



Applying Hidden Markov Models to Voting Advice Applications

Marilena Agathokleous*  and Nicolas Tsapatsoulis

*Correspondence:
mi.agathokleous@edu.cut.ac.cy
Cyprus University of Technology, 30,
Arch. Kyprianos str., Limassol, 3036,
Cyprus

Abstract

In recent times, a phenomenon that threatens the representative democracy of many developed countries is the low voter turnout. Voting Advice Applications (VAAs) are used to inform citizens about the political stances of the parties that involved in the upcoming elections, in an effort to facilitate their decision making process and increase their participation in this democratic process. VAA is a Web application that calls the users and parties to state their position in a set of policy statements, which are based on the current affairs of their country and then it recommends to each user the party that better fits their political views. SVAAs, a social recommendation approach of VAAs, on the other hand, base their recommendation on the VAA community of users in a collaborative filtering manner. In this paper we resort to Hidden Markov Models (HMMs) in an attempt to improve the effectiveness of SVAAs. In particular, we try to model party-supporters using HMMs and then use these models to recommend each VAA user the party whose model best fits his/her answer sequence of the VAA policy statements. HMMs proved to be effective machine learning tools for sequential and correlated data and this is the main rationale behind this study. VAA policy statements are usually correlated and grouped into categories such as external policy, economy, society, etc. As a result, opting from the various answer choices in each policy statement might be related with selections in previous and subsequent policy statements. Given that the order of policy statements is kept fixed within each VAA one can assume that (a) answer patterns (sequences of choices for all policy statements included in the VAA) can be found that characterise 'typical' voters of particular parties, and (b) the answer choice in each policy statement can be 'predicted' from previous answer choices. For our experiments we use three datasets based on the 2014 elections to the European Parliament (<http://www.euvox2014.eu/>), which are publicly available through the Preference Matcher website (http://www.preferencematcher.org/?page_id=18).

Keywords: Hidden Markov Models; Voting Advice Applications; collaborative filtering; expectation maximisation; recommender systems

1 Introduction

The low voter turnout is a phenomenon that affects the most advanced democracies, since many citizens are not represented properly. Ladner and Pianzola [1] specifically mentioned Switzerland, where the voter turnout does not exceed 50% by 1975. Citizens, partly because of their lack of knowledge on the political issues, tend to avoid this democratic decision making process. E-democracy tools provide the means to use Information and Com-

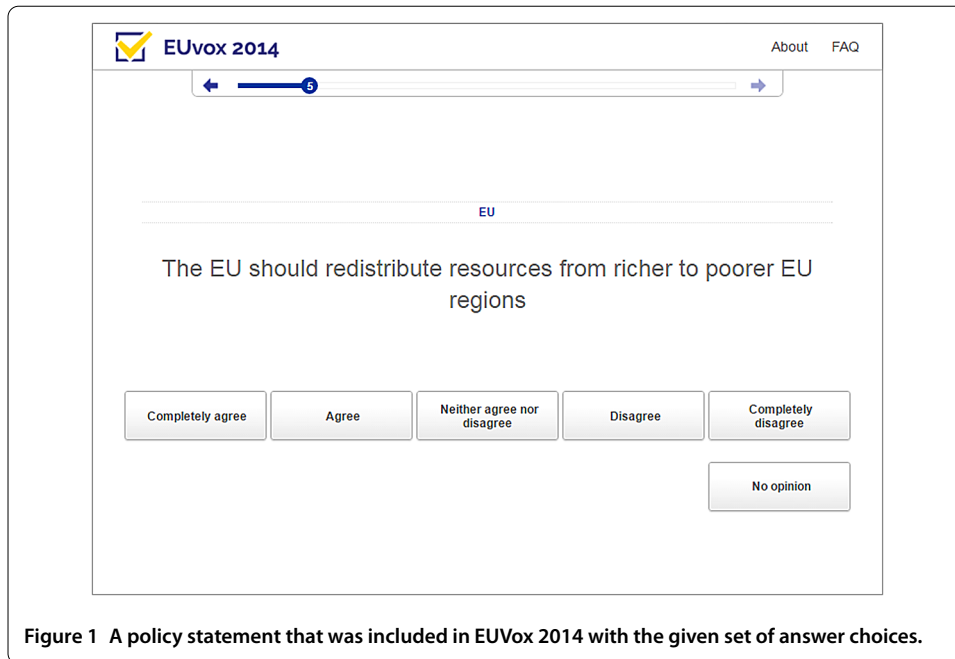


Figure 1 A policy statement that was included in EUVox 2014 with the given set of answer choices.

munications Technologies (ICTs) in order to inform people about the political stances of the parties who take part in the upcoming elections aiming to increase citizen participation and to promote direct involvement in political activities [2].

Voting Advice Applications (VAAs) (according to Fivaz *et al.* [3] this term is widely used since 2007) are e-democracy tools that undertake the role of the ‘tipster’ and facilitate citizens’ decision making process by matching their political stances with those of parties. Findings have shown that VAAs’ recommendations affect the decision making process of a significant part of voters, especially if these people are undecided, women, aged under 34, or they are voting for the first time [3, 4]. In addition, and as reported in some studies [4, 5], in many cases VAAs were responsible for increasing the participation of citizens in elections.

VAAs ask users and parties to fill a specific questionnaire that contains a number of policy statements (see a policy statement example in Figure 1), which are created according to issues that concern the nation in the time of elections and represent important political, economic and social issues [6, 7]. The recommendation process that a VAA traditionally follows contains two main steps: first the users’ answers are matched with the parties’ answers and then the VAA ranks, in decreasing order of the matching score, the parties according to party-user ‘similarity’ (see Figure 2 for party ranking based on party-user similarity as computed in traditional VAAs).

In addition to the policy statements, many VAAs ask users to answer a number of supplementary questions. One of these supplementary (opt in) questions is the vote intention of user (i.e., which party the user intends to vote in the upcoming election). An example of how the supplementary questions appear in the EUVox, the EU-wide VAA that was designed for the 2014 elections to the European Parliament, is shown in Figure 3. Here we can see that the vote intention of the user for the European elections is found in the second question. The vote intention variable is commonly used, to cluster the community of VAA

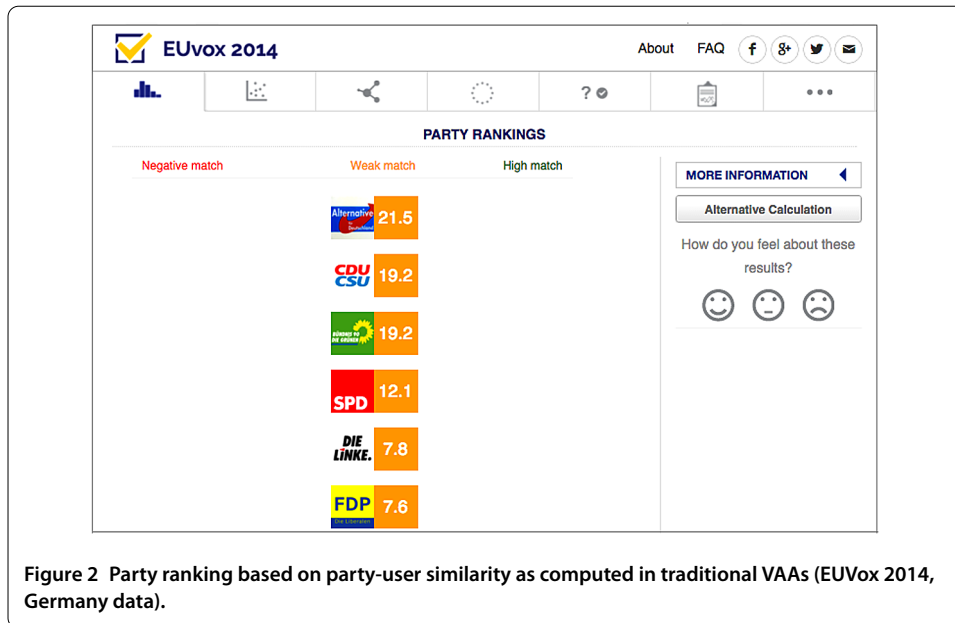


Figure 2 Party ranking based on party-user similarity as computed in traditional VAAs (EUvox 2014, Germany data).

