# TeamSkill Evolved: Mixed Classification Schemes for Team-Based Multi-player Games

Colin DeLong and Jaideep Srivastava

Department of Computer Science,
University of Minnesota
{delong,srivasta}@cs.umn.edu
http://www.cs.umn.edu

**Abstract.** In this paper, we introduce several approaches for maintaining weights over the aggregate skill ratings of subgroups of teams during the skill assessment process and extend our earlier work in this area to include game-specific performance measures as features alongside aggregate skill ratings as part of the online prediction task. We find that the inclusion of these game-specific measures do not improve prediction accuracy in the general case, but *do* when competing teams are considered evenly matched. As such, we develop a "mixed" classification method called TeamSkill-EVMixed which selects a classifier based on a threshold determined by the prior probability of one team defeating another. This mixed classification method outperforms all previous approaches in most evaluation settings and particularly so in tournament environments. We also find that TeamSkill-EVMixed's ability to perform well in close games is especially useful early on in the rating process where little game history is available.

**Keywords:** Player rating systems, competitive gaming, perceptron, passive aggressive algorithm, confidence-weighted learning.

## 1 Introduction

In games, the challenge of ascertaining one player or team's advantage over their opponents continues to be an open research problem. In particular, the rise of online multi-player games has put the task of skill assessment front and center for game developers, wherein the long-term success or failure of a title is linked, in part, to the ability of players to find similarly-skilled teammates and opponents to play against. "Matchmaking", an automated process used to match players together for an online game, depends on accurate estimations of player skill at all times in order to reduce the likelihood of imbalanced matches. If one player or team is far superior to their opposition, the resulting game can frustrate less-skilled players and potentially lead to customer churn.

For games which focus on the online multi-player experience, including popular titles such as Halo, Call of Duty, and StarCraft 2, the task of appropriately matching up *millions* of players and teams of roughly equal skill is crucial - and

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 26–37, 2012.

daunting. With such large player populations, batch learning methods become impractical, neccesitating an online skill assessment process in which adjustments to a player's skill rating happen one game at a time, depending only on their existing rating and the outcome of the game. This task is made more difficult in titles centered around team-based competition, where interaction effects between teammates can be difficult to model and integrate into the assessment process.

Our work is concerned with this particular variant of the skill estimation problem. Although many approaches exist for skill estimation, such as the well-known Elo rating system [1] and the Glicko rating system [2], [3], they were primarily designed for one versus one competition settings (in games such as Chess or tennis) instead of team-based play. They can be altered to accomodate competitions involving teams, but, problematically, assume the performances of players in teams are independent from one another, thereby excluding potentially useful information regarding a team's collective "chemistry". More recent approaches [4] have explicitly modeled teams, but still assume player independence within teams, summing individual player ratings to produce an overall team rating.

"Team chemistry" is a widely-held notion in team sports [5] and is often cited as a key differentiating factor, particularly at the highest levels of competition. In the context of skill assessment in an online setting, however, less attention has been given to situations in which team chemistry would be expected to play a significant role, such as the case where the player population is highly-skilled individually, instead using data from a general population of players for evaluation [4].

Our previous work in this area [6] described several methods for capturing elements of "team chemistry" in the assessment process by maintaining skill ratings for subsets of teams as well as individuals, aggregating these ratings together for an overall team skill rating. One of the methods, TeamSkill-AllK-EV (hereafter referred to as EV), performed especially well in our evaluation. One drawback of EV, however, was that it weighted each aggregate n-sized subgroup skill rating uniformly in the final summation, leaving open the possibility that further improvements might be made through an adaptive weighting process.

