iSee: Interactive Scenario Explorer for Online Tournament Games

Greg Smith, Desney Tan, Bongshin Lee
Microsoft Research
One Microsoft Way
Redmond, Washington 98052, USA
{gregsmi,desney,bongshin}@microsoft.com

ABSTRACT

Fantasy games, in which players compete to correctly predict real-world outcomes in sports, entertainment, and politics, have grown in popularity and now represent a significant portion of online gaming. Pick'em pools, also known as office pools, are a fantasy game specifically focused on tournament-style competitions such as the "March Madness" NCAA basketball championship. Pick'em pool players often spend significant time trying to understand the current state of competition and to anticipate future events that may significantly affect their performance within the pool. Unfortunately, the combinatorial nature of the outcome space makes these tasks extremely challenging, and intuition is often a highly inaccurate guide. In this paper we present iSee, a system that allows players to make these complex calculations and inferences. We describe a variety of interface options for the interactive presentation of tournament outcome visualizations. We also describe in detail the implementation of a set of algorithms for reliably projecting player performance and distilling the complex outcome space to a number of key scenarios. Finally, we report on a pilot study soliciting user feedback on the system.

Categories and Subject Descriptors. H.5.m [Information Interfaces and Presentation (e.g., HCI)]: miscellaneous; I.6.8 [Types of Simulation]: Monte Carlo; K.8 [Personal Computing]: Games.

General Terms. Algorithms, Design, Human Factors.

Keywords. Fantasy games, tournament, office pools, scenario exploration.

1. INTRODUCTION

Fantasy games, in which players try to correctly predict outcomes of real-world competitions, represent a large and growing genre within the online gaming industry. In 2007, the Fantasy Sports Trade Association (FSTA) reported 19.4 million fantasy sports players in the United States and Canada, with a sustained annual growth of 7-10% [7]. Similarly, comScore reports that one popular fantasy soccer website in the United Kingdom had 1.4 million unique users in August 2007 alone [4]. Additionally, FSTA estimates that each fantasy player directly spends about US$500 annually on magazines, online information, contests, and leagues [8], with much more revenue garnered from paid advertising. To tap into this large market, many major online portals (e.g. ESPN, FoxSports/MSN, Yahoo! Sports, CBS Sportsline, etc) have developed dedicated fantasy game offerings. Other sites have expanded the fantasy genre from its traditional sports base to predictions in domains such as Hollywood movies [10], celebrity news [1], and political races [6].

Pick'em pools, also known as tournament or office pools, are a large component of the fantasy industry. In pick'em pools, players predict outcomes in real world tournament-style competitions, such as the Football League Cup in the UK or the "March Madness" NCAA basketball championship in the US. Because of the simplicity of participating, pick'em pools appeal to a wide range of players. Even with little domain knowledge, casual players can make reasonable predictions using rankings, expert opinions, or even frivolous factors such as team mascot preference. On the other hand, expert players can spend large amounts of time researching and applying complex inferences and 'insider knowledge' in order to make their picks.

As soon as the real-world tournament begins, predictions are locked and players can no longer directly influence the outcome. The "fun" of a pick'em pool lies in anticipating outcomes, rooting for real world results that favor certain predictions, and engaging in social interactions with other participants regarding possible outcomes and relative pool performance. Therefore, the ease and accuracy with which players can make sense of future possibilities during the tournament is important and has direct impact on the fun of the game. This principle has been implicitly acknowledged in other entertainment domains. For example, poker broadcasts on TV have become quite popular since they have started providing views of players' cards along with projections of how likely they are to win. These statistics further allow commentators to make more interesting comments about the desired cards for each player and about how the game might unfold.

Unfortunately, as we will demonstrate in this paper, the mapping between real world outcomes and pick'em pool performance is quite complex. The answers to straightforward questions are often unintuitive because of the combinatorial space of possible tournament outcomes and the interactions between various player picks. Even for the most dedicated players, the computations are tedious and difficult. The challenge then, is to design a system that can support this information need and make participation more compelling for players at every level.

In our work, we have developed the Interactive Scenario Explorer for Entertainment, or iSee, a system that automatically highlights interesting scenarios within the tournament and allows pick'em

pool players to project future standings within pools. Additionally, the system allows players to interactively explore scenarios they care about. By eliminating tedious calculations and improving players' understanding of the game, the system enhances the enjoyment of the competition and offers both casual and expert players a shared information context around which to anchor "trash talking" and other inter-player communication.

It is important to note that iSee is *not* designed to help players make picks that increase their chances of winning. In fact, iSee treats the picks of all participating players as input to be specified in advance of performing any calculations, and does not affect the actual outcome of the tournament. This is because the system is designed to increase the enjoyment level of pool players, and removing the human from the game by automating the pick process has the potential to do the exact opposite [12].

In the remaining sections, we present background and related work, demonstrate the difficulty of manual calculations, describe the functionality that iSee provides and the interface with which players interact, document implementation details, and present results from a pilot study we conducted.

