An Analytical Approach to Optimizing The Utility of ESP Games

Chien-Wei Lin, Kuan-Ta Chen, Ling-Jyh Chen
Institute of Information Science
Academia Sinica
{masaki, ktchen, cclljj}@iis.sinica.edu.tw

Irwin King, Jimmy H.M. Lee
Department of Computer Science & Engineering
The Chinese University of Hong Kong
{king, jlee}@cse.cuhk.edu.hk

***Abstract*—In this paper, we propose an analytical model for computing the utility of ESP games, i.e., the throughput rate of appropriate labels for given images. The model targets generalized games, where the number of players, the consensus threshold, and the stopping condition are variable. Via extensive simulations, we show that our model can accurately predict the stopping condition that will yield the optimal utility of an ESP game under a specific game setting. A service provider can therefore utilize the model to ensure that the hosted ESP games produce high-quality labels efficiently, given that the number of players willing to invest time and effort in the game is limited.**

I. Introduction

With the help of Web 2.0 technology and appropriate designs to motivate people, any group of Internet users, who do not know each other, can combine their "computation power" to solve AI-hard problems. Because of this ability, the process is called *social computation*. In [1], Ahn and Dabbish proposed the ESP game, a real-time, web-based, two-player application. To play, in each round, the randomly matched players keep suggesting appropriate labels to describe an image until they achieve a *consensus*, i.e., both players suggest the same label. If the players achieve a consensus, the label they agree on is likely to be an appropriate description of the current image.

In this paper, our objective is to model the performance of the ESP game and optimize its utility by redefining the criteria for finishing a game. The ESP game proposed in [1] only allows two players to participate. Once they achieve a consensus, the current image is considered solved and the game continues with the next image. In our study, we consider a more generalized ESP game that incorporates the following extensions:

1) The number of players, n, can be greater than 2.
2) The consensus threshold, m, can be any positive integer that is not larger than n; that is, a label is considered a consensus decision if it is proposed by m out of n players.
3) The stopping condition, k, can be any positive integer; that is, an image is considered correctly labeled if k consensuses are reached.

In our framework for generalized ESP games, the game proposed by Ahn and Dabbish [1] corresponds to an instance where $n = 2$, $m = 2$, and $k = 1$. Hereafter, we use "ESP games" or "games" to refer to the proposed generalized version. As some variants of ESP games ask players to label audio clips instead of images, we use the term "puzzle" to denote the target object that players must label by consensus.

In our model, we assume that the number of appropriate labels for each puzzle is limited, and all remaining words are considered inappropriate. For example, to label an image containing a red car beside a river, "car," "river," "red" are considered appropriate or good. Other words are considered inappropriate or bad, even if there is a consensus among the players. For example, players may input typos like "cra," "rive," or "rde" by mistake, or words that are too vague or general, such as "picture," "photo," "sea" and still achieve a consensus occasionally. In such cases, we deem that the current game yields a bad label and the quality of the game's output is decreased.

We model the *utility* of generalized ESP games, i.e., the throughput rate of good labels for the puzzles and its relationship with the game's settings, i.e., the number of players, the consensus threshold, and the stopping condition. We find that a tradeoff exists between the efficiency of the consensus achieved and the quality of matched labels. Our contribution in this work is three-fold:

1) We present a generalized ESP game in which the number of players, the consensus threshold, and the stopping condition are variable.
2) We propose a probabilistic model that can predict the efficiency, quality, and utility of an ESP game based on the game's settings.
3) Via extensive simulations, we show that the proposed model can accurately predict the optimal stopping condition, which facilitates the maximal utility of a generalized ESP game. This feature can be used by game service providers to maximize the outcome of games, given that the number of players willing to invest time and effort in the game is limited.

The remainder of this paper is organized as follows. In Section II, we review related works. We present our probabilistic model for generalized ESP games in Section III, and evaluate its performance via simulations in Section IV. Section V details the optimal stopping conditions predicted by our model. Then, in Section VI, we summarize our conclusions.

This research was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under NSC Grants: NSC 96-3113-H-001-010, NSC 96-3113-H-001-011, and NSC 9X6-3113-H-001-012.

II. Related Work

Since Ahn and Dabbish first proposed the concept of the ESP game in [1], a number of social games based on similar ideas have been developed. In the ESP game, players are required to guess the same label for a given image provided by the system. Subsequently, Ahn et al. proposed `Peekaboom` [4], which does not require participants to submit appropriate descriptions for a given image. Instead, players must "circle" a certain object in the image based on a given description. The main difference between an ESP game and `Peekaboom` game is that, in an ESP game, the players guess *what* an image is but they describe *where* an object is in an image in `Peekaboom`.

