Enhancing Web Service Discovery Using Meta-heuristic CSO and PCA Based Clustering

Chapter · January 2018

DOI: 10.1007/978-981-10-3376-6_43

2 authors, including:

Sowmya Kamath S
National Institute of Technology Karnataka
86 PUBLICATIONS **216** CITATIONS

SEE PROFILE

Enhancing Web Service Discovery using Meta-heuristic CSO and PCA based Clustering

Sunaina Kotekar and Sowmya Kamath S

Department of Information Technology,
National Institute of Technology Karnataka, Surathkal
Mangalore - 575 025, Karnataka, India
sunainakotekar@gmail.com,sowmyakamath@nitk.ac.in

Abstract. For service oriented application development, one of the most prominent tasks is Web service discovery. For a targeted objective to be achieved, it is challenging to get all the appropriate services from a pool of many web services. To obtain the most suitable services, it is of utmost importance to get the user's preference of service-specific terms, also, that comply with natural language documentation. This methodology is time consuming due to functional diversity of services. Clustering the web services as per their domain, based on functional similarities would enhance the search engine's ability to recommend relevant services. In this research, we have come up with new approach for web service description documents clustering in a UDDI repository into functionally similar groups using Cat Swarm Optimization (CSO) Algorithm, optimized by Principle Component Analysis(PCA) dimension reduction technique. Results obtained by experimentation show that the proposed approach was useful to enhance service discovery process.

Keywords: Web service discovery; bio-inspired algorithms; Document clustering; Semantics; Swarm intelligence

1 Introduction

Service-oriented computing (SOC) is the computing model that uses services as basic components for application development and is already highly regarded in internet application development. To build a service-centric application model, SOC extends a service oriented architecture (SOA), as a means for reorganizing software applications and frameworks into a set of collaborative services. Web services are currently the most prominent implementations of SOA, as they are standards based, and employ XML based protocols for messaging, service description and data transfer. The capabilities of a Web service are encapsulated in its service description, in the form of a WSDL (Web Service Description Language) document.

The task of searching for relevant Web services for a given requirement is traditionally based on the service name and natural language description [11]. Hence, this search is similar to that performed by search engines and is primarily keyword based. However, several studies have revealed that more than 55%

of published services do not have any natural language documentation, while about 30% of services have less than 10 words [9][10]. Also, basic keyword search overlooks the actual functionality of a Web service, by concentrating only on the service name and documentation (if available).

To overcome these drawbacks, intelligent techniques to automatically capture both the functionality of a service and also its functional domain are crucial. This has to be done on available data, instead of relying on service providers to make such information available explicitly. Since a service's WSDL effectively describes its capabilities and also provide information for a client on how to use the web service, the WSDL document is used as an effective source for determining the functionality of a service. Text mining techniques can be applied to WSDL documents to identify useful components, which describe actual functionality of the corresponding Web service. Using this functionality related information, WSDL documents can be clustered to capture their domain, thus achieving search space reduction during the process of service discovery and selection.

Document clustering is the process of categorization of documents into groups of similar documents, where each group represents a particular domain. The process is performed such that intra-group document distance is to be kept low, while inter-group document distance should be high. A distance measure technique is thus the heart of document clustering. Document clustering significantly reduces the search space/domain when it comes to searching a related document(s) with some keywords. Semantics based techniques that capture the morphological variants of a user provided keyword, can further enhance the retrieval process.

This dissertation presents modified CSO, which is based on the social behavior of cats in nature, for Web service description clustering. Text mining techniques were applied to a real world service dataset to extract their functional information. CSO was applied to a set of Web services to determine similar groups. The clustering accuracy of the CSO algorithm was analyzed with a traditional algorithm like the K-means basic clustering algorithm. Also, PCA was used to scale down the large set of attributes, for further optimizing clustering purity.

The remaining paper is sketched as follows: Section II presents current trend in the field of clustering Web services to enhance service discovery. Section III describes the proposed methodology and the specifics of the Cat Swarm Optimization Algorithm. Section IV explains the experimental results and their analysis, and then conclusion and references.

2 Related Work

Discovering web services based on the functional requirements is need of the hour in search engines. To boost precision in search results, several approaches have focused on clustering Web services based on their functional similarity. Elgazzar et al [6] suggested a technique for finding similarity between web services, by using the service's WSDL document which describes particular web service in detail. Using this computed similarity between each WSDL document corresponding to web service, the services were clustered using QT clustering algorithm. Nayak

et al [8] explained strategy for finding the affinity between web services on the basis of Jaccard Coefficient to cluster similar document for the ease of discovery of services. Liu and Wong [7] gave a idea for web service clustering on the basis of extracted content from WSDL document such as content, hostname, context and service name.

