Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference

Learning Social Calculus with Genetic Programing

Saad Ahmad Khan[†], **Jonathan Streater**[‡], **Taranjeet Singh Bhatia**[†],
Steve Fiore[‡] and **Ladislau Bölöni**[†]
[†]Department of Electrical Engineering and Computer Science,
[‡]Institute of Simulation and Training
University of Central Florida
4000 Central Florida Blvd, Orlando FL 32816
[†]{skhan, tsbhatia, lboloni}@eecs.ucf.edu, [‡]{jstreate, sfiore}@ist.ucf.edu

Abstract

Physical or simulated agents sharing an environment with humans must evaluate the impact of their own and other agents' actions in the specific social and cultural context. It is desirable that this social calculus aligns itself with the models developed in sociology and psychology - however, it needs to be expressed in an operational, algorithmic form, suitable for implementation.

While we can develop the framework of social calculus based on psychological theories of human behavior, the actual form of the algorithms can only be acquired from the knowledge of the specific culture. In this paper we consider social calculus based on culture-sanctioned social values (CSSMs). A critical component of this model is the set of action-impact functions (AIFs), which describe how the actions of the agents change the CSSMs in specific settings. We describe a technique to evolve the AIFs using genetic programming based on a limited set of data pairs which can be obtained by surveying humans immersed in the specific culture. We describe the proposed model through a scenario involving a group of soldiers and a robot acting on a peacekeeping mission.

Introduction

Achieving appropriate behavior in a social-cultural context is one of the most elusive goals of agent research. There are, however, many practical applications where social behavior is necessary. Agents acting in virtual environments, such as games or training must show a believable social behavior. This can often be achieved with careful scripting. However, when agents control autonomous robots which interact with humans in social settings, the requirements are harder and the interactions more open ended. The agent must have a model through which it can evaluate the impact of specific actions on the participants in the social interaction. There are actions which are physically possible, but socially unacceptable in a given culture. We will use the term *social calculus* for this evaluation process.

The fields of sociology and psychology have a rich literature of describing human behavior in specific cultural contexts. Social calculus, however, requires explicit formulas or algorithms which take as input the observable facts of a situation and specific actions, and provide an output in the form of quantitative metrics. The models developed in humanities are rarely expressed in such quantitative form. In recent years, there is an ongoing effort to *operationalize* models from sociology and psychology (Miller, Wu, and Funk 2008; Miller et al. 2009; Bosse, Jonker, and Treur 2008). Alternatively, we can design new models of reasoning in a social-cultural context, which are informed by the sociological models, but designed from ground up to provide an implementable algorithmic framework.

The objective of this paper is to develop a technique for learning algorithmic components of social calculus based on the input from human observers.

The starting point of our model is the framework of culture-sanctioned social metrics (CSSM) (Khan et al. 2012; Khan, Singh, and Bölöni 2012; Bölöni 2012). We assume that the human behavior proceeds through a series of actions a_i. Actions impact the state of the actor, the target of the action, their peers as well as the perception of the general public. In this model, the state of the agent, relevant to its actions in the social-cultural context is described by a collection of metrics. The metrics can be divided into *tangibles* (such as wealth and time) and the socially constructed CSSMs (such as dignity and politeness). CSSMs are not necessarily independent, but they are not arbitrarily convertible to each other.

The change of CSSMs as a result of an action is described by action-impact functions (AIFs). Let us consider a social metric $M_c(A,t)$ showing the value of the metric at time t for agent A. The action-impact function will give the value of the same metric after an action had been performed $a(A_A, A_T, x_1, x_2, \ldots, x_n)$ where A_A is the actor of the action, A_T is the target of the action, and x_i are the parameters which describe the nature of the execution of the action:

$$M_c(A, t+1) = F(M_c(A,t), a(x_1, x_2, \ldots x_n)) \quad (1)$$

We need expressions for this function for various agents: the actor, the target, but also their peers. We shall also consider virtual agents which represent, for instance, the public opinion of the bystanders.

The reader may note that our analysis is essentially just a rewriting of the traditional way in which an agent can be built. What is new here, is the *CSSM bottleneck* - we assume that the behavior of the agents in a social-cultural context can be fully described by the CSSMs and the tangible values. The utility function can be, of course, a complex and possibly non-linear function of these values, but it does not depend on anything else. What makes our model more useful for social-cultural modeling is that the components of the utility function are clearly mapped to values which make sense in a certain culture.

