Learning and Updating User Models for Subpopulations in Persuasive Argumentation Using Beta Distributions

Emmanuel Hadoux and Anthony Hunter
Department of Computer Science
University College London
London, UK

ABSTRACT

Persuasion is an activity that involves one party (the persuader) trying to induce another party (the persuadee) to believe or do something. It is an important and multifaceted human facility both in professional life (*e.g.*, a doctor persuading a patient to give up smoking) and everyday life (*e.g.*, some friends persuading another to join them in seeing a film). Recently, some proposals in the field of computational models of argument have been made for probabilistic models of what the persuadee knows about, or believes. However, they cannot efficiently model uncertainty on the belief of individuals and cannot represent populations. We propose to use mixtures of beta distributions and apply them on real data gathered by linguists. We show that we can represent the belief and its uncertainty using beta mixtures and that we can predict the evolution of this belief after an argument is given. We also present examples of how to use the mixtures in practice to replace general belief update functions.

KEYWORDS

Persuasive Argumentation; Belief Representation

ACM Reference Format:
Emmanuel Hadoux and Anthony Hunter. 2018. Learning and Updating User Models for Subpopulations in Persuasive Argumentation Using Beta Distributions. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Computational models of argument can potentially be used for systems to persuade users to change their behaviour (*e.g.*, to eat less, to exercise more, to vote) [9]. A *persuader* (the proponent) has a dialogue with a *persuadee* (the opponent or user) to make her believe (or disbelieve) some combination of arguments (the persuasion goal), *e.g.*, to eat healthier food. Building upon Dung's abstract argumentation [3], a dialogue concerns an argument graph G without self-attacks where $\text{Args}(G)$ is the set of arguments in G, and $\text{Attacks}(G)$ is the set of attacks in G.

A *persuasion dialogue* is a sequence of moves. In this work, a move consists in positing an argument or querying the user of her belief in an argument $A \in \text{Args}(G)$. By choosing appropriate arguments to present to the user, the system may raise the user's belief in the persuasion goal. However, for the system, there is a problem of how to communicate with the user and get her arguments. We assume – without loss of generality in this work – that the system cannot understand arguments presented in natural language, given the complexity of processing arguments in free text. Hence, the interface with the user is restricted. One solution is for the system to give a menu of arguments that the user might believe, and the user presents agreement/disagreement in each argument by giving it a score (as in a Likert scale [15]). This score is in the unit interval and denotes the belief the user has in the argument.

Asking the user about which arguments she believes allows for each move in the dialogue to be tailored to the user. However, a user might be asked too many questions, which might cause her to disengage. To address this, we can construct a *user model* that contains information such as the belief that the user has in some of the arguments. The system can then harness the user model to choose its moves.

To represent and reason with belief in arguments, we can use the epistemic approach to probabilistic argumentation [1, 13, 19]. Applying this approach to modelling persuadee's beliefs in arguments has produced methods for updating beliefs during a dialogue [10, 12], for efficient representation and reasoning with the probabilistic user model [7], and for harnessing decision rules for optimizing the choice of arguments based on the user model [8]. In this work we consider that the user is not adversarial, meaning that she is not trying to deceive the system nor is playing strategically. This is the case in, for instance, a patient/doctor dialogue, where the doctor is trying to persuade the patient to stop smoking and the patient is not trying to prevent the doctor from doing so.

These developments offer a well-understood theoretical and computationally viable framework for applications such as behaviour change. However, the user model lacks any quantification of the uncertainty associated with any statement of belief in an argument. Indeed, different users have different numerical representation when asked about their belief; or different reactions to the way the queries are formulated depending, for instance, on their personnality (see, *e.g.*, [14, 18]). In a proposal to address this, a probability distribution over belief distributions was proposed [11], but unfortunately, no methodology was given for how to construct the probability distribution over belief distribution in a way that fairly reflects the uncertainty. To address these shortcomings, we provide a new proposal for constructing user models that is based on beta distributions. These distributions offer a well-established and well-understood approach to quantifying uncertainty. Additionaly, they allow for a principled way of representing subpopulations of a population. This is particularly important for applications in persuasion, where different subpopulations may have quite different beliefs in the arguments in a dialogue, and they may have very different ways of updating in response to specific dialogue

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

moves. In this case, easy-to-get or already gathered data (such as a medical record for instance) can be leveraged to match a new user with a particular subpopulation in order to use a more efficient argumentation strategy.

We proceed as follows: we present a dataset gathering belief values of individuals before and after an argument on a specific topic is given to them; We review the definitions for probabilistic user models; We review beta distributions and provide a methodology for constructing a user model from data; We evaluate our methods on the dataset; We show how a user model using beta mixtures can be updated in response to specific moves in a dialogue; And we finally conclude on the contributions of this paper.

2 PSYCHOLOGICAL DATA

To motivate and evaluate our framework, we use real data on participants, gathered by linguists. We use the data provided by Lukin et al. [16] [1]. It contains the belief on five different topics, for 637 participants (the persuadees), before and after a complex, expert-made, monological argument is given to them. The topics are: "abortion", "climate change", "death penalty", "same sex marriage" and "illegal immigration". In our paper, we refer to the belief before (resp. after) the argument is given as the *initial* (resp. *posterior*) belief. Note that these monologues can be seen as being a single move in a dialogue: the posit of an argument and the query of the belief. Additionaly, usual dialogue moves are often less complex arguments, making the monologues a more interesting choice for the sake of the analysis of the belief. Traditional dialogues are only a sequence of single moves. Their particularities will be discussed in Section 7.

