# A Bio-Inspired Fuzzy Agent Clustering Algorithm for Serach Engines

Radu D. Găceanu rgaceanu@cs.ubbcluj.ro Babeş-Bolyai University, Cluj - Napoca, Romania Eötvös Loránd University, Budapest, Hungary

UNIUNEA EUROPEANĂ | GUVERNUL ROMÂNIEI MINISTERUL MUNCII, FAMILIEI ŞI PROTECŢIEI SOCIALE AMPOSDRU | Fondul Social European POSDRU 2007-2013 | Instrumente Structurale 2007-2013 | MINISTRUL EDUCAŢIEI, CERCETĂRII, TINERETULUI ŞI SPORTULUI OIPOSDRU | UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA

The SECTORAL OPERATIONAL PROGRAMME HUMAN RESOURCES DEVELOPMENT, Contract POSDRU 6/1.5/S/3 --- "Doctoral studies: through science towards society". Babeş – Bolyai University, Cluj - Napoca, Romania.

## Contribution

We propose an Adaptive Fuzzy Agent Clustering Algorithm based on the ASM (Ants Sleeping Model) in order to resolve the clustering problem.
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 and the parameters are selected and adjusted adaptively. In order to avoid the agents to be trapped in local minima, they are also able to directly communicate with each other. Furthermore, the agent moves are expressed by fuzzy IF-THEN rules and hence hybridization with a classical clustering algorithm is needless. Also, 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.

## Motivation

The possibility to cluster web search results so that the output would be a list of labeled clusters would be very helpful in our opinion. The best way to explain 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". We claim 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" and hence the importance of finding proper clustering algorithms that cluster web search results.

## Token weights

Term frequency inverse document frequency:

$$tfidf_{t,d} = tf_{t,d} \cdot idf_t \tag{1}$$

$$idf_t = log\frac{N}{df_t} \tag{2}$$

$N$ — total number of documents in the collection
$df_t$ — document frequency, the number of documents in the collection containing the term $t$.

## Conclusions and future work

Finding proper clustering algorithms for search engines is an important issue. Our algorithm already behaves well on datasets about the size of Iris. In order to perfrom properly in the area of serach engines an algorithm should also be context aware and perhaps incremental.

## References

[1] L. Chen, X. H. Xu, and Y. X. Chen. An adaptive ant colony clustering algorithm. In *Machine Learning and Cybernetics* , pages 1387–1392, 2004.

[2] C. Chira, D. Dumitrescu, and R. D. Găceanu Stigmergic agent systems for solving np-hard problems. In *KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES* , pages 177–184, June 2007.

[3] R. D. Găceanu and H. F. Pop. An adaptive fuzzy agent a clustering algorithm for search engines. In *MACS2010: Proceedings of the 8th Joint Conference on Mathematics and Computer Science, SELECTED PAPERS, EDITORS: Horia F. Pop and Antal Bege*, Komárno, Slovakia, July 14–17, 2010, pages 342–349.

[4] S. Schockaert, M. D. Cock, C. Cornelis, and E. E. Kerre. Fuzzy ant based clustering. In *Ant Colony Optimization and Swarm Intelligence*, pages 342–349, 2004.

## Fuzzy sets

Let $X$ be the universe of discourse. The sets $S$, $D$, $VD$ of $Similar$, $Different$ and $VeryDifferent$ documents respectively are defined in the following way:

