Crowdsourcing Formal Decision Making Using Generalized Semantic Games

Ahmed Abdelmeged and **Karl Lieberherr** and **Yizhou Sun**
Northeastern University

Abstract

We are after a Wikipedia for formal scientific knowledge; a crowdsourcing system where the crowd decides whether formal science claims hold. Formal science claims (a.k.a. claims) are expressed as logical statements interpreted in a "rich" computable structure where several predicates and functions are implemented in a programming language that is more expressive than the logic used for describing the claims. The purpose is to bring formal science knowledge to an active, objectively refutable form on the web. We call our system the Scientific Community Game (SCG).

Our approach is to use a class of logical game called the Semantic Games (SGs, a.k.a. quantifier games, Tarski games or Hintikka games) to ensure that user decisions are well justified. Users must justify their decisions by winning an SG.

We describe several configurations for our system but more importantly we describe a novel approach, based on synthetic users, to assess different configurations of our crowdsourcing system.

Introduction

We are after a Wikipedia for formal scientific knowledge; a crowdsourcing system where the crowd decides whether formal science claims hold. Formal science claims (a.k.a. claims) are expressed as logical statements interpreted in a "rich" computable structure where several predicates and functions are implemented in a programming language that is more expressive than the logic used for describing the claims. The purpose is to bring formal science knowledge to an active, objectively refutable form on the web. We call our system the Scientific Community Game (SCG).

The problem of deciding formal science claims is fundamentally different from typical problems solved through crowdsourcing such as image labeling (von Ahn and Dabbish 2004) and web page classifiers (Ipeirotis et al. 2010) which are only informally specified. Humans partially specify *what* the problem is while they are solving it. In our case, humans are needed to only provide *how* to solve the problem. Another example of formally specified problems that are solved through crowdsourcing is protein folding (Cooper et al. 2010a). Protein folding can be expressed as the formal science claim $optimal(folding) := \forall folding2 : better(folding2, folding)$, where $better$ is a predicate implemented through a computer simulation of the natural phenomena of protein folding. Natural sciences such as physics and biology are becoming increasingly about simulation models (Johnson 2001).

In addition to providing their solutions to decision problems (either $true$ or $false$), we require users to *justify* their decisions. For the protein folding example, a $false$ decision can be justified by providing a counterexample, a protein folding that is better than the folding claimed to be optimal. But, what justification can be provided for a $true$ decision? a proof based on how $better$ is implemented can be provided. However, proofs are normally beyond the abilities of average individuals in the crowd. It is also possible to take the failure of other users, arriving the $false$ decision, to provide a counterexample as a justification for the truth of the underlying claim. This later approach has a long tradition in logic and has been extensively studied.

Our approach is to use a class of logical game called the Semantic Games (SGs, a.k.a. quantifier games, Tarski games or Hintikka games) to ensure that user decisions are well justified. Users must justify their decisions by winning an SG.

Semantic Games

Logical games have a long history going back to Socrates. More recently, they became a familiar tool in many branches of logic. Important examples are Semantic Games (SGs) used to define truth, back-and-forth games used to compare structures, and dialogue games to express (and perhaps explain) formal proofs (Marion 2009), (Hodges 2009), (Keiff 2011).

SGs are played between two players, the *verifier* and the *falsifier* [1]. We illustrate SGs through an example here. A more extensive description and precise definition can be found at (Kulas and Hintikka 1983).

Given the formula $\forall x \in [0, 1] \exists y \in [0, 1] : x \cdot y + (1 - x) \cdot (1 - y^2) \geq 0.5$ and the verifier is ver and the falsifier is fal. According to the rules, fal must provide a value for universally quantified variables. Suppose that

[1] Other names have been also used in the literature such as *I* and *Nature*, *Proponent* and *Opponent*, and *Alice* (female) and *Bob* (male).

fal provided 0, then the game proceeds on the formula $\exists y \in [0,1] : (1 - y^2) \geq 0.5$. According to the rules, ver must provide a value for existentially quantified variables. Suppose that ver provided 0, then the game proceeds on the formula $1 \geq 0.5$. According to the rules, this is a true primitive formula and therefore the verifier wins. The rules for and-compounded formulas is that the falsifier chooses one of the subformulas. For or-compunded formulas, the verifier chooses one of the subformulas. For negated formulas, the game proceed on the subformula under the negation but with both players exchanging their roles.