CSSMs are consistent in a given culture, but they vary between cultures. A given culture assigns a name, a calculation method and a series of behavior rules to these metrics. Agents not immersed in a particular culture would not know about, or would not know how to calculate these values. Even an agent which is immersed in the culture might choose to ignore the rules associated with these values (but it would be aware of the transgression). Finally, an agent might not be able to accurately observe or compute the values (which frequently require a significant cognitive load and accurate observation of the environment). Agents might also make mistakes when planning their actions - especially in cases when they interact with agents which use a different set of values. The latter cases constitute cases of bounded rationality.

In order to apply the model to a given scenario involving one or more cultures, we need to (a) choose a set of CSSMs appropriate to the culture and (b) acquire the action-impact functions for all the feasible action combinations.

Part (a) of the task is clearly a task for a social anthropologist. CSSMs are strongly tied to human cultures: they do not follow mathematical rules, and they cannot be inferred from first principles. The translation of the name into a foreign language or its use in a different cultural context might not transfer the evaluation algorithms and rules of conduct. The term "dignity" has different evaluation methods and rules of conduct in the African-American culture compared other English-speaking cultures. The sociological concept of "face" has three different words in Chinese: *mian*, *lian* and *yan* (Haugh and Hinze 2003). The relatively well established terms of "loosing face" and "saving face" are Chinese lexical borrowings, which entered English in the late 19th century. In other languages, such as Hungarian, these concepts can be explained only through circumlocutions.

For part (b) the task of designing the AIFs, the situation is different. AIFs are multiparameter mathematical functions, we can not directly ask them from human informants. Knowledge engineering these functions for every possible action is a difficult challenge, because the design space is very large. (Bölöni 2012) had modeled the Spanish Steps flower selling scam using the CSSM mechanism. The scenario only has two participants - yet there are 20 different actions, 14 different CSSMs (if we consider self, peer and public perceptions separately). This is already a significant knowledge engineering task. As we are moving to more open-ended scenarios, with a larger number of participants, the number of AIFs and their respective complexity increases at least quadratically. Finding efficient methods to acquire the AIFs is thus a critical step in making the CSSM approach applicable to medium size real world interaction scenarios.

Figure 1: The private P is interacting with vendor V, with the sergeant S and robot R in the background.

In this paper we describe a method which acquires AIFs from a survey of human respondents for specific spot values of the actions. The AIFs will be evolved using a genetic programming mechanism. The objective is to find functions which match the input, provide realistic results when becoming part of the agents and can be expressed in mathematically simple forms. We also hope that the mathematical form of the evolved functions will have explanatory power about the human social-cultural behavior in the given context.

Running example: the market checkpoint scenario

To anchor our modeling work in a plausible real world scenario, we shall use a running example which is a situation frequently encountered in peacekeeping missions. We assume the location to be a Middle Eastern country (although the scenarios would unfold roughly similarly in other parts of the world - with the necessary adaptations for the cultural specifics). The scenario takes place at a military checkpoint at the entrance of a busy market.

The POV of the checkpoint team (sergeant (S), a private (P) and a robot (R)): the efficiency of the checkpoint and their personal security require maintaining a free and uncluttered area around the checkpoint. On days with a high alert level the perceived security is lower, and due to the more thorough inspections the traffic through the checkpoint slows down. The presence and location of the food vendor affects the security risks. Security threats can come from the street vendor itself, from creating additional crowding near the checkpoint, and from blocking lines of sight (either directly, or through the crowding).

The checkpoint team considers desirable to maintain good relations with the local population (in general), and the food vendor (in particular). interaction (informal conversations, exchange of gifts) increase friendship and trust. Unfriendly actions (such as ordering around or threatening) negatively impact the relations.

The POV of the street vendor: it is in the financial interest of the vendor to position his cart close to the checkpoint.