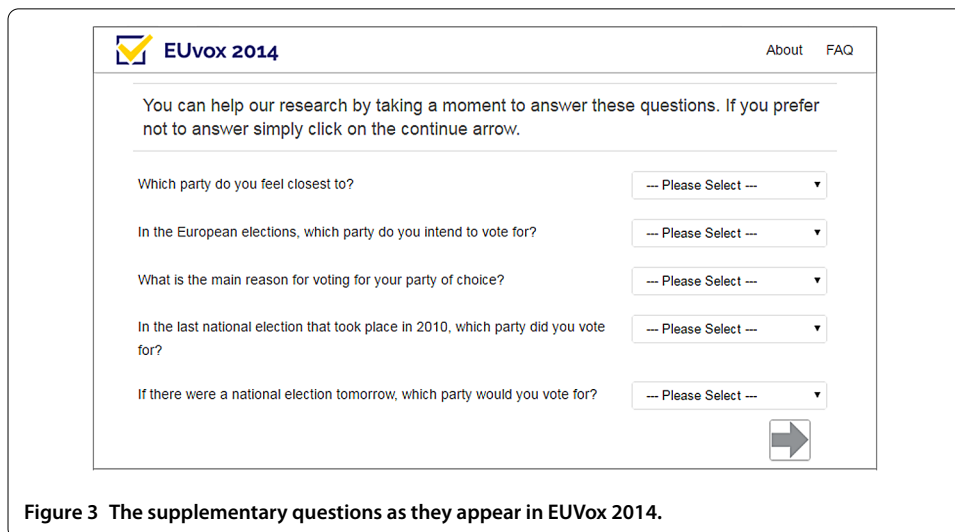
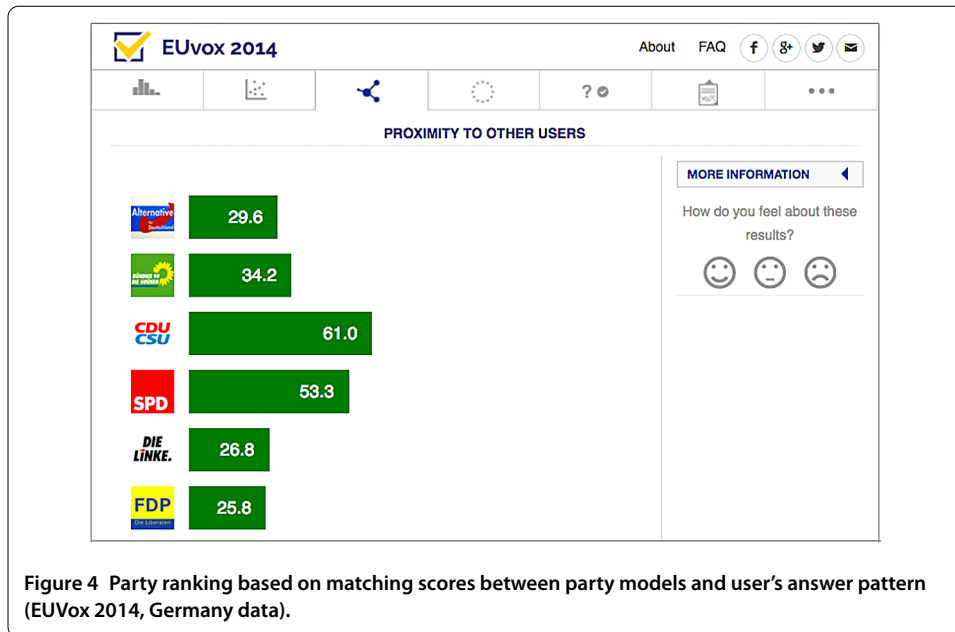


Figure 3 The supplementary questions as they appear in EUvox 2014.

users into disjoint groups, in a social variation of VAAs, called SVAA, that is based on the collaborative filtering [8] philosophy.

SVAA has the same policy questionnaire with the traditional VAAs but provide a radically different matching technique. In SVAA recommendation, parties' answers are ignored and recommendation is given based on the community of VAA users using collaborative filtering techniques [8, 9]. During the training phase of the SVAA design, a number of groups is created corresponding to the parties that participate in the elections. Each group includes the users that expressed vote intention for a specific party; thus, in the training phase only the filled questionnaires of the VAA users that answered the vote intention supplementary question and expressed preference for a particular party (i.e., they did not choose answers like 'not decided yet' or 'I will not vote' and similar) are included. VAA users who are clustered into the same group, which means they support the same party, are likely to create similar answer patterns (i.e., fill in the policy questionnaire in



a similar way), since they commonly share the same political opinions. So, party models can be created for each group to show the common way, if any, in which the users in each group fill the online questionnaire. Once party models are created, the SVAA starts its operation phase. The sequence of answers (i.e., filled in questionnaire) of a VAA user feed the party models and produce in their outputs party-user similarity scores. Recommendation is given to the user in a decreasing order of these similarity scores. An example of the matching scores presented to the user according to the SVAA philosophy is shown in Figure 4. SVAAAs proved to make better voting predictions than the traditional VAA matching schemes between users' and parties' profiles [10, 11]. In addition, as recorded by users' feedback through the emoticons shown in the right part of Figure 2 and Figure 4, SVAA recommendation surpasses VAA recommendation in terms of users' satisfaction [12].

In order to tackle the recommendation problem of SVAAAs, machine learning techniques [13, 14] can be used to indicate the likelihood of a user to belong to a class of the VAA community of users; as already mentioned each class corresponds to a specific party. In essence what is accomplished with machine learning is the modelling of parties groups based on their users' answer patterns, which are created by their responses to the online questionnaire. In this paper we investigate the application of Hidden Markov Model (HMM) classifiers for party-user similarity estimation, in an effort to improve the effectiveness of social vote recommendation. HMM classifiers provide a way to apply machine learning to data represented as a sequence of correlated observations [13]. Although the order in which policy statements are displayed to users is not important in VAAs, the policy statements are usually correlated and grouped into categories (e.g., external policy, economy, society, etc). Thus, opting from the various answer choices in each policy statement might be related with selections in previous and subsequent ones. Given that the order of policy statements is kept fixed within each VAA one can assume that (a) answer patterns, i.e., sequences of choices for all policy statements included in the VAA, can be found that characterise 'typical' voters of particular parties, and (b) the answer choice in each policy statement can be 'predicted' from previous answer choices. When users answer the policy

statements, they are incrementally producing a sequence of symbols. Whenever a process includes a sequence of dependent observations, HMM classifiers can be used to model input sequences as generated by a parametric random process. This is our basic rationale for employing HMMs for obtaining similarity matching between parties and users in SVAA.

An HMM classifier models a sequence of symbols from observed data without knowing the sequence of states it has to follow to generate these observations [13]. We assume that VAA users, who support the same party, produce similar sequences of symbols (answer patterns). Therefore, HMM classifiers can be used (a) to create simple and compact models for each party to show the ‘path’ that users, who support the same party, follow to answer the online questionnaire, and (b) to classify every new user into the party in which is most likely to belong, according to their given answers/symbols on the policy statements.