In this paper, we build on our previous work by introducing five algorithms which address this drawback in various ways, TeamSkill-AllK-Ev-OL1 (OL1), TeamSkill-AllK-Ev-OL2 (OL2), TeamSkill-AllK-Ev-OL3 (OL3), TeamSkill-AllK-EVGen (EVGen), and TeamSkill-AllK-EVMixed (EVMixed). The first three - OL1, OL2, and OL3 - employ adaptive weighting frameworks to adjust the summation weights for each n-sized group skill rating and limit their feature set to data common across all team games: the players, team assignments, and the outcome of the game. For EVGen and EVMixed, however, we explore the use of EV's final prediction, the label of the winning team, as a feature to be included along with a set of game-specific performance metrics in a variety of online classification settings [7], [8], [9]. For EVMixed, a threshold based on EV's prior probability of one team defeating another is used to determine whether or not to include the metrics as features and, if not, the algorithm defers to

EV's predicted label. EVGen, in contrast, always includes the metrics during classification.

Evaluation is carried out on a carefully-compiled dataset consisting of tournament and scrimmage games between professional Halo 3 teams over the course of two years. Halo 3 is a first-person shooter (FPS) game which was played competitively in Major League Gaming (MLG), the largest professional video game league in the world, from 2008 through 2010. With MLG tournaments regularly featuring 250+ Halo teams vying for top placings, heavy emphasis is placed on teamwork, making this dataset ideal for the evaluation of interaction effects among teammates.

We find that EVMixed outperforms all other approaches in most cases, often by a significant margin. It performs particularly well in cases of limited game history and in "close" games where teams are almost evenly-matched. These results suggest that while game-specific features can play a role in skill assessment, their utility is limited to contexts in which the skill ratings of teams are similar. When they are not, the inclusion of game-specific information effectively adds noise to the dataset since their values aren't conditioned on the strength of their opponents.

The outline of this paper follows. Section 2 briefly describes some of the work related to the problem of skill assessment. In Section 3, we introduce our proposed approaches - OL1, OL2, OL3, EVGen, and EVMixed. In Section 4, we describe some of the key features of the dataset, our evaluation testbed, and share the results of our evaluation in terms of game outcome prediction accuracy. We then conclude with Section 5, discussing the results and future work.

## 2 Related Work

The foundations of modern skill assessment approaches date back to the work of Louis Leon Thurstone [10] who, in 1927, proposed the "law of comparitive judgement", a method by which the mean distance between two physical stimuli, such as perceived loudness, can be computed in terms of the standard deviation when the stimuli processes are normally-distributed. In 1952, Bradley-Terry-Luce (BTL) models [11] introduced a logistic variant of Thurstone's model, using taste preference measurements for evaluation. This work in turn led to the creation of the Elo rating system, introduced by Arpad Elo in 1959 [1], a professor and master chess player who sought to replace the US Chess Federation's Harkness rating system with one more theoretically sound. Similar to Thurstone, the Elo rating system assumes the process underlying each player's skill is normally-distributed with a constant skill variance parameter $\beta^2$ across all players, simplifying skill updates after each game.

However, this simplification was also Elo's biggest drawback since the "reliabilty" of a rating was unknown from player to player. To address this, the Glicko rating system [3], a Bayesian approach introduced in 1993 by Mark Glickman, allowed for player-specific skill variance, making it possible to determine the

confidence in a player's rating over time and produce more conservative skill estimates.

With the release of the online gaming service Xbox Live in 2002, whose player population quickly grew into the millions, there was a need for a more generalized rating system incorporating the notion of teams as well as individual players. TrueSkill [4], published in 2006 by Ralf Herbrich and Thore Graepel, used a factor graph-based approach to meet this need. In TrueSkill, skill variance is also maintained for each player, but in contrast to Glicko, TrueSkill samples an expected performance given a player's skill rating which is then summed over all members of a team to produce an estimate of the collective skill of a team. Because the summation is over individual players, player performances are assumed to be independent from one another, leaving out potentially useful group-level interaction information. For team-based games in which highly-skilled players may coordinate their strategies, this lost interaction information can make the estimation of a team's advantage over another difficult, especially as players change teams.

Several other variants of the aforementioned approaches have also been introduced, including BTL models [12], [13], [14] and expectation propagation techniques for the analysis of paired comparison data [15].