The drawback of these approaches is that, a traditional algorithm like QT clustering can result in too many clusters of small size. Initially, some clusters of big size will be formed and then the remaining data will get clustered into smaller clusters and all remaining dissimilar service will form a cluster, because of this, purity of the clustering will be very low. Also, creating a similarity matrix of size nxn for all n WSDL documents is time consuming.

Chu and Tsai [3], first discussed the concept of computational artificial life or computational intelligence, which primarily emulate animal behavior in nature for solving computationally hard problems. These algorithms are usually employed as optimization techniques, and several swarm intelligence based methods that simulate the intelligent behavior of animals are currently available. Particle Swarm Optimization[2] (PSO), Ant Colony optimization (ACO) [5], and also a recent development in the form of Cat Swarm optimization (CSO)[3][4], that makes use of the social herding behavior of cats in nature. Some problems which are addressed using these nature-inspired algorithm are scheduling, Vehicle routing problem (VRP), Shortest Path problem, Travelling salesman problem (TSP) and data mining, particularly clustering problem etc. Chu et al [4] proposed the CSO algorithmic rule that extends two submodels based on the two most important behavioral character of cats, the "seeking mode" and the "tracing mode". Santosa [1] recommended a method for clustering records in a standard dataset according to their classes using CSO clustering algorithm, which was found to be better than several other clustering techniques.

Our approach is to applying the CSO based optimization to traditional clustering algorithm like K-means, using the computed functional similarity of service documents. Natural language processing methods like tf-idf (term frequency - inverse document frequency) and morphological analysis are used to obtain the similarity and dissimilarity between service documents. This similarity value is used to effectively cluster service documents into functionally similar groups.

3 Proposed System

The proposed methodology is aimed at describe the problem of extracting the functional information of services, and using this to automatically categorize a set of Web services in a domain specific manner. Figure 1 shows the workflow of the proposed work.

A WSDL document is an inherent source of the functional details of a Web service, so these are first pre-processed. All the natural language terms in the WSDL document are considered as a feature list and extracted by a process called content extraction. Filters like stopword removal, stemming are applied in preprocessing. The outcome of this step is set of keywords from each WSDL document, along with their frequency in dictionary format. This dictionary of

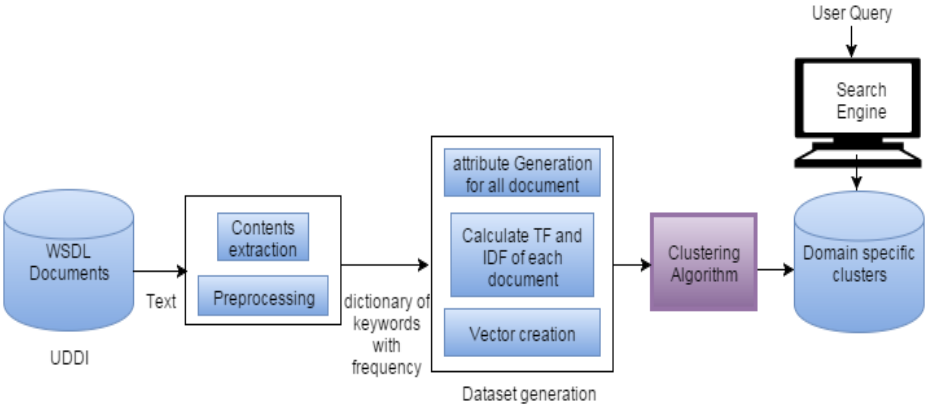

Fig. 1: Proposed methodology

words from all documents combined are taken as *attributes*. Next, the tf-idf value is calculated using the equation (1).

$$tf - idf_{i,j} = tf_{i,j} * idf_i \tag{1}$$

where i represents the i^{th} document and j represent j^{th} word in the attribute list $tf_{i,j}$ is frequency of attribute j appears in document i as shown in equation (2). where as idf_j is calculated using equation (3) explains how important an attribute is.

$$tf_{i,j} = \frac{number\ of\ times\ attribute_j\ in\ document_i}{Total\ number\ of\ words\ in\ document_i} \tag{2}$$

$$idf_j = log \frac{N}{|d \epsilon D : j \epsilon d|} \tag{3}$$

where N is number of documents. idf_j is calculated as logarithmic fraction of total number of documents to number of documents containing j^{th} attribute.

