Matchin: Eliciting User Preferences with an Online Game

Severin Hacker
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
shacker@cs.cmu.edu

Luis von Ahn
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
biglou@cs.cmu.edu

ABSTRACT
Eliciting user preferences for large datasets and creating rankings based on these preferences has many practical applications in community-based sites. This paper gives a new method to elicit user preferences that does not ask users to tell what they prefer, but rather what a random person would prefer, and rewards them if their prediction is correct. We provide an implementation of our method as a two-player game in which each player is shown two images and asked to click on the image their partner would prefer. The game has proven to be enjoyable, has attracted tens of thousands of people and has already collected millions of judgments. We compare several algorithms for combining these relative judgments between pairs of images into a total ordering of all images and present a new algorithm to perform collaborative filtering on pair-wise relative judgments. In addition, we show how merely observing user preferences on a specially chosen set of images can predict a user's gender with high probability.

Author Keywords
Web Games, Preference Elicitation, Human Computation

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g. HCI): Miscellaneous

INTRODUCTION
This paper introduces a game that asks two randomly chosen partners "which of these two images do you think your partner prefers?" If both partners click on the same image, they both obtain points, whereas if they click on different images, neither of them receives points. Though seemingly simple, this game presents players with a strangely recursive conundrum that seems to come straight out of *The Princess Bride*: "should I pick the one I like best, or the one I think my partner likes best, or the one I think my partner thinks I like best, etc."

CHI 2009, April 4–9, 2009, Boston, MA, USA.

We study the consequences of observing tens of thousands of people play this game on the Internet. Our findings suggest that the game can collect many types of useful data from the players. First, it is possible to extract a global "beauty" ranking within a large collection of images. This may appear surprising, since perfectly combining relative preferences between pairs of images into a total ordering is technically impossible because pair-wise preferences are not necessarily transitive: the fact that users prefer image A to image B and image B to image C does not necessarily imply that they prefer A to C.

Nevertheless, a good global ranking can be extracted because, interestingly, this is the same problem as inferring the skill of chess players by just looking at their wins and losses. Extracting global rankings of large collections of images based on "beauty" has applications to image search and computer vision. In the case of image search, knowing which images are more appealing could allow for a search engine that displays the more appealing images first. In the case of computer vision, this data could be used to train algorithms that automatically assess the quality of an image (e.g., a camera that tells the user how good a picture is).

Our second finding is that, after a person has played the game on a small number of pairs of images, it is possible to extract the person's general image preferences. This problem is known as collaborative filtering and is well-studied for the case of users giving absolute ratings (e.g., assigning a numerical score to each image). We present a new algorithm for collaborative filtering that needs only relative judgments between pairs of images, and we show that our algorithm is better at predicting the users' behavior than global rankings that do not distinguish among different users. This implies that user preferences on images are, as expected, subjective.

Third, we use the players' preferences between images to create a simple gender model. Based on only ten pair-wise judgments, our model can determine a player's gender with high probability. This shows that responding to a request for seemingly benign information, such as which of two images a user prefers, can actually reveal significant information about a person. Under these circumstances, it becomes questionable whether people really can protect their privacy online.

Although we concentrate on the specific application of images, the wider implication of our findings is that asking partners in a two-player game to guess which of two options their partner will choose represents a viable mechanism for extracting user preferences and data. The game we concentrate on, called Matchin (see Figure 1), has been played by tens of thousands of players for large periods of time.

Matchin follows the spirit of "Games with a Purpose," i.e., games that are fun to play and at the same time collect useful data for tasks that computers cannot yet perform. The ESP Game [1] is probably the most well-known game in this category. Whereas the ESP Game collects labels for images, Matchin collects information about how appealing an image is. Similar games have been proposed for tagging music [2], object detection in images [3], [4] and collection of common-sense facts [5].

PRIOR WORK

A Taxonomy of Methods

There are several methods to elicit user preferences. In this paper we only consider methods that involve more than one person. We call the people who give ratings "judges" and their ratings "judgments" or "decisions." The goal is to combine the ratings from all the judges. We will first look at different desirable features of rating mechanisms.

Absolute Versus Relative Judgments

First, we make a distinction between absolute and relative judgments. An absolute judgment is a judgment that assigns an absolute score to an item, such as a star rating from 1-5 where 1 is worst and 5 is best. On the other hand, a relative judgment only compares items, i.e., "this image is better than that image." Absolute ratings have two problems: *calibration* (or better, the lack of calibration) and *limited resolution*.

Calibration is the problem of defining what a particular rating means compared with previous ratings and compared with other people's ratings. For example, if I usually assign "1" or "2," my "4" might have the same meaning as someone else's "5." Also, judgment may change during the rating process: for example, a user might in their first rating give a "5" to a good image, only to discover later that there are far better images. Thus, they may want to change their first rating to a "4." In practice, however, users rarely adjust their ratings. This creates systematic errors in the data.