In short, the purpose of our paper is to introduce an SVAA method for similarity matching between parties and users based on HMMs and investigate its performance based on the accuracy of predicting their voting intention. We show that, even if the order in which the policy statements are answered in a VAA does not really matter, the HMM classifier performs quite well in estimating vote intention of unseen users. Nevertheless, the HMMs’ performance relies on the smooth distribution of samples per party and on the consistency between the answers of the users, who are classified as belonging to these parties. Therefore, in the cases where these conditions are not met, the results may not be satisfactory and outlier and/or rogues detection may be required [15]. Experiments were also conducted for a comparison of HMM classifier with the traditional VAA party-user matching method and other SVAA’s native solutions to recommend parties to users. We observe that HMM classifier shows the highest performance between the traditional party-coding recommendation method and other party-supporters modelling methods.

To the best of our knowledge this is the first time HMMs are used to compute party-user similarity either in VAAs or in SVAA. For our experiments we use three datasets derived from EUVox 2014. EUVox was sponsored financially by the Open Society Initiative for Europe (European Elections 2014) and the Directorate-General for Communication of the European Parliament (area of Internet-based activities/online media 2014) to help voters to have quick access to information according to the political positions of the parties that took part in the 2014 elections to the European Parliament (see <http://www.euvox2014.eu/>). The datasets differ in size, in the number of parties participating in the elections and in the population’s distribution percentage among the various parties. The corresponding datasets are available to other researchers working in the areas of VAA and Web based recommender systems through the Preference Matcher website.^a

2 Problem formulation

The basic aim of a traditional VAA is to recommend parties to users. In such a case there is a set of N users $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, a set of U policy statements (or issues) $Q = \{q_1, q_2, \dots, q_U\}$, and a set of D political parties $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_D\}$. Each user $\mathbf{x}_j \in X$ and each political party $\mathbf{p}_i \in P$, has answered each policy statement $q_k \in Q$.

Based on their answers, every political party or user can be represented in a vector space model:

$$\mathbf{x}_j = \{x_{(j,1)}, x_{(j,2)}, \dots, x_{(j,k)}, \dots, x_{(j,U)}\} \tag{1}$$

$$\mathbf{p}_i = \{p_{(i,1)}, p_{(i,2)}, \dots, p_{(i,k)}, \dots, p_{(i,U)}\}, \tag{2}$$

where $x_{(j,k)}, p_{(i,k)} \in L$ are the answers of the j -th user and i -th party, respectively, to the k -th policy statement. The vectors \mathbf{x}_j and \mathbf{p}_i are, usually, named user and party *profiles* respectively.

A typical set of answers is a 6-point Likert scale: $L = \{1$ (Completely disagree), 2 (Disagree), 3 (Neither agree nor disagree), 4 (Agree), 5 (Completely agree), 6 (No opinion) $\}$. In several cases, and in the majority of SVAA methods proposed so far, the sixth point is not taken into consideration since it does not correspond to a particular stance and it is usually replaced with the third point (Neither agree nor disagree). In this work we keep the sixth point as a distinct emission symbol but not as a distinct state (see also Section 3). As a result the set L , in the context of this work, becomes: $L = \{1, 2, 3, 4, 5, 6\}$. In Figure 1 an example, of how the answer choices appear to VAA users, is shown.

The VAA recommendation task tries to approximate the unknown relevance $h(j, i)$ of user j to party i given the user and party answers \mathbf{x}_j and \mathbf{p}_i respectively, and then to suggest a ranking of political parties based on user-party relevance. In machine learning terms, the task is to approximate the hidden function $h(j, i)$ with a function $\hat{h} : \mathbb{R}^U \times \mathbb{R}^U \rightarrow \mathbb{R}$, where $\hat{h}(\mathbf{x}_j, \mathbf{p}_i)$ is the estimation of the relevance of user j with political party i . Typically $\hat{h}(\mathbf{x}, \mathbf{p}) \in [0, 1]$. In any case, the top suggestion p_q^j for user j should be:

$$p_q^j = \underbrace{\operatorname{argmax}}_i (\hat{h}(\mathbf{x}_j, \mathbf{p}_i)). \tag{3}$$

One of the supplementary (opt in) questions (see Figure 3) that many VAAs ask users to answer, in addition to the U policy statements, is the vote intention of user, i.e., which party the user intends to vote in the upcoming election. SVAAAs usually use the vote intention variable v_{ij} to a-priori cluster VAA users into parties, i.e., party-supporters as indicated by the vote intention variable are clustered together. Then statistical or machine learning approaches are used to create party models. Thus, for every party i a model \mathbf{M}_i is created using as training examples the subset \mathbf{Tr}_i of user profiles who expressed voting intention for party i , that is $\mathbf{Tr}_i = [\mathbf{x}_j | v_{ij} = i]$. Then these models can be exploited to provide a recommendation based on collaborative filtering [16] that takes advantage of VAA's user community. In this case the top recommendation p_q^j for user j is given by:

$$p_q^j = \underbrace{\operatorname{argmax}}_i (\hat{h}(\mathbf{x}_j, \mathbf{M}_i)). \tag{4}$$

In this work we use Hidden Markov Models to create the party models \mathbf{M}_i . Thus, Eq. 4 becomes:

$$p_q^j = \underbrace{\operatorname{argmax}}_i (\hat{h}(V^j, \lambda_i)), \tag{5}$$

where V^j is the set of observations corresponding to user profile \mathbf{x}_j and λ_i is the HMM for party i created (see Section 3). The solution of Eq. 5 is obtained with the aid of Viterbi algorithm as usually happens in HMM classifiers [13].

An HMM is a double stochastic process that models data evolving in time. It is defined by a latent Markov chain, which consists of a finite number of states, and a number of observation probability distributions for each state. At each discrete time instant, the system

switches from one state to another, while an observation is produced by the probability distribution according to the current state [17]. In an HMM, the states are not observable (they are ‘hidden’), but an observation is generated as a probabilistic function of the state, when the system visits the state [13, 18].

An HMM is described by three parameters: $\lambda = (A, B, \pi)$ (see more details in Section 3), which can be estimated based on specialised Expectation Maximisation techniques, such as the Baum-Welch algorithm [19]. These parameters are calculated through several training iterations, by using the entire training data set at each time, until an objective function is maximised. To avoid knowledge corruption, the data should be stored in memory and be trained from start at each iteration; this is costly and time consuming. Therefore, in real life, the datasets for training HMMs are often limited and this can significantly reduce their performance since it heavily depends on the availability of a sufficient quantity of representative training data to calculate the model’s parameters [17].

HMMs have not been used in SVAAAs so far; this is probably due to the fact that within a VAA the observations corresponding to user’s answer choices are not time dependent. However, as we already mentioned, user answer choices can be considered as a sequence of *correlated* observations while the answer options (‘Completely disagree’, ‘Disagree’, ‘Neither agree nor disagree’, ‘Agree’, ‘Completely agree’) can be used as the HMM states. Under these circumstances the HMMs can be applied to VAA, as we have a sufficient number of states and a fairly rich set of data.

3 Methodology

An HMM is characterised by the following ([13, 18]):

- A set of W discrete states $S = S_1, S_2, S_3, \dots, S_W$, with $G = g_1 g_2 \dots g_T$ to be the state sequence (i.e., if we have $g_t = S_i$ that means at time t the system is in state S_i).
- A set of E observations $V = v_1, v_2, v_3, \dots, v_E$, with $O = O_1 O_2 \dots O_T$ to be the sequence of observations corresponding to states G .
- A state transition matrix A , that shows the probability of going from state S_i to state S_j : $A \equiv [a_{ij}]$ where $a_{ij} \equiv P(g_{t+1} = S_j | g_t = S_i)$.
- An observation emission matrix B , that describes the probability of observing v_e in state S_j : $B \equiv [b_j(e)]$ where $b_j(e) \equiv P(O_t = v_e | g_t = S_j)$.
- The probability distribution of being in the first state of a sequence: $\pi \equiv [\pi_i]$ where $\pi_i \equiv P(g_1 = S_i)$.