## 3 Proposed Approaches

In our previous work [6], we sought to explicitly model group-level interaction effects during the skill assessment process, introducing four methods which took varying approaches to addressing this issue - TeamSkill-K, TeamSkill-AllK, TeamSkill-AllK-EV, and TeamSkill-AllK-LS. These approaches had in common the idea that ratings themselves need not be limited to individual players, but subsets of teams as well. Here, we modified the Elo, Glicko, and TrueSkill rating systems to be used as generic learners which maintained skill ratings for groups of players. In doing so, both group and player-level skill could be captured, producing a clearer picture of a team's collective skill. The key differences between these approaches was the amount of subgroup rating information used and the ways in which aggregate group skill ratings were weighted during the summation to produce a team's skill rating.

One of the approaches, EV, performed especially well during evaluation, improving on the unaltered versions of Glicko and TrueSkill, and, in most test cases, the other TeamSkill approaches as well. The main idea behind EV is to use all available group-level history, from groups of size $k = 1$ (individual players) to $k = K$ (the size of the team), and sum together the expected skill rating corresponding to each set of $k$-sized group ratings, weighting each uniformly:

$$s_i^* = \frac{K}{\sum_{k=1}^{K} (|h_i(k)| > 0)} \sum_{k=1}^{K} \frac{E[h_i(k)]}{k} \tag{3.1}$$

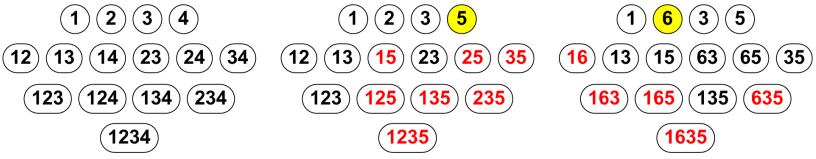


**Fig. 1.** The group history problem. This figure illustrates the group history available for a team of four players at three different time instances, proceeding chronologically from left to right. Black font indicates that history is available for a given group while red font indicates that history is not available.

In this notation, $s_i^*$ is the estimated skill of team $i$ and $h_i(k)$ is a function returning the set of skill ratings for player groups of size $k$ in team $i$, including the empty set $\emptyset$ if none exist. When $h_i(k) \to \emptyset$, we let $E[h_i(k)] = 0$.

Despite its excellent results, EV is a "naive" approach, lacking a means of updating the summation weights, potentially leading to suboptimal performance. To that end, we introduce three adaptive frameworks which allow the summation weights to vary over time - TeamSkill-AllK-Ev-OL1 (OL1), TeamSkill-AllK-Ev-OL2 (OL2), and TeamSkill-AllK-Ev-OL3 (OL3).

### 3.1 TeamSkill-AllK-Ev-OL1

When attempting to construct an overall team skill rating, one key challenge to overcome is the fact that the amount of group history can vary over time. Consider figure 1: after the first game is played, history is available for all possible groups of players. Later, player 4 leaves the team and is replaced by player 5, who has never played with players 1, 2, or 3, leaving only a subset of history available and none for the team as a whole. Then in the final step, player 2 leaves and is replaced by player 6, who has played with player 3 and 5 before, but never both on the same team, resulting in yet another variant of the team's collective group-level history. The feature space is constantly expanding and contracting over time, making it difficult to know how best to combine the group-level ratings together. In OL1, we address this issue by maintaining a weight $w_k$ for each aggregate group skill rating of size $k$, contracting $\mathbf{w}$ during summation by uniformly redistributing the weights from indicies in the weight vector not present in the available aggregate group skill rating history. Given the winning team $i$, $w_k$ is updated by computing to what extent each of the aggregate rating's prior probability of team $i$ defeating some team $j$ according to TeamSkill-K [6], $P_k(i > j)$, is better than random, increasing the weight of $w_k$ for a correctly-predicted outcome.