`Verbosity` [3] collects so-called commonsense decisions of the game's two participants, which can be used for further research on knowledge or commonsense reasoning and analysis. A similar game called `Phetch` [2] is designed to capture users' natural language expressiveness about an image. The system will record the textual statements, and we can use them as a corpus for research on natural language understanding.

Our work differs significantly from previous studies because we do not propose a new game to test the participants' knowledge. As the ESP game is an effective social computation platform that can "extract" users' knowledge during game play, we believe that it can be "optimized" in terms of "outcomes" through appropriate design.

III. Modeling of ESP Games

In this section, we describe the proposed probabilistic model for generalized ESP games. First we detail our assumptions and define the variables of the model. We then estimate the number of rounds required to solve a puzzle, as well as the number of good and bad labels suggested by participants before a puzzle is finally solved. Finally, based on our model, we evaluate the productivity of an ESP game by three characteristics, namely, efficiency, quality, and utility.

A. Assumptions

Our model of an ESP game is based on the following assumptions:

1) *Round-based play*. We assume that the game play is round-based rather than continuous. In each round, a player can only make one guess about the current puzzle, and the system checks whether the players' guesses match at the end of each round.
2) *Independent guess*. For model tractability, we assume that a player's current guess is not affected by his/her guesses in previous rounds.
3) *Good and bad words*. We assume that the number of "good" labels for each puzzle is limited, so all remaining words are considered "bad", i.e., inappropriate. We assume that players will do their best to guess good words in the vocabulary. However, there is a possibility that they will fail to pick the right words.
4) *Uniform guess*. We assume that players' guesses are drawn uniformly from both the good and bad vocabulary pools.

In our model, we assume that n players participate in a game. In addition, the consensus threshold is set to m, and the stopping condition is set to k. For a certain puzzle, the size of the good vocabulary is denoted by v_{good}, while that of the bad vocabulary is denoted by v_{bad}. Thus, the total number of words that players can choose from is $d = v_{good} + v_{bad}$. The probability that a player will guess a word in the good vocabulary is $prob_{good}$; and the probability that a player will guess a bad word is $prob_{bad}$, which is equal to $1 - prob_{good}$. The default value of $n = 2$, $m = 2$, $v_{good} = 20$, $d = 1000$, $prob_{good} = 0.8$.

B. Time Required to Solve Puzzles

We begin by modeling the number of rounds required to solve a puzzle, i.e., how many rounds it takes to satisfy the specified stopping condition k. The terms "consensus" and "match" are used interchangeably to indicate that a label has been proposed by m players, and denote the label as a *matched label*. In addition, we define a discrete random variable, S, to represent the number of rounds needed to solve a puzzle, and write the probability mass function of S as follows:

$$\begin{aligned} f_S(s) &= \Pr(\text{no. of matches} \geq k \text{ in the } s_{th}\text{round}) \\ &= \Pr\left(\text{no. of matches} \geq k \text{ in the first } s \text{ rounds}\right) \\ &\quad - \Pr\left(\text{no. of matches} \geq k \text{ in the first } (s-1) \text{ rounds}\right). \end{aligned}$$

We assume the probability that exactly i matches will occur in the first s rounds is $P(i; s)$, and that the i matches will comprise i_{good} matches from good words and i_{bad} matches from bad words. The number of good matches, i_{good}, must be in the range 0 and $min(i, v_{good})$, and $i_{good} + i_{bad} = i$.

Now we focus on computing the probability of i_{good} matches in the first s rounds. On average, each player in the first s rounds proposes $s_{good} = s \cdot prob_{good}$ good words and $s_{bad} = s \cdot prob_{bad}$ bad words. We can model the probability of one good match occurring in the first s rounds as

$$\begin{aligned} &P_{good}(1) \\ &= 1 - \sum_{q=0}^{m-1} \binom{n \cdot s_{good}}{q} \left(\frac{1}{v_{good}}\right)^{q} \left(1 - \frac{1}{v_{good}}\right)^{n \cdot s_{good} - q}. \end{aligned} \tag{1}$$

Next, we model the probability of i_{good} good matches occurring in the first s rounds. Therefore, the probability of i_{good} good matches in the first s rounds can be computed by

$$\begin{aligned} &P_{good}(i_{good}) \\ &= C_{i_{good}}^{v_{good}} P_{good}(1)^{i_{good}} [1 - P_{good}(1)]^{v_{good} - i_{good}}. \end{aligned} \tag{2}$$

Similarly, the probability of i_{bad} bad matches in the first s rounds can be computed by

$$\begin{aligned} &P_{bad}(i_{bad}) \\ &= C_{i_{bad}}^{v_{bad}} P_{bad}(1)^{i_{bad}} [1 - P_{bad}(1)]^{v_{bad} - i_{bad}}. \end{aligned} \tag{3}$$