Based on the calculated values of $tf - idf$, the nearness and dissimilarity between the documents are calculated using the Euclidean distance formula as given by equation (4).

$$d(x, y) = ||x - y||^2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4}$$

Using this value, the K-means algorithm is applied to the service documents. Here, the number of clusters (K) into which the service documents are to clustered is specified. During the first iteration, K documents are randomly chosen as cluster centers. Based on the Euclidean distance formula given in Equation (4), the distance between each document and the center chosen is computed and the document if assigned to the nearest center. The calculation of mean of each cluster leads to the discovery of new centroid of the cluster and the process of clustering is continued. Finally, the clusters formed are stored in the repository. Whenever a user searches for a web service we can find the relevant cluster of WSDL documents to provide them a better result. Each cluster can be tagged with keywords for ease of search. After finding relevant clusters for query, WSDL documents in clusters can be sorted based on relevance.

3.1 Cat Swarm Optimization based Clustering

To get the better set of clusters determined by K-means clustering algorithm; hence enhancing the purity of the clusters formed, the CSO technique was applied to the service documents. As per the standard of CSO algorithm, two sub-models that are based on real-world cat behavior while hunting for food have been proposed. These sub-models are named as "seeking mode" and "tracing mode".

In CSO, the number of cats to be used in each iteration should be decided initially. Every cat owns D dimensions that represent its position, velocities pertaining to every dimension, a dependency of fitness operator (in terms of degree) known by the fitness worth of each cat, and boolean flag to determine cats' mode (seeking or tracing). Algorithm 1 illustrates CSO clustering applied to given service documents. The two sub modules of algorithm 1 are explained in detail in procedures.

Algorithm 1: CSO Algorithm for Clustering

Data: Dataset for clustering, number of cluster K, Number of copy
Result: Getting clusters of data
Randomly choose K data as cluster centers $c_1, c_2, ..., c_K$;
repeat
  Enter Seeking mode;
  if *seeking SSE* $< SSE_i$ **then**
    SSE_{i+1}=new SSE
  else
    $SSE_{i+1} = SSE_i$
  Enter Tracing mode;
  if *seeking SSE* $< SSE_i$ **then**
    SSE_{i+1}=new SSE
  else
    $SSE_{i+1} = SSE_i$
until *Records are not stable in cluster*;
return *List of records in different cluster*

Seeking mode: Four fundamental pillars of seeking mode are: the selected dimension (SRD), seeking memory pool (SMP), self position consideration (SPC), seeking range of counts of dimension to change (CDC) (shown in procedure: seeking mode). The SSE is computed using Equation (5) and the distance for reassigning the clusters is computed as per Equation (4). The seeking mode imitates the process of cats moving around while they are near a prey. So, in CSO, the cluster centers are changed slightly after each iteration, after which the SSE is computed and updated to the smallest value.

$$SSE = \sum_{i=1}^{k} \sum_{x \in D_i} (||x - m_i||^2) \tag{5}$$

Tracing mode: The tracing mode imitates the behavior of real-world cats, when they intend to attack their prey, by jumping on the prey. So, in CSO, the cluster centers with least SSE are considered and the *velocity* is calculated and

Procedure Seeking mode

Data: Parameters SPM, SRD, SPC
Result: Getting intermediate clusters of data
for *i=0 to k* **do**
  create SMP times copy of cluster $center_i$ position
  Determine i value
  Compute shifting parameter ($clustercenter_i * SRD$)
  for *x=1 to SMP* **do**
    At random add or subtract cluster centroids with the shifting parameter
    ((SMP*k) cluster candidates are obtained at this step)
    Compute distance
    Group data into clusters based on distance calculated
    Compute SSE
    Determine the potential candidate to be recognized as new cluster
    center by roulette wheel selection
return *List of records in different intermediate cluster*

updated using Equation (6). Using this computed velocity, the position of the cat is updated using Equation (7). For the new position, the SSE is calculated again as before, and we check if the new value is the lowest. If the SSE value is lower than earlier value, then the new position is the updated cluster center.

$$V_{k,d} = v_{k,d} + r_1 * c_1 * (x_{best,d} - x_{k,d}) \tag{6}$$

$$x_{k,d} = x_{k,d} + v_{k,d} \tag{7}$$

Procedure Tracing mode

Data: Intermediate clusters and centers
Result: Getting intermediate clusters of data
for *i=1 to k* **do**
  Modify $velocity_i$
  Modify $position_i$
  Find new cluster $center_i$
Compute distance
Assign data points into clusters based on distance calculated
Compute SSE
return *List of records in different intermediate cluster*

3.2 Dimensionality Reduction using PCA

In the proposed clustering methodolgoy, a large WSDL dataset is used, due to which the constructed tf-idf matrix results in large number of attributes. In this tf-idf matrix, most of the values are zero, as a particular term may be relevant only to a few services in the same domain, due to which the tf-idf matrix is highly sparse. To reduce this sparseness, dimensionality reduction technique is used.