Limited resolution is the problem of assigning a rating to an image that is only marginally better than a different image. Assuming that the rating system only has 5 levels, the user might give it the same rating even though they clearly think it is better (but not good enough for the next level). In this case we lose information. To overcome the problem of limited resolution, one could simply use a rating system with finer granularity, say from 1 to 100. However, many judges will not adapt to this system, but instead keep a scale of 1 to 10 in their mind and map "8" to "80" and so forth.

Figure 1. The Matchin game

Relative judgments have the advantage that they are usually easy to make. In most cases, they do not change after seeing new information, i.e., a user who prefers image A over image B will still do so after they have seen other images. Even if the absolute ratings of image A and image B change over time, their relative ordering is likely to stay the same. Therefore, old absolute ratings are more likely to be inaccurate than old relative ratings.

Total Versus Partial Judgments

By total judgments, we mean that the judges are required to make judgments about all of the images. In the case of absolute ratings, the user is required to rate all n images. In the case of relative ratings, the user is required to compare every image with every other image, which is on the order of $O(n^2)$ comparisons. Total judgments are, therefore, infeasible for large datasets. Partial judgments, however, have the problem of how to deal with incomplete data.

Random Access Versus Predefined Access

By random access, we mean that the judges are allowed to search for particular items and then rate them. This has the advantage that the judges can focus on rating things in which they are most interested. However, it has a major drawback: it opens the door to malicious manipulation. Judges could easily search for their own pictures and always give them the highest ratings. This behavior cannot easily be stopped on the Internet since the cost of creating new "fake" identities is extremely low and it is not (currently) possible to tell "fake" accounts from real accounts. Another drawback of random access methods is that some items receive many ratings while others receive few. In such cases, combining the ratings becomes difficult.

By predefined access, we mean methods where the users are given images to rate in a *predefined sequence*. Thus, the users cannot influence which images they can rate. While theoretically it is still possible to cheat just by waiting for one's own images, it is much harder. In a method employing random access, the chance of being able to rate one's own images is 1. For a method that randomly shows

one out of n images (with replacement) to rate, the chance of being able to rate one's own image is $1/n$, and the expected time to wait until one sees their own image is n. Therefore, methods that use predefined access have the desirable property that the possibility of cheating decreases as the amount of data increases.

"I Like" Versus "Others Like"

Another important distinction between methods is whether the judges are asked "what do you like?" or "what do you think others will like?" Although the difference looks subtle at first, it has major implications. We can compare this to the problem of predicting elections. The most common way is to poll potential voters and ask them who they would vote for in the upcoming election. One then takes the sample average as an estimate of the future election result. This is the "I like" case. The other option is to ask them "who do you think is going to win the election?" In this case they will not only consider their own opinion but also the opinion of their friends and relatives in combination with external information (polls, news, etc.). This is the "others like" case. Here, every voter automatically becomes a *weak predictor*, because every voter only has a limited amount of information at his/her disposal. In this "others like" case we can make a further distinction between methods that ask for what one particular partner might prefer and what other people in general prefer.

Direct Versus Indirect

By direct, we mean methods that ask the judges about the "beauty" of an image. Indirect methods would infer "beauty" through meta-information. Examples of meta-information are number of views, number of comments, number of tags, and number of pages linking to a particular image. Indirect methods have the disadvantage that, once the methods are known, their ratings can quickly besubjected to cheating. People could easily create many comments on their own images, add lots of tags, create dummy pages linking to their images, etc. This means that even though an indirect method might use predefined access (for example by crawling images and counting incoming links, etc.), people still have random access to the meta-information and can change it in any way they want.

Existing Methods

We have categorized several existing methods using our taxonomy in Table 1.

Flickr Interestingness

The popular online photo sharing Web site Flickr [6] has developed its own algorithm to rank images. Although their algorithm has not been published, we know from their patent application [7] that it is at least partly based on meta-data such as "the quantity of user-entered meta-data concerning the media object, the number of users who have assigned metadata to the media object, an access pattern related to the media object, a lapse of time related to the media object, and/or; on the relevance of metadata to the media object." (We note that all these meta-data can easily be faked.)

	Flickr	Voting	Hot/Not	Matchin
Partial	Yes	Yes	Yes	Yes
Direct	No	Yes	Yes	Yes
Predefined access	No	No	Partly	Yes
Others like	Partly	No	No	Yes
Relative	No	No	No	Yes

Table 1. A taxonomy of methods

This means that Flickr's "interestingness" does not measure "beauty" directly. Some of the meta-data measures how much other people will possibly like an image. A link to an image, for example, is usually created because the authors think the image might be interesting to their readers. However, the problem with all methods that rely on meta-data (like number of comments) is that established long-term users who have many friends on that network have an advantage. Ultimately, it is not clear whether "interesting-ness" measures the "interestingness" of the image or the popularity of its author.