$$1 \leq \beta \leq \infty, w_k^0 = \frac{1}{K}, K' = \min\left(\max_{k \leq K}(|h_i(k)| > 0), \max_{k \leq K}(|h_j(k)| > 0)\right) \tag{3.2}$$

$$u = \frac{1}{K'} \sum_{k > K'} w_k^t \tag{3.3}$$

$$w'^t_{(k \leq K')} = w^t_{(k \leq K')} + u \tag{3.4}$$

$$s_i^* = \sum_{k=1}^{K'} w'^t_k E[h_i(k)] \tag{3.5}$$

$$w^{t+1}_{(k \leq K')} = w^t_{(k \leq K')} \beta^{\frac{1}{2} + P_k(i>j)} \tag{3.6}$$

$$w_k^{t+1} = \frac{w_k^{t+1}}{\sum_{l=1}^{K} w_l^{t+1}} \tag{3.7}$$

The main drawback of this approach is that the weight for $k = 1$ eventually dominates the weight vector as it is the element of group history present in every game and, therefore, the weight most frequently increased relative to the weights of $k > 1$. Given enough game history, this classifier will converge to exactly $k = 1$ - the classifier corresponding the an unmodified version of the general learner (Elo, Glicko, or TrueSkill) it employs.

### 3.2 TeamSkill-AllK-Ev-OL2

OL2 attempts to remedy this by maintaining a weight matrix corresponding to the lower triangular of a $K$x$K$ grid, or one weight vector $\mathbf{w}$ for each of the $K$ possible summation situations given a team's group-level game history. This ameliorates the issue of the $k = 1$ weight increasing faster relative to the weights of $k > 1$ since each row in the $K$x$K$ grid pertains to a situation where the length of the non-zero row elements equals $K'$ (as defined previously).

$$s_i^* = \sum_{k=1}^{K'} w^t_{(K',k)} E[h_i(k)] \tag{3.8}$$

$$w^{t+1}_{(K',k \leq K')} = w^t_{(K',k \leq K')} \beta^{\frac{1}{2} + P_k(i>j)} \tag{3.9}$$

$$w^{t+1}_{(K',k)} = \frac{w^{t+1}_{(K',k)}}{\sum_{l=1}^{K'} w^{t+1}_{(K',l)}} \tag{3.10}$$

### 3.3 TeamSkill-AllK-Ev-OL3

OL3 works similarly to OL1 in most respects, but instead uses a predefined window of the $d$ most recent games in which $k$-sized group history was available to compute its updates. In this way, the weights "follow" the most confidently-correct aggregate skill ratings for each window $d$. In the following, let $L_{d,k}$ be the

number of games in the window $d$ in which, for some $k$, TeamSkill-K incorrectly predicted the outcome of a game.

$$s_i^* = \sum_{k=1}^{K'} w_k'^t E[h_i(k)] \tag{3.11}$$

$$w_{(k \leq K')}^{t+1} = w_{(k \leq K')}^{t} \beta^{\frac{1}{2}+(d-L_{d,k})/d} \tag{3.12}$$

$$w_k^{t+1} = \frac{w_k^{t+1}}{\sum_{l=1}^{K} w_l^{t+1}} \tag{3.13}$$

### 3.4 Using Game-Specific Data during Classification

OL1, OL2, and OL3 - like the other TeamSkill approaches - only use data available in all team-based games, namely the players, their team associations, and game outcome history. One natural question to ask is how well could we do if we included game-*specific* data during the step in which the label of the winning team is predicted. Though not ideal from a general implementation perspective, it is reasonable to assume that a carefully-chosen set of game-specific performance metrics might help produce a more accurate prediction. Here, we introduce two such methods - TeamSkill-AllK-EVGen (EVGen) and TeamSkill-AllK-EVMixed (EVMixed).