In the theory of SGs, logical statements interpreted in a computable structure (a.k.a. claims) derive their meaning from the games played by the rules prompted by the logical connectives encountered in the claims (Pietarinen 2000). The existence of a winning strategy for the *verifier* implies that the underlying logical statement is indeed *true* and the existence of a winning strategy for the *falsifier* implies that the underlying logical statement is indeed *false*.

Motivating Example

Suppose that we have the following claim family $sp(c) := \forall x \in [0,1] \exists y \in [0,1] : x \cdot y + (1 - x) \cdot (1 - y^2) \geq c$. And we want to enlist a crowd (of 10 users for example) to help us figure out whether the claim $sp(0.6)$ holds or not. Furthermore, we want to identify the individuals that are skilled enough to make the correct decision with appropriate justification.

One starting point is to ask the users for their positions on $sp(0.6)$. What if all of the 10 users took the verifier position, should we take their unjustified decisions and consider $sp(0.6)$ to be true? or should we force positions on users?

Let's say 3 users decided to be verifiers and 7 users decided to be falsifiers. Suppose that we decided that we are not going to force positions on users, then there is 21 different potential SGs to be played. Should all of them be played? or is it enough to play only a portion of the SGs? Should we decide the subset of games to be played? or should we let users choose their opponents?

Suppose that 15 SGs were played. In 10 of them the verifier won and in the remaining 5 the falsifier won. Should we conclude that $sp(0.6)$ is true? What if the falsifier was forced in 6 of the 10 games where the verifier won, should we still conclude that $sp(0.6)$ is true?

How do we fairly assess the skills of users or even rank them? taking into account that some of them might have been at a disadvantage given that only a subset of possible games where played and that they might have been forced more often?

Each of these three groups of questions is answered by a particular component in our system. Questions regarding which SGs to be played is answered by a component named the Crowd Interaction Mechanism (CIM). Questions regarding the truth likelihood of a claim given a history of SGs played on that claim are answered by the Claim Evaluator (CE). Questions regarding the skill of a given user at deciding a particular claim given a history of SGs played at that claim are answered by the User Evaluator (UE).

Below, we describe a fair CIM, two CEs and two UEs but more importantly we describe a novel approach, based on synthetic users, to assess different configurations – a configuration consists of a CIM, a CE and UE – of our crowdsourcing system.

Applications

Computational Problems Our system can be applied to solve, and to develop algorithms to solve, complex computational problems. A task that has been traditionally handled through crowd sourcing competitions such as Harvard Catalyst Competitions (har).

A computational problem can be logically specified as a claim about the relation between either (1) the input properties and the output properties, or (2) the input properties and the output finding process properties such as resource consumption.

Typically, the resulting logical specification is trivially true, however players need to, efficiently, solve the underlying *computational problems* to get the examples and counterexamples they need during the course of SGs. For example, the falsifier of the $prime(7) = \forall k\ s.t.\ 1 < k < 7 : \neg divides(k, 7)$ needs to compute the factors of 7.

Education The collaborative and self-evaluating nature of SGs provides a peer-based evaluation system for MOOCs and on-line courses on formal science topics. The peer-based evaluation is guaranteed to be fair, and it saves significant time for the teaching staff.

Contributions

This paper makes the following contributions: (1) We bring semantic games to crowdsourcing to evaluate users and claims. (2) We not only map logical formulas to semantic games but we also map them to probability formulas which predict outcomes of the semantic games for given skill levels. (3) We contribute the design and analysis of a broad crowdsourcing platform for model checking in a structure. (4) We introduce the novel concept of synthetic scholars with a given skill level to evaluate design decisions before the system is used with humans.

Proposed System

We describe a *fair* CIMs, two alternative CEs and two alternative UEs.

Fair CIMs

The CIM can put a user at a disadvantage by enrolling it in either more or fewer games than average (ISSUE1), enrolling it against more or fewer strong users than average (ISSUE2), enrolling it against more forced users than average (ISSUE3) or forcing it to take a particular position more often than average(ISSUE4).

ISSUE1 can be avoided by enrolling users in a round robin tournament. One approach to get users holding the same position to play against each other, is to force a position on one of the players. However, this raises issues 3 and 4. We propose Contradiction Agreement Games (CAGs) as an alternative approach to forcing only one of the players.