2. BACKGROUND

2.1 Terminology

A **tournament** refers to a number of **competitors** from a single sport (or other domain of competition) vying to be crowned the overall champion. Depending on the particular tournament, a competitor can be a single person (e.g. athlete), or a group of people (e.g. team). Throughout the illustrative examples in this paper, we use *Japan*, *China*, the *United States*, and *South Korea* as our canonical tournament competitors. Each tournament consists of a sequence of head-to-head **contests** (sometimes referred to as matches, ties, fixtures, or heats) between competitors that lead to some result (i.e. one competitor winning and one losing). The basic goal of a tournament is to winnow multiple competitors down to a single champion. In a single elimination tournament (also known as a knockout or sudden-death tournament), competitors who lose a match are immediately eliminated from the tournament (or at least from winning the tournament), and only winning competitors move on and vie to be the champion. For the purpose of simplifying discussion, the illustrative examples in this paper are all single elimination tournaments, though the principles apply equally well in other formats. A **bracket** is the common term for a tournament visualization in the form of a tree, in which leaf nodes represent the initial configuration of competitors and the root represents the eventual champion. The structure of the bracket defines which competitors will play each other as they progress through the tournament towards the championship.

Within the fantasy games themselves, we use the term **player** to refer to a person who is taking part in the fantasy competition – in other words, a person who has completed a set of contest predictions. This should not be confused with a single-person tournament competitor, who is often also called a player in colloquial language. Throughout the examples in this paper, we use *Alex*, *Beth*, and *Chuck* as our canonical players. The tournament contest predictions made by participating fantasy players are called **picks**. Most pick'em tournaments require that players pick all outcomes within the bracket, and for the purposes of simplicity, this is what we describe in this paper. Again, the extension to other formats is trivial. Multiple players who are competing against each other in the pick'em game form a **pool** or **league**. Each pool applies a

scoring system to reward correct picks, and at the end of the tournament the players' final **placements** (1st, 2nd, 3rd, etc.) are determined by their scores. Pools can comprise just two players, or thousands, and often there are prizes or other recognition for those players finishing in one or more of the top placements. The **placement probability** represents the probability that a given player will finish the pool in a given placement. For example, Alex might have a 75% chance of finishing in 3rd place.

2.2 Related Work

The roots of modern fantasy sports are often traced back to the "Baseball Seminar" in 1960. During this event, people formed rosters of Major League Baseball players and earned points based on various statistics such as RBI (runs batted in) and ERA (earned run average) [14]. In 1980, a small group of dedicated fantasy baseball players, who called themselves Rotisserie League Baseball, began getting mainstream press coverage, and throughout the 1980's fantasy sports continued to grow in popularity. These leagues quickly spread to other sports, such as football, and in 1989 Fantasy Sports Magazine debuted as the first regular publication covering more than one fantasy sport [5].

In the late 1990s, the Internet boom created a revolution in fantasy sports, as the tedious paper-based management and tracking of players and teams went online. There are now dozens of major services supporting the fantasy player market. For example, Yahoo! Sports' StatTracker allows players to track live scores in their fantasy football leagues. However, most existing fantasy services focus primarily on reducing the burden of manual tracking and we believe there is opportunity in allowing players to look beyond the current state of the game and to project what may happen in the future.

The huge number of possible outcomes in a fantasy tournament represents an enormous challenge for sense-making tools. It is very difficult to analyze large data sets with dozens of attributes and discover meaningful information, such as patterns and trends. Data mining is the science of extracting useful information from large data sets or databases [9]. It has been applied to a broad range of fields from business intelligence and financial analysis to eScience, which often generates enormous data sets. Data mining is most useful in exploratory analysis scenarios in which there are no predetermined notions about what will constitute an "interesting" outcome [18]. Since data mining is an iterative process to uncover interesting patterns, trends, and correlations between data items, visualization also plays an important role in these systems. For example, many interactive visualization systems help users visually explore complex relationships in large information databases [18]. With iSee, we attempt to apply analogous techniques to the domain of fantasy tournaments.

While many data mining systems require users to mathematically manipulate data and information, we had to make iSee accessible to non-technical users. Hence, we employ direct manipulation, which provides users with rapid and incremental feedback as they directly manipulate objects in the interface [15]. For example, Spotfire [16], the commercial version of starfield displays [3], is an interactive visualization system that updates color- and size-coded points in a two-dimensional graphical display as users adjust control widgets, such as sliders, buttons, and check boxes. Similarly, to provide this continuous feedback in iSee and to allow users to efficiently explore the information space, we continuously run and display calculations after each user action.
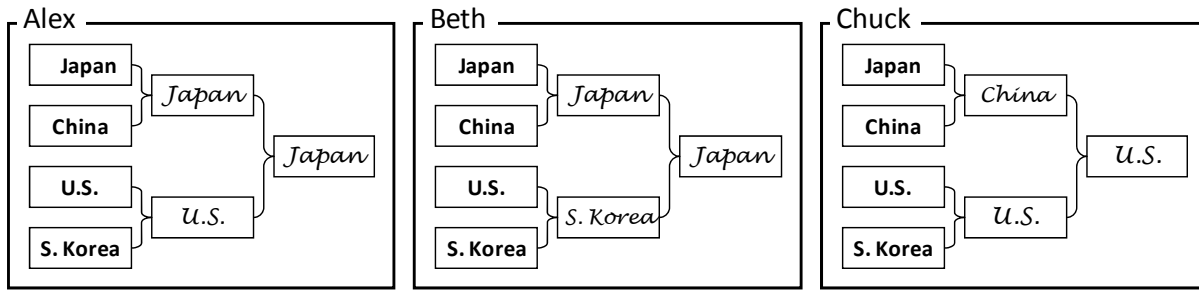
Figure 1. Our three example players have filled out their brackets for the office pool.