In our implementation we consider three states of the HMMs, i.e., $W = 3$, $S = \{S_1, S_2, S_3\}$ labeled as S_1 : ‘Negative’, S_2 : ‘Neutral’ and S_3 : ‘Positive’ corresponding to answer choices S_1 : (Completely disagree, Disagree), S_2 : (Neither agree nor disagree, I have no opinion), and S_3 : (Agree, Completely agree) that could be given in the U policy statements of the VAA questionnaire. We chose grouping the answer choices, since separation between the direction (agree/disagree) and intensity (completely) is often difficult to be done by respondents, who are asked to think along multiple dimensions. In addition VAA users tend to avoid taking ‘extreme’ positions in the Likert scale; as a result differences between ‘Completely Agree’ and ‘Agree’ and ‘Completely Disagree’ and ‘Disagree’ create noise rather than more subtle classification. This phenomenon can lead to measurement contamination [20]. In addition, during initial experimentation we observed that HMMs can be more easily created if only three states are considered, in terms of efficiency (training time) and effectiveness (performance).

Every state sequence G has length equal to the number of policy statements, i.e., $T = U = 30$ while the mapping from a user profile \mathbf{x}_j (see also Eq. 1) to an emission sequence $V^j = \{v_1^j, v_2^j, v_3^j, \dots, v_E^j\}$ is obtained as follows:

$$v_q^j = x_{(j,q)} + |L| \cdot (q - 1), \tag{6}$$

where $x_{(j,q)}$ is the answer choice of user j to policy statement q ($q = 1, \dots, E$), L is the set of answer options (see also Section 2) and $|L|$ is its cardinality (number of answer options in the policy statements, in our case $|L| = 6$). For instance if a VAA user selected ‘Completely Disagree’ in the 1st policy statement, then the recorded observation in the 1st place of the sequential answers of the voter would be: $1 + 6 * (1 - 1) = 1$; whereas if the answer choice in the 23rd policy statement was ‘I agree’, then the observation $4 + 6 * (23 - 1) = 136$ would be registered in the 23rd place of the V^j sequence.

As already mentioned an HMM is fully described by three parameters: $\lambda = (A, B, \pi)$. In this work we consider that each party users can be modelled by an HMM λ_i . The way VAA users respond to the first policy statement differs among users, who support different parties, reflecting into different π_i ; the same holds for any other policy statement reflecting in different B_i while the way answer choices are given in two consecutive policy statements also varies among different party supporters reflecting into different A_i .

4 Related work

4.1 Recommender systems in politics

Recommender Systems (RSs) are software tools and techniques, which recommend products or services to be exploited by a user, in an effort to help them decide what they really need from the sheer volume of data that many modern online applications manage. Focusing in the problem of the information overload [21], these systems are widely used in e-commerce [8, 9], where they make proposals for consumers of products to buy, as well as in e-government, e-business, e-library, e-learning, e-tourism, e-resource services and e-group activities [22].

E-government is a way to use the combination of information technology, structural changes and new skills in public administration to improve the quality of public services, reinforce the democratic process and support community objectives [23]. Teran *et al.* [24] used a fuzzy recommender system for e-elections in e-government to inform citizens about candidates and enhance their participation in democratic processes. They introduced the fuzzy clustering analysis, which provides a graphical representation of political parties distributed in clusters, so as to give the opportunity to citizens to examine the behaviour of candidates and find similarities among them.

Dyczkowsk and Stachowiak [25] presented a content-based recommender system of elections that suggest a candidate to a voter, according to the intuitionistic fuzzy (IF)-set theory. They found that IF-set theory can sufficiently model incomplete knowledge about the political positions of a candidate and operate successfully on that information. They also developed a Web application that is intended to be a universal platform for creating recommender systems for elections.

4.2 SVAA methods

Researchers from different research fields deal with many aspects of VAAs [26]. Some of them concern for whether VAAs urge citizens to vote and whether the recommendation

made by these systems affects the final vote decision [3, 4]. Some others show interest in the design of VAAs and especially to party-user similarity estimation methods that can be adopted to predict voting intention [27–30].

Katakis *et al.* [6] noticed that voters often do not agree with the political positions of the party that they intend to vote, but they support it since they are affected by family, friends and community. Therefore, they inspired by community's influence and proposed an alternative matching technique in VAAs that compares the answers sequence of a user (i.e., its profile x_j) to those of other VAA users to find similar ones. Then, recommendation is given based on the distribution of voting intentions of the similar users. Their rationale was that VAAs are, in essence, recommendation tools applied in e-politics scenarios. Thus, use of collaborative filtering approaches could be easily adopted. Based on this conceptualisation they presented the so-called Social VAA (SVAA), which proved to make better voting predictions than the traditional matching schemes between users' and parties' profiles. In their paper they resorted to clustering and classification approaches for generating vote advice in SVAA. They showed that party-supporters modelling based on data mining classifiers and Support Vector Machines, achieve the best performance.

Tsapatsoulis and Mendez [31] dealt with building party models for SVAA based not on voting intention but on the probability to vote each one of parties participating in the German elections in 2013. They compared a Mahalanobis Classifier, a Weighted Mahalanobis Classifier and function approximation approaches (i.e., regression) and they concluded that there is no much gain when using the probability to vote instead of the vote intention. Among the compared methods they noticed that non-linear party modelling techniques, such as neural network based ones, outperform the linear methods like Mahalanobis.

Tsapatsoulis *et al.* [29] in an effort to provide practical design guidelines for SVAA dealt with the problem of finding the minimum number of VAA users required to build effective party models. They limited their analysis to the Mahalanobis Classifier for minimise the factors influencing their research questions. They found that, as the number of parties modelled increases the performance of recommendation decreases, in terms of the Mean Average Precision (MAP) [32] and F-measure [33]. In addition they showed that effective party-supporters models can be built based on a rather small number of user profiles.

4.3 HMM for similar problems

When there is a sufficient number of hidden states with a rich class of observation distributions, the HMMs can accurately represent probability distributions in complex real world problems and create simple and compact models [34]. Thus there are various applications of HMMs in different research areas such as in diverse sequence recognition tasks [13], in the design of automatic speech recognition systems [18, 35], in natural language processing [36], in online character recognition of handwriting [13] and signature verification [37], in bioinformatics [38], as well as in automatic translation tasks [39].

Netzer *et al.* [40] referred to an HMM that relates the latent relationship states to the observed buying behaviour of a customer, investing in customer relationship management. The proposed HMM enables the company to update the customer profile over the time by understanding the evolution of customer relationships through the time and makes it possible to create a long term purchasing behaviour by recognising the marketing activities that are preferred in building customer-brand relationships and by predicting the future choices better than other benchmark models.

Sahoo *et al.* [41] assumed that the user's behaviour changes with time and proposed HMMs to correctly interpret the blog reading behaviour of users and make personalised article recommendations. They found that the proposed method leads to better article recommendation than the existing recommender systems do. Sahebi *et al.* [42] applied HMMs and other clustering algorithms in Web usage-based recommender systems. By comparing their performance, they showed that HMMs outperformed the other algorithms.

Although there is enough evidence about the appropriateness of HMM classifiers for SVAA recommendation, they have not been applied so far, probably because there are simpler machine learning techniques that can be used instead. However, we strongly believe that HMMs have an advantage compared to those methods: they can capture the correlation between answers in different policy statements.