The data also contains the value for each dimension of the OCEAN model: "Openness to experience", "Conscientiousness", "Extraversion", "Agreeableness" and "Neuroticism" [5]. This model is widespread in the psychology literature as a representation of the personality of an individual. The value of each dimension is assessed using the "Ten Item Personality Inventory" (TIPI) [6] prior to any interaction with the arguments. It consists in giving to participants two adjectives associated with each dimension of the OCEAN model. For instance, "extraverted" and "reserved" are associated with "Extraversion" while "anxious" and "calm" are associated with "Neuroticism". The participant is then asked to choose a value on a 7-point Likert scale meaning how much she agrees herself as being, for instance, extraverted. The results are then compiled into a score for each dimension. We refer to these scores as the OCEAN scores.

OCEAN scores set aside, Figure 1 shows the posterior belief given the initial belief on the "abortion" topic. Interestingly, we can see that a significant proportion of the participants do not change their belief. Indeed, 198 out of 637 participants stay in a ±5% band around their original belief. However, we can also see that the pairs (initial, posterior) beliefs are not uniformly spread on the diagonal. Indeed, the more extreme the initial belief is, the less likely it changes in the posterior. Therefore, we will be able to analyse efficiently these data, as we will see later.

In this paper, we only consider belief. Topics such as "abortion" can leverage other factors such as preferences of the participants. However, dealing with preferences is a whole domain by itself, out of the scope of this paper. While the impact of the preferences on the belief is not obvious, they can still be taken into account in the update function (see Section 7).

[1]The dataset is available at https://nlds.soe.ucsc.edu/persuasion_persona

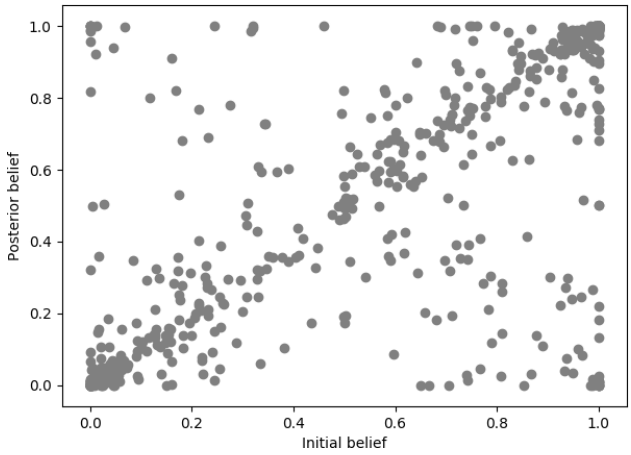

Figure 1: Belief before and after the "abortion" argument for each participant

In addition to gathering the initial and posterior beliefs, an interesting part of Lukin et al. work is about predicting the belief change. Given the initial belief and the individuals' OCEAN scores, they aim at anticipating the effect of the argument on the belief of each individual. They first start by dividing the values into three "bins": low values, medium values and high values, using rigid, sharp bounds for each. They then use a Naive Bayes classifier to predict, given the initial belief and the OCEAN scores, in which bin the posterior belief is, *i.e.*, after the argument has been given. They reach a F1-score of 0.52 with this method.

In the next section we present a new framework to model the belief (initial or posterior) of individuals in a more efficient and comprehensive way than using a sharp value. The shortcomings when using sharp values are also discussed in the following section.

3 PROBABILISTIC USER MODELS

In the general case, when dealing with argumentative dialogues, each odd (resp. even) move in the dialogue is a persuader (resp. persuadee) move. However, the persuadee moves are played with respect to the arguments she believes in, in reaction to the persuader positing an argument. Therefore, an efficient strategy needs to take into account the possible subsets of arguments the persuadee believes in. Indeed, an agent is unlikely to posit arguments she does not have faith in. To that end, the persuader keeps and updates a *belief model* of the persuadee and uses it in her decision process. We use the epistemic approach to probabilistic argumentation [1, 13], defining a model as a *belief distribution* over all possible subsets of believed arguments.

Definition 3.1. A **belief distribution** B over all subsets X of $\text{Args}(G)$ is such that $\sum_{X \subseteq \text{Args}(G)} B(X) = 1$. The **belief in an argument** A is:

$$B(A) = \sum_{X \subseteq Args(G) \text{ s.t. } A \in X} B(X).$$

For a belief distribution B and $A \in \text{Args}(G)$, $B(A)$ is the belief that an agent has in A (*i.e.*, the degree to which the agent believes A is true). When $B(A) > 0.5$, the agent believes the argument to some degree, whereas when $B(A) \leq 0.5$, the agent disbelieves the argument to some degree.

The persuader uses a belief distribution B as a belief model of the persuadee and updates it at each stage of the dialogue. To update a user model during a dialogue, a belief redistribution function takes a belief distribution and returns a revised belief distribution.

In order to do that, we consider the notion of an update method $\sigma(B_{i-1}, D(i)) = B_i$ generating a belief distribution B_i from B_{i-1} based on the move $D(i)$ in dialogue D. If the move is a posit of argument A, the belief in A should be modified. Likewise, if the move is a query, the belief in the argument should be set to the value given by the user.

However, if the update is applied on all possible subsets of arguments, it may lead to a computationaly intractable problem. To address this issue, we can for instance exploit the structure of the argument graph G [7] or define the belief directly on the singleton arguments. We choose the latter in this work.

Having a single belief value for each argument is a computationaly efficient way to model a user. However, we cannot represent uncertainty on the belief. Although this is already addressed if the number of possibilities is finite and discrete [11], this method cannot account for continuous uncertainty. Moreover, different users have a different interpretation of the same numerical value they may give as an answer to a query or give different values depending on the formulation of the query (see, *e.g.*, [14, 18]). We thus need to handle intervals rather than precise values.