Voting

Perhaps the simplest method of eliciting user preferences is just to let users vote on images, using either approval/ disapproval or a rating scale (e.g., 1 to 5 stars). Users can search for particular items and vote on them (random access). This is possibly the most frequently used method on the Web: Digg [8], YouTube [9] and others all use variants of this scheme to rate and rank their content.

These methods, since they are based on random access, share the common characteristic that some items receive many votes while others receive few. This leads to a new problem of combining these ratings into a global ranking. If two items have the same average rating, but one has 1,000 votes while the other one only has 10 votes, the one with more votes should probably be ranked higher. However, generalizing this principle is non-trivial.

Hot or Not

The Internet site "Hot or Not" [10] uses a voting system from 1 to 10. It is limited in that it ranks only images of people. The most important difference from the previously mentioned sites is that a normal user is given random images to rate; they cannot search for them. However, it is still possible for people to send a link to an image to a friend who can then rate the picture. Therefore, it is still easy for malicious users to cheat and rate their friends' pictures higher than they might rank otherwise.

Related Work

Gajos and Weld [11] studied preference elicitation for user interface optimization. Their method works both with passive feedback (users can change the interface to their need) and active feedback ("this user interface is better than that user interface"). In this work, we only consider active

feedback. Also, in their work they assume that the preferences depend on some measurable features (like whether a certain button is visible or not), while we treat the images as a black box (i.e., we never look at the pixel values of the images).

OUR GAME

The Mechanism

Matchin is a two-player game that is played over the Internet. At the beginning of the game, a player is matched randomly with another person who is visiting the game's Web site at the same time. If there is no other player available at the same time, we pair them with a bot (a computer that plays as if it was a human). After the player is matched with its partner (either human or machine), they play several rounds. In each round, the two players see the same two images and both are asked to click on the image that they think their partner likes better. If they agree, they both receive points.

Thus, if the players want to score many points, they not only have to consider which image they prefer, but also which image their partner might prefer. Every game takes two minutes. One pair of images, or "one round," usually takes between two to five seconds, thus a typical game consists of 30-60 rounds. To make the game more fun, the players are given more points for consecutive agreements. More specifically, Matchin uses a sigmoid function for scoring games. The scoring function is shown in Figure 2. While the first match only earns a few points, the second and third match in a row earn exponentially more points until the seventh match at which point the growth of the function decreases.

At the end of the game, the two players can review all of their decisions and chat with each other. All clicks are recorded and stored in a database. We also store the time it took for the users to make a decision.

The bot uses these stored clicks to emulate a human as closely as possible. When it "sees" two images, it "clicks" on the image that was considered to be better by a human in an earlier game. The bot mimics the same person for the entire game. Also, the bot waits exactly as long as the human did. (Note, however, that the bot's "clicks" are not recorded.)

For the results in this paper, we use a collection of 80,000 images from Flickr that were gathered during a one-week time period in October 2007. In every round, we show two random pictures from this collection, favoring images for which we have not yet collected enough data.

One of the underlying design objectives of Matchin was to remove any systematic errors in the resulting data, i.e., all of the judgments should be correct in that they truly reflect the judge's opinion. The judgments should also be robust in the sense that they should still be considered "valid" after a long time. Matchin is thus a rating system that uses the previously defined concepts of relative judgment and

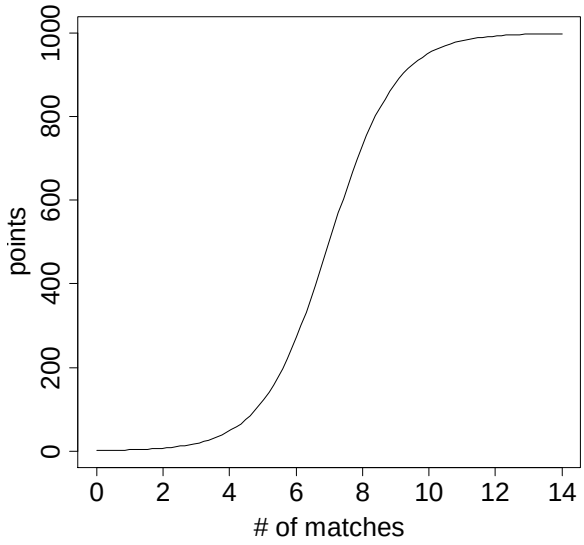

Figure 2. The scoring function used in Matchin.

predefined access. Matchin gives an incentive for the judges to consider not only their own opinion, but also the possible opinions of others in making their judgments.

Because of predefined access, Matchin is very hard to cheat. In fact, with 80,000 images, it would take malicious users on average more than a week of game-play until they could see their target image to improve its rating. Even then, because the user can only compare images, the impact of a single malicious judgment is minimal. We also note that to minimize cheating, images are presented to both players in random order (i.e., the image on the left for one player might be the image on the right for the other player).