He will try to maintain friendly relations with the members of the checkpoint team, and will remember past interactions with the individual soldiers, appropriately reciprocating friendly or unfriendly behavior. He is aware of factors such as high alarm level (which can mitigate a specific intransigence from the checkpoint team). On the other hand, impolite behavior from a soldier who is considered a friend is perceived more negatively than, for instance, impolite behavior from the robot. The vendor will follow his cultural norms in his behavior - for instance, it is not acceptable to refuse a polite request from a friend.

The choice of CSSMs

Let us now analyse and model out scenario using the CSSM model. We shall use the following collection of metrics:

- **Perceived security level (S, P, R):** is a metric of the level of threat as perceived by the soldiers. It depends on the alarm level, on the level of traffic, and the crowd created by the vendor.
- **Dignity (S, P, V):**. The perception of the personal dignity by the soldiers and the vendor. For the sake of simplicity we shall call both of them dignity, but the two parties apply different evaluation algorithms. The soldiers use a generic Western cultural model adapted to their status as soldiers (being defied on an open order decreases dignity). The seller uses his own cultural model - for the actions of this scenario, for instance involves that being ordered around decreases dignity. Similarly the refusal of an offered gift is an offense to the vendor.
- **Politeness (S,P, V):** The perceived politeness metric is evaluated according to culture specific algorithms by the vendor and the soldiers.

Action repertoire

We model the possible scenarios using a series of possible actions. An action is performed either by a single actor (e.g. the vendor V moving from location 1 to location 3) or is the interaction between an actor and a recipient (the vendor V giving a gift to sergeant S). From the point of view of our model, the actions are fully described by their impact on the values of the actor and (if applicable) the recipient. Our modeling approach here is to define a relatively small number of actions, but to characterize them with detail variables which describe, for instance, the destination of a movement or the verbal style in which a request or command is delivered. These actions are listed in Table 1.

Table 1: Possible actions for the participants in the Market Checkpoint scenario (with specific possibilities for actor and target)

	Action	Actors	Targets	Param.
A1	moves	V		L
A2	declines-to-move	V		
A3	offers-gift	V	S, P	
A4	initiates-conversation	V,S,P	V,S,P	
A5	accepts-conversation	V,S,P		
A6	orders-to-move	S,P,R	V	x,y
A7	passes-order	S,P	P, R	s
A8	accepts-gift	S, P	V	
A9	declines-gift	S, P	V	
A10	pushes	S, P, R	V	x

Case study: detail variables and impact model of action A6

One of the most critical and interesting actions is A6, where the representative of the soldiers (S, P or R) requests the vendor V to move the cart to a farther location. This requests goes against the financial interests of the vendor. What we need to investigate is how this request (and the response to it) affect the values of the participants.

First of all, we need to discuss the detail variables of the action A6. This request can be made at various levels of politeness. To find a numerical metric of the politeness level of a request, we will use the *mitigation level* of the order - according to the classification recently popularized by Malcolm Gladwell (Gladwell 2008)[1]. To the 6 mitigation levels discussed by Gladwell, which culminate in command, we add three more levels which model the threat of and actual physical actions, respectively.

Table 2: The parametrization of action A6 in terms of offensiveness at various levels of mitigated speech (L1-L9)

Name	Example
L1: Hint	Seems like the shade under the tree at position X provides you better shade today
L2: Preference	I think that moving to position X instead of position Y would be a better option to sell goods
L3: Query	Wouldn't position X be a better option to run business today?
L4: Suggestion	You should push the cart to position X before noon, as it would get crowded after two hours.
L5: Obligation statement	You need to move the cart to position X by noon as we would need this space for patrol.
L6: Command	Move to position X!
L7: Threat of physical action	Move to position X or else I'll move you!
L8: Minor physical action	pushing the cart manually away
L9: Major physical action	taking the vendor in custody

Survey-based calibration of the model

Assigning numbers to social values is an inherently inexact science. However, the working assumption is that the culture enforces a more or less uniform method to calculate the sanctioned social values. This means that we can validate (and, if necessary calibrate) the CSSM model by perform-

[1] Note however, that similar ideas are present in the literature for a long time - e.g. in Brown and Levinson's politeness model(Brown and Levinson 1987)

ing a survey in which persons cognizant with the respective culture will judge the impact on the social values.