Finally, this representation is fitted to a particular user. When dealing with user experiments and real data (as the one present in the previous section), we need to be able to represent populations. Indeed, as discussed earlier, data are easily accessible and are thus available to use. In this case, different personality profiles can be extracted from the data and belief models associated with these profiles. When dealing with a new user, a more efficient initial model can be associated with her, depending on the closest personality profile. It can then be refined to suit this particular user.

4 BETA DISTRIBUTION AND MIXTURE

One of the many interpretations that can be given to a belief b on an argument A is the probability that argument A is believed by the user. In other words, b can be seen as the value of the parameter p of a Bernoulli trial, the probability that the trial yields a successful outcome. In our context, the outcome of this trial gives whether argument A is succesfully believed or not by the user. When $p = 0.5$, the trial represents a coin flip with an unbiaised coin. In terms of belief, it means the user has one chance over two to answer that she believes the argument if asked.

In a real situation, when dealing with actual users, the value of the parameter p cannot be given with certainty. In such a case, we can use beta distributions to handle the uncertainty on p.

Definition 4.1. A **beta distribution** $\mathcal{B}(\alpha, \beta)$ of parameters α and β is a probability distribution on the parameter p of a Bernoulli trial. The parameters α and β constrain the distribution. They can be interpreted as: given $\alpha - 1$ successes and $\beta - 1$ failures over $n = \alpha + \beta - 2$ Bernoulli trials, what is the probability distribution over all possible p.

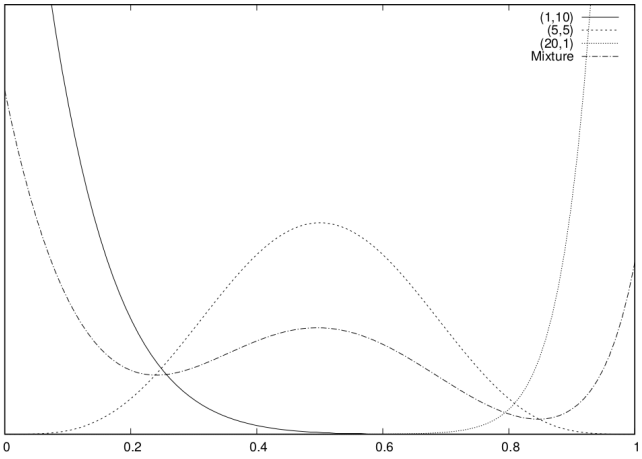

Figure 2: Examples of beta distributions with parameters $(\alpha = 1, \beta = 10), (\alpha = 5, \beta = 5), (\alpha = 20, \beta = 1)$ (zoomed in for visualization)

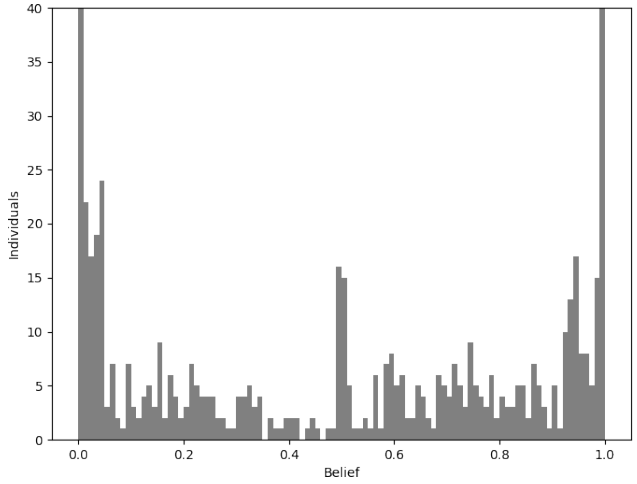

Figure 3: Initial belief of the population on the "abortion" topic

By extension, in this work we use it as a probability distribution on the belief in an argument.

Figure 2 shows three examples of beta distributions with parameters, from left to right, $(\alpha = 1, \beta = 10), (\alpha = 5, \beta = 5), (\alpha = 20, \beta = 1)$.

Using beta distributions allows us to address all the points presented in the previous section. The distribution can handle the uncertainty on the belief (the distribution over the parameter p) whether it comes from the lack of prior knowledge or from the discrepancies in the cognitive evaluation of the belief. They are also meant for representing populations, if we consider each belief as a single Bernoulli trial. Interestingly, we can also use them to detect subpopulations with homogeneous behaviours (*i.e.*, similar belief).

The histogram in Figure 3 shows the initial belief of all the participants before an argument on the "abortion" topic is given. We see that a unimodal distribution (*i.e.*, containing only one "bell") cannot accurately represent the data. Indeed, it is composed of several high values (on the y-axis), separated by low values. For this reason, we use a mixture of unimodal beta distributions in order to create a multimodal distribution. Each unimodal distribution is called a component, and all components are weighted and summed as a linear combination.

Definition 4.2. A **beta mixture** is characterized by a triple $\langle \boldsymbol{\alpha} = (\alpha)_{\{1,\dots,C\}}, \boldsymbol{\beta} = (\beta)_{\{1,\dots,C\}}, \boldsymbol{\pi} = (\pi)_{\{1,\dots,C\}} \rangle$ where C is the number of components, α_c, β_c and π_c are respectively the parameters α, β and the weight associated with component $c \in \{1, \dots, C\}$.

Therefore, a mixture M is calculated as follows:

$$M(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}) = \sum_{c=1}^{C} \pi_c \times \mathcal{B}(\alpha_c, \beta_c).$$

By extension, the probability of a belief $x \in X$ (also called a sample when talking about a value in a dataset) under the mixture $M(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})$ is:

$$M(x; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}) = \sum_{c=1}^{C} \pi_c \times \mathcal{B}(x; \alpha_c, \beta_c)$$

with $\mathcal{B}(x; \alpha_c, \beta_c)$ the function giving the probability that sample x has been drawn from the beta distribution of parameters α_c and β_c.