Game	forced	winner	payoff (p_1, p_2)
Agreement T1	p_2	p_1	(0, 0)
	p_2	p_2	(0, 1)
Agreement T2	p_1	p_1	(1, 0)
	p_1	p_2	(0, 0)
Contradiction	–	p_1	(1, 0)
	–	p_2	(0, 1)

Table 1: The Contradiction-Agreement Game

The Contradiction-Agreement Game CAGs remove the restriction that scholars must take contradictory positions on claims. In case scholars take contradictory positions, CAG reduces to one SG. Otherwise, CAG reduces to two testing SGs. In a test SG, one of the scholars, the tester, is forced to take the opposite position of the position it chose. The two scholars switch their testing roles between the two games. Even though the tester is forced to take a particular position, CAG-based evaluation remains fair.

SGs with forced scholars can cause unfairness in two different ways:

1. Winning against a forced scholar is not the same as winning against an unforced scholar. Giving both winners a point for winning would be unfair.
2. The forced scholar is at a disadvantage.

To overcome these two problems, we adopt the rule that the scholar winning an SG *scores* a point only if its adversary is not forced. Although, this solves the two problems, it, oddly enough, puts the winner at a disadvantage because it has no chance of *scoring* a point. Luckily, considering both test games together, the evaluation (i.e. payoff) is fair because both scholars have an equal chance of scoring. Furthermore, scholars remain properly incentivised to win under the payoff. This is important to ensure the fairness of user evaluation as well as the potential correctness of the contributions of the unforced winners. Our readers can verify these properties by inspecting Table 1 which summarizes CAGs. The columns of the table indicate the name of the SG being played, the forced scholar (if any), the SG winner and the payoffs.

User Evaluators

We devised an algorithm to evaluate users by estimating their strength based on scores derived from winning SGs. The most straight forward naïve approach is to use the scores as strength estimates $Str_1(U_i) = \sum \mathit{Payoff}(U_i, U_j)$. More sophisticated (and hopefully more accurate) approaches take into account some of the fairness issues mentioned in the previous section. For example, taking into account that users might not have played the same number of games. We can use the following approach, we call it SIMPLE, to estimate user strengths $Wins_2(U_i) = \sum \mathit{Payoff}(U_i, U_j)$, $Losses_2(U_i) = \sum \mathit{Payoff}(U_j, U_i)$ and $Str_2(U_i) = Wins_2(U_i)/(Wins_2(U_i) + Losses_2(U_i))$.

It is also possible to add further sophistication by taking into account the strengths of the opponents. Winning against a strong opponent should give a larger boost to the estimated strength. Losing against a strong opponent should only give a small hit to the estimated strength. Following these assumptions, we arrive at the following approach, we call it ITERATIVE, to estimate user strengths:

Informally, the algorithm starts with the user strength estimates derived by the SIMPLE algorithm. Then it computes the weighted wins and losses for each user based on the payoffs and the strength of their opponents. Then it computes strength as the fraction of weighted wins divided by the sum of weighted wins and losses. The last two steps are iterated to a fixpoint.

$$\begin{aligned}
Str_3^0(U_i) &= Str_2(U_i) \\
Wins_3^{(k)}(U_i) &= \sum \mathit{Payoff}(U_i, U_j) * Str_3^{(k-1)}(U_j) \\
Losses_3^{(k)}(U_i) &= \sum \mathit{Payoff}(U_j, U_i) * (1 - Str_3^{(k-1)}(U_j)) \\
Total_3^{(k)}(U_i) &= Wins_3^{(k)}(U_i) + Losses_3^{(k)}(U_i) \\
Str_3^{(k)}(U_i) &= \begin{cases} 0.5, \text{if } Total_3^{(k)} = 0 \\ Wins_3^{(k)}(U_i)/Total_3^{(k)}(U_i), \text{o/w} \end{cases}
\end{aligned}$$

Claim Evaluators

The naïve approach relies on three assumptions:

1. the verifier (falsifier) winning an SG of claim c as a truth (falsehood) evidence of c.
2. all evidences have equal weights.

Based on these assumptions, the truth likelihood of c is $Prc\,holds = E_v/(E_f + E_v)$ where E_v (E_f) is the number of times the verifier (falsifier) has won SGs of claim c.

A more sophisticated approach takes estimated strength of the losing user as the weight of an evidence. Furthermore, both approaches should filter out the evidences where the winners are forced into their winning positions. The rationale is that the justification provided by the winners through the SG does not match with their initial position.

