

Fuzzy Methods for Customer Relationship Management and Marketing: Applications and Classifications

Andreas Meier
University of Fribourg, Switzerland

Laurent Donzé
University of Fribourg, Switzerland

Managing Director: Lindsay Johnston
Senior Editorial Director: Heather Probst
Book Production Manager: Sean Woznicki
Development Manager: Joel Gamon
Acquisitions Editor: Erika Gallagher
Typesetters: Adrienne Freeland
Print Coordinator: Jamie Snavelly
Cover Design: Nick Newcomer, Greg Snader

Published in the United States of America by
Business Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: <http://www.igi-global.com>

Copyright © 2012 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Fuzzy methods for customer relationship management and marketing: applications and classifications / Andreas Meier and Laurent Donze, editors.

p. cm.

Summary: "This book explores the possibilities and advantages created by fuzzy methods through the presentation of thorough research and case studies"-

-Provided by publisher.

Includes bibliographical references and index.

ISBN 978-1-4666-0095-9 (hardcover) -- ISBN 978-1-4666-0096-6 (ebook) -- ISBN 978-1-4666-0097-3 (print & perpetual access) 1. Customer relations--Management. 2. Marketing. 3. Fuzzy logic. I. Meier, Andreas, 1951- II. Donzi, Laurent.

HF5415.5.F89 2012

658.8'12--dc23

2011042182

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 6

Using a Fuzzy-Based Cluster Algorithm for Recommending Candidates in E-Elections

Luis Terán

University of Fribourg, Switzerland

Andreas Ladner

Institut de Hautes Études en Administration Publique, Switzerland

Jan Fivaz

University of Bern, Switzerland

Stefani Gerber

University of Bern, Switzerland

ABSTRACT

The use of the Internet now has a specific purpose: to find information. Unfortunately, the amount of data available on the Internet is growing exponentially, creating what can be considered a nearly infinite and ever-evolving network with no discernable structure. This rapid growth has raised the question of how to find the most relevant information. Many different techniques have been introduced to address the information overload, including search engines, Semantic Web, and recommender systems, among others. Recommender systems are computer-based techniques that are used to reduce information overload and recommend products likely to interest a user when given some information about the user's profile. This technique is mainly used in e-Commerce to suggest items that fit a customer's purchasing tendencies. The use of recommender systems for e-Government is a research topic that is intended to improve the interaction among public administrations, citizens, and the private sector through reducing information overload on e-Government services. More specifically, e-Democracy aims to increase citizens' participation in democratic processes through the use of information and communication technologies. In this chapter, an architecture of a recommender system that uses fuzzy clustering methods for e-Elections is introduced. In addition, a comparison with the smartvote system, a Web-based Voting Assistance Application (VAA) used to aid voters in finding the party or candidate that is most in line with their preferences, is presented.

DOI: 10.4018/978-1-4666-0095-9.ch006

1 MOTIVATION

The rapid increase of information on the Internet is currently a key issue when one is looking for relevant information. In the political sector, the amount of available information about candidates and political parties is also drastically increasing. This is becoming a significant issue for voters when they face election processes that require them to select their representatives from a big list of candidates since, in many cases, the candidates are relatively unknown to their constituents.

In this chapter, the use of recommender systems for eElections is presented as an alternative to solve the problems of information overload.

Recommender systems are computer-based techniques that attempt to present information about products that are likely to be of interest to a user. This technique is mainly used in eCommerce in order to provide suggestions on items that a customer is, assumable, going to like.

Yager (2003) distinguishes between recommender systems and targeted marketing by considering that a recommender system is a “participatory” system in which the user intentionally provides information about his preferences. In a targeted marketing effort, the recommendation is based on extensional information, which is nothing but information predicated upon the actions or past experiences with respect to specific objects.

A recommender system for eCommerce specifies two basic entities, which include the user (i.e., customer) and the item (i.e., product). The main goal of this type of recommender system that is used in eCommerce is to basically increase the sales of products. The main problems of recommender systems, according to Vozalis et al. (2003), include the following:

- **Quality of Recommendations:** The information received from a recommender system must be reliable; for that reason, recommender systems should minimize the

number of false positive results (i.e., the products that the customer does not like).

- **Sparsity:** A recommendation system is related to the number of recommendations made by customers. The sparsity problem of recommender systems emerges when the number of rated items is small compared to the total number of items, which leads to weak recommendations since the recommender systems are based on similarities between individuals.
- **Scalability:** Increasing the number of users and products elevates the cost in terms of computations in recommender systems.
- **Lost of Neighbor Transitivity:** The correlations between users cannot be expressed unless they have purchased and rated common items.
- **Synonymy:** Recommender systems generally cannot link products with different names that belong to the same category.
- **First Rater Problem:** A product cannot be recommended unless another customer has previously rated it.
- **Unusual User Problem:** This problem refers to users who cannot define their opinion about a product. This causes inconsistent recommendations.

The most-used techniques in recommender systems are based on collaborative filtering technologies according to Guo et al. (2007) and Sarwar et al. (2001). They include collaborative filtering algorithms that are memory-based (i.e., user-based) and model-based (i.e., item-based).

- **Memory-based collaborative filtering algorithms:** These techniques are based on the computation of “neighborhood formation” that uses the user-item matrix R , which contains the ratings of items by users (users are not required to provide their opinion on all items). This may cause the previously mentioned problem of sparsity.

The most common techniques used to reduce the effect of sparsity consist of default voting, preprocessing using averages, the use of filterbots, and the use of dimensionality reduction techniques.

- **Model-based collaborative filtering algorithms:** The model-based (i.e., item-based) collaborative filtering algorithm uses the set of items that the active user u_a has ranked to compute the similarities between this item and a target item (i_j) and to select the n most similar items. In order to compute similarities between items i_i and i_j , model-based techniques isolate the users that have rated both items.

According to Yager (2003), recommender systems, which are used in eCommerce, can be classified as “targeted marketing” since they use information that is based on the actions or past experiences of users. The accuracy of the recommendation in this type of method depends directly on users’ participation. In targeted marketing, the main objective of the recommendation is to increase the margin of sales by recommending products that the users are likely to find appealing.

Given that we focus on recommender systems, which could contribute to improved democratic processes in eGovernment, the definition of Yager (2003) for recommender systems is used in this chapter with the assumption that, in eGovernment systems, the users are willing to participate in the process of providing information about their preferences.

2 ELECTRONIC GOVERNMENT AND ELECTRONIC DEMOCRACY

The European Commission (2010) refers the term *eGovernment* to the use of information technologies to improve the interaction between public administrations, citizens, and the private sector. Three types of relationships are defined

for eGovernment: Administration-to-citizens (A2C), Administration-to-Business (A2B), and Administration-to-Administration (A2A).

Meier (2009) describes an eGovernment framework developed at the University of Fribourg that consists of three levels: Information and Communications, Production, and Participation. It is shown in Figure 1(a).

The lowest level provides information and communication for eGovernment. It focuses on the design of communal Web portals. The second level consists of the actual public services (e.g., electronic procurement, taxation, and electronic payments, among others). The third level refers to citizen participation. This chapter focuses on the participation level; more specifically, eDemocracy.

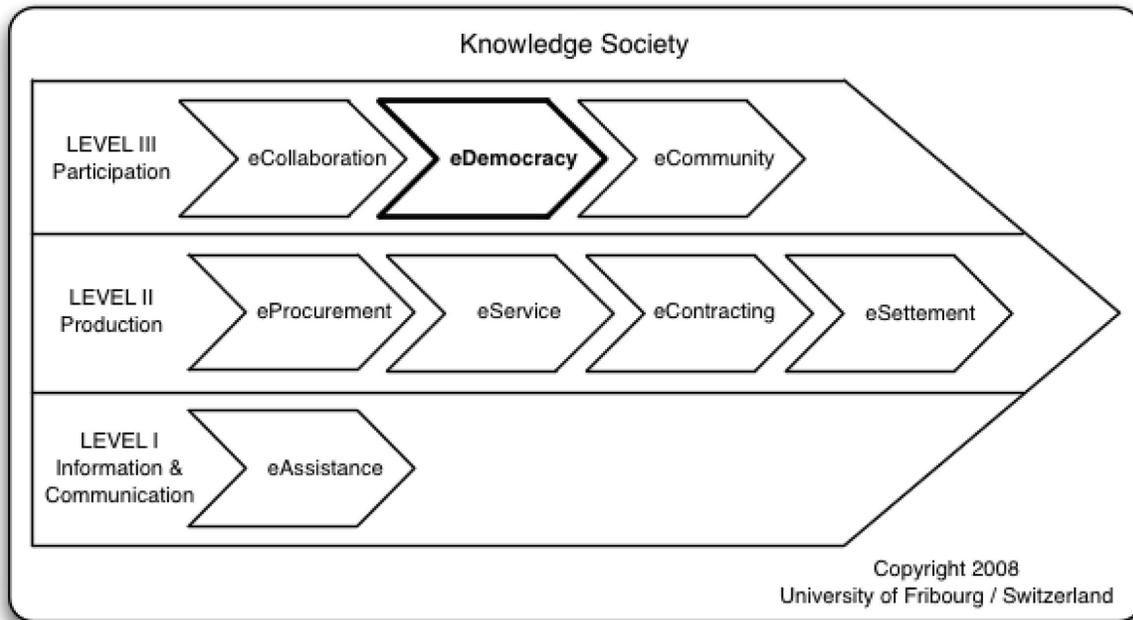
The term *eDemocracy* refers to the use of information and communication technologies that enable citizens to exercise their rights and fulfill their obligations in the information and knowledge society in a time- and place-independent manner.