Combining Eq. 2 and Eq. 3, we can derive the probability of i matches in the first s rounds as

$$P\left(i; s\right) = \sum_{i_{good}=0}^{\min(i, v_{good})} P_{good}\left(i_{good}\right) P_{bad}\left(i_{bad}\right).$$

After rewriting the probability mass function of S, the number of rounds needed to solve a puzzle becomes

$$f_S(s) = \left[1 - \sum_{i=0}^{k-1} P\left(i; s\right)\right] - \left[1 - \sum_{i=0}^{k-1} P\left(i; s-1\right)\right].$$

Finally, we obtain the expected number of rounds needed to solve a puzzle as follows:

$$E\left(s\right) = \sum_{s=1} s \cdot f_S\left(s\right).$$

C. Number of Matches

Here we model how many good labels and bad labels are matched. We treat the question of whether a certain word is a match or not as a Bernoulli event, where "success" indicates that the label is matched and "fail" indicates a non-match. The probability of a good label being matched in the first s rounds is shown in Eq. 1. Consequently, the sum of the Bernoulli random variable of each good word will be a binomial random variable with a success probability equal to Eq. 1. It can be computed as

$$\sum_{v_i \in V_{good}} I(v_i \text{ matched}), \tag{4}$$

where V_{good} denotes the set of good words, and $I(\cdot)$ denotes the indicator function. Let $N_{good}(s)$ be the expected value of Eq. 4, i.e., the expected number of good matches in the first s rounds. The value can be derived by

$$N_{good}(s) = v_{good} \cdot P_{good}(1).$$

$N_{bad}(s)$, the expected number of bad matches in the first s rounds, can be derived similarly by

$$N_{bad}(s) = v_{bad} \cdot P_{bad}(1).$$

D. Efficiency, Quality, and Utility

Here we explain how we evaluate the productivity of an ESP game. We define *the efficiency of an ESP game as the rate that labels are matched for the given images*. If the number of participants remains the same, higher efficiency indicates that the system is more "productive" given the same amount of resources. In addition, we define *the quality of an ESP game as the proportion of good labels among all the matched labels*. Higher quality indicates that the matched labels are more likely to be appropriate descriptions of the target puzzle.

However, there is often a trade-off between efficiency and quality in a real system because configurations that yield higher efficiency often lead to lower quality; conversely, settings that yield higher quality may impact on the level of efficiency. For this reason, we define *the utility of an ESP game as the product of the game's efficiency and quality*. This definition enables us to explain utility as *the throughput rate of good labels produced by an ESP game*.

Based on the probabilistic model presented in this section, we can write the formula of the efficiency, quality, and utility of an ESP game as follows:

$$\begin{aligned} \textit{Efficiency} &= \frac{E\left(N_{good}\left(s\right) + N_{bad}\left(s\right)\right)}{E\left(s\right)}; \\ \textit{Quality} &= \frac{E\left(N_{good}\left(s\right)\right)}{E\left(N_{good}\left(s\right) + N_{bad}\left(s\right)\right)}; \\ \textit{Utility} &= \frac{E\left(N_{good}\left(s\right)\right)}{E\left(s\right)}. \end{aligned} \tag{5}$$

IV. Model Validation

In this section, we describe the simulations used to validate our model. After explaining the simulation setup, we compare the utility computed by our model with that derived in the simulations.

A. Simulation Setup

To investigate the accuracy of our model under different settings, we change the parameters and observe whether the simulated quantity of good and bad matches is identical to or close to that computed by our analytical model. Specifically, we change the four major variables, i.e., n, m, v_{good} and $prob_{good}$. When evaluating the effect of one variable, the other three are set to their default values. Moreover, when we adjust the consensus threshold, we set the number of players at 20, as the consensus threshold must be no greater than the number of players.

B. Validation by Utility Curves

In the following, we investigate how the utility of an ESP game changes under different stopping conditions, k. As shown in Fig. 1, the utility reaches its maximum when $n = 2$ and $k = 10$. As the number of participants increases, the shapes of the utility curves change slightly, and the optimal stopping condition shifts slightly to the lower k values. The concave shape of the utility curve indicates that, as k increases, there should be a tradeoff between the efficiency and quality of ESP games such that the utility curve is not monotonic.

We now consider the effects of the other parameters on the utility curves of ESP games, and check the correspondence between the results derived by our model and those of the simulations. The effects of m, v_{good}, $prob_{good}$ are investigated. However, because of space limitations, we only show the conclusion we have. For all the parameters, the utility curves computed by our model are very close to those derived by the simulations. We observe that m and v_{good} have a strong effect on the optimal k, while n and $prob_{good}$ have relatively little effect.

V. Optimal Stopping Conditions

In this section, we focus on how to set the stopping condition to maximize an ESP game's utility. We discuss how they change under different configurations and examine how our optimization method improves the game's utility.