System Evaluation

To evaluate our system, we test it with a crowd of synthetic users with a predetermined skill level, on a claim with known truth. Should the system be sound, we expect the probability of correctly classifying a claim to be positively correlated with the skill level of the crowd. Moreover, the estimated user strength should be consistent with the preset skill level of users.

Synthetic Users

Users are expected to take positions (either verifier or falsifier) on claims, select a subformula of a compound formula and to provide values for quantified variables. To select a position, the user can play an SG against itself and select the winning position. To select a subformula, the user may examine the subformulas left to right, select a position to take on each subformula and select the first subformula on which it would take the same position as its position on the compound formula. Based on this approach, it suffices to supply functions to provide values for quantified variables (i.e. Skolem functions [2]) in order to define a user. We call these set of Skolem functions, a strategy.

A synthetic strategy with quality p is expected to provide the correct examples (and counterexamples) with probability p. To provide a correct example for $\exists x p(x)$, the provided example x_0 must satisfy $\exists x p(x) \Rightarrow p(x_0)$ (i.e. when $\exists x p(x)$ is true, $p(x_0)$ should better be true). To provide a correct counterexample for $\forall x p(x)$, the provided example x_0 must satisfy $\forall x p(x) \Leftarrow p(x_0)$.

We define synthetic strategy s out of three components, a perfectly winning strategy w , a perfectly losing strategy l and the quality p. To define a perfect winning strategy for SGs of the claim family $sp(c) := \forall x \in [0,1] \exists y \in [0,1] : x \cdot y + (1-x) \cdot (1-y^2) \geq c$. we need to supply two Skolem functions $provideX(c)$ and $provideY(x,c)$. We observe that the best y should maximize $x \cdot y + (1-x) \cdot (1-y^2)$ where both x and y are in $[0,1]$. Utilizing basic knowledge of calculus, we arrive at:

$$provideY(x,c) = \begin{cases} 1 & \text{if } x = 1 \\ x/(2 - 2 * x) & \text{otherwise} \end{cases}$$

and $provideX(c) = 0.552786$ which is the x-coordinate of the saddle point of $x \cdot y + (1-x) \cdot (1-y^2)$.

Soundness of Synthetic Users Suppose that we have two synthetic strategies for the $sp(c)$ claim family, sv with quality pv and sf with quality pf. Given a true claim such as $sp(0.6)$, what is the probability that a verifier ver, using sv, wins an SG against a falsifier fal, using sf? What is the probability that fal wins? What are the probabilities for a false claim such as $sp(0.9)$? Should our approach to constructing synthetic users be sound, we expect the probabilities of ver winning SGs on correct claims and fal winning SGs on false claims to be positively correlated to pv and pf.

Pr$\{ver$ wins $|$true claim$\}$ = pv because the claim is true and therefore the falsifier selection of counterexamples cannot affect the correctness of subclaims and that there is only one verifier action. Pr$\{fal$ wins $|$false claim$\}$ = $pf + (1 - pf)(1 - pv)$ because the claim is false and therefore if the falsifier selection of counterexamples is correct then the resulting claim would be false and the remaining verifier action cannot affect the correctness. And if the falsifier selection of counterexample is incorrect, then the resulting claim would be correct. However the falsifier can still win should the verifier select the wrong example.

To summarize, the probability of correctly classifying true claims monotonically increases with pv. The probability of correctly classifying false claims is monotonically increases with pf and monotonically decreases with pv. This is consistent with the view that the verifier is responsible for showing the claim true and the falsifier is responsible for showing the claim false.

Configuration (CIM-UE-CE)	Classification Quality	Inconsistent skill estimates
AL-SM-UNW	0.807	0.185
AL-SM-W8D	0.851	0.185
AL-IT-UNW	0.807	0.231
AL-IT-W8D	0.848	0.230
TH-SM-UNW	0.808	0.444
TH-SM-W8D	0.837	0.446
TH-IT-UNW	0.807	0.442
TH-IT-W8D	0.836	0.443

Table 2: Experimental Results

Experiments

We evaluated the quality of claim and user evaluation produced by 8 different configurations of our system. The quality of claim evaluation for true claims is the estimated truth likelihood produced by the CE ect. For false claims it is $1 - ect$. For user evaluation, we report the fraction of pairs of users with consistent rankings. A pair of users has inconsistent rankings if the estimated skill level of the first user is higher than the estimated skill level of the second user while the second user has a higher probability of taking correct action and vice versa.