3. DIFFICULTY OF CALCULATIONS

Consider the example of Alex, Beth, and Chuck competing in a pick'em pool based on a small soccer tournament. The tournament involves four teams: Japan, China, the United States, and South Korea. In a four-team single-elimination tournament, there are only three games: two semi-finals and a final. Players each fill out a bracket, predicting the winner of each of the three games. Their picks are shown in Figure 1. After each real-world match is played, players earn points according to a scoring function. In our example, players earn 10 points for each outcome correctly predicted. While an escalating scoring functions, such as 10 points for each of the semi-final games and 20 for the finals is more common, we use the same points for every game case for simplicity. With this simple function, each player will have a score between 0 (none correct) and 30 (all three picks correct) at the end of the tournament. These scores determine their final placements.

Placement probability is perhaps the most basic of inferences, but even this is difficult to intuit. Placement probabilities describe players' relative chances of winning the pool, or placing in a certain position, at the end of the tournament. Currently, players use factors such as current scores or trends in scores to determine how well they are doing, but also as an indicator of how well they will do in the future. This approach is highly prone to error.

For example, one might intuit that the odds of each player winning the pool are roughly equivalent early in the tournament. This might seem especially true if we assume all teams are evenly matched (i.e. each team has an equal chance of winning each game) and no games have yet been played. However, many people are surprised to learn that this is not the case.

The most straightforward way to calculate the true likelihood of each player winning is to generate all possible real-world tournament outcomes. The eight possible outcomes for our tournament can be seen in Figure 2. Since we have assumed that each game outcome is equally likely, each tournament outcome is also equally likely to occur. Hence, placement probabilities can be generated simply by applying the scoring function to player picks, and counting how many times each player places in a certain position.

In this example, Alex comes in 1st in two out of the eight outcomes, while Chuck comes in 1st in five of them (some of these are ties, but that is irrelevant for this example). Hence, Chuck is more than twice as likely as Alex to finish in 1st place, even before any games have been played. Intuitively, this can be explained by the specific overlap in player picks. This calculation becomes extremely hard (if not impossible) for people to do when there are more players in the pool, more games to be played, and when teams are not evenly matched, which is often the case in real-world pools. To our knowledge, there is no closed form analytical solution that could make this calculation faster. Various versions of the problem have been shown to be at least #P-Hard [2].

4. FUNCTIONALITY AND INTERFACE

We now describe the functionality and usage of the iSee system in alleviating the challenges presented. We ground our description within a specific interface we have built as a proof-of-concept to demonstrate the underlying approach of performing interesting calculations that provide common ground for social interaction. Where relevant, we also describe possible alternatives to the interface. We show a simplified mock-up screenshot in Figure 3, with constituent components described in additional sections.
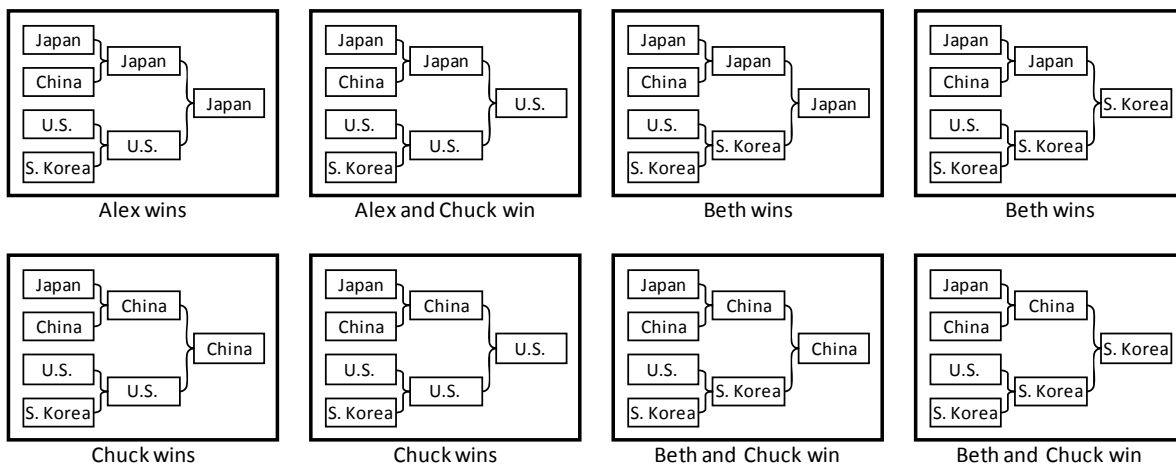