5 Experimental results

5.1 Datasets

In this paper, experiments were conducted to measure the performance of voting prediction, i.e., the accuracy of predicting the vote intention of the users, by applying an HMM classifier to VAA data (filled in questionnaires). Comparisons with the traditional VAA party-user matching method and other party modelling techniques were also done. Three datasets derived from EUVox 2014 and corresponding to three different countries electorate (Cyprus, Germany and Greece) were used in the experiments. EUVox is an EU-wide VAA that was designed for the 2014 elections to the European Parliament. Its questionnaire consists of 30 policy statements and it is based on European-wide issues, issues that are salient for citizens in a particular region, and country-specific issues. All policy statements devised using the same criteria across all cases, which ensure that users can find many issues that are relevant to their daily lives and at the same time the policy statements capture both the supranational and the national dynamic of party competition. The policy statements are clustered into three groups according to the main issue in which they refer: (a) European Union, (b) Economy and (c) Society.

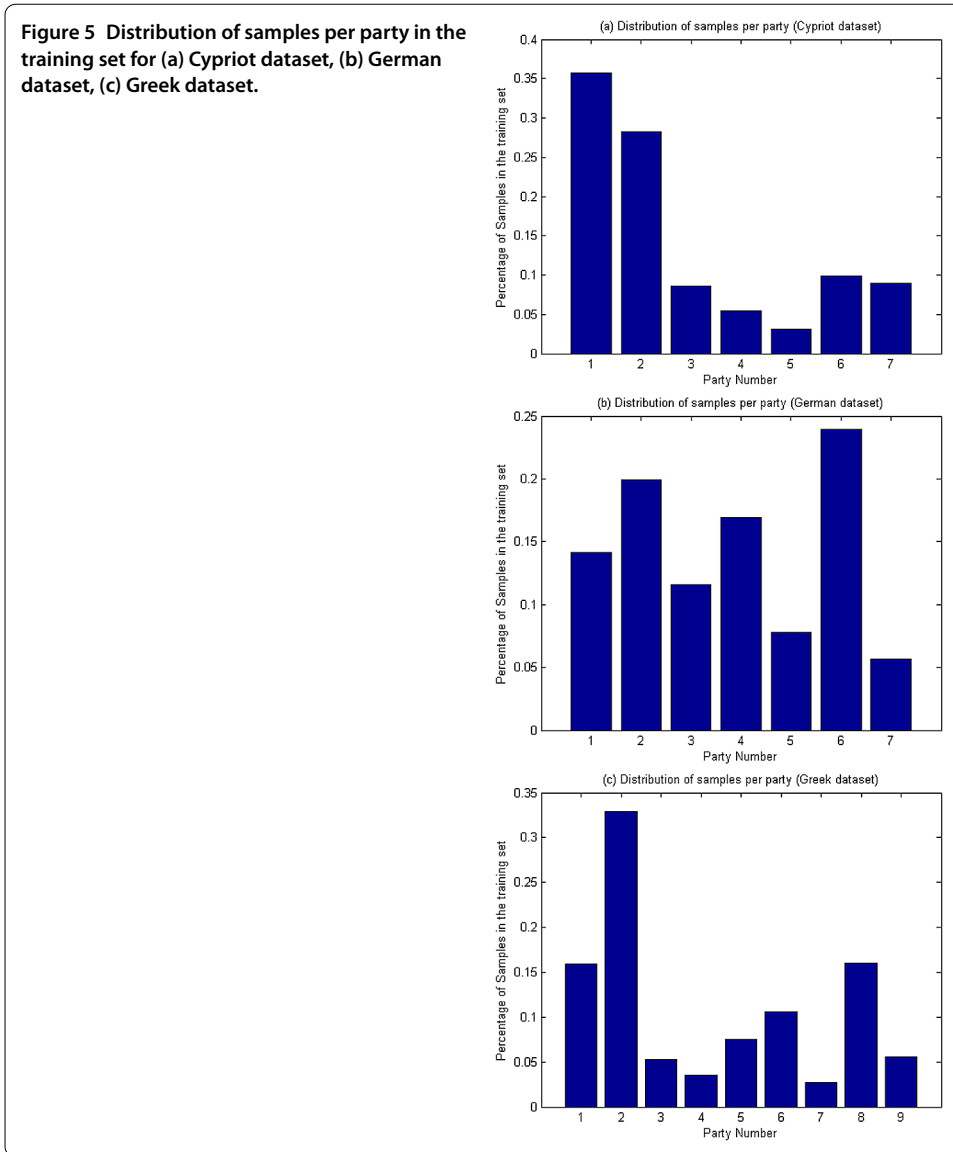
To be able to calculate the performance of the voting prediction, we took into consideration only the users who expressed vote intention for a specific party. Therefore, the questionnaires of the users, who did not answer the supplementary question on voting intention, or answered either 'not decided yet' or 'I will not vote' were exempted. Approximately 40% of the VAA users expressed voting intention for a specific party. The main characteristics of the filtered datasets are summarised in Table 1. For the evaluation, we randomly divided the users in each one of the datasets into a training set and a test set [43] in a 60:40 proportion. Figure 5 presents the distribution of samples per party in the training sets of the three datasets.

As training set $Tr = \{(\mathbf{x}_j, v_{ij}) | j = 1, \dots, N_i, v_{ij} \neq \emptyset\}$ we set the vectors \mathbf{x}_j corresponding to users' answers to the online questionnaire, along with the corresponding vote intention

Table 1 Datasets' characteristics

Dataset	# samples (Questionnaires)	# samples in the training set	# samples in the test set	# parties modelled
Cypriot	1,897	1,138	759	7
German	5,180	3,108	2,072	7
Greek	26,243	15,746	10,497	9

Figure 5 Distribution of samples per party in the training set for (a) Cypriot dataset, (b) German dataset, (c) Greek dataset.



v_{ij} that refers to the party number. After training, the created HMM classifier (i.e., the set of party models) was used to predict the vote intention of the users in the test set $Te = \{(\mathbf{x}_t, v_{i_t}) | (\mathbf{x}_t, v_{i_t}) \notin Tr, t = 1, \dots, N_t, v_{i_t} \neq \emptyset\}$, which is a set of vector and vote intention pairs (\mathbf{x}_t, v_{i_t}) not used in the training set.

The datasets were chosen such as to differ in size. The samples of the Cypriot dataset are few, since less than 2,000 users of the Cypriot dataset filled properly the online questionnaire and expressed at the same time their vote intention. The Greek dataset is approximately 13 times bigger than the Cypriot dataset. The number of samples in German dataset is slightly larger than the number in the Cypriot dataset, but it was preferred because it is characterised by a rather smooth distribution of samples per party, which is not the case in the Greek and Cypriot datasets. Furthermore, the number of parties varies among the selected datasets while the same happens for the population's distribution percentage among the various parties. The mentioned differences helped us to examine the

behaviour of HMMs when there is no sufficient number of data points per party and when the number of samples varies among parties.

5.2 Evaluation framework

To evaluate the voting prediction performance of HMMs, we resort to well-known measures defined in the field of information retrieval [33]. Specifically Precision, Recall, F-measure [44] and Mean Average Precision [33] are computed for all users, who support a particular party and a weighted average is calculated. The Appendix provides a clear definition of these metrics in the context of the current work.

5.3 Results and discussion

Experiments were designed to investigate the performance of social voting recommendation by using HMMs for similarity matching between parties and users in VAAs. The HMMs were trained with the aid of the HMM Toolbox of Matlab, which was built by Kevin Murphy in 1998 and it uses the Baum-Welch algorithm for estimating parameters of HMMs with discrete outputs [45]. We created an HMM $\lambda_i = (A_i, B_i, \pi_i)$, for every party in each one of the datasets. Thus, we concluded with seven HMMs for the Cypriot and German datasets and nine HMMs for the Greek dataset.