We repeated the experiment with three different crowd distributions BIMODAL, NORMAL, and UNIFORM and for the true claim $sp(0.2)$ and the false claim $sp(0.75)$. The NORMAL crowd had 25 synthetic users with the following probabilities of taking correct actions 0, 0.1, 0.2, 0.2, 0.3, 0.3, 0.3, 0.4, 0.4, 0.4, 0.4, 0.4, 0.5, 0.5, 0.5, 0.5, 0.5, 0.6, 0.6, 0.6, 0.7, 0.7, 0.8, 0.9 and 1. The BIMODAL crowd had 21 synthetic users with the following probabilities of taking correct actions 0, 0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.3, 0.3, 0.4, 0.5, 0.6, 0.6, 0.7, 0.7, 0.7, 0.7, 0.8, 0.8, 0.9 and 1. The UNIFORM crowd had 10 users with the following probabilities of taking correct actions 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.

The numbers we report on Table 2 are for the UNIFORM crowd on $sp(0.2)$ and are averages over 10000 experiments. However, the findings we report below are consistent across other crowds and for $sp(0.75)$ as well. A configuration consists of a CIM, a UE and a CE. We tested two CIMs, AL and TH. AL is a round robin of CAGs. TH is a round robin of CAGs where only a randomly chosen third of the games where played. The two UEs we tested are SM and IT. SM is the simple UE defined above as Str_2. IT is the iterative UE defined above as Str_3. The two CEs we tested are UNW and W8D. UNW is the naïve CE with games where winners were forced are ignored. W8D is the strength weighted CE described above again with games where winners were forced are ignored. We published our implementation at (IMP) and hardcoded the seeds for random number generators to make our results perfectly repreducable.

Findings

We found the strength weighted claim evaluator W8D to enhance the classification quality. We also found the full round

[2] Technically, $\forall x p(x)$ can be rewritten as $\neg\exists x \neg p(x)$. The verifier (falsifier) provides examples for existentially quantified variables under an even (odd) number of negations.

robin to be produce fewer inconsistent rankings than the partial round robin. Surprisingly, we found that the simple user evaluator to produce significantly fewer inconsistent rankings than the iterative evaluator.

Experience with SCG

We report on our experience using SCG in teaching algorithms classes (Kar). The most successful course (using (Kleinberg and Tardos 2005) as textbook) was in Spring 2012 where the interaction through the SCG encouraged the students to solve difficult problems. Almost all homework problems were posted as claims, and the students posted both their exploratory and preformatory actions (Linderoth 2010) [3] on Piazza (Pia). The students collaboratively solved several problems such as the problem of finding the worst-case inputs for the Gale-Shapely stable matching algorithm.

We do not believe that, without the SCG, the students would have created the same impressive results. The SCG effectively focuses the scientific discourse on the problem to be solved.

The SCG proved to be adaptive to the skills of the students. A few good students in a class become effective teachers for the rest thanks to the SCG mechanism.

Related Work

Crowdsourcing and Human Computation

There are several websites that organize competitions. Examples include, TopCoder.com and Kaggle.com. We believe that we provide a foundation to such websites.

A comprehensive study of crowdsourcing is in (Kittur et al. 2013). They argue that an ideal crowd work system would offer peer-to-peer and expert feedback and encourage self-assessment. Such a system would help workers to learn, and produce better results. In SCG, we have this suggestion built in. Users communicate through the SGs and give each other feedback which leads to learning. Self-assessment is based on counting the wins. SCG provides also significant autonomy for the workers, as long as they follow the SG rules.

We provide a specific, but incomplete proposal of a programming interface to work with the global brain (Bernstein, Klein, and Malone 2012). What is currently missing is a payment mechanism for scholars and an algorithm to split workers into pairs based on their background. Our approach can be seen as a generic version of the "Beat the Machine" approach for improving the performance of machine learning systems (Attenberg, Ipeirotis, and Provost 2011) as well as other scientific discovery games, such as FoldIt and EteRNA. (Cooper et al. 2010b) describes the challenges behind developing scientific discovery games. (Andersen et al. 2012) argues that complex games such as FoldIt benefit from tutorials. This also applies to our system, but a big part of the tutorial is reusable across formal scientific disciplines.

