Estimating party-user similarity in Voting Advice Applications using Hidden Markov Models

Marilena Agathokleous
Cyprus Univ. of Technology
30, Arch. Kyprianos str.
CY-3036, Limassol, Cyprus
mi.agathokleous@edu.cut.ac.cy

Nicolas Tsapatsoulis
Cyprus Univ. of Technology
30, Arch. Kyprianos str.
CY-3036, Limassol, Cyprus
nicolas.tsapatsoulis@cut.ac.cy

Constantinos Djouvas
Cyprus Univ. of Technology
30, Arch. Kyprianos str.
CY-3036, Limassol, Cyprus
costas.tziouvas@cut.ac.cy

ABSTRACT

Voting Advice Applications (VAAs) are Web tools that inform citizens about the political stances of parties (and/or candidates) that participate in upcoming elections. The traditional process that they follow is to call the users and the parties to state their position in a set of policy statements, usually grouped into meaningful categories (e.g., external policy, economy, society, etc). Having the aforementioned information, VAA can provide recommendation to users regarding the proximity/distance that a user has to each participating party. A social recommendation approach of VAAs (so-called SVAAs) calculates the closeness between each party's devoted users and the current user and ranks parties according the estimated 'party users' - user similarity. In our paper we stand on this approach and we assume that 'typical' voters of particular parties can be characterized by answer patterns (sequences of choices for all policy statements included in the VAA) and that the answer choice in each policy statement can be 'predicted' from previous answer choices. Thus, we resort to Hidden Markov Models (HMMs), which are proved to be effective machine learning tools for sequential and correlated data. Based on the principles of collaborative filtering we try to model 'party users' using HMMs and then exploit these models to recommend each VAA user the party whose model best fits their answer pattern. For our experiments we use three datasets based on the 2014 elections to the European Parliament[1].

CCS Concepts

•Computing methodologies → Machine learning;

Keywords

Hidden Markov Models; Voting Advice Applications; collaborative filtering; expectation maximization; recommender systems

[1]http://www.euvox2014.eu/

1. INTRODUCTION

Citizens, partly because of their lack of knowledge on the political issues, tend to avoid the democratic decision making process contributing in low voter turnout that affects the most advanced democracies. Ladner and Pianzola [18] specifically mentioned Switzerland, where the voter turnout does not exceed 50% by 1975. E-democracy tools and services can be used to inform people about the political stances of the parties (and/or candidates) who take part in the upcoming elections, aiming at increasing citizen participation and promoting direct involvement in political activities [22]. Voting Advice Applications (VAAs) are specifically designed e-democracy tools that further serve this purpose [17, 26]. They have been applied to facilitate citizens' decision making process by matching their political stances with those of parties and/or candidates. Findings have shown that VAAs' recommendations affect the decision making process of a significant part of voters, especially those who are undecided or belong to specific categories, such as people under 34 years old and/or first time voters [9, 26].

Recommender Systems (RSs) are software tools and techniques, which recommend products or services to users, in an effort to help them decide what they really need from the sheer volume of data that many modern online applications manage [14, 24]. Although the recommender systems are strongly affiliated with the field of e-marketing, several other application areas were also emerged. Recently, several researchers used recommender systems for e-elections in e-government to inform citizens about candidates and enhance their participation in democratic processes [7, 28], while Katakis *et al.* [15] introduced SVAAs (Social Voting Advice Applications), an extended form of VAAs that is based on the principles of collaborative filtering.

VAAs ask users and parties to fill a specific questionnaire that contains a number of policy statements, which are selected according to issues that concern the nation in time of elections and represent important political, economic and social issues [15, 19]. Figure 1 shows an example of such a policy statement along with the set of possible answers a user can select. The recommendation process that a VAA traditionally follows contains two main steps: first, it calculates the similarity scores utilizing the user's and the parties' and/or candidates' answers in the policy statements and then, the VAA ranks the parties according to party-user 'similarity'. Figure 2 presents an example taken from the German VAA of the elections to the European parliament in 2014.