The experimental process that was adopted for each one of the datasets involves the following steps: First the parameters of the party models were initialised by random guess. Then the algorithm was updating the parameters iteratively until convergence, by using the training set Tr . After training the party models, the created HMM classifier was applied to the test set Te to classify unseen users into the most probable party class, i.e., if the user's answer pattern was most likely to fit with the i 'th party model, then the user was classified into the party p_i . In the end, to examine the voting prediction performance of HMMs, the real vote intention of each user in the test set was compared to their predicted vote intention (the party id of the party in which they were classified) and an overall score of how well the algorithm performed was calculated. For that the Precision, Recall and F-measures were computed for all users in a particular party and then a total weighted average was estimated. In Tables 2-4 the results for each party of the Cypriot, German and Greek dataset, respectively, are shown.

Additional experiments were also conducted to compare the performance of the HMM classifier with the traditional party-user matching method of VAAs and other classifiers that were applied for similarity matching between parties and users in SVAAs. Tables 5-7 show the total weighted averages for Precision, Recall and F-measure as well as the Mean Average Precision (MAP) by applying various algorithms to each one of the datasets. The traditional VAA method of voting recommendation is referred as 'Party Coding', while the

Table 2 The results of HMMs for each party - Cypriot dataset

Party	Recall	Precision	F-measure
1	0.7569	0.7842	0.7703
2	0.7198	0.9030	0.8011
3	0.2258	0.2029	0.2137
4	0.1538	0.2308	0.1846
5	0.2174	0.0694	0.1053
6	0.2593	0.4468	0.3281
7	0.6780	0.3922	0.4969

Table 3 The results of HMMs for each party - German dataset

Party	Recall	Precision	F-measure
1	0.7801	0.3412	0.4747
2	0.3079	0.5177	0.3861
3	0.4573	0.4438	0.4505
4	0.6227	0.6592	0.6404
5	0.2621	0.7500	0.3885
6	0.7807	0.8380	0.8083
7	0.1736	0.2500	0.2049

Table 4 The results of HMMs for each party - Greek dataset

Party	Recall	Precision	F-measure
1	0.3807	0.6176	0.4711
2	0.5150	0.7434	0.6085
3	0.0799	0.2252	0.1179
4	0.4593	0.1390	0.2134
5	0.4451	0.4057	0.4245
6	0.4935	0.5971	0.5404
7	0.6341	0.0880	0.1546
8	0.0691	0.3542	0.1156
9	0.6612	0.2539	0.3670

Table 5 The aggregate results of party models by applying various algorithms to the Cypriot dataset

Method	Recall	Precision	F-measure	MAP
Party Coding	0.4875	0.6247	0.5746	0.6554
HMM	0.5968	0.6525	0.6234	0.7306
Mahalanobis	0.6100	0.5936	0.6017	0.7384
KNN	0.6166	0.5543	0.5838	0.6992

Table 6 The aggregate results of party models by applying various algorithms to the German dataset

Method	Recall	Precision	F-measure	MAP
Party Coding	0.4913	0.4731	0.4820	0.6788
HMM	0.5458	0.5811	0.5629	0.7203
Mahalanobis	0.5106	0.5341	0.5221	0.6856
KNN	0.4913	0.4995	0.4954	0.6625

Table 7 The aggregate results of party models by applying various algorithms to the Greek dataset

Method	Recall	Precision	F-measure	MAP
Party Coding	0.3574	0.4783	0.4091	0.5627
HMM	0.3966	0.5237	0.4514	0.6099
Mahalanobis	0.3732	0.5178	0.4338	0.5798
KNN	0.4794	0.4545	0.4666	0.6336

KNN refers to k -nearest neighbour classification [44, 46]. The aggregate results of HMMs obtained in the German and Cypriot datasets outperform the ones obtained in the Greek dataset. Also, HMM classifier achieved better overall performance than the other applied methods.

The HMM classifier achieved a very good prediction performance for the Cypriot dataset. Actually it responded extremely well on the two first parties, which concentrate

the major percentage of users (see Figure 5(a)) and their users seem to have consistency on their answer patterns. The low performance in vote prediction for the users of small parties is mainly due to insufficient number of samples (see the results in Table 2). However, even if the two first parties hold the majority of users, these users are less than the samples $N = 20U$ (U is the number of policy statements in the questionnaire) that Tsapatsoulis *et al.* [29] mentioned as the required number for training party models when the Mahalanobis classifier is used. This makes us understand that HMMs can be effectively trained even with few training samples, when these samples form a single cluster in the U -dimensional hyperspace.

The quite smooth performance across parties in the German dataset, as it can be seen in Table 3, occurs due to the smooth distribution of samples per party (see Figure 5(b)) along with the homogeneity of the answer patterns among the users in each party. Even so, the prediction performance for the sixth party, which holds the majority of the users, exceeds the performance of the others. Consequently, the results for the seventh party, which has the smallest distribution of samples in the training set, are the worst.

The vote prediction performance of HMMs for the Greek dataset is controversial (see Table 4) and varies significantly among parties. Once again the HMM for the party with the highest number of users, i.e., the second party, achieved the best score. The non-accurate results for the small parties caused mainly due to insufficient number of samples. Nevertheless, there are cases of parties with fewer samples, such as the fifth and sixth, whose HMMs performed better than parties with more samples such as the first and eighth party. By carefully examining these cases in Table 4 we see that the low number of samples reflects in unbalanced recall and precision scores, which in turn lead to low F-scores, while the poor performance for the other parties is possibly due to non-homogeneity of user profiles, which leads to low scores in both recall and precision. Non-homogeneity within the users in parties occurs for various reasons, such as different political background and different view for the various categories of policy statements. For instance, the supporters of the same party might have a common view on economy but totally different in EU policy issues. As we explain later in the Conclusion section, within party clusters can be investigated separately by modelling data from each specific cluster through a Gaussian distribution and then generating mixture of Gaussians taking into account the ratio of each source [47, 48]. It is known that whenever the distributed data are asymmetric and multi-modal, a mixture of Gaussians can be used to model them [35].

The overall performance of the HMM classifier in predicting vote intention in SVAA is very satisfactory (see the aggregate results of HMMs obtained in each one of the datasets in Tables 5-7). Thus, the use of HMMs, which are based on the conditional probabilities of the VAA users' answers, seems to be quite effective. That was expected since the policy statements in VAA questionnaires are usually correlated and grouped into categories representing specific political issues. Therefore the answer choice to each policy statement can be 'predicted' from previous answer choices. Also the policy statements are answered with a specific display order, from the first to last one, and is kept constant for a specific VAA creating sequences of symbols. The users who support the same party are likely to create similar sequences of choices to policy statements (answer patterns), since they commonly share the same political opinions. So, an HMM classifier, by recognising the given answer patterns of users, who are classified in each party according to their vot-

ing intention, is able to create simple and compact models for each party and make quite well predictions on unseen data. We noticed, however, that imperfect modelling might happen due to insufficient number of samples in the party or because of no or low coherence between the profiles of users, who are classified into the same party. Even so, the non-accurate results for small parties do not critically affect the design of social recommendation, something that was also reported by Tsapatsoulis *et al.* [29].

By applying HMMs to SVAAAs we realised that HMM classifier's performance is closest to that of Mahalanobis classifier's (see the aggregate results in Tables 5-7). However, the HMM classifier achieved better performance than the Mahalanobis classifier and the other machine learning algorithms applied. In almost all cases machine learning techniques outperform the traditional VAA party-user matching method; this is in agreement with conclusions of previous studies (see Agathokleous *et al.* [10], Katakis *et al.* [6], Tsapatsoulis and Mendez [31], Tsapatsoulis *et al.* [29]). Finally, we can see in Table 7 that the k -nearest neighbour classification (KNN) has better Recall, F-measure and MAP scores than the HMM classifier in the Greek dataset. The KNN classifier finds the k -nearest users in the training set, whose answer patterns are nearest to the answer pattern of the user under question. Then it assigns to this user the party id of the party that has the smallest expected misclassification cost among the parties with the k -nearest users [13]. In our case we chose to take into consideration the three nearest users (i.e., $k = 3$) and use the Euclidean distance metric. The good results of KNN classifier reinforce, even more, the remark made previously about the multi-modal distribution of user profiles within the same party in Greek dataset, since it takes into account only the nearest users and not all the users in a party group.