Figure 2 also presents the mixture of the three components, combined with a weights vector $\pi = [0.4, 0.5, 0.1]$.

5 FITTING A MIXTURE TO DATA

In order to be able to fit a mixture to the data, we need to find, given a number C of components, the parameters α_c and β_c for each component c, as well as the weight π_c to attribute it for the combination. The objective is to find the parameters maximizing the likelihood, *i.e.*, the probability that the data have been generated by a given mixture:

$$l(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} M(x_i; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})$$

with n the number of samples in the dataset X. However, for convenience, we maximize the log-likelihood on the completed data:

$$ll(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \mathbf{z}) = \sum_{i=1}^{n} \sum_{c=1}^{C} z_{ic} \times [\ln \pi_c + \ln \mathcal{B}(x_i; \alpha_c, \beta_c)] \qquad (1)$$

where $z : \{1, \dots, n\} \times \{1, \dots, C\} \to \{0, 1\}$ is a function completing the data such that $z_{ic} = 1$ if sample x_i belongs to component c, 0 otherwise.

To this end, we use an *Expectation-Maximization* (EM) algorithm. EM algorithms are a class of methods operating as follows:

(1) We start with an initial (random or not) assignment of the data to the different components, *i.e.*, one possible function z.
(2) Maximization: we calculate the parameters $\boldsymbol{\alpha'}$ and $\boldsymbol{\beta'}$ maximizing the likelihood of this assignment (Equation 1).
(3) Expectation: we calculate, for each component, the probability for each sample that it belongs to this component. We then calculate the new assignment maximizing this probability for each sample:

$$z'_{ic} = \begin{cases} 1 \text{ iff } c = \underset{c' \in \{1,\dots,C\}}{\arg\max}\ \mathcal{B}(x_i; \alpha'_{c'}, \beta'_{c'}) \\ 0 \text{ otherwise.} \end{cases}$$

The weights vector $\boldsymbol{\pi'}$ is calculated by taking the ratio of samples assigned to each component.
(4) We iterate step 2 and 3 until convergence, *i.e.*, until a stable assignment is reached.

Intuitively, we can see that this method finds a local optimum instead of a global one as it depends on the initial assignment. It needs to be started several times with different initial assignments or use some knowledge on the data to find the best initial set of parameters.

Unfortunately, no analytic form exists for directly maximizing the parameters of beta mixtures (step 2). It means that the maximization has to be performed numerically, which is a computationaly intensive process. However, we can approximate the beta parameters using the *method of moments*. The objective is to approximate them with the parameters of normal distributions that can be maximized easily using existing analytic forms for them. If we define $\mu = \sum_{i=1}^{m} y_i / m$ as the empirical mean of the samples $(y)_{\{1,\dots,m\}}$ in the dataset X assigned to a given component and $V = \sum_{i=1}^{m} y_i^2 / m - \mu^2$ the variance, we can approximate the parameters α and β of this component as follows:

$$\alpha = \mu \left(\frac{\mu(1-\mu)}{V} - 1 \right), \beta = (1-\mu) \left(\frac{\mu(1-\mu)}{V} - 1 \right).$$

Using the method of moments, we can use the analytical formulae for normal distributions and subsequently calculate the beta parameters. This calculation replaces the maximization of Equation 1 in Step 2 of the EM algorithm. This method has been independently created and analyzed by Schröder and Rahmann [17].

The drawback of the method detailed in this section is the necessity to know the number of clusters a priori. One can run the fitting with different numbers of clusters and take the maximum likelihood but the complexity of the model has to be taken into account. Indeed, a perfect mixture would be composed of as many Dirac functions as there are different values in the sample set. We propose to apply the *Normalized Entropy Criterion* (NEC) which is a balance between the likelihood of the mixture and its complexity, *i.e.*, the number of components needed for a given set of samples [2]. While the best mixture is the one fitting the best to the data, the more components it contains, the longer it takes to compute. The NEC criterion accounts for this by penalizing mixtures where the components are overlapping too much and thus forces a small number of components. To use it, the fitting procedure needs to be run with different numbers of components. The best mixture is eventually the one minimizing the NEC.

Definition 5.1. The **NEC** for a mixture with C components is as follows:

$$\mathrm{NEC}(C) = \frac{\mathrm{E}(C)}{\mathrm{L}(C) - \mathrm{L}(1)}$$

with:

$$\mathrm{E}(C) = \sum_{c=1}^{C}\sum_{i=1}^{n} t_{ic} \times \ln t_{ic} \text{ and } t_{ic} = \frac{\pi_c \mathcal{B}(x_i;\alpha_c,\beta_c)}{\sum_{k=1}^{C}\pi_k \mathcal{B}(x_i;\alpha_k,\beta_k)}$$

$$\mathrm{C}(C) = \sum_{c=1}^{C}\sum_{i=1}^{n} t_{ic} \ln(\pi_c \mathcal{B}(x_i;\alpha_c,\beta_c)).$$

Finally, $\mathrm{L}(C) = \mathrm{C}(C) + \mathrm{E}(C)$. Moreover, by convention, $\mathrm{E}(1) = 0$.

Therefore, for each number of components $c_1, \ldots, c_m$ to test, once the mixture is fitted with this number of components, one can calculate the NEC. The best number of components c^* is such that

$$c^* = \underset{c_j \in \{c_1,\ldots,c_m\}}{\arg\min} \mathrm{NEC}(c_j).$$

6 EXPERIMENTS

In this section we present the methodology to fit a mixture to data. We applied it on the monologues provided by Lukin et al. [16]. We show that we can be more precise and flexible in the separation of the values than having low/medium/high bins with rigid, predetermined bounds. We also apply more efficient learning methods to predict the posterior belief intra and inter topics.