In his work, Meier (2009) mentions the importance of citizen participation in eDemocracy (e.g., eElection and eVoting). Meier defines the term *eDiscussion* as a stage where citizens could know more about the candidates or the subject in a voting process. It uses information and communication technologies, such as forums, decision aids, and subscription services, among others, to aid voters in making decisions.

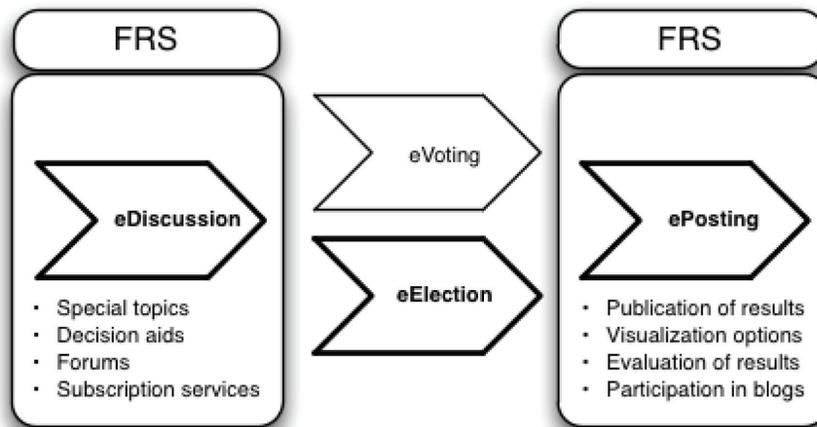
In the same way, once an eVoting or eElection process has been completed, Meier defines the term *ePosting* as another stage that is required on eDemocracy. This stage facilitates the publication of results, and it gives voters the possibility to open discussion channels about the process. Figure 1(b) shows the stages of eVoting and eElection as part of a process chain.

In this chapter, a fuzzy recommender system (FRS) for eElections introduced by Terán et al. (2010) is used. The FRS provides a user-friendly bi-dimensional interface, which can help voters to establish the most similar candidates according

Figure 1. e-Government framework



(a) Knowledge Society



(b) Stages of eVoting and eElection as part of a Process Chain

to their preferences and tendencies. Figure 1(b) shows that the FRS could also be employed on eDiscussion as an add-on tool so that users may become better acquainted with the candidates who are involved in an election. Different types of tools similar to the FRS have been used in

different countries. In this chapter, a Web-based application called *smartvote*, which has been used in Switzerland since 2003, is described.

In addition, Figure 1(b) shows that the FRS could also be used on ePosting as an add-on tool to analyze the congruence between pre- and post-

election periods. This tool could help citizens to improve what is described in the work of Meier (2009) as “public memory” and to enhance the so-called “political controlling.” In the work of Schwarz et al. (2010), an analysis of congruence in the Swiss lower house between 2003 and 2009 indicates that 85% of elected authorities voted in the parliament according to what they claimed on *smartvote* when they were candidates. The FRS could be used with the approach proposed by Schwarz et al. (2010) in which 34 *smartvote* questions came up in the parliament. For that reason, an appropriate design of the questions used to generate candidates’ profiles is extremely important if analysis of congruence is required. In this chapter, the generation of profiles is not covered; rather, it is only assumed that such a profile can be generated and evaluated by using the FRS for both pre- and post-election periods.

Finally, according to Yager (2003) and with the assumption that users are willing to collaborate in providing information about their preferences, recommender systems that are used in eDemocracy are classified as “participatory” systems. These types of recommender systems are suitable for the one-and-only item, according to Guo et al. (2007), where the *recommendation target* is a unique item/event. Examples of one-and-only items include the sale of a house, trade exhibitions, elections, voting, and community building efforts, among others, where recommendations make no use of past actions since these events occurred only once.

3 SMARTVOTE: A VOTING ADVICE APPLICATION

3.1 About Smartvote

The amount of data available on the Internet is growing exponentially and creating what can be considered an almost infinite and ever-evolving network with no discernable structure. This phenomenon not only affects our daily lives, but

it also has an important impact on politics and electoral campaigns. In the work of Fivaz et al. (2010), clear evidence of the increasing popularity of Voting Advice Applications (VAAs)¹ is shown. In this work, Fivaz et al. (2010) describe the use of the *smartvote* (2010) project, which is an online VAA for local, cantonal, and national elections in Switzerland

smartvote was developed in 2002 and 2003 by the Swiss non-profit organization Politools². The core of *smartvote* is the issue-matching module. A couple of months before an election, all candidates receive the *smartvote* questionnaire, either by e-mail or post, and they are asked to mark their responses and return it. The 2007 questionnaire consisted of more than 70 questions on the most important political issues (e.g., “Do you think that nuclear power plants should be shut down?”). The possible answers are “yes,” “rather yes,” “rather no,” and “no”. Candidates do not have the possibility to opt out. They must answer all questions and confirm their answers before they are saved in the *smartvote* database.

About two months before the election, the *smartvote* Web site is made accessible to the voters and leads them through three steps in order to arrive at their individual voting recommendation. First, voters must specify their political profile. They are asked to answer the same questionnaire that the candidates completed, but they can choose between a “deluxe version” that consists of all questions and a “rapid version” that only includes 36 questions. Unlike the candidates, the voters have also a “no answer” option if they wish to leave out a number of questions, and they can weigh the answers according to the level of importance that the issues hold for them. The Web site provides voters with additional background information, including pros and cons for each question. Second, voters have to select the constituency for which they want to receive a voting recommendation, and they must also decide whether they wish to receive a voting recommendation on the individual candidate or party level. Third, *smartvote*

compares the voters' answers with the answers of parties or candidates, including the voters' weighting factors. As a result, the voters receive voting recommendations in the form of individualized "matching lists" with a decreasing ranking of parties or candidates according to their match to the voters' answers (see Figure 2(a))

The Web site also provides visualizations for political profiles:

- The *smartmap* shows political positions in a two-dimensional coordinate system, where the North-South axis represents a liberal-conservative tendency, and the West-East axis represents a left-right tendency. Figure 2(b) shows an example of *smartmap* in which the dots represent a

specific candidate and the colors signify political parties.

- The *smartspider* expresses the strength of attitudes and positions of political candidates based on themes. The *smartspider* has eight axes that are oriented from the perspective of their content to areas of Swiss politics. Figure 2(c) shows an example of a *smartspider* that indicates the political tendencies of a specific candidate and the voter.
- Additionally, *smartvote* offers a comprehensive database of all candidates that consists of a political profile, information about their political careers and agendas, details about their education, professional and family backgrounds, as well as links

Figure 2. Smartvote system outputs

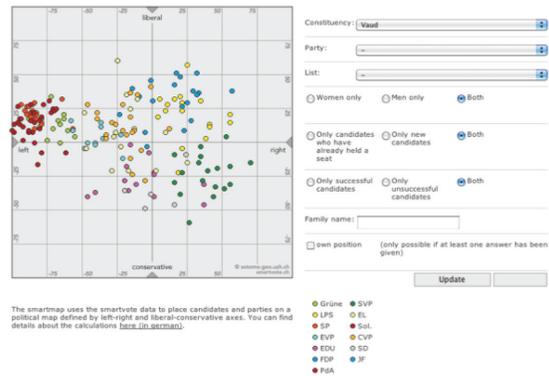
Voting recommendation

Election: Elections to the National Council - 21.10.2007
 Constituency: Vaud
 Number of seats: 13

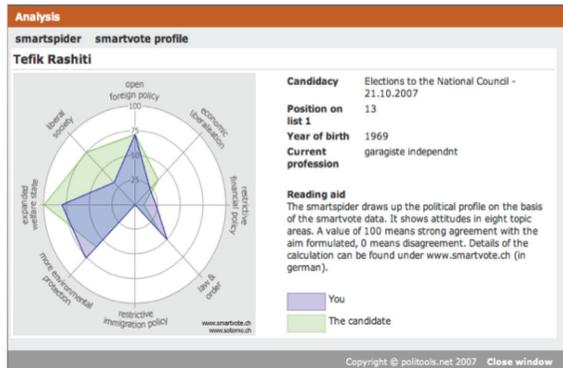
Candidates who have not replied to the questionnaire
 Print the voting recommendation

Name	Match	Analysis
1. Rashiti Tefik Year of birth 1969 (NA)	79.2%	
2. Huguenin Marianne Year of birth 1950 (NA), so far	78.3%	
3. Dwyer Didier Year of birth 1964 (NA)	78.6%	
4. Pascual Isabelle Year of birth 1973 (Sd)	78.6%	
5. Topkiser Nadine Year of birth 1984 (Sd)	78.6%	
6. Delino Jean-Michel Year of birth 1981 (Sd)	78.4%	
7. Walker Marie-Claude Year of birth 1951 (Sd)	78.1%	
8. Schaffner-Jankin Janine Year of birth 1973 (Sd)	78.1%	
9. Jaccard Ludovic Year of birth 1988 (Sd)	77.6%	
10. Zwyssig Josef Year of birth 1956 (NA), so far, elected	77.4%	
11. Freund Sarah Year of birth 1985 (NA)	77.4%	
12. Montagner Stéphane Year of birth 1971 (SP)	77.2%	
13. Zwahlen Pierre Year of birth 1955 (SP)	76.7%	
14. Wirthli Damien Year of birth 1985 (NA)	75.0%	
15. Eysa Olivier Year of birth 1958 (Droit)	75.0%	
16. Terrier Diane Year of birth 1981 (NA)	74.5%	
17. Guex Sébastien Year of birth 1957 (Sd)	74.5%	
18. Aparicio-Mayorez Isabelle Year of birth 1959 (Sd)	74.3%	
19. Jaccard-Berger Christiane Year of birth 1937 (NA)	74.3%	

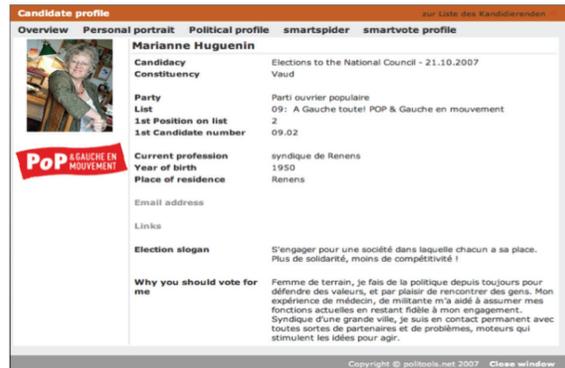
(a) Voting Recommendation



(b) SmartMap



(c) SmartSpider



(d) Candidate Profile

to their personal Web sites and video files. Figure 2(d) shows an example of this type of output.