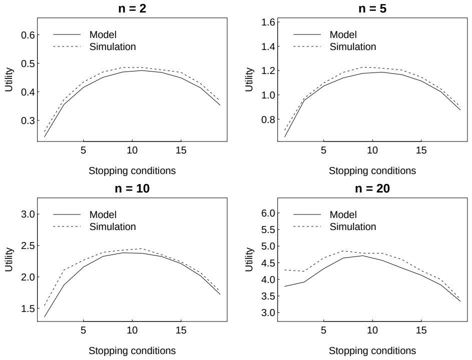

Fig. 1. The relationships between utility and stopping conditions under different *n*.

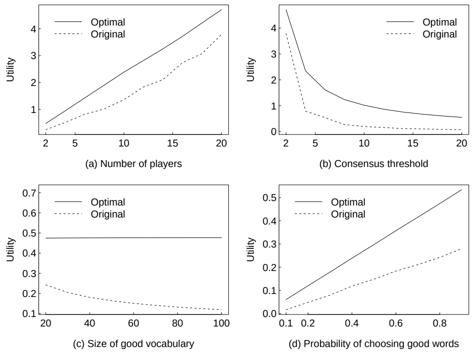

Fig. 3. The effect of the parameters on the improvement in utility

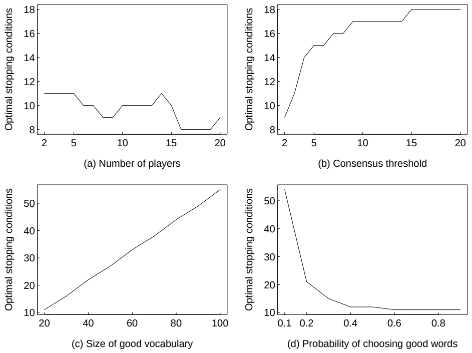

Fig. 2. The effect of the parameters on the optimal stopping conditions

A. *Effect of Parameters*

Here, we consider the effect of different parameters on the optimal stopping conditions. Interestingly, the number of participants does not affect the optimal stopping conditions, as shown in Fig. 2(a). This is reasonable because the probability of good matches and bad matches remains the same regardless of the number of players, which only affects the rate of label matching. The consensus threshold, on the other hand, affects the optimal stopping conditions significantly when it increases, as shown in Fig. 2(b). Raising the consensus threshold makes label matching more difficult; however, the advantage is that matching bad labels will become relatively more difficult than matching good labels. Therefore, when the consensus threshold increases, the matching rate of good labels will grow faster than that of bad labels; consequently, the optimal stopping condition is deferred to allow more good words to be matched before finishing the puzzle.

Both increasing the number of good words and reducing the probability of choosing good words increase the optimal stopping conditions because they make matching good labels more difficult. Thus, a relatively late stopping condition is required in order to increase the proportion of good matches.

B. *Benefit of Optimization*

To demonstrate how optimization improves the game's achieved utility, we examine the gain derived by adopting the optimal stopping condition suggested by our model. We define the *utility gain* as the ratio of the utility of an optimized game to that of a simple ESP game, i.e., with the stopping condition set to 1.

The relationships between the utility gain and various game parameters are shown in Fig. 3. We observe that, the optimization achieved by adopting the optimal stopping condition generally provides a utility boost that is 2 or more times higher than that of the simple ESP game. Even if we consider a more conservative scenario, where only two participants play the game and the consensus threshold is set to 2, the utility gain will be around 2, assuming the number of good words is 20 and the probability of choosing good words is 0.8. Moreover, the utility gain increases rapidly as either the consensus threshold or the size of the good vocabulary increases. The utility gain is only significantly lower than 2 when the number of participants is much higher than 2. However, we can still achieve a utility gain of around 1.3, even the number of players is as high as 20. These findings demonstrate that the utility optimization provided by our analytical model can generally provide twice as much utility as a non-optimized game, which stops immediately after a label has been matched.

VI. Conclusion

We have proposed a generalized ESP game in which the number of players, the consensus threshold, and the stopping condition are variable. In addition, we have presented an analytical model that computes the efficiency, quality, and utility of an ESP game given the game's settings. Via extensive simulations, we show that by applying the optimal stopping condition predicted by our model, the game's utility will be usually be at least 2 times higher than that of a non-optimized game. This feature can be leveraged by game service providers to improve the utilization of finite player efforts in order to maximize both the efficiency and quality of the matched labels.

References

[1] L. von Ahn and L. Dabbish. Labeling images with a computer game. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004.

[2] L. von Ahn, S. Ginosar, M. Kedia, and M. Blum. Improving Image Search with PHETCH. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 4, 2007.

[3] L. von Ahn, M. Kedia, and M. Blum. Verbosity: a game for collecting common-sense facts. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78, 2006.

[4] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64, 2006.