Researchers from different research fields deal with many

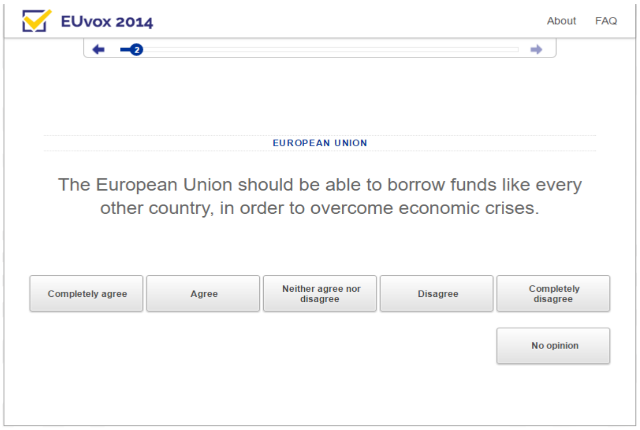

Figure 1: A question that was included in EUVox 2014 along with the given set of answer options.

aspects of VAAs [25]. Some of them investigate whether VAAs urge citizens to vote and whether recommendations made by these systems affect the final vote decision [9, 26]. Other researchers are interested in the design of VAAs dealing with practical issues such as the derivation of optimal party-user similarity estimation methods that accurately predict users' voting intention [20, 21, 29]. We note here that the estimation of similarity between users based on their choices from a set of products is a core problem in Recommender Systems as well.

Recently Katakis *et al.* [15] coined the term 'Social VAAs' (SVAAs) in an effort to describe VAAs, whose recommendation is based on the collaborative filtering philosophy that is widely used in RSs [12, 13]. SVAAs in addition to parties' answers to the policy statements, they also utilize models that capture the behavior - in respect to the policy statements - of each party voters. Thus, a social VAA has the same policy questionnaire with the traditional VAA but also party voters models created by estimating the joint probability of answer patterns and vote intention of each user. Vote intention is an opt in question which is included in VAAs as one of the supplementary questions. An example of supplementary questions included in VAAs is shown in Figure 3, where the vote intention question is the second one.

In SVAAs users are classified into groups according to their voting intention, i.e., party or candidate choice, and then models are created for each party to 'show' the common way, if any, in which party supporters fill the online questionnaire producing their own answer pattern. Then, the SVAA recommends new user with the party or the candidate whose users' model matches better their answer patterns. Figure 4 presents an example of the matching scores presented to a user based on the SVAA philosophy. SVAAs proved to make better voting predictions than the traditional matching schemes between users' and parties' profiles [1]. In addition, as recorded by users' feedback through the emoticons shown in the right part of Figures 2 and 4, SVAA recommendation surpasses VAA recommendation in terms of users satisfaction [6].

In order to tackle the recommendation problem of SVAAs, machine learning techniques [2] can be used to indicate the likelihood that a user belongs into a class, where each class

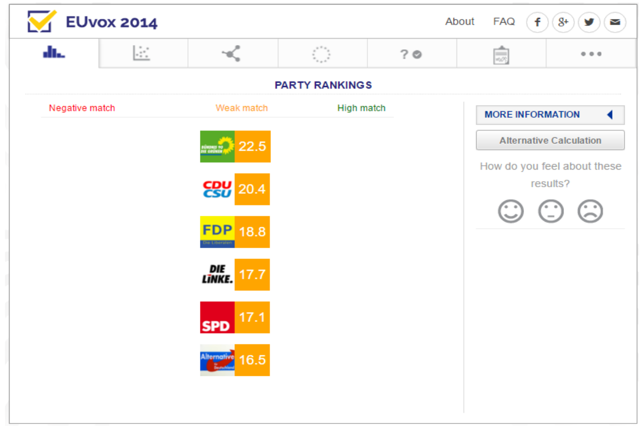

Figure 2: Party ranking based on party-user similarity as computed in traditional VAAs (EUVox 2014, Germany data).

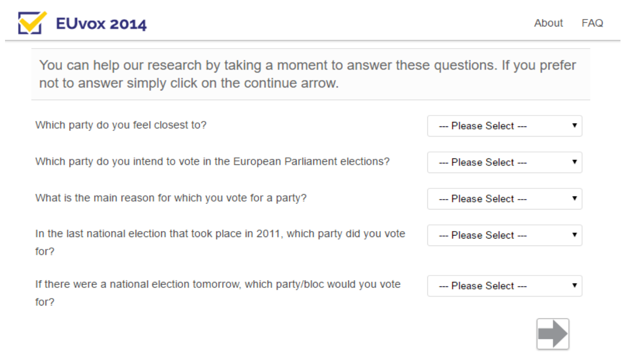

Figure 3: The supplementary questions as they appear in EUVox 2014.

corresponds to a specific party. In essence, what is accomplished with machine learning is to model each party cording to its supporters' answer patterns to policy statements. Thus, if a user is classified into a party, it is more likely this user has the same political positions with people who are already classified to the same party. Katakis *et al.* [15] resorted to clustering and classification approaches for generating vote advice in SVAAs and they showed that party voter modeling based on data mining classifiers and Support Vector Machines, achieve the best performance.