3.1.1 Participation

smartvote went online for the first time in June 2003 at the start of the national election campaign. Slightly more than 50% of the candidates participated and answered the questionnaire. In the following years, *smartvote* offered its services to several dozen of cantonal and local elections. With every election that it covered, the Web site could increase its popularity and gain more media partners to the extent that, in 2007, *smartvote* was considered a regular part of the electoral campaign. More than 30 media partners (e.g., print media as well as TV and radio broadcasters) supported *smartvote* and integrated the tool and its analyses, such as the *smartspider* graphs of important candidates, into their own news coverage. Due to the cooperation with media partners, *smartvote* present both on- and offline via the media. With regard to this broad coverage it is not surprising that, in the 2007 elections, the number of participating candidates increased considerably: Out of the 3,100 candidates, 85% revealed their political preferences by answering the *smartvote* questionnaire. The program's use by voters also increased. The number of generated voting recommendations grew from 255,000 in 2003 to nearly one million in 2007.

3.1.2 Impacts on Voters' Decision-Making

After the 2007 national elections, a survey among *smartvote* users was conducted. Among other aspects, this survey data allows a look at the impact *smartvote* has had on voters and their decision-making process. Initial analysis of this data showed that its users regarded *smartvote* as an important channel of information (Fivaz & Felder, 2009).

In actuality, they regarded *smartvote* as their most important source of information (see Figure 3).

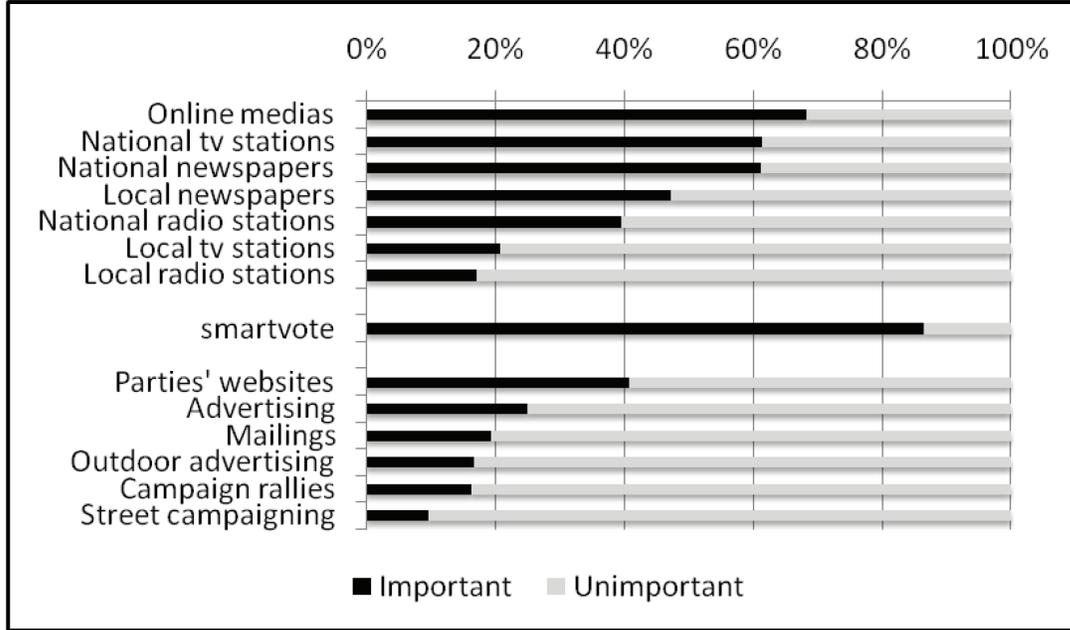
Furthermore, Ladner et al. (2010) found evidence, based at the same survey data, that *smartvote* also directly affected voters' decisions: 67% of the *smartvote* users stated that the voting recommendation that they received influenced their electoral choice. This finding is supported by additional evidence. Ladner et al. (2010) could also show that among *smartvote* users, the amount of swing-voters, or those who voted for a different party than in the previous elections, is significantly higher than among voters who did not use *smartvote*. However, since these are only initial results, they should be treated with caution. Nevertheless, they clearly indicate that the *smartvote* voting recommendation does affect the electoral decision. Thus, it is very important to take a closer look at the applied methods for calculating *smartvote*'s data.

3.2 Smartvote Profile Generation

smartvote's recommendations are based on the similarities that exist between voters and candidates. In a first step, candidates and voters must specify their own political profiles through their responses on a questionnaire, which consists of 60-80 questions regarding political issues. The *smartvote* questionnaire can be divided into groups of questions that each addresses a specific political field. The *smartvote* questionnaire defines two types of questions:

- **Standard Questions**, which are related to the acceptance or rejection of a specific political issue.
- **Budget Questions**, which inquire whether you would spend more or less in certain areas.

Figure 3. Importance of media and campaigning



3.3 Smartvote Match Points Computation

To generate its recommendations, the *smartvote* system uses a statistical method to compute the “match points” by using equation (1):

$$MP_i(v,c) = 100 - |a_{iv} - a_{ic}| + b \quad (1)$$

where $MP_i(v,c)$ represents the number of points of agreement (i.e., match points), a_{ic} and a_{iv} represent the numerical answers given by voter v and candidate c to questions i , and b represents a bonus which applies if, and only if, we are in the presence of a perfect match (“Yes-Yes” and “No-No” combinations).

The next step in the matching calculations is to take into account the relevance that each voter gives to each question. All the questions also have a weighting, which consist of “+”, “=”, and “-.” Depending on the weighting assigned by the voter, the corresponding match points are multiplied by

the factors 2, 1, or 0.5 (weighting value “+” corresponds to factor value 2, weighting value “=” corresponds to factor value 1, and weighting value “-” corresponds to factor value 0.5), as shown in equation (2):

$$MP_i(v,c,w) = (100 - |a_{iv} - a_{ic}| + b) \times w_i \quad (2)$$

where w_i is the weighting value that citizen c gives to a given questions i . A matching value, which is the percentage between voter v and candidate c , is calculated using equations (3) and (4),

$$MP(v)_{Max} = \sum_{i=1}^n a_{iv} \times w_i \quad (3)$$

where $MP(v)_{Max}$ is the theoretical maximum possible match score, which depends only on the answers and weights of voter v .

$$MP(v,c,w) = \frac{MP(v,c,w)}{MP(v)_{Max}} \times 100 \quad (4)$$

where $MP(v,c,w)$ represents the matching value as the percentage between voter v and candidate c .

smartvote can also be used in order to generate recommendations by full lists; in this case, the matching values are computed by using the mean average of all candidates on the list.

3.4 Smartvote Recommendation Output

The output that relevant for this article is the voting recommendations shown in Figure 3(a), which provides a list of candidates that are closest to the voter's political profile. The voting recommendations can be displayed either by candidate or full list.

3.5 Symmetry Problem

There is a drawback in using *smartvote*'s matching point computation. Known as "the symmetry problem," this challenge can be illustrated with the following example:

Two individuals, p_1 and p_2 , both answer the *smartvote* questionnaire as both voters v_1 and v_2 and candidates c_1 and c_2 , respectively. The responses to all of their questions as candidate and voter are the same for p_1 and p_2 . Assume that the answer to a specific question of p_1 is "Yes" (score = 100), and p_2 is "Probably Yes" (score = 75).

The relation between pairs v_1-c_2 and v_2-c_1 are

$$MP(v_1, c_2) = \frac{100 - |100 - 75|}{150} = 0.5$$

$$MP(v_2, c_1) = \frac{100 - |100 - 75|}{100} = 0.75$$

As it is shown, the computation of matching points depends on the maximum possible match score, which, in turn, depends on the answer that is provided by the voter.

4 FUZZY-BASED CLUSTER ALGORITHM FOR RECOMMENDING E-ELECTIONS

Although collaborative filtering-based approaches are more widely used, they are only suitable in the repeat-appeared scenario, which is described by Vozalis (2003). As mentioned in section 2.2, recommender systems for eGovernment must also be suitable in the one-and-only items scenario, in which the *recommendation target* is a unique item/event (i.e., a voter v wants to receive a recommendation of n candidates that are close to his preferences in an election E).

In the voting/election scenario, the recommendation makes no use of past events, given the fact that candidates could be different for each election or they could change their political orientation.

