Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 7 (2011) 305–307

Procedia Computer Science

The European Future Technologies Conference and Exhibition 2011

A Bio-inspired Fuzzy Agent Clustering Algorithm for Search Engines

Radu D. Găceanu [a,b,1]

[a] *Department of Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania*
[b] *Department of Programming Theory and Software Engineering, Eötvös Loránd University, Budapest, Hungary*

Abstract

In general, web search engines respond to queries by returning a list of links to web pages that are considered relevant. However, these queries are often ambiguous or too general and the users end up browsing through a long list of items in order to find what they are actually looking for. And hence the idea to cluster web search results so that the output would be a list of labelled clusters. An algorithm based on the ASM (Ants Sleeping Model) is proposed. In the ASM model each data is represented by an agent, its environment being a two dimensional grid. The agents will group themselves into clusters by making simple moves according to some local environment information. At any step an agent can pro-actively decide to directly communicate with one of its fellows and choose to move accordingly, the moves being expressed by fuzzy IF-THEN rules. Thus the chance of getting trapped in a local optimum is minimized and hybridization with a classical clustering algorithm becomes needless.

Keywords: Ants Sleeping Model; Fuzzy logic; Clustering

Nowadays web search engines offer good results and users generally find what they are looking for even in the first pages of the search results. Nevertheless, it also happens that users end up browsing through several pages until getting the result they were looking for. Ongoing research is done in order to make the process easier and several features like searching in a time interval or using related searches have already been added to search engines. However this is still not enough because the users cannot always express what they need in a consistent way so sometimes their queries are ambiguous or too general and hence the search results are not really relevant. So the possibility to cluster web search results so that the output would be a list of labelled clusters suddenly turns out to be quite useful. But the best way to show this is through an example. Suppose a user enters the query “mouse” in a search engine. The result will usually be a list containing sites about “mouse — the animal”, but also sites about “mouse — the device”. It could be claimed that usually a user is not searching for both. So the idea would be to offer the user the possibility to browse through a list of either “mouse — the animal” or “mouse — the device”. This work presents a clustering approach which could be used by search engines in order to group search results in different categories. This way the user could navigate directly to a specific category and find what he needs faster than browsing page by page using a traditional search engine.

E-mail address: rgaceanu@cs.ubbcluj.ro

[1] The authors wish to thank for the financial support provided from programs co-financed by The Sectorial Operational Programme Human Resources Development, Contract POSDRU 6/1.5/S/3 “Doctoral studies: through science towards society”.

doi:10.1016/j.procs.2011.09.060

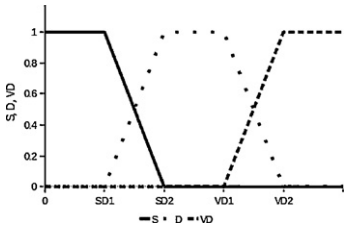

Figure 1. The fuzzy sets *S*, *D* and *VD* corresponding respectively to the linguistic concepts *Similar*, *Different* and *VeryDifferent* are called the *states* of the fuzzy variable *Similarity*. The limits *SD*1, *SD*2, *VD*1, *VD*2 are application specific.

The skeleton of this approach is based on the ASM-like algorithms from [1,3] embelished with features from [2] and [4]. In the ASM model, due to the need for security, the ants are constantly choosing for more comfortable environment to sleep in. The ants feel comfortable among individuals having similar characteristics. In ASM, each data item is represented by an agent, and his purpose is to search for a comfortable position for sleeping in his surrounding environment. While it does not find a suitable position to have a rest, he will actively move around to search for it and stop when he finds one. Similar to the approach from [2] the agents can directly communicate with each other and unlike the approach from [3] they can pro-actively decide to do this at any time in the clustering process. This minimizes the chance of agents getting trapped into local minima and it also speeds-up the clustering process.

The agents decide upon the way they move on the grid according to their similarity with the neighbours, using fuzzy IF-THEN rules [4]. Thus two agents can be similar (*S*), different (*D*) or very different (*VD*). If two agents are similar they would get closer to each other. If they are different they will get away from each other or if they are very different they will get even further away from each other. The number of steps they do each time they move depends on the similarity level. So if the agents are *VD* they would jump many steps away from each other; if they are *D* they would jump less steps away from each other. In the end the ants which are *S* will be in the same cluster. A graphical representation of a fuzzy variable *Similarity* is shown in Figure 1.

In order to evaluate the clustering algorithm a Java application containing the following components has been made: a web crawler, a weighing component and a clustering component. The Web crawler browses the World Wide Web in a methodical, automated manner with the purpose of parsing and indexing the HTML pages. To explore the Web graph the breadth-first algorithm is used. The crawler receives as input a set of starting pages and it extracts the text and the links. The weighting component has two parts: a MySQL procedure and a Java thread that executes the procedure at a given interval of time. When performing a normal search, the dot product between the query vector and the documents from the index is computed and the documents are returned in decreasing order. A matrix of document similarities is given to the clustering component which outputs the clusters and the documents from each cluster.

In order to test the algorithm in a real-world scenario, the well-known Iris dataset was considered. The data set contains 3 classes of 50 instances, each class referring to a type of iris plant. This dataset is appropriate for rather testing classification, but it was preferred for clustering too because the class attribute is given and hence there is a straight forward way to evaluate the algorithm. So apparently it would be ideal for the algorithm to produce 3 clusters of 50 instances each, the 3 clusters corresponding to the given 3 classes. But another reason for choosing this dataset is the fact that two of the classes are not linearly separable and a good clustering algorithm should sense this aspect.

After the algorithm execution, in the resulted final grid configuration, all agents are grouped in clusters. Compared with the classical k-means algorithm where 17 misclassifications are reported, the considered approach misclassified bellow 10 items. Compared to the approaches from [1,3] the number of iterations is reduced from thousands to hundreds by letting the agents to pro-actively decide to directly communicate with each other at any time like in [2]. In [4], the fuzzy IF-THEN rules are used for deciding if the agents are picking up or dropping an item. In this model the fuzzy rules are used for deciding upon the direction and length of the movement. Because no a priori information on the number of clusters is required, this algorithm is a good solution for the web search results clustering problem. Ongoing research is done for improving the fuzzy IF-THEN rules that govern the movements.

References

[1] L. Chen, X.H. Xu, Y.X. Chen, An adaptive ant colony clustering algorithm, in: Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, Vol. 3, 2004, pp. 1387–1392.