Tsapatsoulis and Mendez [30] dealt with building party voter models for SVAAs based 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, and they concluded that there is no much gain when using the probability to vote instead of the vote intention. They also noticed that non-linear party modeling 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 SVAAs dealt with the problem of finding the minimum number of VAA users required to build effective party's voter models. They limited their analysis

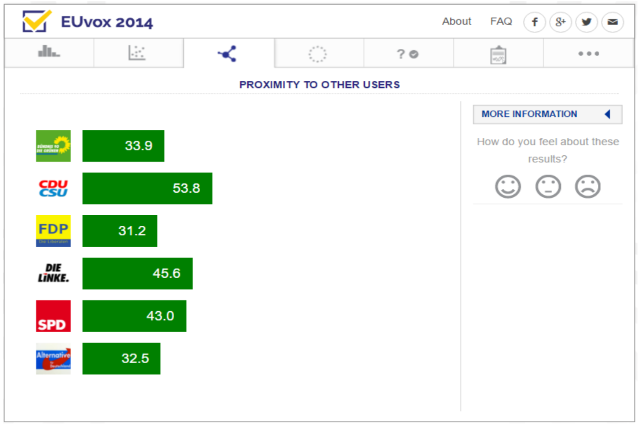

Figure 4: Party ranking based on matching scores between party models and user's answer pattern.

to the Mahalanobis Classifier for minimize the factors influencing their research questions. They found that, as the number of parties modeled is increased the performance of recommendation is decreased. In addition they showed that effective party voter models can be built based on a rather small number of user profiles.

In this work we adopt the social approach of VAAs and 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 [2].

In VAAs the order in which policy statements are displayed to users is not important; however, 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 is 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, that are sequences of choices for all policy statements included in the VAA that characterize 'typical' voters of particular parties can be found, and (b) the answer choice in each policy statement can be 'predicted' from previous answer choices. When users answer the questions, 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 for SVAAs.

We assume that VAA users, who support the same party, produce similar sequences of symbols, i.e., answer patterns. Thus, HMM classifiers can be used to predict and identify the 'path' that users, who support the same party, follow to answer the online questionnaire, and to create simple and compact models for each party, so as to be able to classify new users into the most probable party class. Although there is enough evidence about the appropriateness of HMM classifiers for SVAA recommendation, they have not been applied so far. This is probably due to the fact that there are simpler machine learning techniques that can be used instead. However, we strongly believe that HMMs have an advantage compared to other machine learning methods: they can capture the correlation between answers in different 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 questions 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; in such case datasets used for training should be cleaned using outlier and/or rogue detection techniques [5].

To the best of our knowledge this is the first time HMMs are used to compute party-user similarity either in VAAs or in SVAAs. For our experiments we use three datasets derived from EUVox 2014. EUVox is an online application that was sponsored 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 âĂŞ 2014). Its purpose was to help voters to have quick access to information related to the political positions of the parties participated in the 2014 elections to the European Parliament (see more information at http://www.euvox2014.eu/). The chosen datasets differ in size, in the number of parties participating in the elections and in the population's distribution percentage among the various parties. An important, possible, contribution to researchers belonging to the Recommender Systems community is that the corresponding datasets, as well as many other VAA datasets, are freely available through the Preference Matcher Website[2]. One of the aims of the current work is to mobilize researchers of RSs to investigate the performance of their techniques on VAA data.

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 $X = \{\vec{x_1}, \vec{x_2}, \ldots, \vec{x_N}\}$, a set of U policy statements $Q = \{q_1, q_2, \ldots, q_U\}$, and a set of D political parties or candidates $P = \{\vec{p_1}, \vec{p_2}, \ldots, \vec{p_D}\}$. Each user $\vec{x_j} \in X$ and each political party $\vec{p_i} \in P$, has answered each policy question $q_k \in Q$.