Kittur and Chi and Suh (Kittur, Chi, and Suh 2008) argue that we should make creating believable invalid responses as effortful as completing the task in good faith. They also show that introducing verifiable questions improves the response quality significantly. The recommendation for micro-task markets is: "It is extremely important to have explicitly verifiable questions as part of the task."

In SCG all questions are verifiable through the SGs and CAGs. Indeed, in SCG it is hard to create believable invalid responses. For example, if you intensionally misclassify a claim as false, you will be stuck to defending a false claim. You will only succeed against a weak user.

Crowdsourcing for consensus tasks is studied in (Kamar, Hacker, and Horvitz 2012) by managing the tradeoff between making more accurate predictions about the correct answer by hiring more workers and the costs for hiring. A task is called a consensus task if it centers on identifying a correct answer that is not known to the task owner. The model checking tasks of SCG fall into this category. Our work is different because of our utilization of SGs.

Mechanism Design

The high-level goal of mechanism design is to design a protocol, or mechanism, that interacts with participants so that self-interested behavior yields a desirable outcome (Papadimitriou 2001). The strategy domains in SCG are the legal definitions of the Skolem functions. A specific strategy is expressed by an avatar. The outcome of a SCG tournament when a specific CIM is applied is the social welfare: The claims with their likelihood that they are true or false.

Logic and Games

Logic has long promoted the view that finding a proof for a claim is the same as finding a defense strategy for a claim. Logical Games (Marion 2009), (Hodges 2009) have a long history going back to Socrates. The SCG builds on Paul Lorenzen's dialogical games (Keiff 2011).

Recursive Winning Strategies The intuitionistic logic community has studied (Berardi 2007) winning strategies in the context of Tarski games. While those investigations are focusing on specific domains, like arithmetic, they contain useful abstraction useful to SCG.

Foundations of Digital Games

A functioning game should be deep, fair and interesting which requires careful and time-consuming balancing. (Jaffe et al. 2012) describes techniques used for balancing that complement the expensive playtesting. This research is relevant to SCG lab design. For example, if there is an easy way to refute claims without doing the hard work, the lab is unbalanced.

Architecting Socio-Technical Ecosystems

This area has been studied by James Herbsleb and the Center on Architecting Socio-Technical Ecosystems (COASTE) at CMU http://www.coaste.org/. A socio-technical ecosystem supports straightforward integration of contributions from many participants and allows easy configuration.

[3] Choosing a claim and a position are exploratory actions, supporting actions are performatory actions.

Our proposed system has this property and provides a specific architecture for building knowledge bases in (formal) sciences. Collaboration between scholars is achieved through the scientific discourse which exchanges instances and solutions. The structure of those instances and solutions gives hints about the solution approach. An interesting question is why this indirect communication approach works.

The NSF workshop report (Scacchi 2012) discusses socio-technical innovation through future games and virtual worlds. The SCG is mentioned as an approach to make the scientific method in the spirit of Karl Popper available to CGVW (Computer Games and Virtual Worlds).

Online Judges

An online judge is an online system to test programs in programming contests. A recent entry is (Petit, Giménez, and Roura 2012) where private inputs are used to test the programs. Topcoder.com includes an online judge capability, but where the inputs are provided by competitors. This dynamic benchmark capability is also expressible using our approach: The claims say that for a given program, all inputs create the correct output. A refutation is an input which creates the wrong result.

Educational Games

Our proposed system can be used as an educational tool. One way to create adaptivity for learning is to create an avatar that gradually poses harder claims and instances. Another way is to pair the learner with another learner who is stronger. (Andersen 2012) uses concept maps to guide the learning. Concept maps are important during lab design: they describe the concepts that need to be mastered by the students for succeeding in the game.

Formal Sciences and Karl Popper

James Franklin points out in (Franklin 1994) that there are also experiments in the formal sciences. One of them is the 'numerical experiment' which is used when the mathematical model is hard to solve. For example, the Riemann Hypothesis and other conjectures have resisted proof and are studied by collecting numerical evidence by computer. In our case, experiments are performed during the play of SGs.

Karl Popper's work on falsification (Popper 1969) is the father of non-deductive methods in science. Our approach is a way of doing science on the web according to Karl Popper.

Scientific Method in CS

Peter Denning defines CS as the science of information processes and their interactions with the world (Denning 2005). Our approach makes the scientific method easily accessible by expressing the hypotheses as claims. Robert Sedgewick in (Sedgewick 2010) stresses the importance of the scientific method in understanding program behavior. Using our system, we can define labs that explore the fastest practical algorithms for a specific algorithmic problem.