6 Conclusion

This study was conducted in order to investigate whether HMMs could improve the effectiveness of social voting recommendation. We based on the idea that while the users are answering the VAA policy statements, they are incrementally producing sequences of observations (answer patterns) that might characterise 'typical' voters of particular parties; thus, an HMM classifier, whose ability to capture correlations in symbol sequences would be beneficial.

The performance of the HMM classifier in SVAA, according to Recall, Precision and F-measure is quite promising. We observed that, even if the order in which policy statements are displayed in VAAs does not actually matter, the HMMs perform very well in estimating the vote intention of users taking into account the intra-sequence correlations. This is not a surprise as the SVAAAs are based on the party-supporters models and HMM classifier creates simple and compact models by identifying the 'path' that users, in each party, follow to answer the online questionnaire. Also, the policy statements in VAAs are grouped according to the issue category that they represent. The statements that refer on the same subject are correlated and are evaluated in a similar way by users having similar political views. Therefore, what answer is going to be given on a policy statement is dependent on what was observed on a previous one from the same subject category. By finding the conditional probability in which a statement is given according to a statement that has already occurred, the HMMs can effectively provide vote recommendation.

From our experiments we realised that the HMM classifier outperforms the traditional party-coding recommendation method and other party-supporters modelling methods.

In addition we noticed that the prediction performance of HMMs depends on the consistency between the answers of the users in each party and the distribution of samples per party. In general, the parties, with the majority of users achieved the best performance in all three datasets, even in the case of the Cypriot dataset, where the two first parties had the biggest distribution of samples in the training set, but the number of these samples is small. This lead us to the observation that HMMs can be effectively trained even with few training samples, when these samples form a single cluster in the policy statements hyperspace. In cases, where the profiles of party-supporters create a multi-modal clustering in the policy statements hyperspace due to different political backgrounds and different views in the various categories of policy statements, the results tend to be poor. Under these circumstances, the use of mixture of Gaussians [35, 47] or different clustering techniques [46, 49, 50] could be beneficial. In the near future we plan to tackle this problem by using per party and per category of policy statements HMMs. Thus, a combination of HMMs for party-supporters modelling will be pursued to account for the multi-modal distribution of VAA user profiles within the same party.

7 Declarations

List of abbreviations: This list shows the abbreviations in the order they appeared in the text:

VAA Voting Advice Application
 HMM Hidden Markov Model
 ICTs Information and Communications Technologies
 EU European Union
 SVAA Social Voting Advice Application
 RSs Recommender Systems
 IF-set Intuitionistic Fuzzy set
 MAP Mean Average Precision
 KNN k -nearest neighbour

Appendix: Formal definition of evaluation metrics

Let us define X_q as the set of users that expressed a voting intention for party p_q , that is:

$$X_q = \{j | j : v_j = p_q\}. \quad (7)$$

Let us also define as X^v the set of users who expressed a voting intention for any one of the parties (i.e., the set of users that answer the corresponding supplementary question in the VAA):

$$X^v = \bigcup_{q=1:D} X_q. \quad (8)$$

We also define the set G_q as the set of users who expressed a voting intention for party p_q and the social VAA recommendation coincides with their voting intention, and F_q the set of users who expressed a voting intention for a party other than p_q but were recommended

p_q by the social recommendation, i.e.,

$$G_q = \{j|j : v_{ij} = p_q, p_q = \underbrace{\operatorname{argmax}}_i(L(j, i | \mathbf{x}_j))\}, \tag{9}$$

$$F_q = \{j|j : v_{ij} \neq p_q, p_q = \underbrace{\operatorname{argmax}}_i(L(j, i | \mathbf{x}_j))\}. \tag{10}$$

With the aid of the definitions above we can formally define the per party Precision (Pr^q) and Recall (Re^q) measures as follows:

$$Pr^q = \frac{|G_q|}{|G_q| + |F_q|}, \tag{11}$$

$$Re^q = \frac{|G_q|}{|X_q|}, \tag{12}$$

where $|A|$ denotes the cardinality of set A .

The F-measure for a particular party p_q is defined as usual with the aid of Pr^q and Re^q :

$$Fm^q = \frac{2 \cdot Pr^q \cdot Re^q}{Pr^q + Re^q}. \tag{13}$$

The overall Precision (Pr), Recall (Re) and F-measure (Fm) are computed as weighted sums of the per party corresponding quantities:

$$Pr = \frac{1}{|X^v|} \sum_{q=1:D} (|X_q| \cdot Pr^q), \tag{14}$$

$$Re = \frac{1}{|X^v|} \sum_{q=1:D} (|X_q| \cdot Re^q), \tag{15}$$

$$Fm = \frac{1}{|X^v|} \sum_{q=1:D} (|X_q| \cdot Fm^q). \tag{16}$$

Let us consider the social voting recommendation list for user j as an ordered set O_j , which is comprised by the social VAA recommendations in decreasing order:

$$O_j = \{p_{q_1}^j, p_{q_2}^j, \dots, p_{q_D}^j | p_{q_1}^j > p_{q_2}^j > \dots > p_{q_D}^j\}, \tag{17}$$

where $p_{q_1}^j$ is the top SVAA recommendation for user j (i.e., $p_{q_1}^j = \underbrace{\operatorname{argmax}}_i(L(j, i | \mathbf{x}_j))$). Let us also consider as r_j the rank of user's actual voting intention v_j in the set O_j , i.e.:

$$r_j = \underbrace{\operatorname{arg}}_i (v_{ij} = p_{q_i}^j). \tag{18}$$

Every time the actual voting intention of user j appears first in his/her social voting recommendation list O_j , we have the best prediction and $r_j = 1$ (i.e., $v_j = p_{q_1}^j$). However, even if the real voting intention is ranked second or third the user feels rather satisfied. By con-

sidering this notation Average Precision [51] of user j is given by:

$$\text{AvP}(j) = \frac{1}{r_j}. \quad (19)$$

Averaging $\text{AvP}(j)$ across all VAA users j in the test set give us the Mean Average Precision [33]:

$$\text{MAP} = \frac{1}{N_t} \sum_{j=1}^{N_t} \text{AvP}(j), \quad (20)$$

where N_t is the number of users in the test set.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Authors contributed to the manuscript with the order they appear. All authors discussed the experiments and the final results as well as read and approved the final manuscript.

Acknowledgements

The work presented in this paper is an outcome of collaborations established through the framework of the EC project "NOTRE: Network for sOcial compuTing ResEarch" (<http://notre.socialcomputing.eu/>) under the contract H2020-TW INN-2015/692058. The work in this paper was partially funded by the Preference Matcher network (<http://www.preferencematcher.org/>).