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

$$\vec{x_j} = \{x_{(j,1)}, x_{(j.2)}, \ldots, x_{(j,k)}, \ldots, x_{(j,U)}\} \qquad (1)$$

$$\vec{p_i} = \{p_{(i,1)}, p_{(i,2)}, \ldots, p_{(i,k)}, \ldots, p_{(i,U)}\} \qquad (2)$$

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 question. The vectors $\vec{x_j}$ and $\vec{p_i}$ 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

[2]http://www.preferencematcher.org/?page_id=18

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 is usually replaced with the third point, i.e., with 'neither agree nor disagree'. In this work we decided to keep the sixth point as a distinct emission symbol (see also Section 3) in order to avoid a common criticism by political scientists who strongly argue about the difference between these two categories [3]. As a result the set L, in the context of this study, becomes: $L = \{1,2,3,4,5,6\}$. Figure 1 shows an example of the way the policy statements in the EUVox 2014 appear and how the answer options are presented to VAA users.

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

$$p_q^j = \underbrace{argmax}_{i}(\hat{h}(\vec{x_j}, \vec{p_i})) \tag{3}$$

In many VAAs, the users are asked to answer a number of supplementary questions in addition to the U policy statements. 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 the type of supplementary questions and how they appear in the EUVox 2014 is shown in Figure 3.

The main idea behind the SVAA is to use the vote intention variable y_j and model each party's voters using statistical or machine learning approaches. Thus, for every party i a model $\vec{M_i}$ is created using as training examples the subset $\mathbf{T_i}$ of user profiles who expressed voting intention for party i, that is $\mathbf{T_i} = [\vec{x}_j | y_j{=}i]$. Then, these models can be exploited to provide a recommendation based on collaborative filtering [11] that takes advantage of a VAA's voter community. In this case the top recommendation p_q^j for user j is given by:

$$p_q^j = \underbrace{argmax}_{i}(\hat{h}(\vec{x_j}, \vec{M_i})) \tag{4}$$

In this work we use Hidden Markov Models to create the party-voter models $\vec{M_i}$ (see Section 3). Thus, Eq. 4 becomes:

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

where V^j is the set of observations corresponding to user profile $\vec{x_j}$ and λ_i is the party-voters model for party i created using HMM training. The solution of Eq. 5 is obtained with the aid of Viterbi algorithm as usually happens in HMM classifiers [2].

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 [16]. In an HMM, the states are not observable, i.e., they are 'hidden', but an observation is generated as a probabilistic function of the state, when the system visits the state [2].

An HMM is described by three parameters: $\lambda = (A, B, \pi)$, which can be estimated based on specialized Expectation Maximization (EM) techniques, such as the Viterbi or the Baum-Welch algorithm. The parameters are calculated through several training iterations, by using the entire training data set at each time, until an objective function is maximized. To avoid knowledge corruption, the data should be storage in memory and be trained from the start at each iteration, a costly and time consuming process. Therefore in real life, the datasets used for training HMMs are often small and this might significantly reduce their performance since the effectiveness of HMMs depend heavily on the availability of a sufficient quantity of representative training data to calculate the model parameters [16].

As already stated, in this work we try to optimize SVAA recommendation with the aid of a Hidden Markov Model classifier. This is, probably, the first time the HMMs are used in SVAAs and one of the very few times used in Recommender System applications in general. A possible explanation is the fact that within a VAA, and in many RSs, the observations corresponding to user (answer) choices are not time dependent. However, as we already mentioned, in VAAs user answer choices can be considered as a sequence of *correlated* observations while HMM states could correspond to the set of permissible answer options ('Completely disagree', 'Disagree', 'Neither agree nor disagree', 'Agree', 'Completely agree'). 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 characterized by [2]:

- A set of W discrete states $S = S_1, S_2, S_3, ..., S_W$, with $G = g_1 g_2 ... g_T$ 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 $V = v_1, v_2, v_3, ..., v_E$, with $O = O_1 O_2 ... O_T$ 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 HMMs with three states, 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. Furthermore, there are six possible observations $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, where v_1: 'Completely disagree', v_2: 'Disagree' v_3: 'Neither agree nor disagree', v_4: 'I have no opinion', v_5: 'Agree', and v_6: 'Completely agree'.