Figure 2. In a very small tournament, we can enumerate all the outcomes and count the winning placements.
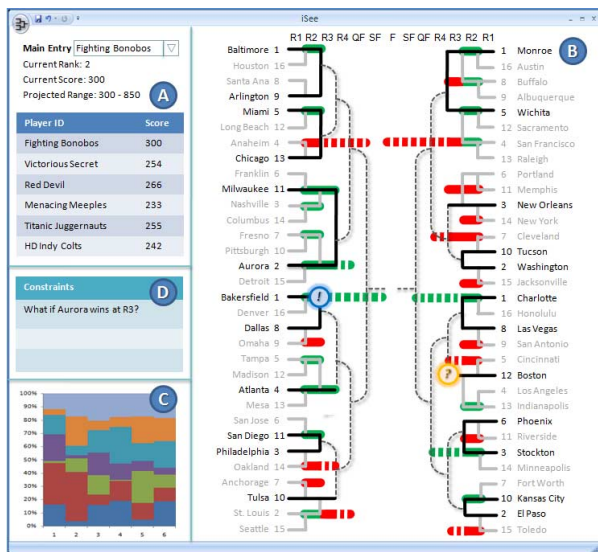
Figure 3. Prototype iSee interface components. A: player list and scoring summary. B: Bracket view (detail in Figure 4). C: Player placement (detail in Figure 5) and competitor performance matrices. D: Constraints.

4.1 System Input

iSee expects several inputs from the pool owner or from individual players. First, the system needs the structure of the tournament and scoring scheme of the pool. Second, iSee requires all player picks, since the overlap in these are critical in making interesting inferences. Third, iSee expects that real-world outcomes will be added as they become available, since it is able to generate new projections with each new piece of information. Most portals already collect and store this information. In pilot feasibility tests, we successfully partnered with MSN/FoxSports to get the required inputs from data that was already stored.

Finally, iSee requires a matrix of how likely each team is to beat each other team. We call this the **prior probability matrix**. By default, iSee derives this matrix by applying a linear function to the rankings within the tournament. For example, a top-ranked team is much more likely to beat a tenth ranked team than to beat a second-ranked team, and so on. But the system also allows users to customize this matrix, either by selecting a different algorithm or by manually changing individual values. This allows players to factor expert picks, betting odds, 'insider knowledge,' personal hunches, or any other such information into their projections.

With these raw inputs, iSee performs its calculations and presents the results in different forms.

4.2 Basic Bracket and Scores

Central to any tournament game is a view of the bracket, which displays real-world outcomes as well as player picks and the correctness of those picks. The bracket is also used to display other information within the structure of the game.

The classic tournament bracket contains a series of match-ups, with competitors who are playing each other connected by a vertical line. The winner of each contest progresses to the next round and their name is filled in on the result line coming out of the matchup. This process continues until there is only one competitor left, the tournament champion.

Unfortunately, this representation consumes large amounts of screen space, which becomes a problem in our interface. Hence we use an alternate representation of the bracket called AdaptiviTree [17]. Rather than repeating textual labels for each win, AdaptiviTree deforms the bracket to present tournament outcomes in a non-textual way (see Figure 4). This is extremely space efficient and studies have shown this visualization to be more consumable. In AdaptiviTree, the correctness of player picks is shown by overlaying simple graphs on top of the tournament brackets. Correct picks are represented by green line segments, and incorrect picks by red ones. A line segment corresponding to a still-viable pick in a future contest is dotted green.

We also display a listing of players and contact information, status messages, current scores, projected scores, and score histories in a separate pane (Region A in Figure 3).

4.3 Placement Probabilities

Placement probabilities represent how likely it is for each player to place in each position when the tournament ends. Each player can end the pool in each rank, and thus the placement probability can be represented as a matrix. Rows represent players, columns represent ranks, and the values in each cell represents the placement probability. To help players identify interesting patterns, we can also show the placement probability using bar graphs or a stacked bar. These three options are shown in Figure 5.

4.4 Competitor Performance

In addition to the placement probabilities, we can also visualize competitor performance. Competitor performance is a view of the likelihood that a given real-world tournament contender will make it to a certain round. In the simplest case, this is a straightforward mathematical derivation of the prior probability matrix and can be described by a matrix with a row for each competitor, a column for each round, and a number in each cell that represents the probability that the competitor will make it to that round of the tournament. We have also explored a more visual representation, overlaying bars on each competitor in the bracket view. In this view, we use visual properties such as the opacity or saturation of the bars to represent the probabilities.
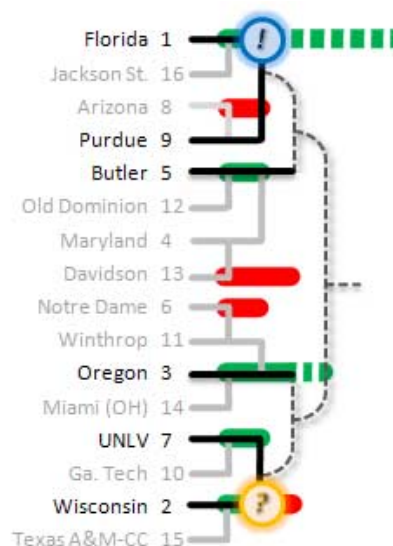

Figure 4. AdaptiviTree layout of a bracket. A player's picks are superimposed on the results with a colored overlay. Icons are overlaid to indicate two possible scenarios of interest.