As explained in Section 2, the data are monologues, created by experts of the domain. Conveniently, they allow us to study our method on the equivalent of one step of a dialogue, while being substantial enough to trigger a change of belief in only one step. Moreover, dealing with dialogues requires the instantiation of additional functions such as the update method, the reinstatement function, etc. The choice of these methods and the methods themselves require a paper on their own. However, we discuss them and provide some examples in Section 7.

Likewise, the focus of these experiments is more on the evolution of the belief rather than on whether the users have actually been persuaded or not. Indeed, persuasion is a combination of several factors that need to be carefully studied in isolation.

6.1 Fitting a mixture

As a first experiment, we applied our method to fit a mixture to the initial belief distribution on the "abortion" topic. We ran the method with three different numbers of components for the mixture: three, four and five. Figure 4 shows the individual distributions for the components of each mixture. As we can see, the difference between the 3-component and the 4-component mixtures are substantial. The left-most component of the 3-component mixture has a bigger area under the curve (and actually overestimate the probability over this interval) than the one of the 4-component mixture. For the 4 and 5-component mixtures, the four first components are almost identical. However, we can see that the fifth component of the 5-component mixture is almost always 0-valued except at the extreme right of the interval. On one hand, it allows for the fourth component to start later on the belief interval but, on the other hand, it can be seen as a form of overfitting. This mixture will thus be less efficient to represent new data not available during the fitting process.

This visual analysis is corroborated by the calculation of the NEC criterion. As a reminder, we aim at minimizing the value of the criterion calculated for each mixture. In our case, the value of the NEC criterion for the mixture with 3 components is 2.486. It is -2.41 for 4 components and -1.47 for 5 components. Therefore, the best mixture according to the NEC criterion is the one with 4 components: the mixture that is not overfitted (unlike the 5-component) and that is not overestimating the probability (unlike the 3-component). Figure 5 shows the initial belief for the whole population, as well as the 4-component mixture calculated previously. The mixture is the linear combination of the distributions shown in the middle graph of Figure 4, combined with the weight vector learnt during the process.

The parameters for each component are as follows:

(1) $\alpha = 0.452, \beta = 14.986, \pi = 0.283$
(2) $\alpha = 6.886, \beta = 24.417, \pi = 0.106$
(3) $\alpha = 6.962, \beta = 4.674, \pi = 0.207$
(4) $\alpha = 2.866, \beta = 0.254, \pi = 0.404$

We have applied this method to the other topics and found that the best mixtures have respectively 5, 5, 6 and 4 components for the "climate change", "death penalty", "same sex marriage" and "illegal immigration" topics. It means that the number of components is application-dependent and the whole method has to be run for each domain. On the other hand, it also means that the mixture is tightly fitted to the domain (because of the difference in the number of components) and thus represents it more efficiently than a more general method.

6.2 Predicting the belief

Our second experiment consisted in finding the component an individual's posterior belief belongs to, depending on the initial belief and the OCEAN scores of this individual. We used the machine learning models available in the Scikit-learn Python library[2] and compared the F1-score associated with each of these models. We present here the methodology we applied on four types of classifier:

(1) a *gradient tree boosting* classifier (the best performing method in the library on this task),
(2) a *naive Bayes* classifier,
(3) a classifier choosing the most crowded component,
(4) a *status quo* classifier assuming the belief stays in the same component.

The classifiers 3 and 4 are used as baselines.

The high-level idea of the *gradient tree boosting* [4] is to first learn a simple decision tree on the data. It then takes all the samples misclassified by the initial decision tree and learns a new decision tree on these samples. This process is iterated until the desired number of simple classifiers is reached. In our case, we use two classifiers for each class (*i.e.*, for each component in the mixture representing for the posterior belief). Note that the purpose is not to advocate for the gradient tree boosting but rather to show we can achieve good performance using machine learning methods on the data and our mixtures.

For all of the four methods listed previously we use a 10-fold cross validation on the "abortion" data. In other words, the sample set is divided in ten parts, one part is chosen as the test set while learning on the nine others. Therefore, during the learning process

[2] http://scikit-learn.org

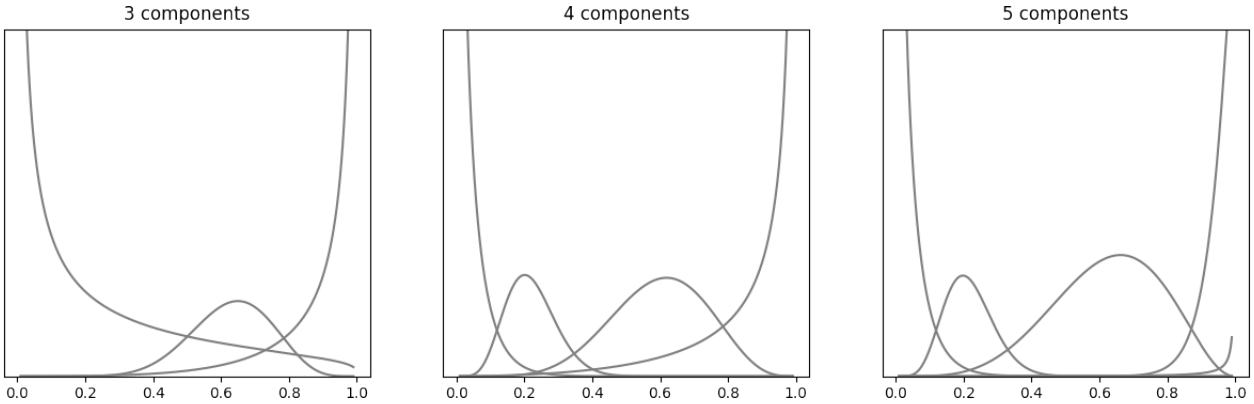

Figure 4: Individual component distributions for different mixtures on the "abortion" data