Principal Component Analysis (PCA) is a technique used for feature reduction, which involves mapping of data from high dimensional space to lower dimensional space. To adopt this technique, we compute covariance matrix for

the data, using which the Eigen values are calculated. Based on Eigen vectors obtained from largest Eigen values, we reconstruct the new data matrix with lower dimension. Eigen vectors are called as principal components.

3.3 Similarity based WSDL dataset generation

When the tf-idf matrix is generated, only the existence of the attribute and its frequency in that particular document is considered. To consider the semantic similarity between terms used in functionally similar services, the morphological variants and synonyms of terms must also be considered. For example, car, vehicle, four wheeler etc are related words, while tf-idf would consider these are completely different words. We incorporate a word similarity measure to determine functionally similar documents inside a cluster. To calculate the data matrix, we modify tf-idf equation as shown in (8).

$$tf - idf_{i,j} = tf_{i,j} * idf_j * MaxSimilarity_{i,j} \tag{8}$$

Where $MaxSimilarity_{i,j}$ is 1 if the i^th attribute is present in j^{th} document. Else, it is maximum of all similarity found between words in i^{th} document to j^{th} attribute as shown in equation (9). Here n being total number of words in i^{th} document. Similarity is computed using Wordnet [12].

$$MaxSimilarity_{i,j} = \max_{1 \leq k \leq n} Similarity(word_k, attribute_j) \tag{9}$$

4 Experimental Results

To calculate effectiveness of the CSO based clustering, several standard datasets (like Iris, Glass, Balance Scale, Soybean Small and Wine[1]) as well as on the WSDL dataset[2] were taken and experimented upon. The purity or accuracy of clustering is calculated using equation (10), where j is number of classes and k is number of clusters.

$$Purity(\%) = \frac{\sum_0^k \max_0^j (Documents\ belonging\ to\ each\ class)}{Total\ number\ of\ documents} \tag{10}$$

Table 1 tabulates the accuracy of clustering using K-means and CSO on the various standard datasets and on the WSDL dataset. There were 3 classes of Iris flowers available in the given standard dataset, namely "Iris-versicolor", "Iris-setosa" and "Iris-virginica". In the case of WSDL documents, the domain to which a web service belongs to is taken into account for calculating the purity. Domains are: "education", "travel", "food", "geography", "medical", "economy", "weapon", "communication", and "simulation". This is obtained from the folder hierarchy of the OWL-S TC dataset.

As can be seen from the results, purity level varies with the number of attributes or number of records. It was observed that purity level almost inversely proportional to number of attributes and records collectively. Figure 2

[1] Available at `https://archive.ics.uci.edu/ml/datasets.html`
[2] Available at `http://projects.semwebcentral.org/projects/owls-tc/`

Table 1: Purity of clusters for different datasets

Dataset Name	No of Records	Attributes	Classes	K-means Purity	CSO Purity
Iris	150	4	3	67%	90%
Glass	214	9	6	54%	58%
Balance scale	625	4	3	61%	78%
Soybean small	47	35	4	79%	83%
Wine	178	13	3	70%	72%
WSDL Documents	1083	644	9	56%	63%

shows comparative analysis of cluster purity for K-means and CSO on different datasets. Table 2 shows the different results based on datasets after the PCA dimensionality reduction has been applied to the sparse data. The number of attributes are more in the case of Wine and WSDL dataset, and also the data is sparse, so PCA was applied for reducing number of attributes. As a results, clustering purity increased from 63% to 69% and also the run time of algorithm reduced greatly due to less dimension of data.