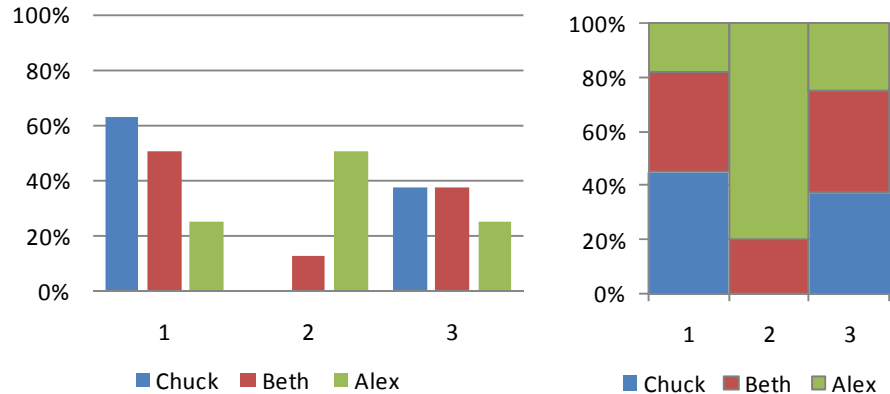|	1st	2nd	3rd
Chuck	62.5%	0.0%	37.5%
Beth	50.0%	12.5%	37.5%
Alex	25.0%	50.0%	25.0%

Figure 5. Three alternative representations for the player placement probability matrix.

4.5 Key Scenarios

iSee provides automated detection of key scenarios, calculated independently for each player in the pool. Players can flip to any-one else's bracket to see their key scenarios. For example, as shown in Region B of Figure 3 as well as in Figure 4, iSee demar-cates certain games on the bracket with icons indicating scenarios of potential importance. These scenarios include different catego-ries of significance, distinguished by icon type. A tooltip on each icon explains how a certain scenario affects an individual player's standings. Sometimes the scenario involves an upcoming game – for example, "If Japan beats China then Alex's probability of winning jumps from 25% to 50%." Sometimes the scenario high-lights when results lead to guarantees in the placements – for ex-ample, "If China wins the first round, Chuck is guaranteed to win the pool." While there is prior work on calculation of importance metrics on individual games in a sporting tournament, that work was focused on aggregate measures of interest to the public rather than personalized metrics based on pool performance [13].

4.6 Scenario Exploration

While the placement probabilities and key scenarios serve as good starting points, expert players often want to interactively explore particular scenarios. iSee supports this functionality by allowing players to set constraints within the inferences. With each new constraint, iSee recalculates and refreshes the probabilities.

There are two basic classes of constraints, game constraints and player placement constraints. Game constraints allow players to explore what-if scenarios for games that have not yet occurred. For example, Beth may like to know how her placement proba-bilities change if China wins in the first round of the tournament, or perhaps what happens if China wins the entire tournament. She can also set multiple unrelated constraints, so long as they do not conflict with each other (i.e. China and its first round opponent cannot both win).

Player placement constraints allow a player to explore the scena-rios in which they are most likely to place in a certain position. For example, Chuck may want to know the most likely scenarios in which he places first. By setting this constraint and looking at the competitor performance matrix, he can tell, for example, that he has the greatest chance of placing first if certain teams make it to certain round, or alternatively, that other teams must not make it beyond a certain point in the tournament. He can also see how other player placement probabilities change in the case that he

places first. For example, Chuck might discover that when he comes in first place, Alice always finishes in last place.

These constraints can be added and removed in a lightweight manner. In the simplest instantiation, we use a series of combo boxes for the player to add and remove constraints. In a more complex version of the system, we use direct manipulation to allow the player to interact directly with the bracket or the com-petitor performance matrix to set constraints, as well as the placement probability component for player placement con-straints. In either case, we keep a running list of current con-straints, but also visually represent them on the relevant compo-nents (Region D in Figure 3). This allows players to quickly dis-tinguish real-world results from what-if explorations.

5. IMPLEMENTATION

The core iSee calculation engine is implemented in C# as a multi-threaded .Net Framework component. Below we describe the derivation and calculation of the two most important outputs of the system, placement probabilities and key scenarios.

5.1 Placement Probabilities

The basic building block of the iSee calculation engine is the placement probability. We begin by describing how the core cal-culation is performed, before discussing a sampling technique and the potential sources of error in the calculations.

5.1.1 Calculation

In a single-elimination tournament, each contest can be won by one of two competitors. Therefore, the number of possible tour-nament outcomes is 2^n, where n is the number of contests remain-ing to be played. In our illustrative four-team tournament, there are eight unique tournament outcomes. If all possible tournament outcomes are all equally likely (e.g. teams in the tournament are evenly matched), then the placement probability is the placement count expressed as a fraction of the outcome space. For example, Alex has a 25% chance of coming in first place.

However, tournament outcome probabilities are rarely uniform. Some tournament competitors are stronger and some are weaker. To account for this, we weight the counted outcomes by the a priori probabilities of those outcomes. Using the prior probability matrix, the table containing the probabilities of any team beating any other team, we multiply the probabilities of the individual game results to get the overall probability of a given full tourna-ment outcome. In the case of evenly-matched teams in a three-

game tournament, each possible tournament outcome would have a probability of $0.5 \times 0.5 \times 0.5 = 12.5\%$. But if a player's winning outcome involved two long-shot competitors and one evenly-matched competitor winning, the outcome probability might be more like $0.3 \times 0.3 \times 0.5 = 4.5\%$. It is the normalized sum of these weighted tournament outcome probabilities, and not the raw count, that represents the realistic placement probability for a given scenario.