The main difficulty in creating games with a purpose is to make them enjoyable. We have studied several variations of the game in order to make it more fun. Our most notable finding is regarding the scoring function. When we first designed the game we used a constant scoring function: 100 points for every agreement, the same scoring function as in the ESP Game. In play testing, we found that players could get many points by quickly picking the images at random (e.g., always choosing the one on the left). This allowed players to get 100 points with 50% probability in every round even without taking the time to look at every image and thus made the game less enjoyable. We then switched to a scoring function that gave a higher reward to consecutive agreements. At first, we used a linear scoring function where the first match earned 100 points; the second earned 200, the third 300, etc. Informal player testing showed that this made the game more fun. We later switched to an exponential scoring function and test players got even more excited about playing. The rewards, however, sometimes became too high (after 20 matches in a row, players were getting 2^{20} points), so we had to limit the amount of points that could be earned in a single round.

That is why we chose the sigmoid function that has a maximum at 1,000 points. The exponential and sigmoid scoring functions create an artificial ladder from which players can fall if they make a mistake, similar to TV's *Who Wants to Be a Millionaire?*

THE DATA

We launched the game on a dedicated Web site, called GWAP [12] (for "Games With A Purpose") on May 15, 2008. Within only four months, 86,686 games had been played by 14,993 players. In total, there have been 3,562,856 individual decisions (clicks) on images. Since the release of the game, there has been on average a click every three seconds (including nights!). This shows that the game is both very enjoyable and works well for collecting large amounts of data.

An individual decision/record is stored in the form:

<id, game_id, player, better, worse, time, waiting_time>

Where *id* is a number assigned to identify the decision, *game_id* is the ID of the game, *player* is the ID of the player who made the decision in this record, *better* is the ID of the image the player considered better, *worse* is the ID of the other image, *time* is the date and time when the decision was made, and *waiting_time* is the amount of time the player looked at the two images before making a decision.

GLOBAL RANKINGS

We first look at several methods to combine the relative judgments into a global ranking. For the global ranking, we consider the data as a multidigraph $G = (V, E)$, a directed graph which is permitted to have multiple arcs between two nodes. The nodes V are the images. For every decision made by a user to prefer image $i \in V$ over a different image $j \in V$ there exists a directed edge $(i, j) \in E$. The goal then is to produce a global ranking $\boldsymbol{r} = \boldsymbol{r_1} > \boldsymbol{r_2} > \boldsymbol{r_3} > \cdots > \boldsymbol{r_n}$ over all of the images. The following methods all have in common that they use a *ranking function* $f: V \mapsto \mathbb{R}$ that maps every image to a real number first, called its *rank value*, and then applies sorting. For this *induced ranking* $\hat{\boldsymbol{r}}_{\boldsymbol{f}}$ it holds that an image is ahead of a different image if its rank value is larger:

$$\hat{\boldsymbol{r}}_{\boldsymbol{f_x}} > \hat{\boldsymbol{r}}_{\boldsymbol{f_y}} \Leftrightarrow f(x) > f(y)$$

Methods

We compare three different ranking functions: empirical winning rate (EWR), ELO rating, and TrueSkill rating.

Empirical Winning Rate (EWR)

Perhaps the simplest form of a ranking function is to use the empirical winning rate as an estimate for an image's quality. The empirical winning rate is the number of times an image was preferred over a different image, divided by the total number of comparisons in which it was included. In graph terms, the empirical winning rate of an image is just its out degree divided by its degree:

$$f_{EWR}(i) = \frac{\deg^+(i)}{\deg(i)}$$

The empirical winning rate is easy to understand, but has two problems. For images that have a low degree (because they have taken part in few comparisons), the empirical winning rate might be artificially high or low. The second problem is that it does not take the quality of the competing image into account, i.e., "winning" against a bad image is worth the same as "winning" against a good image.

ELO Rating

The ELO rating tries to overcome the latter problem. The ELO rating system [13] was introduced by Arpad Elo for rating chess players. In this model, each chess player's performance in a game is modeled as a normally distributed random variable. (We note that later studies showed logistic distribution to be a better model for chess rankings.) The mean of that random variable should reflect the player's true skill and is called the player's ELO rating. If a player wins, his/her ELO rating goes up, otherwise it goes down. The actual difference depends on how good the other player is, i.e., how surprising a win or loss is.

For learning, we first initialize each image's ELO rating R_i to 1,600. Before each comparison between two images i and j we compute their *expected scores* (i.e., their expected chance of winning in this comparison) E_i and E_j according to a scaled logistic function:

$$E_i = \frac{1}{1 + 10^{\frac{R_i - R_j}{400}}}$$

$$E_j = \frac{1}{1 + 10^{\frac{R_j - R_i}{400}}}$$

The factor 400 is chosen such that a player whose ELO score is 200 higher than another player's ELO score has a chance of winning of about 75%. After the comparison, we know that either image i or image j won, i.e., we know the *true score* S_i: $S_i = 1$ if image i won and $S_i = 0$ if image i lost. The prediction error is $S_i - E_i$. We then update the image's ELO rating R_i accordingly:

$$R_i \leftarrow R_i + K(S_i - E_i)$$

Thus, if the *expected score* of image i is above its *true score* the image's ELO rating will be adjusted downward, otherwise it will be adjusted upward. K is a model parameter that defines by how much the scores of the two images are changed. A large value of K makes the scores more sensitive to "winning" or "losing" a single comparison. To compute the ELO ratings, we iterate over all comparisons in our training set and update the R_i's accordingly. We then use the image's ELO ratings as our ranking function:

$$f_{ELO}(i) = E_i$$

We compared models with different values of K and found that the prediction error is relatively insensitive with respect to K. For all experiments below we chose $K = 16$.