Endnote

^a http://www.preferencematcher.org/?page_id=18

Received: 8 June 2016 Accepted: 17 November 2016 Published online: 13 December 2016

References

- Ladner A, Pianzola J (2010) Do Voting Advice Applications have an effect on electoral participation and voter turnout? Evidence from the 2007 Swiss Federal Elections. In: Electronic participation. Springer, Berlin, pp 211-224
- Milakovich ME (2010) The Internet and increased citizen participation in government. *JeDEM* 2(1):1-9
- Fivaz J, Pianzola J, Ladner A (2010) More than toys: a first assessment of Voting Advice Applications' impact on the electoral decision of voters. Technical report 48, National Centre of Competence in Research (NCCR): Challenges to Democracy in the 21st Century
- Ruusuvirta O, Rosema M (2009) Do online vote selectors influence electoral participation and the direction of the vote. In: ECPR general conference, 10-12 September
- Ladner A, Fivaz J, Pianzola J (2012) Voting Advice Applications and party choice: evidence from smartvote users in Switzerland. *Int J Electron Gov* 5(3-4):367-387
- Katakis I, Tsapatsoulis N, Mendez F, Triga V, Djouvas C (2014) Social Voting Advice Applications - definitions, challenges, datasets and evaluation. *IEEE Trans Cybern* 44(7):1039-1052
- Lefevre J, Walgrave S (2014) A perfect match? The impact of statement selection on Voting Advice Applications' ability to match voters and parties. *Elect Stud* 36:252-262
- Ricci F, Rokach L, Shapira B, Kantor PB (2011) Recommender systems handbook. Springer, Berlin
- Jannach D, Zanker M, Felfernig A, Friedrich G (2010) Recommender systems: an introduction. Cambridge University Press, Cambridge
- Agathokleous M, Tsapatsoulis N, Katakis I (2013) On the quantification of missing value impact on Voting Advice Applications. In: Engineering applications of neural networks. Springer, Berlin, pp 496-505
- Katakis I, Tsapatsoulis N, Triga V, Tziouvas C, Mendez F (2012) Clustering online poll data: towards a voting assistance system. In: Proceedings of the 2012 seventh international workshop on semantic and social media adaptation and personalization. SMAP '12. IEEE Comput. Soc., Los Alamitos, pp 54-59
- Djouvas C, Tsapatsoulis N (2014) A view behind the scene: data structures and software architecture of a VAA. In: Semantic and social media adaptation and personalization (SMAP), 2014 9th international workshop on. IEEE Press, New York, pp 136-141
- Alpaydin E (2014) Introduction to machine learning. MIT Press, Cambridge
- Witten IH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques. Kaufmann, Los Altos
- Djouvas C, Gemenis K, Mendez F (2014) Weeding out the rogues: how to identify them and why it matters for VAA-generated datasets. In: Proceedings of the 2014 European consortium for political research general conference. ECPR, Colchester, pp 1-7. <https://ecpr.eu/Filestore/PaperProposal/c8a72d32-cc2d-4102-95df-bb0f6950d423.pdf>
- Tsapatsoulis N, Georgiou O (2012) Investigating the scalability of algorithms, the role of similarity metric and the list of suggested items construction scheme in recommender systems. *Int J Artif Intell Tools* 21(4):12-40

17. Khreich W, Granger E, Miri A, Sabourin R (2012) A survey of techniques for incremental learning of HMM parameters. *Inf Sci* 197:105-130
18. Rabiner LR (1989) A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc IEEE* 77(2):257-286
19. Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann Math Stat* 41(1):164-171
20. Gemenis K (2013) Estimating parties' policy positions through Voting Advice Applications: some methodological considerations. *Acta Politica* 48(3):268-295
21. Walter FE, Battiston S, Schweitzer F (2008) A model of a trust-based recommendation system on a social network. *Auton Agents Multi-Agent Syst* 16(1):57-74
22. Lu J, Wu D, Mao M, Wang W, Zhang G (2015) Recommender system application developments: a survey. *Decis Support Syst* 74:12-32
23. Molnár S (2007) eGovernment in the European Union. European Union, Budapest
24. Terán L, Ladner A, Fivaz J, Gerber S (2012) Using a fuzzy-based cluster algorithm for recommending candidates in e-elections. In: *Fuzzy methods for customer relationship management and marketing*
25. Dyczkowski K, Stachowiak A (2012) A recommender system with uncertainty on the example of political elections. In: *Advances in computational intelligence*. Springer, Berlin, pp 441-449
26. Rosema M, Anderson J, Walgrave S (2014) The design, purpose, and effects of Voting Advice Applications. *Elect Stud* 36:240-243
27. Louwse T, Rosema M (2014) The design effects of Voting Advice Applications: comparing methods of calculating matches. *Acta Politica* 49(3):286-312
28. Mendez F (2014) Modelling proximity and directional logic in VAAs. Paper presented at ECPR, 5-7 September
29. Tsapatsoulis N, Agathokleous M, Djouvas C, Mendez F (2015) On the design of social voting recommendation applications. *Int J Artif Intell Tools* 24(3):1550009. doi:10.1142/s0218213015500098
30. Wagner M, Ruusuvirta O (2012) Matching voters to parties: Voting Advice Applications and models of party choice. *Acta Politica* 47(4):400-422
31. Tsapatsoulis N, Mendez F (2014) Social vote recommendation: building party models using the probability to vote feedback of VAA users. In: *Semantic and social media adaptation and personalization (SMAP), 2014 9th international workshop on*. IEEE Press, New York, pp 124-129
32. Croft WB, Metzler D, Strohman T (2010) *Search engines: information retrieval in practice*. Pearson Education, Upper Saddle River
33. Manning CD, Raghavan P, Schütze H (2008) *Introduction to information retrieval*, vol 1. Cambridge University Press, Cambridge
34. Bilmes JA (2006) What HMMs can do. *IEICE Trans Inf Syst* 89(3):869-891
35. Gales M, Young S (2008) The application of Hidden Markov Models in speech recognition. *Found Trends Signal Process* 1(3):195-304
36. Manning CD, Schütze H (1999) *Foundations of statistical natural language processing*, vol 999. MIT Press, Cambridge
37. Justino EJ, Bortolozzi F, Sabourin R (2005) A comparison of SVM and HMM classifiers in the off-line signature verification. *Pattern Recognit Lett* 26(9):1377-1385
38. Eddy SR (1998) Profile Hidden Markov Models. *Bioinformatics* 14(9):755-763
39. Deng Y, Byrne W (2008) HMM word and phrase alignment for statistical machine translation. *IEEE Trans Audio Speech Lang Process* 16(3):494-507
40. Netzer O, Lattin JM, Srinivasan V (2008) A Hidden Markov Model of customer relationship dynamics. *Mark Sci* 27(2):185-204
41. Sahoo N, Singh PV, Mukhopadhyay T (2012) A Hidden Markov Model for collaborative filtering. *Manag Inf Syst Q* 36(4):1329-1356
42. Sahebi S, Oroumchian F, Khosravi R (2009) Applying and comparing Hidden Markov Model and fuzzy clustering algorithms to web usage data for recommender systems. In: *IADIS European conference on data mining*, pp 179-181
43. Segaran T (2007) *Programming collective intelligence: building smart web 2.0 applications*. O'Reilly Media, Sebastopol
44. Han J, Kamber M, Pei J (2011) *Data mining: concepts and techniques*. Elsevier, Amsterdam
45. Murphy K (1998) Hidden Markov Model (HMM) toolbox for Matlab. <http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html>
46. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv* 31(3):264-323
47. Dasgupta S (1999) Learning mixtures of Gaussians. In: *Foundations of computer science, 1999. 40th annual symposium on*. IEEE Press, New York, pp 634-644
48. Obermaier B, Guger C, Neuper C, Pfurtscheller G (2001) Hidden Markov Models for online classification of single trial EEG data. *Pattern Recognit Lett* 22(12):1299-1309
49. Georgiou O, Tsapatsoulis N (2010) Improving the scalability of recommender systems by clustering using genetic algorithms. In: *Artificial neural networks - ICANN 2010. Lecture notes in computer science*, vol 6352. Springer, Berlin, pp 442-449
50. Berkhin P (2006) A survey of clustering data mining techniques. In: *Grouping multidimensional data*. Springer, Berlin, pp 25-71
51. Yue Y, Finley T, Radlinski F, Joachims T (2007) A support vector method for optimizing average precision. In: *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*. ACM, New York, pp 271-278.