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 $\vec{x_j}$ (see also Eq. 1) to an emission sequence $V^j = \{v_1^j, v_2^j, v_3^j, ..., v_E^j\}$ 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 ($q = 1, ..., E$), L is the set of answer options (see also Section 2) and $|L|$ is its cardinality, i.e., the number of answer options in the policy statements. Thus, in our case $|L| = 6$.

As an example consider that a VAA user selected 'Completely Disagree' in the 1st policy statement; then, according to Eq. 6 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.

An HMM is fully described by three parameters: $\lambda = (A, B, \pi)$. In the framework of this work we consider that each party voters can be modeled by an HMM λ_i since the way VAA users respond to the first policy statement differs among supporters of 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. EXPERIMENTAL RESULTS

4.1 Datasets

As in the majority of VAA and SVAA methods, in this work we set the performance criterion to be the accuracy of predicting a user's vote intention. This also aligns with the approach followed in Recommender Systems where the criterion is the accuracy of predicting users' ratings. Thus, we carried out experiments to measure the performance of voting prediction by applying the HMM classifier on three EUVox datasets derived from Denmark, Bulgaria and Czech Republic. EUVox is an EU-wide voting advice application that was utilized during the 2014 European Parliament elections. Its questionnaire consists of 30 policy statements and it is based on European-wide issues, issues that are salient for voters in a particular region, and country-specific issues. The policy statements are clustered into three groups; to those that refer to European Union issues, to those dealing with economy, and to those related to societal issues.

The three datasets were chosen such as to differ in size. The number of samples of the Bulgarian dataset is quite small; approximately 2800 entries were correct and also contained a voting intention answer. The Czech dataset is approximately 5 times larger than the Bulgarian while the Danish dataset is the largest; it contains almost 4 times more samples than the Czech dataset. In addition the number of parties participating in the elections varies among the selected datasets while the same holds for the population distribution among the various parties. The Danish dataset is characterized by a rather smooth distribution of samples per party which is not the case in the Bulgarian and Czech datasets (see Figure 5). These differences helped us to examine the behavior of HMMs when there is no sufficient number of data points per party and when the number of samples varies among parties.

In order to measure the performance of voting prediction using HMMs, we took into consideration only the users who expressed a voting 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. In all three datasets approximately 40% of the VAA users expressed voting intention for a specific party. The main characteristics of the used datasets are summarized in Table 1.

4.2 Results and Discussion

Experiments were designed to investigate the performance of social voting recommendation using HMMs for estimating party-user similarity. For the evaluation we divided the users of dataset into a training and a test set [8]. A HMM is built against the training set $T_r = \{(\vec{x}_j, y_j)|j = 1...N_l, y_j \neq \emptyset\}$ consisting of the profile vectors $\vec{x}_j$ corresponding to user answers to the online questionnaire along with the user's expressed vote intention y_j. Evaluation of the trained HMMs on unseen data was facilitated using the test set $T_e = \{(\vec{x}_t, y_t)|(\vec{x}_t, y_t) \notin T_r, t = 1...N_t, y_t \neq \emptyset\}$ which is a set of profiles and voting intention pairs ($\vec{x}_t$, y_t) not used in the training set.

In order to perform our experiments we resorted to Matlab's HMM toolbox. This toolbox was built by Kevin Murphy and it uses the Baum-Welch (BW) algorithm for estimating the parameters of HMMs with discrete outputs [23]. We created an HMM $\lambda_i = (A_i, B_i, \pi_i)$, for every party included in each one of the datasets. Thus, we ended up with seven HMMs for the Danish and Bulgarian datasets and ten models for the Czech dataset. After training the party models using the training set T_r the test set T_e was used to classify unseen users, expressed through their profiles, into the party in which the user most likely belongs to, i.e., the user's answer pattern most accurately fits i-th party's model. In the end, to examine the voting prediction performance of HMMs, the real voting intention of each user in the testing set was compared to the predicted voting intention, that is the party id of the party in which they were classified. At the end an overall score of how well the algorithm performed was calculated using the Precision, Recall and F-measure scores and then a total weighted average was estimated [29].

In Tables 2-4 we can see the results for each party of the three datasets while Table 5 shows the total weighted averages for Precision, Recall and F-measures, and the Mean Average Precision (MAP) for each dataset. The aggregate results of HMMs obtained in the Danish and Czech datasets are better than the ones obtained in the Bulgarian dataset, but without a marked difference. The HMM classifier achieved a similar overall prediction performance for the Danish and Czech datasets, with the former to be slightly better.