The datapoints obtained through administering a survey to 91 respondents was to be used as an input into the learning process of the AIFs. Our objective will be that the genetic programming model will evolve functions closely matching those used by the target population when updating their CSSMs.

Representativeness of the survey

One of the important considerations is the representativeness of the survey: are the results of the survey representative of the CSSMs of the target population? It is well known that many academic surveys suffer from the problem of using respondents who are in many ways divergent from the general population and are, in certain ways, “weird” (Henrich, Heine, and Norenzayan 2010).

In the following we will discuss some of the obstacles we perceive in the representativeness of our results.

- The culture of the survey takers (Pakistan) might not be an exact match of the target culture. This is an unavoidable bias - for a perfect localization, one would need to use respondents from the exact geographical location we model.
- There might be a possible misunderstanding between the culture-sanctioned metrics covered by the specific names. Our modeling target was a hypothetical, Arabic speaking Middle-Eastern environment. Our respondents have been primarily Urdu speaking, with a good knowledge of English, and many with at least some level of Arabic.
- The distorting factor of social class: the survey subjects have been drawn from a significantly higher social strata (students, engineers, doctors) than the average composition of the market. It is to be determined whether the social class affects the calculations of CSSMs.

Survey results

While space limits us from analyzing the full output of the survey here, Figure 2 shows a representative case. The figure shows the histogram of answers for the public and peer politeness values for action A(7, 5) - order to move using mitigation level 7 and moderate voice level and A(1, 5) using maximally mitigated speech. The graph shows that there is a remarkable consistency in the estimated CSSM values, but also some level of distribution around mean values.

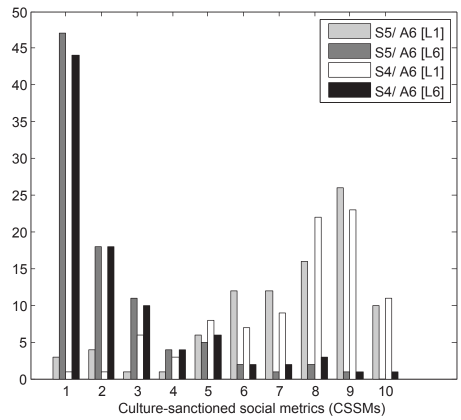

Figure 2: The survey histogram for public politeness [S4] and peer politeness [S5] in view of the vendor when the sergeant performs action [A6] (order to move)

Symbolic regression using genetic programming

Rationality for the use of GP

Finding the AIFs can be seen as a *symbolic regression:* a process through which measured data is fitted with a suitable mathematical formula. Symbolic regression can be performed through manual knowledge engineering. However, there are also several techniques to automatize it, genetic programming being one of the several possibilities. Genetic programming (Koza 1990) is an evolutionary algorithm where the individual units of evolution are programs. Poli et al. (Poli, Langdon, and McPhee 2008) lists a series of circumstances where GP has been found to show good results. Out of these, there are two criteria which strongly applies to the search for AIFs:

- **The interrelationship among the relevant variables is unknown or poorly understood:** this is clearly the case of the various parameters of human interaction. As we have said above, there is no guarantee that CSSMs form an independent set of variables. In fact, there is normally a strong correlation between the self, peer and public CSSMs.
- **Conventional mathematical analysis does not, or cannot, provide analytic solutions:** there is no mathematical theory behind social calculus. What the assumptions behind the CSSM model say is only that different members of the same culture will evaluate the values similarly. We can make only very loose assumptions about the mathematical form of the AIFs - for instance we can infer that they are monotonic in certain variables, or that they are not periodic in certain variables.
- **Significant amounts of test data are available in computer readable form.** In our case, we have a relatively large data set acquired through our survey. Furthermore, the CSSM assumption that any person immersed in a given culture will provide the same evaluation allows for relatively efficient ways to collect data.

Based on these considerations, we conclude that GP is a good choice for the acquisition of AIFs through symbolic regression.