Table 2: Purity of clusters for different datasets after applying PCA

Dataset Name	No of Records	Attributes	Classes	K-means Purity	CSO Purity
Wine	178	4	3	83%	89%
WSDL Documents	1083	4	9	60%	69%

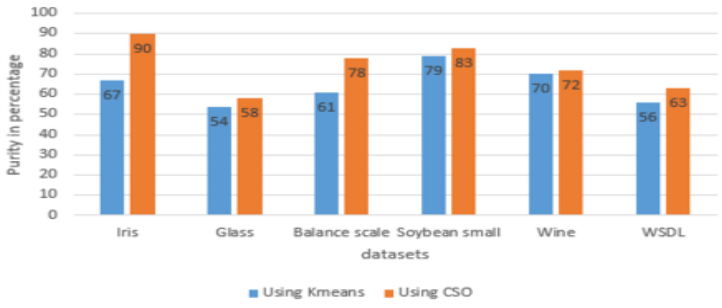

Fig. 2: Observed Purity before PCA - K-Means vs. CSO

Figure 3 shows the result comparative analysis graph between simple WSDL dataset and modifying WSDL dataset using PCA, modified WSDL dataset using Wordnet Similarity, modified WSDL dataset after applying PCA and Wordnet Similarity, it was observed that purity of clusters increased when the similarity factor is added to the dataset.

5 Conclusion and Future Work

Here a modified CSO for Web service description clustering was discussed. Text mining techniques were applied to a real world service dataset to extract their functional information and CSO was applied to these to determine similar groups. The clustering accuracy of CSO was analyzed with traditional K-means basic clustering algorithm. It was found that the purity of clusters obtained by CSO was better than K-means clustering algorithm by about 7%. When PCA based dimension reduction was applied, the purity of clustering increased to 69%

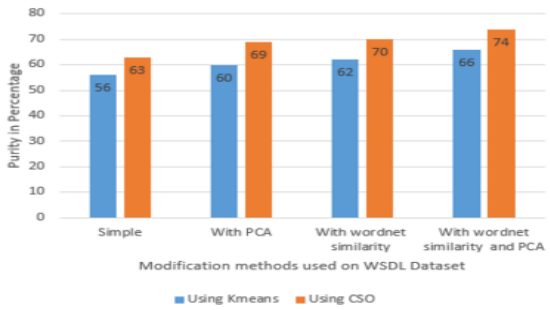

Fig. 3: Observed Purity after modifying WSDL dataset - K-Means vs. CSO

from 63%. This is because there is randomness involved in K-means initial center selection and the algorithm terminates when the centers are stable. But in the case of CSO tracing mode, drastic change of centers are carried out for possible better clustering. As part of future work, this purity can be further enhanced by making use of feature selection techniques, along with feature reduction techniques, to capture the best features based on which clustering can be performed.

References

1. Santosa, Budi, and Mirsa Kencana Ningrum. "Cat swarm optimization for clustering." Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of. IEEE, 2009.
2. A. Abraham, S. Das, S. Roy: Swarm Intelligence Algorithms for Data Clustering. Soft Computing for Knowledge Discovery and Data Mining, 2008: 279-313.
3. S. C. Chu, and P. W. Tsai, Computational Intelligence Based on the Behaviour of Cat, International Journal of Innovative Computing, Information and Control, 3 (1), 2007, pp.163-173.
4. S. C. Chu, P. W. Tsai, and J. S. Pan, Cat Swarm Optimization, LNAI 4099, 3 (1), Berlin Heidelberg: Springer-Verlag, 2006, pp. 854 858.
5. Shelokar, P. S., Valadi K. Jayaraman, and Bhaskar D. Kulkarni. "An ant colony approach for clustering." Analytica Chimica Acta 509.2 (2004): 187-195.
6. Khalid Elgazzar, Ahmed E. Hassan, Patrick Martin, Clustering WSDL Documents to Bootstrap the Discovery of Web Services, 8th IEEE International Conference on Web Services (ICWS'10), Miami, Florida, USA, pp. 147-154, July 2010.
7. Wei Liu, Wilson Wong, Web service clustering using text mining techniques, International Journal of Agent Oriented Software Engineering, Vol. 3, No. 1, pp. 6-26, 2009
8. Richi Nayak, Data mining in Web services discovery and monitoring , International Journal of Web Services Research, Vol. 5, No. 1, pp. 63-81, January, 2008.
9. Fan, J. and Kambhampati, S., 2005. A snapshot of public web services. ACM SIGMOD Record, 34(1), pp.24-32.
10. Kim, Su Myeon, and Marcel-Catalin Rosu. "A survey of public web services." E-commerce and web technologies. Springer Berlin Heidelberg, 2004. 96-105.
11. Bachlechner, Daniel, et al. "Web service discovery-a reality check." 3rd European Semantic Web Conference. Vol. 308. 2006.
12. Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.