Games and Learning

Kevin Zollman studies the proper arrangement of communities of learners in his dissertation on network epistemology (Zollman 2007). He studies the effect of social structure on the reliability of learners.

In the study of learning and games the focus has been on learning known, but hidden facts. In our case, learning is about learning unknown facts, namely new constructions.

Origins

We started this line of work with the Scientific Community Game (SCG). A preliminary definition of the SCG was given in a keynote paper (Lieberherr, Abdelmeged, and Chadwick 2010). (Lieberherr 2009) gives further information on the SCG. The original motivation for the SCG came from the two papers with Ernst Specker: (Lieberherr and Specker 1981) and the follow-on paper (Lieberherr and Specker 2012). Renaissance competitions are another motivation: the public problem solving duel between Fior and Tartaglia, about 1535, can easily be expressed with the SCG protocol language.

Conclusion and Future Work

The paper presented a novel approach to crowdsource the decision of formal science claims, and a novel approach to evaluate our crowdsourcing system using synthetic users. The paper also presented a number of alternative implementations of our system components and the evaluation of different configurations of our system. The paper also presented our experience with applying an earlier informal version of the system in teaching.

We plan to further extend our approach to encourage skilled users to submit a) proofs for claims that can be used to enchance estimates for claim truth likelihood and b) claims about the relationships between structures that can be used to also enhance estimates for claim truth likelihood as well as to enable further interactions between users. We also plan to further extend our current CIM to enable other useful interaction patterns between users. And to further make CEs and UEs resilient to potential attacks.

References

[Andersen et al. 2012] Andersen, E.; O'Rourke, E.; Liu, Y.-E.; Snider, R.; Lowdermilk, J.; Truong, D.; Cooper, S.; and Popovic, Z. 2012. The impact of tutorials on games of varying complexity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, 59–68. New York, NY, USA: ACM.

[Andersen 2012] Andersen, E. 2012. Optimizing adaptivity in educational games. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '12, 279–281. New York, NY, USA: ACM.

[Attenberg, Ipeirotis, and Provost 2011] Attenberg, J.; Ipeirotis, P.; and Provost, F. 2011. Beat the machine: Challenging workers to find the unknown unknowns. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[Berardi 2007] Berardi, S. 2007. Semantics for intuitionistic arithmetic based on tarski games with retractable moves. In *Proceedings of the 8th international conference on Typed lambda calculi and applications*, TLCA'07, 23–38. Berlin, Heidelberg: Springer-Verlag.

[Bernstein, Klein, and Malone 2012] Bernstein, A.; Klein, M.; and Malone, T. W. 2012. Programming the global brain. *Commun. ACM* 55(5):41–43.

[Cooper et al. 2010a] Cooper, S.; Treuille, A.; Barbero, J.; Leaver-Fay, A.; Tuite, K.; Khatib, F.; Snyder, A. C.; Beenen, M.; Salesin, D.; Baker, D.; and Popović, Z. 2010a. The challenge of designing scientific discovery games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG '10, 40–47. New York, NY, USA: ACM.

[Cooper et al. 2010b] Cooper, S.; Treuille, A.; Barbero, J.; Leaver-Fay, A.; Tuite, K.; Khatib, F.; Snyder, A. C.; Beenen, M.; Salesin, D.; Baker, D.; and Popović, Z. 2010b. The challenge of designing scientific discovery games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG '10, 40–47. New York, NY, USA: ACM.

[Denning 2005] Denning, P. J. 2005. Is computer science science? *Commun. ACM* 48(4):27–31.

[Franklin 1994] Franklin, J. 1994. The formal sciences discover the philosophers' stone. *Studies in History and Philosophy of Science* 25(4):513–533.

[har] Algorithm development through crowdsourcing. `http://catalyst.harvard.edu/services/crowdsourcing/algosample.html`.

[Hodges 2009] Hodges, W. 2009. Logic and games. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Spring 2009 edition.

[IMP] Website. `https://github.com/amohsen/gsg`.

[Ipeirotis et al. 2010] Ipeirotis, P.; Provost, F.; Sheng, V.; and Wang, J. 2010. Repeated labeling using multiple noisy labelers. *This work was supported by the National Science Foundation under GrantNo. IIS-0643846, by an NSERC P, Vol.*

[Jaffe et al. 2012] Jaffe, A.; Miller, A.; Andersen, E.; Liu, Y.-E.; Karlin, A.; and Popovic, Z. 2012. Evaluating competitive game balance with restricted play.