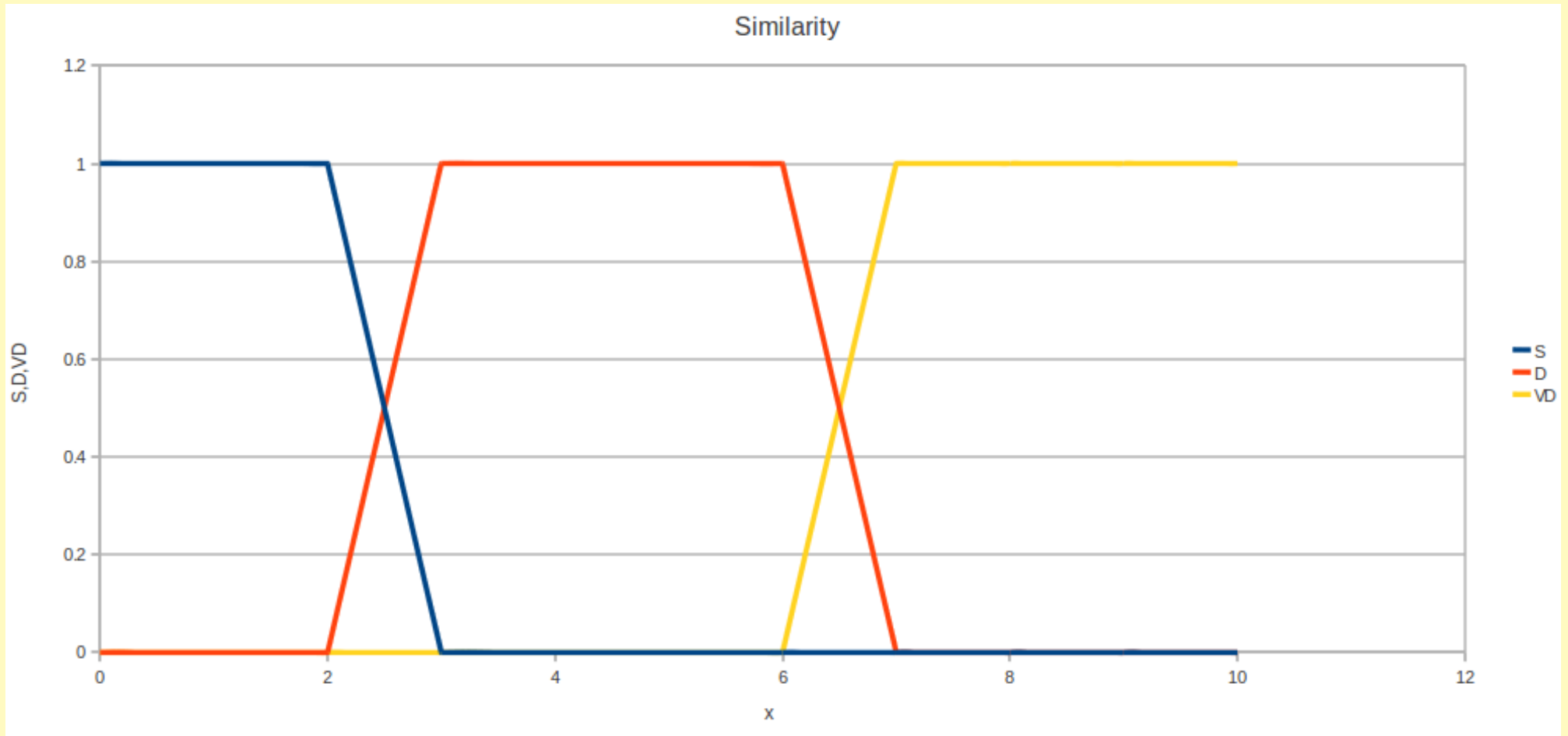
$$S, D, VD : X \to [0, 1] \tag{3}$$

$$S(x) = \begin{cases} 1 & , x \in [0, SD1] \\ (SD1 - x) + 1 & , x \in [SD1, SD2] \\ 0 & , otherwise \end{cases} \tag{4}$$

$$D(x) = \begin{cases} (x - SD2) + 1 & , x \in [SD1, SD2] \\ 1 & , x \in [SD2, VD1] \\ (VD1 - x) + 1 & , x \in [VD1, VD2] \\ 0 & , otherwise \end{cases} \tag{5}$$

$$VD(x) = \begin{cases} (x - VD2) + 1 & , x \in [VD1, VD2] \\ 1 & , x > VD2 \\ 0 & , otherwise \end{cases} \tag{6}$$

## Fuzzy variables – $Similarity$



In the above figure 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 $Similarity$.

## Fuzzy IF-THEN rules

By fuzzy if-then rules we mean conditional statements that comprise fuzzy logic (bellow $a_i$ and $a_j$ denote agents):

if $similarity(a_i, a_j)$ is $S$ then move closer
if $similarity(a_i, a_j)$ is $D$ then move far
if $similarity(a_i, a_j)$ is $VD$ then move further.

## Case study

We have applied our algorithm [3] for clustering web search results. In order to do this we have made a Java application containing the following components: a web crawler, a weighing component and a clustering component. 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 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. The clustering component uses the algorithm defined in this paper [3] and outputs the clusters and the document ids from each cluster. For example, suppose we are searching for the word "mouse". We take only the first 5 search results and send them to the clustering component:

First 5 search results

| Id | Site |
|---|---|
| 0 | http://computer.howstuffworks.com/<br>mouse.htm |
| 1 | http://en.wikipedia.org/wiki/Mouse |
| 2 | http://www.newegg.com/ |
| 3 | http://animal.discovery.com/ |
| 4 | http://computer.howstuffworks.com/<br>&share-redirect?type=facebook&cid=1106 |

The clustering component will output the following clusters: $Cluster0$ (0, 2, 4 ) and $Cluster1$ (1, 3) which seems quite natural as well.

In order to compare our approach with other clustering methdods we have first considered the follwing data set:

Test dataset

| Id | A1 | A2 | A3 | A4 |
|---|---|---|---|---|
| 0 | 0.11 | 0.11 | 0.12 | 0.13 |
| 1 | 0.12 | 0.12 | 0.14 | 0.11 |
| 2 | 0.11 | 0.11 | 0.11 | 0.11 |
| 3 | 0.41 | 0.41 | 0.41 | 0.42 |
| 4 | 0.43 | 0.43 | 0.41 | 0.41 |
| 5 | 0.81 | 0.81 | 0.82 | 0.82 |
| 6 | 0.81 | 0.81 | 0.83 | 0.81 |

We have evaluated our algorithm against this dataset and we have obtained the folowing clusters: $Cluster0$ (0, 1, 2), $Cluster1$ (3, 4) and $Cluster2$ (5, 6). On the same dataset the k-means implementation from Weka reported 3 misclassifications. This suggests that our approach is a promising one.

We also perform tests on bigger datasets like Iris and the results are better compared to k-means: the k-means implementation from Weka has 17 misclassifications while our algorithm has, by now, bellow 10.

## Acknowledgments

The authors wish to thank for the financial support provided from programs co-financed by The SECTORAL OPERATIONAL PROGRAMME HUMAN RESOURCES DEVELOPMENT, Contract POSDRU 6/1.5/S/3 — "Doctoral studies: through science towards society".