TrueSkill Rating

The ELO ranking system assumes that all players have the same variance in their performance. Thus, a player who consistently plays at a medium level will have the same ELO rating as a player who sometimes plays at a high level but also sometimes plays very poorly.

TrueSkill [14] overcomes this problem by describing every player with *two* variables, a mean skill and a variance around that mean skill. TrueSkill employs a full Bayesian graphical model. In TrueSkill, every player's skill s_i is modeled as a normally distributed random variable centered around a mean μ_i and per-player variance σ_i^2. A player's particular performance in a game then is drawn from a normal distribution with mean s_i and a per-game variance β^2, where β is a constant. Intuitively, the higher μ_i is, the better the player. The larger σ_i^2 the more unstable the player's performance is, sometimes he/she is good sometimes bad. Finally, the larger β^2 the more the game's outcome depends on factors other than skill. For games where skill is important β^2 should be smaller than for games of chance. If a player's performance, as drawn in this process, is higher than another player's performance the model predicts a "win."

Working out the math, one obtains the following update equations for the case where image i wins over image j:

$$\mu_i \longleftarrow \mu_i + \frac{\sigma_i^2}{c} \cdot v\left(\frac{\mu_i - \mu_j}{c}\right)$$

$$\mu_j \longleftarrow \mu_j - \frac{\sigma_j^2}{c} \cdot v\left(\frac{\mu_i - \mu_j}{c}\right)$$

$$\sigma_i^2 \longleftarrow \sigma_i^2 \cdot \left[1 - \frac{\sigma_i^2}{c^2} \cdot w\left(\frac{\mu_i - \mu_j}{c}\right)\right]$$

$$\sigma_j^2 \longleftarrow \sigma_j^2 \cdot \left[1 - \frac{\sigma_j^2}{c^2} \cdot w\left(\frac{\mu_i - \mu_j}{c}\right)\right]$$

$$c^2 = 2\beta^2 + \sigma_i^2 + \sigma_j^2$$

where

$$v(x) = \frac{\mathcal{N}(x)}{\Phi(\mathrm{x})}$$

$$w(x) = V(x) \cdot (V(x) + x)$$

($\mathcal{N}(x)$ and $\Phi(x)$ are the standard normal probability density and cumulative distribution function, respectively.). Thus, the winner/loser's mean is adjusted upward/ downward. Also, both variances become smaller, reflecting our intuition that after every comparison we know the true skill of the two players better than before.

As our ranking function, we use the *conservative skill estimate*, which is approximately the first percentile of the image's quality:

$$f_{TrueSkill}\ (i) = \mu_i - 3\sigma_i$$

Thus, with very high probability the image's quality is *above* the conservative skill estimate. The initial values for μ_i,σ_i^2 and for the constant β were chosen according to [14].

COLLABORATIVE FILTERING

In the previous methods, we treated all users equally. In the collaborative filtering setting, we want to find out not only about general preferences, but also about each individual's preferences. This allows us to recommend images to each user based on his/her preferences. Also, we can compare users and images with each other ("which users are similar?" and "which images are similar?"). Therefore, we have developed a new collaborative filtering algorithm we call "Relative SVD" that uses only comparative judgments as its input.

Relative SVD

This model is based on work by Takács et al. [15] for collaborative filtering with absolute ratings. We adapt their model to work in a setting where we only have information about relative ratings. The name stems from the fact that, in the case of absolute ratings, the model solves for the singular value decomposition of a rating matrix $\boldsymbol{X}$.

In this model, each user i and each image j is described by a feature vector of length K. We store the user feature vectors in a $n_{users} \times K$ matrix $\boldsymbol{U}$, where each row is a user's feature vector. Similarly, we store the image feature vectors in a $n_{images} \times K$ matrix $\boldsymbol{V}$. We say that the amount by which user i likes image j is equal to the dot product of their feature vectors:

$$likes(u_i, v_j) = u_i^T v_j$$

We interpret the data gathered from our game as a set D of triplets (i, j, k) where i is a user and j is the image that was preferred over the image k in a comparison. We set d_{ijk} equal to 1 for each element in the training set:

$$\forall\ (i, j, k) \in D\colon d_{ijk} = 1$$

The error for a particular decision e_{ijk} between a sample from the training data and our model can then be written as

$$e_{ijk} = d_{ijk} - (likes(u_i, v_j) - likes(u_i, v_k))$$

And the total sum of squared errors (SSE) as:

$$SSE = \sum_{(i,j,k) \in D} e_{ijk}^2$$

The goal is to find the feature matrices that minimize the total sum of squared errors:

$$(\boldsymbol{U}, \boldsymbol{V}) = \underset{(\boldsymbol{U},\boldsymbol{V})}{\operatorname{argmin}} SSE$$