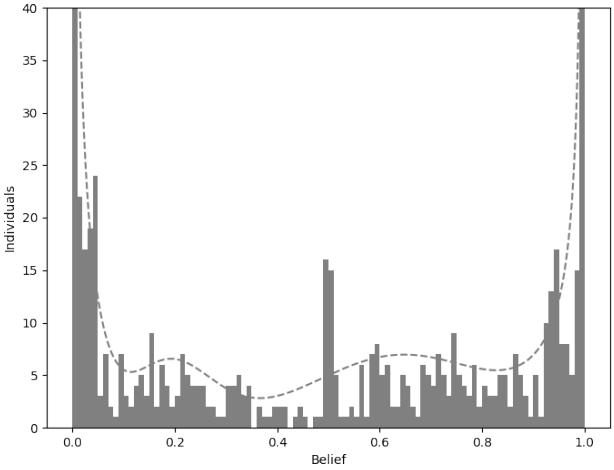

Figure 5: Belief on the "abortion" topic for the whole population before the argument and a 4-component mixture

Method	1	2	3	4
Avg. F1	**0.72**	0.58	0.26	0.64

Table 1: Average F1-score on the 10-fold cross validation for each method

we give the initial belief and the OCEAN scores as input and the component of the posterior belief as expected output for the individuals contained in the nine learning parts. Then, for the test part, we give the initial belief and the OCEAN scores for the individuals contained in the test part and compare the component predicted with the actual component of the posterior belief. This procedure is repeated nine additional times, taking a different part as test set.

In order to compare in a fair way with the method used by Lukin et al., we also ran this procedure with the naive Bayes method (instead of using the results from their paper). Indeed, their methodology and expected output were different. We report the average F1-score on the 10-fold cross validation for each method in Table 1. The difference between the gradient tree boosting and each of the other methods is statistically significant with at least 99.8% confidence using a Student paired t-test on the folds results. It means that using the Gradient boosting method does improve the prediction over the naive Bayes and the baselines. The purpose of the fourth method is to show that even though a significant proportion of the population is not changing its belief, using machine learning performs better than the *status quo* method.

The second part of this experiment consisted in performing a 5-fold cross validation on all five topics in order to learn from four of them to predict the fifth one. Using the gradient tree boosting with four estimators for each component, we achieve an average F1-score of 0.718. This result means that participants behave consistently (but not identically) across topics given their personnality profile and their initial belief.

As a conclusion to these experiments, we have shown that we can fit a mixture to data in order to represent a population. We can also use the mixture in order to learn, intra topic and inter topic, how the belief evolves depending on the initial belief and the psychological features of the individuals composing the population. Using real data such as that provided by Lukin et al., a persuader can use this method to select the best argument to give a particular persuadee or to choose a subpopulation to tackle given a topic. The inter topic learning has also shown that one can have a model learning on a set of arguments and have an insight on the evolution of the beliefs on a new argument before considering to use it in a dialogue.

7 UPDATING THE MIXTURE

Once a mixture is fitted to the data, we can use it in place of the former one-valued belief for an argument. For this, we need to define the allowed moves and the belief update function associated with each of them. In this paper, we define three moves: positing an argument or its negation, reinstating a belief value on a defended argument and querying the belief of an argument. Note that they are just examples of instantiations of existing functions on beta mixtures. Different methods can be defined for different applications.

7.1 Posit

A *posit* consists in asserting an argument or its negation. The belief in this argument is expected to change as a result of the posit, increasingly or decreasingly.

Example 7.1. A refinement function [10] can be adapted to the beta mixtures (and more particularly to their means and variances) as follows:

$$\mu'_c = \mu_c \times (1-k) + k \text{ and } V'_c = V_c \times (1-k), \forall c \in \{1, \ldots, C\}$$

when the argument is positively asserted and:

$$\mu'_c = \mu_c \times (1-k) \text{ and } V'_c = V_c \times (1-k), \forall c \in \{1, \ldots, C\} \quad (2)$$

when it is its negation, where C is the number of components, μ_c (resp. V_c) is the mean (resp. variance) of component c and $k = 0.75$ (resp. 1) in case of the ambivalent (resp. strict) method [10].

In other words, it means that when an argument (resp. its negation) is posited, the mean of each component of the mixture increases (resp. decreases). However, the variance decreases in both cases as we can consider that performing more moves reduces the uncertainty.

Note that, most refinement functions can only be applied if no attacker of the argument being updated is believed, *i.e.*, has a belief bigger than 0.5. This restriction can also be implemented using beta mixtures.

Example 7.2. Let us define two arguments A and C such as C attacks A. The current move to take into account is the posit of A. Therefore, before updating A, we need to check whether C is believed. Let $M(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})$ be the mixture representing the belief on C. We say argument C is believed iff:

$$\int_{x=b}^{1} M(x; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}) > t$$

where b is the threshold for an argument to be considered believed when using sharp beliefs (traditionally 0.5). Parameter t is a threshold defining how conservative is the definition. For instance, $t = 0.8$ means that 80% of the mixture needs to be above b for the argument to be believed. It corresponds to a very conservative definition of the belief. In other words, it can mean that despite the uncertainty, we are 80% sure that the belief if above 0.5. Alternatively, it can also mean that 80% of the population believes argument C with a degree superior to 0.5.

