time after a web service has been instantiated, are used so that there will be a better consistency in the data. So for rectifying the earlier issues, in the final stage the PSO algorithm is proposed. The merits being (1) Simple implementation, (2) Easily parallelized for concurrent processing (3) Few algorithm parameters. The results through ART and PSO were much passable and optimality is established through query time and a credit based ranking technique is applied for presenting a better result.

6. Conclusion

In this paper, users are given recommendations about the web services available for a specific application by analysing the service generated data such as domain logs of web services by performing parallel PSO based clustering using Map Reduce technique and then extracting reports by further refinement to achieve optimality. Application of ART algorithm helped significantly in handling different classes of unfragmented data and as ART takes binary data representing multidimensional nonlinear data was simplified. Medium range of vigilance values between 0.5 and 0.8 are used to set the threshold values for the degree of similarity and it is observed that the cluster parameters are phenomenally increasing both in number and quality.

Usage of PSO algorithm helped to achieve optimality in a faster rate and the computational over head is controlled significantly due to the two phase approach. Optimality is achieved in a better way in PSO as there is no overlapping and mutation. Formation of outliers was also avoided. The self-organizing and scalable nature of swarm algorithms is well suited to complement unsupervised clustering of ART.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Gao H, Stucky W, Liu L. Web services classification based on intelligent clustering techniques. J Inform Technol Appl. 2009;3(5):242–245.

[2] Abramowicz W, Haniewicz K, Kaczmarek M, et al. Architecture for Web services filtering and clustering. In: Proceedings of 2nd IEEE International Conference on Internet and Web applications and services, pp. 18–18, 2011.

[3] Su K, Xiao B, Liu B, et al. A personalized trust-aware QoS prediction approach for web service recommendation. J Knowl-Based Syst. 2017;115(5):55–65.

[4] Chen CL, Tseng FS, Liang T. An integration of Word Net and fuzzy association rule mining for multi-label document clustering. Data Knowl Eng. 2010;69(11):1208–1226.

[5] Dasgupta S, Bhat S, Lee Y. Taxonomic clustering of web service for efficient discovery, Proceedings of the 19th ACM international conference on information and knowledge management, ACM, pp. 1617–1620, 2010.

[6] Dong H, Hussain FK, Chang E. A survey in semantic search technologies, Proceedings of IEEE 2nd International Conference on Digital Ecosystems and technologies, DEST, pp. 403–408, 2008.

[7] Dasgupta S, Bhat S, Lee Y. Taxonomic clustering and query matching for efficient service discovery, Proceedings of IEEE International Conference on Web services (ICWS), pp. 363–370, 2011.

[8] Sukumar AS, Loganathan J, Geetha T. Clustering web services based on multi-criteria service dominance relationship using Peano space filling curve, Proceedings of IEEE International Conference on data Science & Engineering (ICDSE), pp. 13–18, 2012.

[9] Gao H, Liu L. Web services classification based on intelligent clustering techniques, Proceedings of IEEE International Conference on information Technology and applications, IFITA'09, pp. 242–245, 2008.

[10] Li H, Xu X, Hu D, et al. Graph method based clustering strategy for femtocell interference management and spectrum efficiency improvement, Proceedings of 6th International IEEE Conference on Wireless Communications Networking and Mobile computing (WiCOM), pp. 1–5, 2010.

[11] Mohana R, Dahiya D. Optimized service discovery using QoS based ranking: A fuzzy clustering and Particle swarm optimization approach, Proceedings of IEEE Annual Computer Software and applications Conference, pp. 452–457, 2011.

[12] Wang J, Yang X, Long K. Web DDoS detection schemes based on measuring user's access behavior with large deviation', Proceedings of global Telecommunications Conference (GLOBECOM), pp. 1–5, 2011.

[13] Gholamzadeh N, Taghiyareh K. Ontology based fuzzy web services clustering, Proceedings of 5th International Conference on Telecommunications, pp. 721–725, 2010.

[14] Lee YJ, Kim CS. A learning ontology method for restful semantic web services, Proceedings of the International Conference on Web services, pp. 251–258, 2011.

[15] Wen T, Sheng G, Li Y, et al. Research on Web service discovery with semantics and clustering, Proceedings of 6th International Conference on information Technology and Artificial intelligence Conference (ITAIC), Vol. 1, pp. 62–67, 2011.

[16] Paliwal V, Shafiq B, Vaidya J, et al. Semantics-Based Automated service discovery. IEEE Trans Service Comput. 2012;5(2):260–275.

[17] Badr Y, Abraham A, Biennier F, et al. Enhancing web service selection by user preferences of non-functional features, Proceedings of 4th International Conference on Next Generation Web services Practices (NWESP), pp. 60–65, 2008.

[18] Rangarajan SK, Phoha V, Balagani K, et al. Web user clustering and its application to prefetching using ART neural networks. J Comput (Taipei). 2004: 45–62.

[19] Abu Sharkh H, Fung BC. Service-Oriented architecture for Sharing Private Spatial-Temporal data, Proceedings of International Conference on Cloud and service computing (CSC), pp. 40–47, 2011.

[20] Nagy, Oprisa C, Salomie I, et al. Particle swarm optimization for clustering semantic web services, Proceedings of 10th International Symposium on parallel and distributed computing IEEE, pp. 170–177, 2011.