In the Danish dataset the smooth distribution of samples per party (see Figure 5(a)) along with the homogeneity of answer patterns among the supporters of the same party reflects in quite smooth performance across parties as it can be seen in Table 2. However, the prediction performance for the sixth party, which holds the majority of the users, exceeds the performance of the others. The third and the

Table 1: Datasets' characteristics

Dataset	# samples (Questionnaires)	# samples in the training set	# samples in the test set	# parties modeled
Danish	53284	31970	21314	7
Bulgarian	2755	1653	1102	7
Czech	15278	9167	6111	10

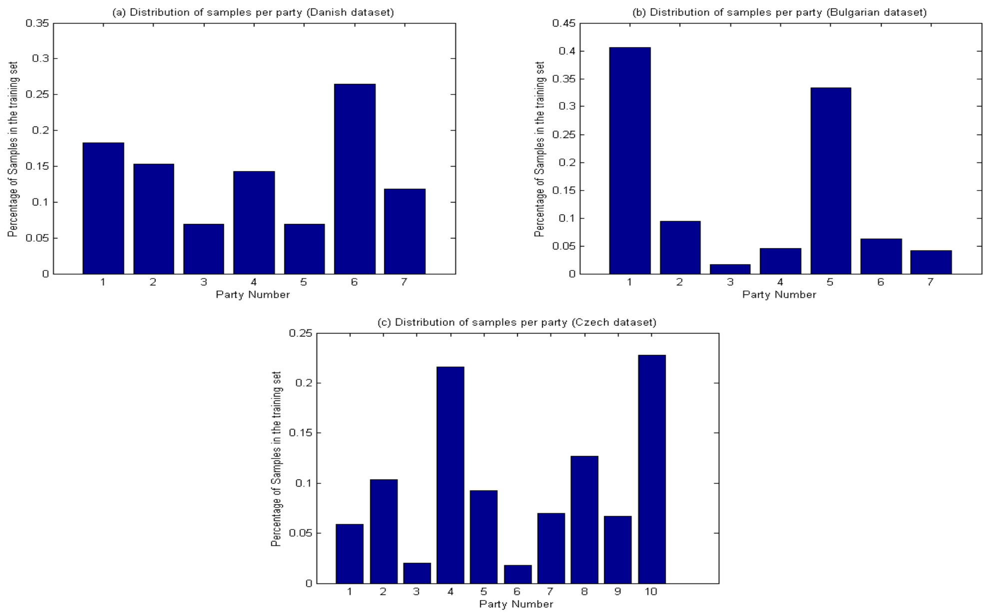

Figure 5: Distribution of samples per party in the training set for (a) Danish dataset, (b) Bulgarian dataset, (c) Czech dataset

Table 2: HMMs performance per party in the Danish dataset

Party Id	Recall	Precision	F-measure
1	0.2915	0.4925	0.3663
2	0.5103	0.4763	0.4927
3	0.7948	0.2014	0.3213
4	0.3955	0.5779	0.4696
5	0.1735	0.6101	0.2702
6	0.6132	0.7837	0.6880
7	0.5419	0.5483	0.5451

fifth parties have the same number of users and the smallest distribution of samples in the training set. Consequently, the HMMs for these parties achieved the worst performance exhibiting high variance between recall and prediction which reflected in low F-score. Even so, the results for the third party were better than the results for the fifth party. This shows that the users in the third party depict higher consistency on their answers and thus the HMM for this party was more effective compared to that of the fifth party.

The vote prediction performance of HMMs for the Czech dataset, shown in Table 3, varies significantly among parties. Once again the HMMs for the parties with the higher number of supporters, i.e., the tenth and fourth (see Figure 5(c)) give the best scores. The relatively low performance in vote prediction for the supporters of small parties is mainly due to insufficient number of samples. However, there are cases