In order to minimize this error, we compute the partial derivatives for each user feature vector and for each image feature vector:

$$\frac{\partial e_{ijk}^2}{\partial u_i} = 2e_{ijk} \cdot (v_j - v_k)$$

$$\frac{\partial e_{ijk}^2}{\partial v_i} = 2e_{ijk}\, u_i$$

$$\frac{\partial e_{ijk}^2}{\partial v_j} = -2e_{ijk}\, u_i$$

Applying ordinary gradient descent with a step size of η while adding a regularization penalty with parameter λ to prevent over-fitting, we obtain the following update equations for the feature vectors:

$$\boldsymbol{u_i \leftarrow u_i + \eta \cdot 2\, e_{ijk}(v_j - v_k) - \lambda u_i}$$

$$\boldsymbol{v_j \leftarrow v_j + \eta \cdot 2\, e_{ijk} u_i - \lambda v_j}$$

$$\boldsymbol{v_k \leftarrow v_k - \eta \cdot 2\, e_{ijk} u_i + \lambda v_k}$$

We also obtain the following algorithm:

1. Initialize the image feature vectors and the user feature vectors with random values. Set λ, η to small positive values.
2. Loop until converged:
   a. Iterate through all training examples $(i, j, k) \in D$.
      i. Compute e_{ijk}
      ii. Update u_i
      iii. Update v_j
      iv. Update v_k
   b. Compute model error.

We experimented with different values for the feature vector length K. In general, the larger K the better, however, the training takes much longer for large values of K. We found $K = 60$ to be a good value. After some experimentation, we found $\eta = 0.02$ and $\lambda = 0.01$ to be good values for the step size and penalty parameters, respectively.

After the user and feature image vectors have been computed, one can easily predict how much a user will like an image just by computing the dot product between their corresponding feature vectors. This allows us to recommend images to users.

ANALYSIS

Comparison of the Models

We split our data into two-thirds for training and one-third for testing. We then trained all four models on the training data. After that, we used the learned models to predict users' behavior on the test data, for which we know the users' actual decisions. Table 2 shows the error on the testing set for different amounts of training data. For all models, the error decreases as we use more training data. For fewer data points, we find that ELO works best, while EWR and Relative SVD perform worst. However, as we increase the amount of training data, Relative SVD beats all the other models. Also, looking at the learning curve in Figure 3, we see that EWR, ELO and TrueSkill seem to converge at an error rate of around 30% while Relative SVD shows no sign of converging at 17%.

Local Minimum

An important question is: Do humans learn while playing the game, i.e., do they learn which type of images are generally preferred? If they adapt too much, it could have unwanted reinforcement effects, i.e., a slight preference for outdoor pictures over indoor pictures at the beginning might lead new users to adapt to this trend and also click on outdoor images so that they can earn more points.

After a while, it would become common knowledge that in this game, outdoor pictures are always preferred over indoor pictures. This would be bad for the validity of our results, since it would not reflect the players' true opinions. This behavior is similar to the well-known problem of an optimization procedure becoming stuck around a local minimum. To test if this was happening, we compared the agreement rate (i.e., the percentage of times a player agrees with his/her partner) of first-time players and other players. We have found that first-time players agree 69.0% of the time with their partner, while the more experienced players agree 71.8% of the time with their partner. This relatively small increase indicates that the players only marginally adapt to the game. This is good news for us because it minimizes the risk of becoming stuck around a local minimum. We have also measured if people learn within a game by measuring the agreement rate in the first half of the game and comparing it to the agreement rate in the second half of the game. In the 100 games we analyzed, the

# judgments	**233,032**	**466,065**	**699,098**	**932,131**	**1,165,164**	**1,398,197**	**1,631,230**	**1,864,263**	**2,097,296**	**2,330,329**
EWR	39.3%	34.3%	32.6%	31.6%	31.2%	30.7%	30.4%	30.1%	30.0%	29.9%
ELO (K=16)	**36.5%**	**33.5%**	32.3%	31.4%	31.0%	30.6%	30.3%	30.1%	30.0%	29.8%
TrueSkill	38.0%	34.4%	33.5%	31.8%	31.3%	30.7%	30.5%	30.2%	30.0%	29.9%
Rel. SVD	45.4%	35.9%	**31.2%**	**28.4%**	**26.6%**	**24.5%**	**22.3%**	**20.7%**	**18.8%**	**16.9%**

Table 2. Testing error of different ranking algorithms as we increase the number of judgments in the training data. While relative SVD distinguishes among users, the other algorithms do not.