Concretely, using the mixture depicted in Figure 5, the value with $b = 0.5$ is around 0.505. If $t = 0.2 < 0.505$, the argument is considered believed. On the other hand, it is considered disbelieved if $t = 0.7 > 0.505$.

Interestingly, we can also use a classifier using the method presented in the previous section to act as an update function. Indeed, if only the component is needed instead of the actual one-valued belief, when an argument is posited, we can give the current belief and the psychological profile of the persuadee to the classifier in charge of this argument. In realistic scenarii such data can be available. For instance, in a healthcare context, one can use medical data on patient. In a more general application, social media interactions can be used (see, for instance, [20] for a personality assessement through Facebook interactions).

Note that the Gradient Tree Boosting method can also be used as a regressor, meaning that it predicts a precise value instead of a class. Applying it on the "abortion" data using our methodology gives us an R^2 score up to 0.75 (on an interval of $(-\infty, 1]$, where 1 is reached with the perfect prediction for all samples) on some folds.

Therefore, using machine learning is an efficient way to define update functions for the posits of arguments. However, in order to be able to compare the result of the learning with the processes defined in the psuchology literature, the machine learning model needs to be carefully chosen. Indeed, the update function that is learnt needs to be extractable out the model (unlike in neural networks for instance).

7.2 Reinstatement

The *reinstatement* move is in fact an indirect move. Indeed, no agent can choose to trigger a reinstatement on purpose. Rather, it is automatically triggered when a previously posited argument, that has been attacked by a counterargument, is defended by the newly posited argument. In this case, the agent chooses to perform a posit move, and the reinstatement is applied as a subsequent step.

A reinstatement consists in increasing the belief in the defended arguments. However, several choices have to be made before building a reinstatement function, for instance:

- Is the belief going back to the original value?
- Is reinstatement occuring when all attackers have been defeated or partially each time one is?

Example 7.3. The reinstatement function below is applied when all attackers of an argument have been defeated. Therefore, it is triggered just after the posit of the last defending argument.

The function is as follows:

$$\mu'_c = (\mu_c \times (1-k) + k) \times \gamma^{d/2} \text{ and } V'_c = V_c, \forall c \in \{1, \ldots, C\}$$

where C, μ_c, V_c and k are previously defined and γ is a discount factor and d is the distance to the last defender, *i.e.*, the length of the chain in the argument graph. In other words, this function means that the reinstated value is a function of the distance to the last defender.

Example 7.4. As another example, let us consider argument A being attacked by argument C_1. We denote μ^0_c, the mean of component c in the mixture for the original belief. The belief in A is updated using Equation 2 in Example 7.1. In the argument graph, A is also attacked by arguments $C_2, \ldots, C_g$. However, in this application domain, the belief is only decreased with the first attack. Moreover, it is partially increased for each attacker that is defeated, back to its original value. The reinstatement function can be defined as follows for each incremental raise of the belief:

$$\mu'_c = \mu_c + \frac{-k\mu^0_c}{g} \text{ and } V'_c = V_c, \forall c \in \{1, \ldots, C\}.$$

We can see that if it is applied g times, one for each attacker, the mean is reverted back to μ^0_c.

Note that this is a simple example only meant for illustration. It needs to be refined if, for instance, only a subset of the attackers is posited or if reinstatements and attacks can be interleaved.

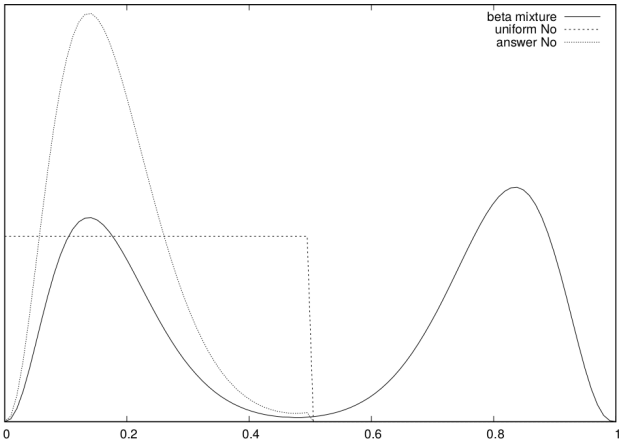

Figure 6: Query with a binary value

7.3 Queries

In order to assess the current belief of the opponent in one particular argument, the proponent can use a special move, a *query*, to ask for this belief. Note that, although a brute forcing strategy such as asking for the belief in all the arguments at each step can give an accurate representation of the opponent, we need to take into account that she can disengage from the dialogue at any point in time. Therefore, we need to keep the dialogues short, efficient and attractive, *i.e.*, not boring nor repetitive.

We can consider two types of query with different expectations on the answers:

(1) Numerical value in [0,1]: "How much do you believe..."
(2) Binary value ({"Yes"/"No"}): "Do you believe..."

7.3.1 Numerical value. When the user answers with a numerical value representing her belief in the argument asked, we can replace the beta mixture by this sharp value. However, as discussed previously, different individuals have different representations of the same value (*e.g.*, [14]). Therefore, we may prefer to transform the mixture in order to have only one component centered on the value given by the user and with a small variance to account for this slight discrepancy. Note that the value of this variance is application-dependent. It depends notably on the amount of risk we are willing to take or the discrepancy we are able to bear in a given context.

7.3.2 Binary value. If the answer given by the user is only binary, we need to transform it back to a specific value or use a more complex method.

In the first case, one possibility is to apply the method for numerical values with pre-chosen ones, for instance, 0.8 when the answer is "Yes", 0.2 otherwise. It means that we fall back into the previous discussion on numerical values, and the need to decide for the "small variance".