Table 3: HMMs performance per party in the Czech dataset

Party Id	Recall	Precision	F-measure
1	0.4778	0.4687	0.4732
2	0.2597	0.3262	0.2892
3	0.5411	0.2970	0.3835
4	0.6953	0.5115	0.5894
5	0.2192	0.2874	0.2487
6	0.4246	0.1836	0.2563
7	0.2889	0.7027	0.4094
8	0.4941	0.5476	0.5194
9	0.2281	0.4171	0.2949
10	0.6980	0.7183	0.7080

of parties with fewer samples, such as the third and sixth, whose HMMs performed better than parties with more samples such as the second, fifth and ninth party. By carefully examining these cases in Table 3 we see that the low number of samples reflects in unbalanced recall and precision scores, which in turn lead to low F-scores. The poor performance for the other parties is possibly due to the non-homogeneity of user profiles which leads to low scores in both recall and precision. Non-homogeneity within party supporters 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

Table 4: HMMs performance per party in the Bulgarian dataset

Party Id	Recall	Precision	F-measure
1	0.3426	0.5114	0.4103
2	0.2832	0.3478	0.3122
3	0.1316	0.3571	0.1923
4	0.6000	0.4636	0.5231
5	0.5706	0.6884	0.6240
6	0.3409	0.1786	0.2344
7	0.3333	0.0955	0.1485

policy issues. As we explain later in the Conclusion section, within party clusters can be investigated separately by modeling data from each specific cluster through a Gaussian distribution and then generating mixture of Gaussians taking into account the ratio of each source [4, 27]. It is known that whenever the distributed data are asymmetric and multi-modal, a mixture of Gaussians can be used to model them [10].

The results for the Bulgarian dataset, shown in Table 4, are more difficult to interpret. The HMM for the first party, i.e., the one with the majority of supporters, achieved a moderate performance, while the HMM for the fourth party, which was trained on less than 100 samples, presents the second best performance. Once more, the observation made in previous datasets is confirmed: HMMs can be effectively trained even with very small training samples when these samples form a single cluster in the U-dimensional hyperspace, where U is the number of policy statements. Nevertheless, if the number of samples is adequate but there is no or low coherence between the profiles of party-supporters then the results tend to be poor.

The overall performance of HMMs in predicting vote intention in SVAAs is quite satisfactory as it can be seen in Table 5. Thus, the use of HMMs, which make use of the conditional probabilities of the VAA user answers, seems to be working. This was expected since the policy statements in VAA questionnaires are usually correlated and grouped into categories representing specific political issues. Therefore, answers to next policy statement can be 'predicted' from previous answers. 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; people who support the same party are likely to create similar sequences, since they usually share same political opinions. Thus, an HMM classifier, by utilizing answer patterns of users supporting the same party, is able to create simple and compact models that perform quite well in terms of prediction scores.

By applying HMMs to VAAs we realized that HMM classifier performance is closest to Mahalanobis classifier behavior in other VAAs, while it surpasses the performance of other machine learning algorithms, which were applied in the past to model user-party similarities (see Agathokleous *et al.* [1], Katakis *et al.* [15], Tsapatsoulis and Mendez [30], Tsapatsoulis *et al.* [29]). We noticed, however, that imperfect modeling happens either due to insufficient number of samples of a party or because of the inconsistency among users classified to the same party. Nevertheless, the non-accurate results for small parties do not critically affect the design of social recommendation, i.e., the overall vote intention predictions remains high, which is in agreement with the results reported by Tsapatsoulis *et al.* [29].

Table 5: The aggregate results of HMMs

Dataset	Recall	Precision	F-measure	MAP
Danish	0.4747	0.5647	0.5158	0.6772
Bulgarian	0.4174	0.4933	0.4522	0.6246
Czech	0.4934	0.5062	0.4997	0.6683

5. CONCLUSION

In this work we use HMM classifier in order to improve the effectiveness of social voting recommendation feature of VAAs. We based on the idea that while the users are answering the VAA policy statements they are incrementally producing sequences of observations, i.e., answer patterns, that might characterize 'typical' voters of particular parties. Thus, the ability of HMMs to capture correlations in symbol sequences would be beneficial. The performance of the proposed technique was evaluated based on the well known Recall, Precision and F-score metrics. 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 SVAAs are based on party-voters models and HMM classifier creates simple and compact models by utilizing the 'path' that users of the same party create when answering the online questionnaire. Also, the policy statements in VAAs are grouped together according to the issue category that they represent. The statements that refer on the same subject are correlated and are answered similarly by the users. Therefore, answering paths are depended and next answers depend on previous answers of same category. By finding the conditional probability in which a statement is given according to category path already occurred, the HMMs can effectively provide vote recommendation.