Symbolic regression for AIFs

Function representation: To start a GP evolution, we need to define the functional space over which the evolution will take place. In our previous experiments with manual knowledge engineering of the AIFs, we have found it useful to restrict them to a combination of constants, polynomials and Heaviside step functions connected through arithmetic operations. In addition, many GP algorithms use periodic functions such as sine and cosine due to their favorable mathematical properties. This set, however, creates a too wide set of combinations, making evolution difficult. We can use some of our a priori knowledge about the problem domain to make simplifying assumptions about the format of the AIFs:

- the functions are not periodical, thus there is no logical need to use periodical functions such as sine functions
- there is a natural aspect of human behavior which achieves saturation
- with appropriate parametrization, sigmoid functions can both emulate linear functions and the Heaviside step function.

The function set was chosen to include only basic arithematic functions with a general form sigmoid function **sigmoid (ax - b)**, where all of the three inputs to the function would be evolved using genetic programming.

Fitness function: The evolved functions have been validated by comparing their values with the reference points provided by the survey results. One of the challenges we have encountered was that the survey had been designed to test the CSM assumptions of cultural consistency, not for AIF elicitation. Thus, despite the comparatively large number of respondents, it covered a relatively small set of the AIFs parameter range. To extend this coverage, we have used a cubic spline interpolated surface of the survey results. The fitness function had been defined based on the euclidean distance between the generated AIF and this surface. For future surveys, explicitly designed for AIF elicitation, this interpolation step might not be necessary.

Genetic operators: Finally, genetic operator probabilities, population size, and the number of generations to evolve were chosen experimentally. For our initial phase we used variable genetic operators of crossover and mutation on the population. The values which seemed to guarantee an exploration of the space and diversity in the population while at the same time insuring selection pressure were using low crossover probability and high mutation rate. The formation of new population in this phase was based on the non-elitist approach (even if children are worse individuals than their parents). The best results were generated using the *tournament selection* for generation of new populations. When the change is relatively small then keeping high level to mutations gives better results in genetic algorithms (O'Neill and Ryan 2001).

One of the problems frequently encountered in genetic programming is *bloat:* the phenomena that the population is gradually taken over by individuals of high complexity (and associated long chromosomes) which offer, at best, minor improvements in fitness. Bloated solutions frequently generalize poorly, and are difficult to interpret by humans. To limit bloat, we have limited the trees to a maximum size of 10. The trees were initially limited to a size of 2 but were allowed to grow only if there was an increase in fitness function.

Results

The workflow described in the previous section had been implemented using the GPLab an open-source toolbox for Matlab (Silva and Almeida 2003).

Table 3: Crossover and mutation probability variation using tournament selection for survival

CrossOver probability	Mutation probability	Fitness	Test fitness
0.05	0.95	82.21	24.73
0.1	0.9	58.07	22.2
0.2	0.8	61.1	22.13
0.25	0.75	67.34	33.28
0.5	0.5	65.04	32.5
0.7	0.3	91	91

In the following we will describe the experimental results for the evolution of the AIF for the dignity CSSM at the action A6. This functions has two parameters, the loudness X1 and the offensiveness X2 (the latter being calibrated with the level of mitigation of the speech).

Figure 3 shows the evolution of the fitness values of the population during the evolution. Using hard limits on the dynamic size of tree not only helped us in minimizing the bloating effect but also we were able to evolve the functions fairly quickly. Evolving the best equations with the optimal parameters took about 80-120 minutes with a population size of 500 individuals and 75 number of generations. The best fitness was 18.85, using the tournament selection procedure for evolving generations.

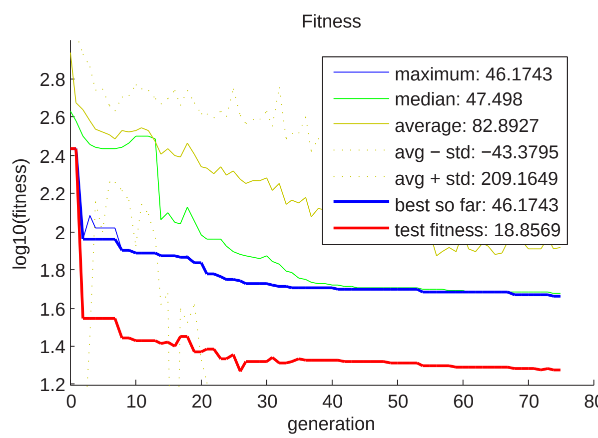

Figure 3: The fitness output for best candidate using tournament selection with variable crossover and mutation probability

The best AIF evolved by the system is shown in Figure 4.