Furthermore, in the recommendation generation, it is necessary to define the elements needed and the output of the system that is developed. As mentioned previously, a recommender system that is used on eElections must be able to recommend a list of n candidates close to the preferences of a specific voter.

In this chapter, a fuzzy-based cluster algorithm for recommending in eElections that was introduced by Terán et al. (2010) is presented. It provides information about the closest candidates to a voter and the distribution of political parties that are organized in fuzzy clusters.

In the section 4.1, the basic concepts of fuzzy logic, fuzzy sets and fuzzy clustering are introduced. Section 4.2 shows the architecture of the fuzzy recommender systems (FRS) and their components, which are better described in sections 4.3 (fuzzy interface), 4.4 (fuzzy profile), 4.5 (recommendation engine), and 4.6 (recommendation output).

4.1 Fuzzy Logic, Fuzzy Sets, and Fuzzy Clustering

Fuzzy logic is a multi-value logic that allows a better understanding of the result of a statement that, in real life, is more approximate than precise. In contrast with “sharp logic,” in which the results of a statement are binary (“true or false,” “one or zero”), fuzzy logic admits a set of truth-values in the interval $[0,1]$. Fuzzy logic is derived from the fuzzy set theory that was introduced by Zadeh (1965) in which a fuzzy set is determined by a membership function with a range of values between 0 and 1. Zadeh (1965) provides a definition of fuzzy sets, which is shown below:

Definition 1. A fuzzy set is built from a reference set that is called the universe of discourse. The reference set is never fuzzy. Assuming $U = \{x_1, x_2, \dots, x_n\}$ as the universe of discourse, then a fuzzy set A ($A \subset U$) is defined as a set of ordered pairs: $\{(x_i, \mu_A(x_i))\}$, where $x_i \in U$, $\mu_A: U \rightarrow [0,1]$ is the membership function of A and $\mu_A(x) \in [0,1]$ is the degree of membership of x in A .

Fuzzy clustering is an unsupervised learning task, which aims to decompress a set of objects into “clusters” based on similarities, where the objects belonging to the same cluster are as similar as possible. In sharp clustering, each element is associated with just one cluster.

The main algorithms used to generate clusters are c-means (sharp clustering) and fuzzy c-means (fuzzy clustering). The fuzzy recommender system that is proposed in this chapter uses the fuzzy c-means algorithm explained below.

The c-means algorithm that was originally proposed by Bezdek (1981) is a method of cluster analysis, which aims to partition n observations into c clusters. Each observation belongs to one and only one cluster. Fuzzy c-means (FCM) is an extension of the c-means algorithm, which allows gradual membership of data points to clusters with different degrees of membership according

to the fuzzy set theory introduced by Zadeh (1965). Thus, fuzzy c-means defines a given set of samples $X = \{x_1, x_2, \dots, x_n\}$, a set of clusters Y_i ($\{i=1, \dots, c\}$ and $\{2 \leq c < n\}$), and a $c \times n$ fuzzy partition matrix $U = [u_{ij}]$. The membership degree u_{ij} of an observation x_i in a cluster Y_i is such that $u_{ij} = \mu_{Y_i}(x_j) \in [0,1]$.

A probabilistic cluster partition defined by the constraints in (5) and (6) guarantees that clusters are not empty, and that the sum of the membership for each x is equal to 1.

$$\sum_{j=1}^n u_{ij} > 0, \forall i \in \{1, \dots, c\} \quad (5)$$

$$\sum_{i=1}^c u_{ij} = 1, \forall j \in \{1, \dots, n\} \quad (6)$$

Thus, the FCM algorithm is based on minimization of an objective function shown in (7),

$$J_m = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|x_j - y_i\|^2 \quad (7)$$

where x_j is the j -th of d -dimensional measured data, y_i is the d -dimensional center of cluster i , m is any real number greater than 1 (m determines the level of fuzziness, and $m=2$ is a typical value that is used), and $\|*\|$ is any norm that expresses the similarity between any measured item and the cluster center.

In (7), $Y=[y_i]$ is a matrix of cluster centers ($i=\{1, \dots, c\}$).

The membership function u_{ij} and the center y_i of clusters are computed in order to take the derivative of the objective function J_m with respect to the parameters to optimize equal to zero, and, taking in to account constraint (6), equations (8) and (9) are obtained.

$$u_{ij} = \frac{1}{\sum_{l=1}^c \frac{\|x_j - y_l\|^2}{\|x_j - y_i\|^2}} \quad (8)$$

$$y_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m} \quad (9)$$

The FCM algorithm is a two-step iterative process that is defined as follows: First, set the input variables c , m , and ε (ε is a termination criteria, and normally $\varepsilon \in [0,1]$). Second, set an iteration number $k=0$. Third, randomly generate a matrix of cluster centers $\mathbf{Y}^{(k)}$. Then, given the initial matrix $\mathbf{Y}^{(k)}$, compute the fuzzy partition matrix $\mathbf{U}^{(k)}$.

Finally, using a repeat-until loop, update $\mathbf{Y}^{(k+1)}$ using $\mathbf{U}^{(k)}$, then update $\mathbf{U}^{(k+1)}$ using $\mathbf{Y}^{(k+1)}$. The process is repeated until the termination criterion is reached ($|\mathbf{U}^{(k+1)} - \mathbf{U}^{(k)}| \leq \varepsilon$). The termination criteria could also be a predefined number of iterations. The FCM algorithm is defined in Table 1.

The outputs of the modified FCM are a fuzzy partition matrix $\mathbf{U}^{(k+1)}$, which contains the membership degree of each element x_j , and a matrix of cluster centers $\mathbf{Y}^{(k+1)}$.

Table 1. FCM Algorithm

<p>Input: c, m, ε Output: $\mathbf{U}^{(k+1)}, \mathbf{Y}^{(k+1)}$ 1: Set iteration number: $k \leftarrow 0$ 2: Generate matrix of cluster centers: $\mathbf{Y}^{(k)} \leftarrow \text{random}$ 3: Compute $\mathbf{U}^{(k)} \leftarrow \mathbf{Y}^{(k)}$ 4: <i>repeat</i> 5: Update $\mathbf{Y}^{(k+1)} \leftarrow \mathbf{U}^{(k)}$ 6: Update $\mathbf{U}^{(k+1)} \leftarrow \mathbf{Y}^{(k+1)}$ 7: <i>until</i> $\mathbf{U}^{(k+1)} - \mathbf{U}^{(k)} \leq \varepsilon$ 8: <i>return</i> $\mathbf{U}^{(k+1)}, \mathbf{Y}^{(k+1)}$</p>

4.2 Architecture

The recommendation process is divided into three steps. In the first step, the voters and candidates must create their profiles by using a fuzzy interface, which is described in greater detail in the following section, to be stored in a database.

In the second step, once all necessary profiles have been created, the user selects the recommendation target and the type of output (i.e., the top-N recommendation or fuzzy cluster analysis). In the final step, once the recommendation engine has computed all information, the user receives the recommendation in the pre-established format. The architecture of the FRS is presented in Figure 4(a). Each element is discussed in depth in the following sections.

4.3 Fuzzy Interface

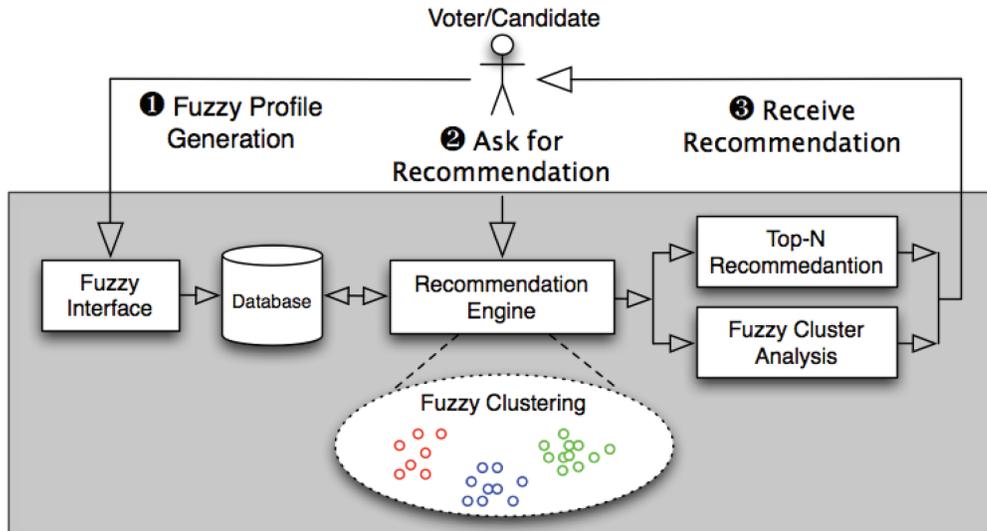
The fuzzy interface is comparable to the *smartvote* interface that is used by candidates and voters to complete a questionnaire regarding political issues (each question has different possible responses). In spite of the flexibility provided by the *smartvote* interface, it can be considered a sharp interface. For this reason, and to guarantee flexibility, a convenient tool is used to determine the level of agreement, disagreement, and relevance for each specific question. Figure 4(b) shows an example of the fuzzy interface.