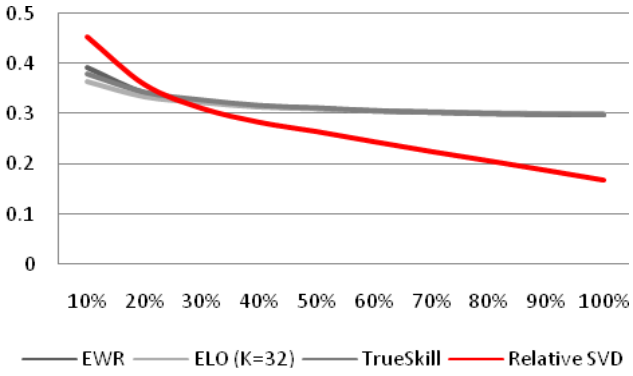

Figure 3. The learning curve for several learning algorithms.

agreement rate goes down slightly from 67% to 64%, which indicates that it is hard to learn the partner's preferences.

Gender Prediction

Intrigued by the fact that Relative SVD performed much better than the other algorithms, we concluded that the decisions are subjective and wondered if we could exploit that fact to create a gender test. We know the gender for 2,594 players from their profile settings (though this information is self-reported and therefore may be wrong). Of these players, 68% are male and 32% female. To find a pair of images (A, B) that has a strong gender bias, we compute the conditional entropy $H[G|X]$ of the player's gender G given that we know the player's decision X, where $A > B$ means that image A was considered better than image B:

$$H[G|X] = \sum_{x \in \{A<B, A>B\}} \Pr(x)\, H[G|X = x]$$

A pair (A, B) has a large gender bias (and is therefore good for determining the gender of a new player) if the conditional entropy $H[G|X]$ is small, i.e., learning the decision tells us a lot about the gender. The necessary conditional probabilities $\Pr(G = g|X = x)$ can be computed with Bayes' rule given the class conditionals $\Pr(X = x|G = g)$. For the class conditionals, we trained two ELO predictors, one with male players only and one with female players only. We then compute $H[G|X]$ for many pairs of images and select pairs for which $H[G|X]$ is smaller than a fixed threshold value.

To predict the gender of new users we sample 10 edges from those with strong gender bias and we ask the users to choose the image they prefer for each pair. In order to make our intentions less obvious, we add some random image pairs. Once we know their decisions on the 10 pairs, we use a simple naïve Bayes classifier to predict their gender. The naïve Bayes classifier assumes that the individual decisions are independent given the gender and chooses the label $\hat{g}$ that maximizes the likelihood of the data:

$$\hat{g} = \text{argmax}_g \Pr(G = g) \prod_{i=1}^{10} \Pr(X_i = x_i|G = g)$$

Figure 4. Women prefer the image on the left while men prefer the image on the right.

Figure 5. Women prefer the image on the left while men prefer the image on the right.
For this pair of images it is hard to guess which one is preferred by women and which one by men.

Figure 4 shows a pair of images that has a strong gender bias. Generally, females prefer the image with the horse rider while men prefer the image with the hut. While many people could have guessed that the image with the horse is the "female" image, not in all cases the pairs with gender bias satisfy common prejudices about the sexes. Figure 5 shows two images for which it is more difficult to guess which one is preferred by men versus women.

We conducted a study with 102 people from Amazon Mechanical Turk [16]. After filtering out people who did not finish the test, we achieved a total accuracy of 78.3%.

DISCUSSION

Figure 6 shows some of the top ranked images by the different global ranking algorithms. Independent of the ranking algorithms, we made the observation that nature pictures are ranked higher than pictures depicting humans.

Sunsets, in particular, are among the very top. Maybe surprisingly, among the 100 highest ranked pictures there is not a single picture in which a human is the dominant element. Animal pictures are also preferred over pictures depicting humans. Animals—especially exotic animals like pandas, tigers, chameleons, fish and butterflies—are highly ranked. Pets, on the other hand, are also ranked high, but usually below the aforementioned animals. Pictures of flowers, churches, and bridges are very highly ranked. Not surprisingly, pictures of famous tourist attractions, like the Sydney Opera, made it into the top 100.

On the other hand, among the worst pictures, almost all were taken indoors and include a person. In addition, many of these pictures are blurry or too dark. Some of the worst pictures are screenshots or pictures of documents or text.

Generally, the pictures that made it into the top 100 are neither provocative nor offensive. This could mean that since the players do not know their partner (or their gender) they go for a "safe" choice. Most of the highly ranked pictures express peaceful and harmonious environs (like a waterfall or a sunset). This suggests that people think a random person will most likely prefer peace and harmony. On the other hand, the pictures that have achieved a high score of "interestingness" on Flickr are often provocative and artistic. While a professional picture of a skull is among the most interesting pictures on Flickr, it would not make it into the top 100 pictures in Matchin.