When tournaments are relatively small, this brute force method works very well. Unfortunately, the exponential nature of the outcome space makes this intractable in most real-world office pools. In a standard 64-team tournament there are 2^{63} (more than 9 million trillion) unique outcomes when the tournament begins, meaning that it would take millions of years of computing time to produce an exhaustive set of placement probabilities for even a single, small set of pool players. To solve this problem, we rely on sampling techniques that approximate these placement probabilities to a very high degree of accuracy in a fraction of the exhaustive compute time.

5.1.2 Monte Carlo Sampling

Monte Carlo techniques are computational algorithms that simulate analytical solutions by repeating a relatively small number of trials using inputs with appropriately randomized distributions. These simulations are popular in a wide variety of domains, and in fact have previously been applied specifically to the domain of pick'em pools for single-elimination sports tournaments [11].

iSee generates a series of random tournament outcomes according to the distributions of the individual contest probabilities. For each contest in the tournament outcome, we use a random number generator weighted by the two competitors' relative chances of beating each other to decide the winner. For each instance of a hypothetical full tournament outcome, we score each player's performance and rank the finishes accordingly. Each player's final placement is recorded, and accumulated across the repeated trials into a placement histogram. When we are done sampling, we have an $N \times N$ histogram of player placements (where N is the number of players and thus also the number of placements). After normalizing all values by the total number of generated samples, the distribution of player placements across the histogram represents the full matrix of player placement probabilities.

Constraints are handled by straightforward adjustment of bracket generation and accumulation. If the player places a game constraint for Japan to win Game 1, iSee runs the placement probability calculations same as before but fixes the probability of Japan winning Game 1 at 100% during sample outcome generation, instead of using the prior contest probabilities. In other words, calculation proceeds exactly as if Japan had already won Game 1. For a player placement constraint, iSee simply discards from accumulation any generated bracket not meeting the constraint.

The sampling operation is CPU-intensive; generating, scoring, ranking, and accumulating thousands of hypothetical outcomes in rapid succession to explore the outcome space. This computation is highly parallelizable, because each sample outcome can be generated, scored, and ranked independently. Therefore, iSee analyzes the number of processors on the target machine and runs multiple simultaneous threads for sampling. The only inter-thread coordination is in incrementing the histogram.

5.1.3 Errors

A traditional pseudo-random Monte Carlo simulation with normally distributed output will have a rate of convergence of $1/N^{1/2}$, where N is the number of samples, regardless of the dimensionality of the input. This means that even with computationally convenient numbers of samples (e.g. 10,000) we can generate estimates with very tight bounds (standard deviation of 0.01 in placement probabilities).

But the algorithmic error alone does not capture the full story. Because iSee outputs numbers for the purpose of user inspection, analysis, and understanding, the discussion of errors goes beyond the numerical accuracy to encompass the user experience. The numbers iSee offers to the user are probabilities, which can only be verified or refuted by conducting many repeated trials. However, since an office pool is not a repeated experiment, if iSee says Beth has a 10% probability of coming in first place, there is no way in the course of a single tournament to "prove" that the correct number was in fact 6%, or 30%. Beth will either come in first, or she will not. The output placement probabilities are also highly sensitive to the team vs. team contest probabilities. If iSee is given as input the proposition that China is a low-likelihood (weak) competitor, this will contribute directly to a lower likelihood of winning the pool for players that predict tournament wins for China. The bottom line is that any inaccuracies due to algorithmic sampling are swamped by the variance embodied in the input probabilities. And yet, in the application of iSee to a single tournament event, we assert that all these sources of "error" are irrelevant to the end user, who is really only looking for interesting snippets of information that form common ground for social interaction. In fact, in some cases it is the very instability of the placement numbers – for example, going from a long shot for first place to a lock with a sequence of unlikely wins – that makes things fun.

There is a crucial exception to this premise. If iSee returns a 0% placement probability, the system is guaranteeing that a particular placement is impossible. This is eminently falsifiable: if ever a player placement that iSee at some point labeled impossible actually does occur, the user will have no reason to trust any of the output numbers again. This is an inherent weakness of sampling, since there is no way to distinguish a true impossibility from an incidence probability arbitrarily close to zero. Unfortunately, the distinction matters a great deal to the user. Players want to know when to root for a particularly unlikely set of teams and outcomes that represent their last hope of a particular placement, or whether instead to truly give up on that placement and focus on another goal (such as finishing in the next highest placement, or besting a particular rival player). This same issue applies equally to placements projected by iSee at 100%, in which iSee is guaranteeing a particular placement.

Since we are aiming to increase fun and not necessarily to optimize mathematical accuracy, we deal with this problem by only reporting approximations (for example, "<1%" and ">99%") until the tournament is small enough that we can perform exhaustive enumeration. For example, when about 15 games remain, representing 32,768 unique outcomes, exhaustive calculations return in under 500 ms on a standard 2009 desktop PC, making it tractable to produce guaranteed results in interactive time. In addition, it is useful to note that the larger the outcome space, the fewer the occurrences of true lock-ins and lock-outs, and the less the players are focused on specific end-placement scenarios. This

minimizes the effect sampling uncertainty has on game play in the early stages of competition, precisely when sampling is required.