4.4 Fuzzy Profile

In the work of Terán et al. (2010), in order to generate a recommendation, the voters and candidates have to generate a profile that describes their preferences. A profile representation called a “fuzzy profile (FP)” is proposed. The FP, a multi-dimensional Euclidean space, is defined by:

$$FP_i = (fpc_{i1}, \dots, fpc_{in})$$

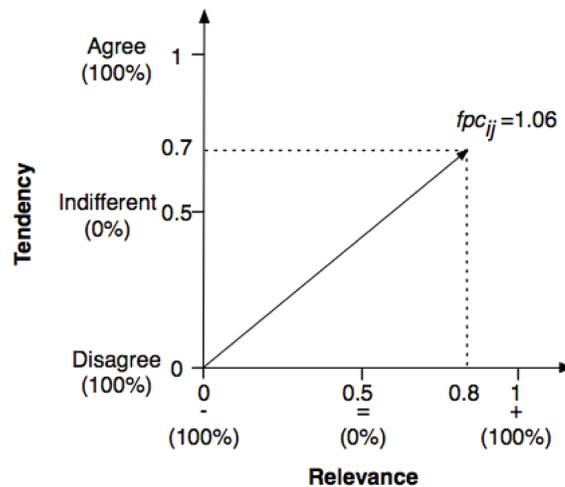
Figure 4. Fuzzy-based cluster algorithm for recommending candidates in e-elections



(a) Fuzzy-based Recommender System Architecture

ENVIRONMENT, TRANSPORT ENERGY		Tendency			No Answer	Relevance		
		Disagree	Indifferent	Agree		-	=	+
3	Should the protection provisions of wolves be relaxed?							

(b) Fuzzy Interface



(c) Fuzzy Profile Component Computation

where FP_i is the FP vector of user i , and fpc_{ij} is the j -th fuzzy profile component (fpc).

Each fpc is equal to the norm of a multi-dimensional Euclidean space defined by:

$$fpc_{ij} = \|(q_{ij1}, \dots, q_{ijl})\| = \sqrt{\sum_{k=1}^l q_{ijl}^2}$$

where fpc_{ij} is the j -th fuzzy profile component of FP_i , and q_{ijk} is the k -th component of fpc_{ij} .

To illustrate the use of a FP, a *smartvote* profile instance of user i (FP_i) that is composed by n questions ($FP_i = (fpc_{i1}, \dots, fpc_{in})$) is used. Each question has two components: “tendency” and “relevance” (q_{ij1} and q_{ij2}), where:

$$fpc_{ij} = \|(q_{ij1}, q_{ij2})\| = \sqrt{q_{ij1}^2 + q_{ij2}^2}.$$

Figure 4(c) shows the results of the fuzzy profile component (fpc) for a general question in the *smartvote* system.

4.5 Recommendation Engine

The FRS is based on the generation of a fuzzy cluster, as shown in Figure 4(a). Once the FP is generated, the next step is to ask for a recommendation. At this point, the user selects a particular event and the type of recommendation (top-N recommendation or fuzzy clustering analysis). The request is sent to the recommendation engine, which processes the query.

To provide a graphical representation of the results that users can easily analyze, the recommendation engine transforms the high-dimensional space of FP to a bi-dimensional space, which reduces the complexity of data analysis. The recommender engine uses a mapping method that was originally proposed by Sammon (1969), which is described in more detail in below.

4.5.1 Sammon Mapping

Clustering-based data mining tools are becoming popular because they are able to “learn” the mapping of functions and systems or explore structures and classes in the data. Sammon’s mapping technique attempts to preserve the inter-pattern distances. It is a well-known technique that is used to transform a high-dimensional space (n -dimensions) to a space with lower dimensionality (q -dimensions) by finding N points in the q -dimensional space.

Denoting the distances between two different points x_i and x_j ($i \neq j$) in the original space as d_{ij} , and the distance between points y_i and y_j in the mapped space as d'_{ij} , then the mapping becomes a minimization problem of the called Sammon’s stress E , as defined in equation (10)

$$E = \frac{1}{\lambda} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{(d_{ij} - d'_{ij})^2}{d_{ij}}, \lambda = \sum_{i=1}^{N-1} \sum_{j=i+1}^N d_{ij} \quad (10)$$

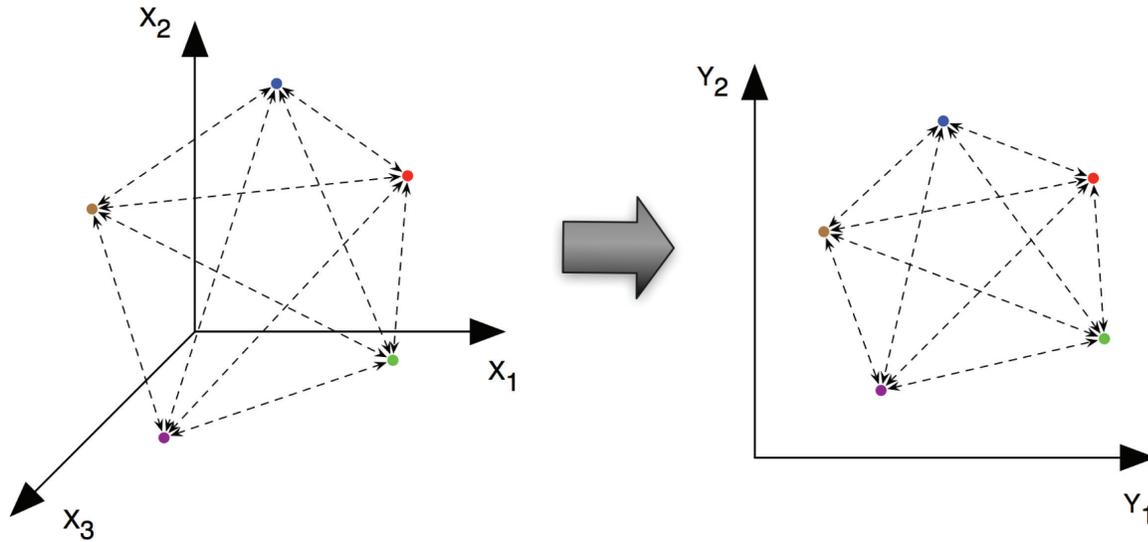
In order to minimize E , Sammon applied a steepest descent technique in which the new y_i at iteration $t + 1$, is given in equation (11)

$$y_i(t+1) = y_i(t) - \alpha \left[\frac{\partial E(t)}{\partial y_i(t)} \right] \left[\frac{\partial^2 E(t)}{\partial^2 y_i(t)} \right] \quad (11)$$

where $y_i(t)$ is the l -th coordinate of point y_i in the mapped space, α is a constant that is empirically computed to be $\alpha \approx 0.3$ or 0.4 . The partial derivatives in (1) are given by:

$$\frac{\partial E(t)}{\partial y_i(t)} = -\frac{2}{\lambda} \sum_{k=1, k \neq i}^N \left[\frac{d_{ki} - d_{ki'}}{d_{ki} d_{ki'}} \right] (y_i - y_{k_i})$$

Figure 5. Illustration of Sammon mapping



$$\frac{\partial^2 E(t)}{\partial^2 y_i(t)} = -\frac{2}{\lambda} \sum_{k=1, k \neq i}^N \frac{1}{d_{ki} d_{ki'}} \left[(d_{ki} - d_{ki'}) - \left(\frac{(y_i - y_{k'})^2}{d_{ki'}} \right) \left(1 + \frac{d_{ki} - d_{ki'}}{d_{ki}} \right) \right]$$

Figure 5 provides an illustration of Sammon mapping from a three-dimensional space to a bi-dimensional space.

This chapter focuses on the Sammon mapping method for the visualization of the clustering results, which preserves inter-pattern distances. Nevertheless, there are three problems that must be taken into account when the Sammon mapping technique is used. They are:

- The prototypes of clusters are usually not known a-priori, and they are calculated along with the partitioning of the data. These prototypes can be vectors that are dimensionally equal to the examined data points, but they also can be defined as geometrical objects (i.e., linear or non-linear subspaces or functions). Sammon mapping is a projection method that is based on the

preservation of the Euclidian inter-point distance norm, so it can be only used by clustering algorithms that are calculated with this type of distance norm. As mentioned in section 4.2.3, fuzzy profiles are defined to be a multidimensional Euclidean space, which fulfills the required condition of the Sammon mapping technique.

- The Sammon mapping algorithm forces one to find, in a high n-dimensional space, N points in a lower q-dimensional subspace, such these inter-point distances correspond to the distances measured in the n-dimensional space. This causes a computationally expensive algorithm, since every iteration step requires the computation of $N(N-1)/2$ distances.
- Finally, this gradient-descent method has the possibility of reaching a local minimum in the error surface, while searching for the minimum of E, so experiments with different random initializations are necessary. In order to avoid this problem, the initialization is estimated using the principal component analysis (PCA) technique,

which maps the data points into a lower dimensional space. The PCA technique is described in the following section.

4.5.2 Principal Component Analysis

The PCA technique, which was introduced by Karl Pearson in 1901, involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. The main objectives of PCA are:

- To identify new meaningful underlying variables.
- Discover or reduce the dimensionality of the data set.

The mathematical background lies in “eigen analysis:” The eigenvector associated with the largest eigenvalue has the same direction as the first principal component. The eigenvector associated with the second largest eigenvalue determines the direction of the second principal component.

In this chapter, we used the second objective. In that case, the covariance matrix of the data set, which is also called the “data dispersion matrix, is defined as follows:

$$\mathbf{F} = \frac{1}{N}(\mathbf{x}_k - \mathbf{v})(\mathbf{x}_k - \mathbf{v})^T$$

where $\mathbf{v} = \bar{\mathbf{x}}_k$ is the mean of the data set. N is equal to the number of objects in the data set.