From our experiments 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. Parties with the majority of users achieved the best performance in the Danish and Czech datasets. In the case of Bulgarian dataset, the HMM for the party with the highest percentage of samples presented moderate results, while the HMM for the fourth party with very few users (less than 100) achieved the second best performance. This lead us to the observation that in some cases the party-supporters profiles 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). In such cases the use of mixture of Gaussians [10] or different clustering techniques 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 modeling will be pursued to account for the multi-modal distribution of VAA user profiles within the same party.

6. REFERENCES

[1] M. Agathokleous, N. Tsapatsoulis, and I. Katakis. On the quantification of missing value impact on voting

advice applications. In *Engineering Applications of Neural Networks*, pages 496–505. Springer, 2013.

[2] E. Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[3] A. Baka, L. Lia Figgou, and V. Triga. 'neither agree, nor disagree': a critical analysis of the middle answer category in voting advice applications. *International Journal of Electronic Governance*, 5:244–263, 2012.

[4] S. Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 634–644. IEEE, 1999.

[5] C. Djouvas, K. Gemenis, and F. Mendez. 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*, pages 1–7. ECPR, 2014.

[6] C. Djouvas and N. Tsapatsoulis. 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*, pages 136–141. IEEE, 2014.

[7] K. Dyczkowski and A. Stachowiak. A recommender system with uncertainty on the example of political elections. In *Advances in Computational Intelligence*, pages 441–449. Springer, 2012.

[8] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.

[9] J. Fivaz, J. Pianzola, and A. Ladner. 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, 10 2010.

[10] M. Gales and S. Young. The application of hidden markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304, 2008.

[11] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

[12] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[13] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[14] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[15] I. Katakis, N. Tsapatsoulis, F. Mendez, V. Triga, and C. Djouvas. Social voting advice applications – definitions, challenges, datasets and evaluation. *Cybernetics, IEEE Transactions on*, 44(7):1039–1052, 2014.

[16] W. Khreich, E. Granger, A. Miri, and R. Sabourin. A survey of techniques for incremental learning of hmm parameters. *Information Sciences*, 197:105–130, 2012.

[17] A. Ladner, J. Fivaz, and J. Pianzola. Voting advice applications and party choice: evidence from smartvote users in switzerland. *International Journal of Electronic Governance*, 5(3-4):367–387, 2012.

[18] A. Ladner and J. Pianzola. Do voting advice applications have an effect on electoral participation and voter turnout? evidence from the 2007 swiss federal elections. In *Electronic participation*, pages 211–224. Springer, 2010.

[19] J. Lefevere and S. Walgrave. A perfect match? the impact of statement selection on voting advice applications' ability to match voters and parties. *Electoral Studies*, 36:252–262, 2014.

[20] T. Louwerse and M. Rosema. The design effects of voting advice applications: Comparing methods of calculating matches. *Acta politica*, 49(3):286–312, 2014.

[21] F. Mendez. Modelling proximity and directional logic in vaas. *Paper presented at ECPR*, 5:7, 2014.

[22] M. E. Milakovich. The internet and increased citizen participation in government. *JeDEM-eJournal of eDemocracy and Open Government*, 2(1):1–9, 2010.

[23] K. Murphy. Hidden markov model (hmm) toolbox for matlab. *online at http://www. ai. mit. edu/˜ murphyk/Software/HMM/hmm. html*, 1998.

[24] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer, 2011.

[25] M. Rosema, J. Anderson, and S. Walgrave. The design, purpose, and effects of voting advice applications. *Electoral studies*, 36:240–243, 2014.

[26] O. Ruusuvirta and M. Rosema. Do online vote selectors influence electoral participation and the direction of the vote. In *ECPR general conference*, pages 13–12, 2009.

[27] Y. Song, L. Zhang, and C. L. Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web (TWEB)*, 5(4), 2011.

[28] L. Terán, A. Ladner, J. Fivaz, and S. Gerber. Using a fuzzy-based cluster algorithm for recommending candidates in e-elections. *Fuzzy Methods for Customer Relationship Management and Marketing*, 2012.

[29] N. Tsapatsoulis, M. Agathokleous, C. Djouvas, and F. Mendez. On the design of social voting recommendation applications. *International Journal on Artificial Intelligence Tools*, 24(3), 2015.

[30] N. Tsapatsoulis and F. Mendez. 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*, pages 124–129. IEEE, 2014.