The results that we obtained from collaborative filtering indicate that there are substantial differences among players in judging images, and taking those differences into account can greatly help in predicting the users' behavior on new images. In fact, we can predict with a probability of 83% which of two images a known player will prefer, compared to only 70% if we do not know the player beforehand. As Figure 3 shows, the error rate of the Relative SVD predictor does not seem to be converging yet. Note that we cannot say that Relative SVD "is better" than the other algorithms since they solve different problems.

Figure 6. Some of the top ranked images by different global ranking algorithms

Interestingly, we found that more experienced players had about the same error rate as new players. This is evidence that the players do not learn much about their partner's preferences by playing the game, either because it might be hard or because they do not care that much about points.

The fact that we can easily create a test to predict the gender of an unknown person that only asks the user to pick their favorite of two images is fascinating. When these pictures do not satisfy common biases about gender preferences (as in Figure 5), it becomes hard for the users to pick the "female" or "male" picture. For people concerned about privacy, it is perhaps scary to think that with so little data, one can get substantial private information about a person. It also opens the question of whether privacy really exists on the Internet.

CONCLUSION

The main contribution of this paper is to provide a new method to elicit user preferences. For two images, we ask users not to tell which one they prefer, but rather which one a random person will prefer. We reward them if they are correct in their prediction. We compared several algorithms for combining these relative judgments into a total ordering and found that they can correctly predict a user's behavior in 70% of the cases. We describe a new algorithm called Relative SVD to perform collaborative filtering on pair-wise relative judgments. Relative SVD outperforms other ordering algorithms that do not distinguish among individual players in predicting a known player's behavior. This suggests that preferences about images are, as expected, subjective. Finally, we present a gender test that asks users to make some relative judgments and, based only on these judgments, we can predict a random user's gender in roughly 4 out of 5 cases.

One area of future work would be to generalize the game to other kinds of media and other types of questions. The game, as it was presented, should work equally well for short videos or songs. Also, instead of asking "which image do you think your partner prefers?" one could ask "which image do you think your partner thinks is more interesting". One could also give prior information about their partner e.g. "given your partner is female which image do you think your partner prefers?" It remains to be investigated how much other personal information can be gathered in the same way as our gender test does.

ACKNOWLEDGEMENTS

We would like to thank Mike Crawford and Edison Tan for their help with the successful deployment of the game, and Susan Hrishenko and the CHI 2009 reviewers for their feedback on this paper. This work was partially supported by generous gifts from the Heinz Endowment and the Fine Foundation. Luis von Ahn was partially supported by a Microsoft Research New Faculty Fellowship and a MacArthur Fellowship.

REFERENCES

1. von Ahn, L. and Dabbish, L. Labeling images with a computer game. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems* (Vienna, April 24-29). ACM, New York, 2004, 319-326.
2. Law, E. and von Ahn, L. Input-Agreement: A New Mechanism for Collecting Data using Human Computation Games. To appear in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems* (Boston, April 4-9). ACM, New York, 2009.
3. Lee, B. and von Ahn, L. Squigl: A Web game to generate datasets for object detection algorithms. In submission.
4. von Ahn, L., Liu, R., and Blum, M. Peekaboom: a game for locating objects in images. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems* (Montréal, April 22-27). ACM, New York, 2006, 55-64.
5. von Ahn, L., Kedia, M., and Blum, M. Verbosity: a game for collecting common-sense facts. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems* (Montréal, April 22-27). ACM, New York, 2006, 75-78.
6. Flickr, a Web site for photo-sharing. http://www.flickr.com
7. Butterfield; Daniel S.; et al. Interestingness ranking of media objects. U.S. Patent application 2006/0242139 A1, filed February 8, 2006.
8. Digg, a Web site for social bookmarking. http://www.digg.com
9. Youtube, a Web site for sharing videos. http://www.youtube.com
10. Hot or Not, a Web site for rating pictures of people. http://www.hotornot.com
11. Gajos, K. and Weld, D. S. Preference elicitation for interface optimization. In *Proc. 18th Annual ACM Symp. on User Interface Software and Technology* (Seattle, Oct. 23-26). ACM, New York, 2005, 173-182.
12. GWAP, a Web site for Games with a Purpose. http://www.gwap.com
13. Elo, Arpad. *The Rating of Chessplayers, Past and Present.* Arco Publications, New York, 1978.
14. Herbrich, R., Minka, T., and Graepel, T. TrueSkill™: A Bayesian Skill Rating System. In *Proc. Advances in Neural Information Processing Systems 19*, MIT Press, Cambridge, MA, 2007, 569-576.
15. G. Takács, I. Pilászy, B. Németh, and D. Tikk. On the Gravity Recommendation System. In *Proc. 13th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining* (San Jose, August 12-15). ACM, New York, 2007, 22-30.
16. Amazon Mechanical Turk. http://www.mturk.com