The PCA technique is based on the projection of correlated high-dimensional data onto a hyper-plane. This mapping uses only the first few q nonzero eigenvalues and the corresponding eigenvectors of the covariant matrix is defined as:

$$\mathbf{F}_i = U_i \Lambda_i U_i^T$$

The covariant matrix \mathbf{F}_i is decomposed to the matrix Λ_i that includes the eigenvalues $\lambda_{i,j}$ of \mathbf{F}_i in its diagonal in decreasing order, and to the U_i matrix that includes the eigenvectors that correspond to the eigenvalues in its columns.

The vector $\mathbf{y}_{i,k} = \mathbf{W}_i^{-1}(x_k) = \mathbf{W}_i^T(x_k)$ is a q -dimensional reduced representation of the observed vector x_k , where the weight matrix \mathbf{W}_i contains the q principal orthonormal axes in its column $\mathbf{W}_i = U_{i,q} \Lambda_{i,q}^{\frac{1}{2}}$.

4.5.3 Fuzzy Cluster Analysis

Once the FP is mapped to a low-dimensional space, FRS generates fuzzy clusters by using the fuzzy c-means algorithm (refer to section 4.1), which requires two main inputs: the number of clusters and a random matrix of cluster centers. For this reason, prior knowledge of the dataset is required. In the eElection scenario, FRS considers the number of clusters to be equal to the number of political parties.

The second input that is required by the fuzzy c-means algorithm is the matrix of initial centers, which is generated randomly. Consequently, the algorithm may converge to a local minimum, given the random nature of the algorithm.

To avoid this problem, a modified version of the FCM algorithm is introduced by Terán et al. (2010). It initializes the matrix of centers with a random member of each political party. The initialization process is based on two assumptions: First, the cluster formation relies on the existence of political parties. Second, the members of political parties have the same ideology according to the ACE project (2006). The modified FCM is presented in Table 2.

The outputs of the modified FCM are a fuzzy partition matrix $U^{(k+1)}$ that contains the membership degree of voters and candidates with respect

Table 2. FCM Modified

Input: c, m, ε Output: $\mathbf{U}^{(k+1)}, \mathbf{Y}^{(k+1)}$ 1: Set iteration number: $k \leftarrow 0$ 2: for $i=1$ to c do 3: $y_i \leftarrow$ random member from P_i 4: end for 5: Compute $\mathbf{U}^{(k)} \leftarrow \mathbf{Y}^{(k)}$ 6: repeat 7: Update $\mathbf{Y}^{(k+1)} \leftarrow \mathbf{U}^{(k)}$ 8: Update $\mathbf{U}^{(k+1)} \leftarrow \mathbf{Y}^{(k+1)}$ 9: until $ \mathbf{U}^{(k+1)} - \mathbf{U}^{(k)} \leq \varepsilon$ 10: return $\mathbf{U}^{(k+1)}, \mathbf{Y}^{(k+1)}$

to each cluster, and a matrix of cluster centers $\mathbf{Y}^{(k+1)}$.

4.5.4 Top-N Recommendation

The top-N candidates similar to voter v are generated by using the bi-dimensional fuzzy profile. The distances of all candidates, with respect to voter v , are computed and the N closest candidates are displayed.

The similarity percentage ($S_{vc_i}(\%)$) of a voter v and the i -th candidate (c_i) is computed using the most distant candidate (d_{max}) as reference. The computation of similarity percentage is shown in equation (12),

$$S_{vc_i}(\%) = 100 - \left(\frac{100 \times d_{vc_i}}{d_{max}} \right) \quad (12)$$

where d_{vc_i} is the distance between voter v and the i -th candidate.

The outputs of this algorithm are the n closest candidates and their similarity percentage with respect to voter v .

4.6 Fuzzy Recommender System Recommendation Output

The objective of the FRS for eElections is to assist voters in making decisions by providing them with information about the candidates that are close to their preferences and tendencies, which could help to improve democratic processes.

Although collaborative filtering-based approaches are more widely used than fuzzy methods, they are only suitable in the repeat-appeared scenario. The recommender systems for eGovernment must also be suitable in the one-and-only items scenario, where the recommendation target is a unique item/event.

In the work of Terán et al (2010), a FRS prototype (FRSP) has been developed to display the results of a recommendation. FRSP has the following input variables: number of clusters (political parties), top-N candidates, total number of candidates, number of questions, number of components of each question, and voter responses. FRSP uses the typical parameters of a fuzzy c-means algorithm ($m=2$, and $\varepsilon=1 \times 10^{-4}$) and Sammon mapping algorithm (PCA is used as initialization method, total iterations = 500, and relative tolerance = 1×10^{-9}).

The recommendation process is given in three steps. First, voters and candidates must create their fuzzy profiles by using a fuzzy interface. The fuzzy interface is a convenient tool that is used to determine the level of agreement/disagreement and relevance of specific questions found in the voter/candidate profiles.

In the work of Terán et al. (2010), the FP of candidates is randomly generated by assuming the answer of candidates and voters. In this chapter, the displayed results correspond to the answers of candidates and voters provided by the *smartvote* (2010) project to the Swiss national elections in 2007. The dataset contains the answers of 257 candidates to the two chambers of the Swiss parliament (i.e., the National Council and the Council

of States), who responded to the 73 questions on the *smartvote* questionnaire. The results that are presented in this chapter consider only the voters who answered the complete questionnaire in order to provide a better accuracy of results. The candidates, who are a part of the dataset, belong to the following political parties:

- Central Democratic Union
- Christian Democratic Party
- Christian-Social Party
- Evangelical Party
- Federal Democratic Union
- Green Party
- Independent Citizens Movement
- Labor Party
- Left Alternative
- Liberal Movement Ecology
- Liberal Party
- Opening Movement
- Radical Democratic Party
- Social Democratic Party
- Swiss Democrats

The recommendation engine used by the FRS transforms multi-dimensional fuzzy profiles into bi-dimensional profiles by implementing the Sammon mapping technique, which attempts to preserve inter-pattern differences, and the PCA technique for the initialization.

The FRS shown in this chapter has two graphical interfaces: the fuzzy cluster analysis graphical interface (FCAGI) and the top-N recommendation graphical interface (TNRGI), which are described in the following sections.

On both graphical interfaces: FCAGI and TNRGI, candidates and the voter are represented using different geometric figures in different colors and orientations. Each political party has a center, this is represented by a geometric figure with the same shape and color of the figures representing the candidates belonging to that political party. The percentage of closeness of a voter to the centers of each political party is presented as percentage and it is placed next to each center of clusters.

The results of the experiments that are presented in the following sections have to be interpreted as mentioned.

4.6.1 Fuzzy Clustering Analysis Graphical Interface (FCAGI)

The FCAGI displays, in a bi-dimensional map, the locations of a voter and the candidates (which are labeled according to their political parties), the clusters that are generated according to each political party, and the percentage of the closeness of voter to each cluster. The FCAGI gives voters the possibility to analyze different political parties and topics by using checkboxes (refer to Figure 6).

The fuzzy cluster analysis is computed by applying the bi-dimensional fuzzy profile. The fuzzy clusters are generated by using a modified version of the fuzzy c-means algorithm with two main inputs: the number of clusters and a random matrix of cluster centers. The number of clusters is equal to the number of political parties, and the random matrix of cluster centers is computed by taking a random member of each political party.

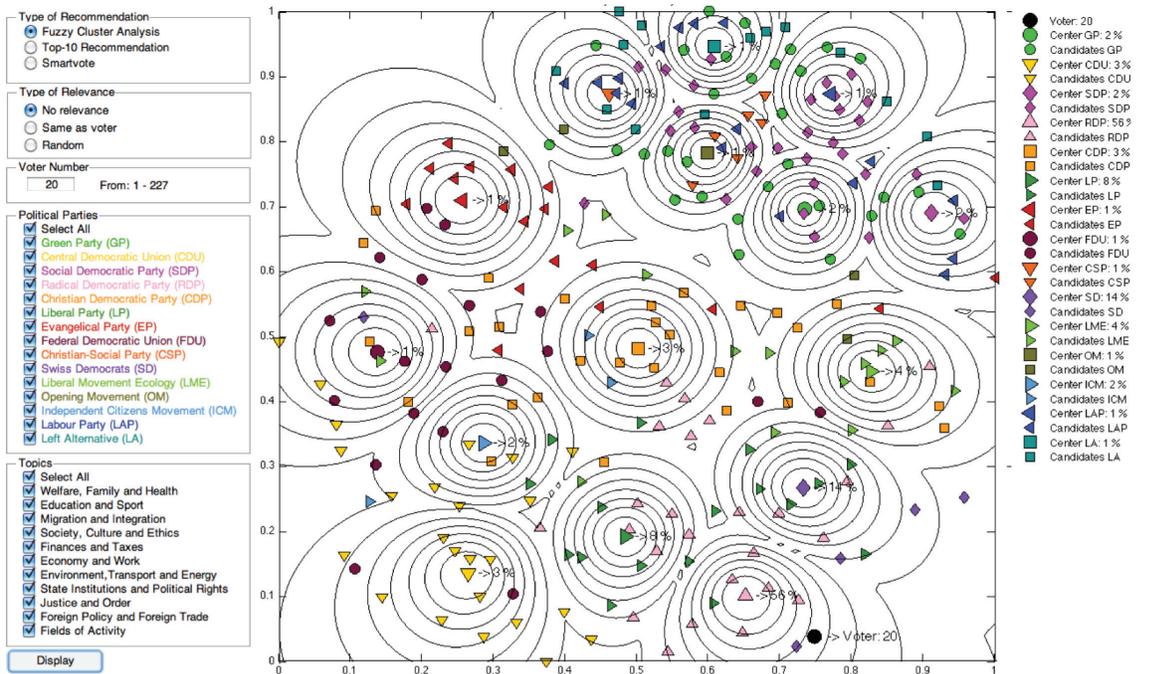
Figure 6(a) shows the FCAGI of a *smartvote* dataset voter by taking into account all of the political parties and topics. The results that are displayed show that candidates, who belong to the same political parties, are located close to each other and tend to form clusters in most cases.