Another more interesting possibility is to mix the current mixture with uniform distributions defined on half of the interval. For instance, if the answer is "No", the mixture is uniformly combined with a uniform distribution defined on [0,0.5[. In this case, all belief above or equal to 0.5 in the initial mixture is decreased to a density of 0 and the density for all the belief below 0.5 is doubled. This can be seen as a form of Bayesian updating with a term such as $P(\text{belief} < 0.5|\text{answer} = \text{"No"}) = 1$.

Figure 6 shows an initial two-component mixture and the new distribution if the user answers "No".

8 CONCLUSIONS

In this paper, we have presented a systematic procedure to fit beta mixtures to beliefs using real world data. This contribution allows us to take into account the uncertainty in the belief in a more flexible way. It is useful to consider uncertainty when dealing with human. Indeed, even if we consider that they are not adversarial in this paper, we cannot count on an exact and consistent numerical representation when it comes to beliefs. Depending on the situation, on the way the query is formulated, etc. there might be discrepancies and inaccuracies that we are now able to handle.

We can also represent the belief on a full population and be able to efficiently determine whether an argument is believed or not by this population and what will be its impact on a group of persuadees. We can also extract subpopulations in order to be able to deal with unknown user more efficiently, by matching her with the closest subpopulation and using this as a starting model in the dialogue.

We showed how the mixtures can be used in argumentation to define update functions with various moves: posits, reinstatements and queries. It makes this framework a tool able to replace the former use of sharp values when dealing, for instance, with strategical argumentation (see, *e.g.*, [8]). We have also demonstrated that predictions on the evolution of the beliefs can be made on the mixtures. We can used previously acquired data to learn a prediction model that can be used to predict and give insights on the impact of arguments on a new user. Interestingly, it also works cross topic. It means that, for instance, if we have data on the efficiency of a previous strategy on a given topic, we can have a flavour of what will be the efficiency of the same strategy on another one. While the best machine learning model to use is still to be defined, the results are very promising.

ACKNOWLEDGMENTS

This research was partly funded by EPSRC grant EP/N008294/1 for the Framework for Computational Persuasion project.

REFERENCES

[1] Pietro Baroni, Massimiliano Giacomin, and Paolo Vicig. 2014. On Rationality Conditions for Epistemic Probabilities in Abstract Argumentation. In *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA)*. 121–132.
[2] Gilles Celeux and Gilda Soromenho. 1996. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of classification* 13, 2 (1996), 195–212.
[3] Phan Minh Dung. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and n-person Games. *Artificial Intelligence* 77 (1995), 321–357.
[4] Jerome H. Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.
[5] Lewis R Goldberg. 1990. An alternative "description of personality": the big-five factor structure. *Journal of personality and social psychology* 59, 6 (1990), 1216.
[6] Samuel D. Gosling, Peter J. Rentfrow, and William B. Swann. 2003. A very brief measure of the Big-Five personality domains. *Journal of Research in personality* 37, 6 (2003), 504–528.

[7] Emmanuel Hadoux and Anthony Hunter. 2016. Computationally Viable Handling of Beliefs in Arguments for Persuasion. In *Proceedings of the 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. 319–326.

[8] Emmanuel Hadoux and Anthony Hunter. 2017. Strategic Sequences of Arguments for Persuasion Using Decision Trees. In *Proceedings of AAAI'17*. AAAI Press, 1128–1134.

[9] Anthony Hunter. 2014. Opportunities for Argument-centric Persuasion in Behaviour Change. In *Proceeding of the 14th European Conference on Logics in Artificial Intelligence (JELIA) (LNCS)*, Vol. 8761. Springer, 48–61.

[10] Anthony Hunter. 2015. Modelling the Persuadee in Asymmetric Argumentation Dialogues for Persuasion. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 3055–3061.

[11] Anthony Hunter. 2016. Two Dimensional Uncertainty in Persuadee Modelling in Argumentation. In *Proceedings of 22nd European Conference on Artificial Intelligence (ECAI)*. 150–157.

[12] Anthony Hunter and Nico Potyka. 2017. Updating Probabilistic Epistemic States in Persuasion Dialogues. In *Proceedings of European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (LNCS)*, Vol. 10369. Springer, 46–56.

[13] Anthony Hunter and Matthias Thimm. 2017. Probabilistic Reasoning with Abstract Argumentation Frameworks. *Journal of Artificial Intelligence Research* 59 (2017), 565–611.

[14] Eric J. Johnson, John Hershey, Jacqueline Meszaros, and Howard Kunreuther. 1993. Framing, probability distortions, and insurance decisions. *Journal of risk and uncertainty* 7, 1 (1993), 35–51.

[15] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).

[16] Stephanie Lukin, Pranav Anand, Marilyn Walker, and Steve Whittaker. 2017. Argument Strength is in the Eye of the Beholder: Audience Effects in Persuasion. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* 1 (2017), 742–753.

[17] Christopher Schröder and Sven Rahmann. 2016. A Hybrid Parameter Estimation Algorithm for Beta Mixtures and Applications to Methylation State Classification. In *International Workshop on Algorithms in Bioinformatics*. Springer, 307–319.

[18] Robert S. Siegler and John E. Opfer. 2003. The Development of Numerical Estimation. *Psychological Science* 14, 3 (2003), 237–250.

[19] Matthias Thimm. 2012. A Probabilistic Semantics for Abstract Argumentation. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI) (Frontiers in Artificial Intelligence and Applications)*, Vol. 242. IOS Press, 750–755.

[20] Wu Youyou, Michal Kosinski, and David Stillwell. 2015. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences* 112, 4 (2015), 1036–1040.