5.2 Key Scenarios

iSee also opportunistically explores various game constraints and helps players find key scenarios. The key scenarios are generated by extending the placement accumulator matrix described earlier to track sub-accumulations of various specific scenarios. For instance, consider the following example, where we are interested in the effect an upcoming game might have on Beth's pool performance. If we are interested in the effect Game 1 (between China and Japan) might have on Beth's chances of coming in first, we need to correlate the two different possible outcomes of Game 1 with the outcomes in which Beth comes in 1st place. We do this by reserving space for four additional accumulators:

- A_{C1}, the number of outcomes where China wins Game 1;
- A_{J1}, the number of outcomes where Japan wins Game 1;
- $A_{B1 \cap C1}$, the number of outcomes where China wins Game 1 and Beth comes in 1st place;
- $A_{B1 \cap J1}$, the number of outcomes where Japan wins Game 1 and Beth comes in 1st place

On each generated sample, in addition to accumulating the result in the overall player placement matrix, we also accumulate the result in any additional accumulators that match the sampled outcome. In the default accumulator, assume that among 10,000 samples, Beth comes in first in 2,000 of them, leading to an overall 1st place placement projection of 20%. Further assume that among those same random samples, in Game 1 China wins 5,400 times, and Japan wins the remaining 4,600 times, because according to the team vs. team prior probabilities, China is slightly favored (54%). Finally, among the sampled outcomes, assume that when China wins, Beth comes in 1st 270 times, and when Japan wins Beth comes in 1st 1,730 times. Applying the rules of conditional probability as shown in Figure 5, we can determine that Beth has only a 5% chance of coming in 1st when China wins Game 1, but has a 37.6% chance of coming in 1st when Japan wins. This is a significant swing, and indicates that the outcome of Game 1 is important to Beth. She would definitely want to watch the game, and should root strongly for Japan to beat China.

The key scenario generation module maintains many such conditional probability accumulators during each sampling run. At the end of the run, iSee scans these sub-accumulators, analyzing the

$$P(Beth\ 1^{st} \mid China\ Win) = \frac{P(Beth\ 1^{st} \cap China\ Win)}{P(China\ Win)}$$

$$P(Beth\ 1^{st} \cap China\ Win) = \frac{A_{B1 \cap C1}}{S_N}$$

$$P(China\ Win) = \frac{A_{C1}}{S_N}$$

$$P(Beth\ 1^{st} \mid China\ Win) = \frac{270/10000}{5400/10000} = \mathbf{0.050}$$

$$P(Beth\ 1^{st} \mid Japan\ Win) = \frac{1730/10000}{4600/10000} = \mathbf{0.376}$$

Figure 6. The calculation to determine the effect of Game 1 on Beth's chances of coming in first. S_N is the total number of samples generated.

relative significance of each swing. Swings of less than a few percent are considered insignificant and are not reported. Lock-ins and lock-outs (where the conditional probability jumps to 100% or drops to 0% with a given outcome) are considered extremely important and are reported with top priority. The remaining conditional dependencies are ranked by the size of the swing.

In theory, there is no limit to the complexities and quantities of sub-accumulators iSee could track in order to locate important scenarios. In practice, there are several strategies iSee uses to trim the space of possibilities. One overall premise is that players care most about the conditional probabilities related to their highest possible placement. In other words, any given player is more interested in a swing in their 1st place chances than in their 3rd place chances. Another strategy iSee uses is to maintain accumulators for the most imminent games in the tournament: this gives users simpler scenario statements, because there are only two possible winners for each game about to be played, and it helps users decide which games to prioritize watching and rooting for in a tournament with many simultaneous or near-simultaneous competitions. Finally, iSee maintains a team-centric matrix of accumulators to spot long, dependent chains of outcomes that collectively have a simple team-centric summarization. For instance, early on in the tournament, iSee may generate the key scenario, "If Japan wins the championship, Chuck cannot win the pool." This collapses a number of different game-based scenarios into a simple, powerful description of an important scenario for Chuck.

6. PILOT STUDY

We conducted a pilot study to solicit early feedback on the information that the iSee system generates. We recruited players from five existing leagues during the 2009 "March Madness" NCAA basketball championship. These leagues were set up and hosted on a variety of online portals, namely FoxSports, Yahoo! Sports, and ESPN. The leagues had seven, eight, nine, ten, and eighteen players respectively, for a total of 52 players. These players reported varying levels of experience with fantasy games as well as engagement with college basketball. For example, some players reported filling out a single bracket for the first time, while others filled out multiple (up to 19) this year and had done so for the last 15 years. Self-report of a representative subset of our players showed that they had average age of about 28 years old, which is relatively consistent with the demographic we would expect for such games.

Before the tournament began, players submitted screenshots of their picks, which were converted and entered into the iSee system. Players received periodic e-mail updates at eight points during the course of the tournament: after round 1, and then midway and at the end of each subsequent round, all the way through the championship game. Each of these updates was generated using the iSee system, and customized for the league. Each included a bar graph with placement probabilities of players, as well as a manually selected subset of textual factoids such as key games, key inter-player match-ups, as well as player trends and commentary. At the end of the tournament, we distributed a short survey with semi-structured questions aimed at soliciting general feedback. 15 players completed the survey.