Another interesting result is that some candidates are closest to political parties other than the one that they belong to. This result clearly shows that, even though candidates belong to a political party, they do not necessarily think in the same manner, which was an expected result.

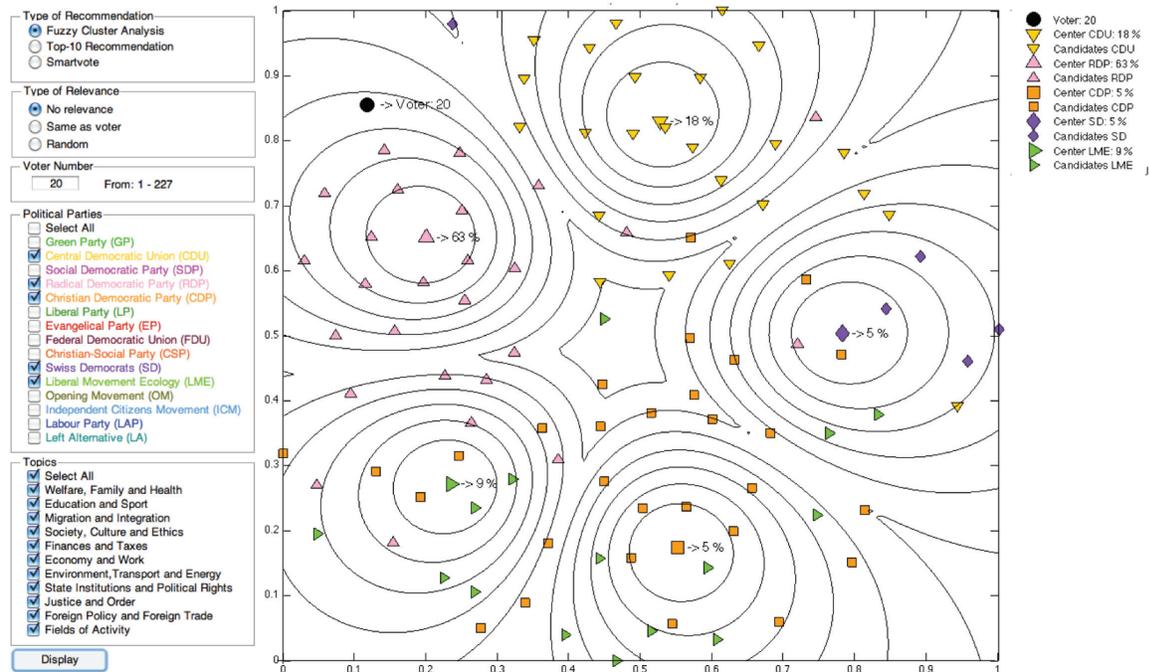
A second experiment presented in this chapter uses the fifth-closest political parties to the same voter used in the experiment that is shown in Figure 6(a). The closest political parties shown in Figure 6(a) are:

- Radical Democratic Party (56% of proximity)
- Swiss Democrats (14% of proximity)

Figure 6. fuzzy clustering analysis graphical interface (FCAGI)



(a) FCAGI of all Political Parties



(b) FCAGI of closest Political Parties to voter

- Liberal Party (8% of proximity)
- Liberal Movement Ecology (4% of proximity)
- Christian Democratic Party (3% of proximity)

Figure 6(b) shows the FCAGI of a *smartvote* dataset user by taking into account all of the topics and the five closest political parties.

The displayed results indicate the formation of clusters. In this experiment, it is also clear that some candidates apparently belong to different political parties. It is possible to explain this observation by realizing that the candidates who belong to a political party do not necessarily think in the same manner.

4.6.2 Top-N Recommendation Graphical Interface (TNRGI)

The TNRGI displays the location of a voter, candidates (labeled by political parties), the clusters generated according to each political party, together with a percentage of closeness of the voter to each cluster, and the N -closest candidates labeled with the percentage of proximity to the voter.

The top- N candidates that are similar to voter v are generated by using the bi-dimensional fuzzy profile. The distances of all candidates, with respect to voter v , are computed and the N closest candidates are displayed.

In order to generate the top- N candidates and similarity percentages, the FRSP computes a vector of distances between the voter and all candidates by using the normalized FP.

Figure 7 shows the results of an experiment by using the closest political parties of the voter that were used in previous experiments and shown in Figure 6(b).

5 DISCUSSION

5.1 Issues, Controversies, and Problems

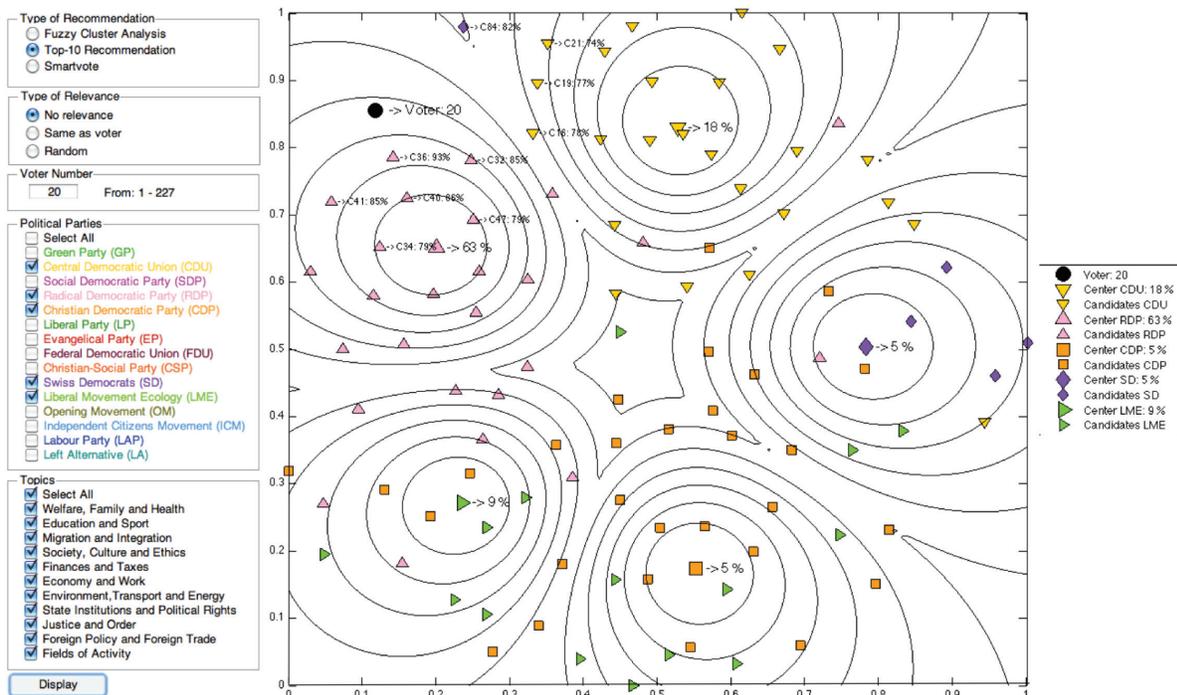
In this chapter, the FRS that was introduced by Terán et al. (2010) is presented. The results shown are using real data that was provided by the *smartvote* project in the Swiss national elections, 2007. Although the obtained results were as expected, there are still some drawbacks that must be taken into account. These drawbacks are related to the generation of profiles, initialization of cluster centers, and the scenario in which the majority of members of each political party do not belong to the same cluster.

Generation of profiles: The fuzzy profile proposed by Terán et al. (2010) considers a multidimensional Euclidean space, which is equal for both voters and candidates. In the case of *smartvote*, the candidates' questionnaire differs from the one that is used by the voters since they can also provide a weighing value to each question.

Initialization of cluster centers: The computation of the centers proposed by Terán et al. (2010) uses a modified version of the fuzzy c-means algorithm, which considers the initial centers by evaluating a random member of the political party. As is shown in Figure 5, the FRSGI displays the locations of politicians in a bi-dimensional space where it is evident that, in some cases, the members of a political party could belong to a cluster that differs from the one that corresponds to his/her political party.

The majority of members do not belong to a cluster: The number of members of a political party must be taken into account for the computation of cluster centers. The fuzzy c-means algorithm moves the centers on each iteration, but it does not consider the case where the majority of members are outside the cluster to which they

Figure 7. Top-N recommendation graphical interface (TNRGI) of closest political parties to voter



belong; otherwise, there could be the case where the majority of members of a political party are outside their own cluster. This case could be presented when having a political party with a small number of members. The fuzzy c-means algorithm moves the center of all clusters until the stop criteria be reached without considering the case when the center of a cluster was moved far away from their members.

5.2 Solutions and Recommendations

In order to overcome the problems mentioned in previous section, the following solutions are proposed.