The evolved output matched with our assumptions about the perceived change in AIF with respect to its variables. The sigmoid (flipped around x-axis) contributes to higher levels of AIF when the input variable has low values, which indicates that being polite maintains better dignity. Similarly, we can see that higher levels of x2 (offensiveness) contributes to lower values of dignity.

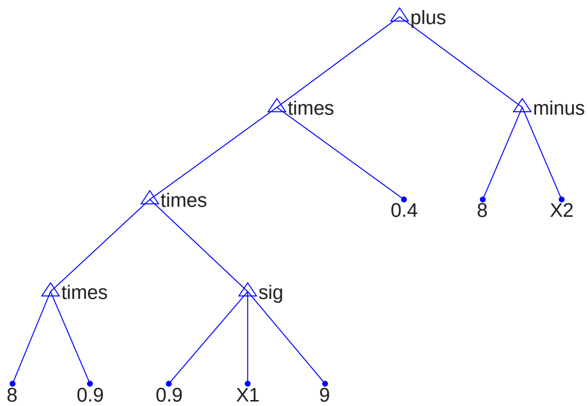

Figure 4: The evolved AIF grammar tree for the CSSM dignity in action A6

Conclusions

In this paper we described an approach for modeling the evolution of cultural values, beliefs and public perceptions for a scenario where peacekeeping soldiers assisted by robots interact with local vendors in a market place. We describe a method which acquires AIFs from a survey of human respondents for specific spot values of the actions. The AIFs were evolved using a genetic programming mechanism. The objective was to find functions which match the input, provide realistic results when becoming part of the agents and can be expressed in mathematically simple forms.

Acknowledgements: The research reported in this document/presentation was performed in connection with Contract Number W911NF-10-2-0016 with the U.S. Army Research Laboratory. The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory, or the U.S. Government unless so designated by other authorized documents.

References

Bölöni, L. 2012. The Spanish Steps flower scam - agent-based modeling of a complex social interaction. In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*.

Bosse, T.; Jonker, C.; and Treur, J. 2008. Formalisation of Damasio's theory of emotion, feeling and core consciousness. *Consciousness and cognition* 17(1):94–113.

Brown, P., and Levinson, S. 1987. *Politeness: Some universals in language usage*, volume 4. Cambridge Univ Pr.

Gladwell, M. 2008. *Outliers: The story of success*. Little, Brown and Company.

Haugh, M., and Hinze, C. 2003. A metalinguistic approach to deconstructing the concepts of "face" and "politeness" in Chinese, English and Japanese. *Journal of pragmatics* 35(10-11):1581–1611.

Henrich, J.; Heine, S.; and Norenzayan, A. 2010. The weirdest people in the world. *Behavioral and Brain Sciences* 33(2-3):61–83.

Khan, S. A.; Singh, T.; Parker, S.; and Bölöni, L. 2012. Modeling human-robot interaction for a market patrol task. In *Proc. of the 25th International FLAIRS Conference*.

Khan, S. A.; Singh, T.; and Bölöni, L. 2012. Soldiers, robots and local population - modeling cross-cultural values in a peacekeeping scenario. In *21th Behavior Representation in Modeling & Simulation (BRIMS) Conference*.

Koza, J. 1990. *Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*. Stanford University, Department of Computer Science.

Miller, C.; Wu, P.; Vakili, V.; Ott, T.; and Smith, K. 2009. Culture, politeness and directive compliance. In *Proc. of the 5th International Conference on Universal Access in Human-Computer Interaction*, 568–577.

Miller, C.; Wu, P.; and Funk, H. 2008. A computational approach to etiquette: Operationalizing Brown and Levinson's politeness model. *IEEE Intelligent Systems* 28–35.

O'Neill, M., and Ryan, C. 2001. Grammatical evolution. *Evolutionary Computation, IEEE Transactions on* 5(4):349–358.

Poli, R.; Langdon, W.; and McPhee, N. 2008. *A field guide to genetic programming*. Lulu Enterprises Uk Ltd.

Silva, S., and Almeida, J. 2003. GPLAB - a genetic programming toolbox for MATLAB. In *Proceedings of the Nordic MATLAB conference*, 273–278.