[Johnson 2001] Johnson, G. 2001. The world: In silica fertilization; all science is computer science. The New York Times.

[Kamar, Hacker, and Horvitz 2012] Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, 467–474. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

[Kar] Website. `http://www.ccs.neu.edu/home/lieber/teaching.html`.

[Keiff 2011] Keiff, L. 2011. Dialogical logic. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Summer 2011 edition.

[Kittur et al. 2013] Kittur, A.; Nickerson, J. V.; Bernstein, M.; Gerber, E.; Shaw, A.; Zimmerman, J.; Lease, M.; and Horton, J. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, CSCW '13, 1301–1318. New York, NY, USA: ACM.

[Kittur, Chi, and Suh 2008] Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, 453–456. New York, NY, USA: ACM.

[Kleinberg and Tardos 2005] Kleinberg, J., and Tardos, E. 2005. *Algorithm Design*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

[Kulas and Hintikka 1983] Kulas, J., and Hintikka, J. 1983. *The Game of Language: Studies in Game-Theoretical Semantics and Its Applications*. Synthese Language Library. Springer.

[Lieberherr, Abdelmeged, and Chadwick 2010] Lieberherr, K. J.; Abdelmeged, A.; and Chadwick, B. 2010. The Specker Challenge Game for Education and Innovation in Constructive Domains. In *Keynote paper at Bionetics 2010, Cambridge, MA, and CCIS Technical Report NU-CCIS-2010-19*. `http://www.ccs.neu.edu/home/lieber/evergreen/specker/paper/bionetics-2010.pdf`.

[Lieberherr and Specker 1981] Lieberherr, K. J., and Specker, E. 1981. Complexity of Partial Satisfaction. *Journal of the ACM* 28(2):411–421.

[Lieberherr and Specker 2012] Lieberherr, K. J., and Specker, E. 2012. Complexity of Partial Satisfaction II. *Elemente der Mathematik* 67(3):134–150. `http://www.ccs.neu.edu/home/lieber/p-optimal/partial-sat-II/Partial-SAT2.pdf`.

[Lieberherr 2009] Lieberherr, K. 2009. The Scientific Community Game. Website. `http://www.ccs.neu.edu/home/lieber/evergreen/specker/scg-home.html`.

[Linderoth 2010] Linderoth, J. 2010. Why gamers don't learn more: An ecological approach to games as learning environments. In Petri, L.; Mette, T. A.; Harko, V.; and Annika, W., eds., *Proceedings of DiGRA Nordic 2010: Experiencing Games: Games, Play, and Players*. Stockholm: University of Stockholm.

[Marion 2009] Marion, M. 2009. Why Play Logical Games. Website. `http://www.philomath.uqam.ca/doc/LogicalGames.pdf`.

[Papadimitriou 2001] Papadimitriou, C. 2001. Algorithms, games, and the internet. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, 749–753. New York, NY, USA: ACM.

[Petit, Giménez, and Roura 2012] Petit, J.; Giménez, O.; and Roura, S. 2012. Jutge.org: an educational programming judge. In *Proceedings of the 43rd ACM technical sympo-*

sium on Computer Science Education, SIGCSE ’12, 445–450. New York, NY, USA: ACM.

[Pia] Website. http://www.piazza.com.

[Pietarinen 2000] Pietarinen, A. 2000. Games as formal tools vs. games as explanations. Technical report.

[Popper 1969] Popper, K. R. 1969. *Conjectures and refutations: the growth of scientific knowledge, by Karl R. Popper*. Routledge, London.

[Scacchi 2012] Scacchi, W. 2012. The Future of Research in Computer Games and Virtual Worlds: Workshop Report. Technical Report UCI-ISR-12-8. http://www.isr.uci.edu/tech_reports/UCI-ISR-12-8.pdf.

[Sedgewick 2010] Sedgewick, R. 2010. The Role of the Scientific Method in Programming. Website. http://www.cs.princeton.edu/~rs/talks/ScienceCS.pdf.

[von Ahn and Dabbish 2004] von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’04, 319–326. New York, NY, USA: ACM.

[Zollman 2007] Zollman, K. J. S. 2007. The communication structure of epistemic communities. *Philosophy of Science* 74(5):574–587.