Generation of profiles: To avoid the problem that arises when using the data provided by *smartvote* (2010) to evaluate the FRS, due to the lack of weighing the values in the questionnaire of candidates, three solutions were proposed:

- Not to take the voter relevance
- Consider that the relevance of voters is equal to relevance of the candidates
- Assume the random relevance of candidates

The first solutions provide results, but the main problem of this approach is the omission of the information that is provided by voters. In the second and third solutions, we are assuming the candidates' answers, which accomplish nothing except adding noise to our results. These options were implemented and evaluated in the FRS that is presented in this chapter.

Computation of cluster centers: As mentioned in previous section, the initialization proposed by Terán et al. (2010) could lead to be in presence of local minimum in the case that the member which was selected as random is located far away of the cluster where the other members of the political party belongs to.

This scenario could be given since that politicians who belong the same political party do not necessarily think in the same manner. In order to avoid this problem, the initialization centers are computed with the mean value of all members of each political party. In this chapter, the FRS was evaluated by using the proposed initialization method.

The majority of members do not belong to a cluster: To avoid the situation in which the majority of members do not belong to the same cluster, it is recommended that stop criteria be added when the majority of members of a political party are out of each cluster. The FRS presented in this chapter does not consider this final recommendation. Therefore, this recommendation should be considered In future research and evaluation.

6 OUTLOOK

It should be noted that, although the prototype is used for eElections, it could be applied for other domains, such as community building and public memory, among others.

In the case of “public memory,” as described in Meier (2009), past behaviors could be taken into account by using voting records to assess the claims of elected authorities prior to the elections, or in the case of candidates who were previously elected. Thus, past behaviors can be used as more reliable profile information.

In the work of Schwarz et al. (2010), an evaluation of congruence of pre-election statements on *smartvote*'s voting advice application and post-election behavior in the Swiss lower house between 2003 and 2009 can be found. In this work, the evaluation is conducted by taking 34 questions from the *smartvote* questionnaire were subsequently voted on in the parliament. The results show an 85% political congruence (i.e., acting in accordance to the positions that were revealed in the *smartvote* questionnaire before the election).

In future research, the FRS could also be used to analyze whether the candidates really acted as they claimed. The FRS could display their location in the bi-dimensional map as candidates and how they behave while they were elected authorities to allow voters to easily understand politicians' behavior.

The FRS has to be evaluated and compared by using different methods for dimensionality reduction and clustering algorithms such as:

- Kmeans (clustering algorithm)
- Gustafson-Kessel Algorithm (clustering algorithm)
- Gath-Geva Algorithm (clustering algorithm)
- Lineal Discrimination Analysis (dimensionality reduction)
- Multidimensional Scaling (dimensionality reduction)
- Fuzzy Sammon Mapping (dimensionality reduction)

Once the FRS has been evaluated and compared with other methods and algorithms, the evaluation of the fuzzy interface, FRSGI, and TNRGI must be made. It is important to evaluate how the users of such system react when using the proposed tool in order to evaluate and improve it before putting it in production.

In future research, the FRS could be also used in so-called community building. The main idea is that when voters have filled their profiles, this information could be used for creating communities, which share similar political points of view. The same idea could be used when creating discussion forums, in which the creator of a forum topic could include only those citizens who are close to him, which could generate more constructive debates that can easily end up in consensus.

7 CONCLUSION

In this chapter, the FRS for eElections used in eGovernment proposed by Terán et al. (2010) is presented. The main objectives of FRS are to increase citizens' participation, and to provide more information to citizens about candidates, which could possibly improve democratic processes.

The recommender system approach differs from collaborative filtering methods in that they are based on past experiences. It is also suitable in the one-and-only scenario in which events such as voting and election processes occur only once.

Another important feature that is introduced in the proposed recommendation system is the fuzzy clustering analysis. It provides a graphical representation of political parties distributed in clusters, helping citizens to analyze politician's behavior according to similarities among them.

Fuzzy clustering analysis differs from classic clustering (i.e., sharp clustering) in that the observations belong to one, and only one, cluster. Moreover, classic clustering makes no use of gradual membership.

The main differences between the proposed recommender system and the *smartvote* system are the computation of similarities and the way that recommendations are displayed to users.

The *smartvote* system computes similarities that are based on "match points" (section 3.1), and the recommendations are displayed as a list of the closest candidates with a percentage of similarity.

The FRS computes similarities based on distances in a high-dimensional space. In addition, it computes fuzzy clusters based on the number of political parties, which are part of an eElection process (section 5.1) where candidates and voters are described in a finest granularity and can belong to several clusters.

The recommendations in the FRS are displayed in a bi-dimensional space, which includes the percentage of similarity of the n -closest candidates. Therefore, relationships to closest "neighbors" can be derived and analyzed.

8 REFERENCES

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York, NY: Plenum Press.

European Commission. (2010). *ICT for government and public services*. Retrieved September 10, 2010, from http://ec.europa.eu/information_society/activities/egovernment/index_en.htm

Fivaz, J., & Felder, G. (2009). Added value of e-democracy tools in advanced democracies? The voting advice application *smartvote* in Switzerland. In Shark, A. R., & Toporkoff, S. (Eds.), *Beyond Egovernment—Measuring performance: A global perspective* (pp. 109–122). Washington, DC: Public Technology Institute and ITEMS International.

Guo, X., & Lu, J. (2007). Intelligent e-government services with personalized recommendation techniques. *International Journal of Intelligent Systems*, 22, 401–417. doi:10.1002/int.20206

Ladner, A., Fivaz, J., & Pianzola, J. (2010). *Impact of voting advice applications on voters' decision-making*. Paper presented at the Conference on Internet, Politics, Policy 2010: An Impact Assessment. Oxford.

Meier, A. (2009). *E-democracy & e-government*. Berlin, Germany: Springer.

Mobashe, R., Burke, R., & Sandvig, J. (2006). *Model-based collaborative filtering as a defense against profile injection attacks*. Paper presented at the 21st National Conference on Artificial Intelligence (AAAI'06), Boston, Massachusetts.

Pearson, K. (1901). Principal components analysis. *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 6, 566.

Project, A. C. E. (2006). *Roles and definitions of political parties*. Retrieved September 10, 2010, from <http://aceproject.org/ace-en/topics/pc/pca/pca01/pca01a>

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *Journal IEEE Transactions on Computers*, 18, 401–409. doi:10.1109/T-C.1969.222678

Sarwar, B., Karypis, G., & Konstan, J. (2001). *Item-based collaborative filtering recommendation algorithms*. Paper presented at the 10th International World Wide Web Conference, China, Hong Kong.

Schwarz, D., Schädel, L., & Ladner, A. (2010). Pre-election positions and voting behavior in Parliament: Explaining positional congruence and changes among Swiss MPs. *Journal Swiss Political Science Review*, 16(4), 533–564. doi:10.1002/j.1662-6370.2010.tb00440.x

Smartvote. (2010). Retrieved September 10, 2010, from <http://www.smartvote.ch/>

Terán, L., & Meier, A. (2010, september). *A fuzzy recommender system for e-election*. Paper presented at the International Conference on Electronic Government and the Information Systems Perspective (EGOVIS 2010), Bilbao, Spain.

Valente de Oliveira, J., & Witold, P. (Eds.). (2007). *Advances in fuzzy clustering and its applications*. West Sussex, England: Wiley. doi:10.1002/9780470061190

Vozalis, E., & Margaritis, K. (2003). *Analysis of recommender systems' algorithms*. Paper presented at the Sixth Hellenic European Conference on Computer Mathematics and its Applications (HERCMA 2003), Athens, Greece.

Yager, R. (2003). Fuzzy logic methods in recommender systems. *International Journal of Fuzzy Sets and Systems*, 136, 133–149. doi:10.1016/S0165-0114(02)00223-3

Zadeh, L. (1965). Fuzzy sets. *Journal of Information and Control*, 8, 338–353. doi:10.1016/S0019-9958(65)90241-X

KEY TERMS AND DEFINITIONS

eGovernment: The use of information technologies and knowledge in the internal processes of government and the delivery of products and services of the state to citizens and industry.

eDemocracy: The use of information and communication technologies to enable citizens to exercise their rights and fulfill their obligations in the information and knowledge society in a time- and place-independent manner.

Fuzzy Cluster Analysis: A method used by the fuzzy recommender system to display, in a bi-dimensional map, the locations of a voter and candidates, as well as the clusters that are generated according to each political party.

Fuzzy Interface: A graphical interface used by candidates and voters in the fuzzy recommender system to complete a questionnaire regarding political issues.

Fuzzy Profile: A multi-dimensional Euclidean space used by the fuzzy recommender system to general profiles of candidates and voters.

Fuzzy Recommender System: An application that is used to recommend candidates who are close to voters' preferences and tendencies in an eElection process.

Top-N Recommendation: A display of the location of voters and candidates, and the clusters that are generated according to each political party, together with a percentage of closeness of voters to each cluster, and the *N*-closest candidates labeled with the percentage of proximity to the voter.

ENDNOTES

- 1 Examples of VAA's: Who do I vote for? (www.whodoivotefor.co.uk), Kieskompas (www.kieskompas.nl), Project Vote Smart (www.vote-smart.org), Glassboth (www.glassboth.org), On the Issues (www.ondthe-

issues.org), Openpolis (www.openpolis.it), Manobalsas (www.manobalsas.lt), smartvote Ottawa (www.smartvoteottawa.ca), Political Compass (www.politicalcompass.org), EU-Profiler (www.euprofiler.eu),

Niqash(www.nicahs.org), and Wahlkabine (www.wahlkabine.at), Politikkabine (www.politikkabinne.at).

² *Politools* is a private association in Bern: www.politools.net