We saw a number of very consistent comments during the course of the tournament as well as in the survey, both in terms of the most engaging parts of the updates, but also expressing possible places for improvement. In general, players enjoyed having the additional information during the tracking part of the tournament

and their comments supported our hypotheses. It seems to have allowed players to target which games they watched or monitored. One player mentioned that "the what-if scenarios allowed me to focus on games that were more important than other games that had less impact on the outcome." It also seemed to fuel discussion and social interaction within the leagues. For example, one player commented that iSee "brought on comparisons between my picks and those who would lose out at my expense" and another that iSee caused them to "pay more attention and get in more arguments/discussions with the other players."

One set of comments hints at the possible negative effects of having more certainty about how one was doing or projected to do within the league. One player that was locked out of 1st early in the tournament expressed that "it would be nice if there was something to keep me interested even after I knew I wasn't going to win." Without the system, this player would likely not have inferred that they were locked out, and would have kept (false) hope. Similarly, another player "had a pretty good lead [current score] for a long time, but the updates kept saying that my chances of winning were low." This player indeed ended up finishing in the bottom half of their league, and it was not clear to us how having this information upfront affected their experience. That said, the reverse also happened, and players who did not do well early on found it "interesting to see that even though people were low in the standings, their chances were still high." iSee "gave everyone that was still mathematically in it hope." We intend to explore these effects more fully in future work.

Another suggestion that was fairly widely expressed was a request for different kinds of information such as "pitting player against player." We believe that this could be a trivial extension by using simple heuristics to infer which subset of players for which to rerun statistics. This would also be much less of an issue if players had direct interactive control of iSee rather than viewing it through an e-mail message. Other players requested knowing "which of my picks went against the rest of the pool and either worked out for me or screwed me over." We believe that this is a statistic that could be calculated and will include this in the next version of iSee.

7. CONCLUSION AND FUTURE WORK

Modeling the outcome possibilities of a fantasy pick'em pool based on tournament play is a difficult and unintuitive task even in the simplest cases. In the common case, it is impossible to do manually and offers no analytical solution. Here we have presented iSee, an interface coupled with a calculation engine for both automatically and interactively exploring the outcome space. We believe iSee can increase player enjoyment by providing real-time results projections and by offering players easily understandable scenarios that affect their standings. To adequately validate the benefits of our system, we are currently working toward a larger-scale deployment to understand its effects on pick'em pool play. We also plan to work on ways to extend the computational methods described in this paper to a wider set of fantasy games.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] British Broadcasting Corporation Celebdaq, http://www.bbc.co.uk/celebdaq/

[2] Chen, Y., Goel, S., and Pennock, D. M. (2008). Pricing combinatorial markets for tournaments. *Proc. of the 40th Annual ACM Symposium on Theory of Computing*, pp. 305-314.

[3] Christopher, A. and Shneiderman, B. (1994). Visual Information Seeking: Tight coupling of dynamic query filters with starfield displays. *Proc. of ACM CHI 94*, pp. 313-317.

[4] comScore, "Start of Premiership Season Sets Off Stampede of Traffic to Fantasy Football Sites," Press Release, October 22, 2007.

[5] Fantasy Sports from Wikipedia, http://en.wikipedia.org/wiki/Fantasy_sports, accessed on June 18, 2009.

[6] Fantasy Politics USA, http://fantasypoliticsusa.com/, accessed on June 18, 2009.

[7] Fantasy Sports Trade Association, "Fantasy Sports Conference Demographic Survey Shows Continued Growth," Press Release, August 2, 2007.

[8] Fantasy Sports Trade Association, "Fantasy Sports Conference to Unveil Demographic Survey and Discuss Legal Battle," Press Release, March 11, 2006.

[9] Hand, D., Mannila, H., and Smyth, P. (2001). Principles of Data Mining. MIT Press, Cambridge, MA.

[10] Hollywood Stock Exchange, http://www.hsx.com/

[11] Kaplan, E.H., and Garstka, S.J. (2001). March Madness and the Office Pool. *Management Science*, Vol. 47, No. 3, pp. 369-382.

[12] Ravaja, N., Saari, T., Turpeinen M., Laarni, J., Salminen, M., and Kivikangas, M. (2006). Spatial Presence and Emotions during Video Game Playing: Does It Matter with Whom You Play? *Presence: Teleoperators and Virtual Environments*, Vol. 15, No. 4, pp. 381-392.

[13] Scarf, P.A. and Shi, X. (2008). The importance of a match in a tournament. *Computers and Operations Research*, Vol. 35, No. 7, pp. 2406-2418.

[14] Schwarz, A. (2004). The Numbers Game: Baseball's Lifelong Fascination with Statistics, Thomas Dunne Books.

[15] Shneiderman, B. (1987). Direct manipulation: a step beyond programming languages. *IEEE Computer*, Vol. 16, No. 8, pp. 57-69.

[16] Spotfire, http://spotfire.tibco.com/

[17] Tan, D., Smith, G., Lee, B., and Robertson, G. (2007). AdaptiviTree: Adaptive Tree Visualization for Tournament-Style Brackets, *IEEE TVCG (InfoVis 2007)*, Vol. 13, No. 6, pp. 1113-1120.

[18] Westphal, C. and Blaxton, T. (1998). Data Mining Solutions: Methods and Tools for Solving Real-World Problems, Wiley Computer Publishing.