### 3.5 TeamSkill-AllK-EVGen

In EVGen, we create a feature set $\mathbf{x}_t$ from a combination of EV's predicted label $\{+1, -1\}$ of the winning team, $\hat{EV}_t$, and a set of $n$ game-specific metrics $\mathbf{m}$. For Halo 3, several logical metrics are available, such as kill/death ratio and assist/death ratio (an assist is given to a player when they do more than half of the damage to a player who is eventually killed by another player), and act as rough measures of a team's in-game efficiency since players respawn after each death throughout the duration of a game. After compiling these metrics for each team, we take the difference between them for use in $\mathbf{x}_t$, adding in $\hat{EV}_t$ as the final feature. EV was chosen because of its superior performance in previous evaluations [6] as well as results from preliminary testing for this work, drawing from the pool of all previous approaches (including OL1, OL2, and OL3).

$$\mathbf{x}_t = (\hat{EV}_t, m_1, m_2, ..., m_n) \tag{3.14}$$

Having constructed the feature set $\mathbf{x}_t$, we use a more traditional online classification framework to predict the label of the winning team $\hat{y}_t$, such as the perceptron [7], online Passive-Aggressive algorithms [8], or Confidence-Weighted learning [9] (Note: substitute $\boldsymbol{\mu}_t$ for $\mathbf{w}_t$ in the latter):

$$\hat{y}_t = sign(\mathbf{w}_t \cdot \mathbf{x}_t) \tag{3.15}$$

After classification, the weight vector over the feature set is then updated according to the chosen learning framework.

### 3.6 TeamSkill-AllK-EVMixed

EVMixed introduces a slight variant to EVGen's overall strategy by selecting a classification approach based on whether or not both teams are considered relatively evenly-matched (that is, if a team's prior probability of winning according to EV, $P^t_{EV}(i > j)$, is close to 0.5). Here, if the prior probability of one team winning is within some $\epsilon$ of 0.5, we use the EVGen model for prediction. Otherwise we simply use EV's label. The approach is simple, as is the intuition behind it: if EV is sufficiently confident in its predicted label, then there is no need for additional feature information.

$$\hat{y}_t = \begin{cases} sign(\mathbf{w}_t \cdot \mathbf{x}_t) & \text{if } |P^t_{EV}(i > j) - 0.5| < \epsilon \\ \hat{EV}_t & \text{otherwise} \end{cases} \tag{3.16}$$

## 4 Evaluation

### 4.1 Dataset

We evaluate our proposed approaches using a dataset of 7,568 Halo 3 multiplayer games between professional teams. Each was played over the Internet on Microsoft's Xbox Live service in custom games (known as scrimmages) or on a local area network at an MLG tournament and includes information such as the players and teams competing, the date of the game, the map and game type, the result (win/loss) and score, and per-player statistics such as kills, deaths, and assists.

Characteristics unique to this dataset make it ideal for our evaluation purposes. First, it is common for players to change teams between tournaments, each of which is held roughly every 1-2 months, thereby allowing us to study the effects of "team chemistry" on performance without the assumption of degraded individual skill. Second, because every player is competing at such a high level, their individual skill isn't considered as important a factor in winning or losing a game as their ability to work together as a team.

### 4.2 Overall Results

The prediction accuracy of OL1, OL2, OL3, EVGen, and EVMixed were evaluated using a number of different subsets of the Halo 3 dataset:

- Games played in tournaments only, scrimmage games only, and both tournament and scrimmage games.
- All of the games, or just those games considered "close" (i.e., prior probability of one team winning close to 50%).

For comparison, we include results from the previous TeamSkill approaches as well. To compute the prior probability of $t_1$ defeating $t_2$, we use the negative CDF evaluated at 0 for the distribution corresponding to the difference between two independent, normally-distributed random variables (as in [6]). Games were labeled as "close" using a variant of the "challenge" method [4] in which the top

**Table 1.** Overall prediction accuracy for all test cases. **Bold cells** = highest accuracy; ***bolded/italicized*** = 2nd-highest accuracy.

| Learner | Data | Close? | k=1 | k=2 | k=3 | k=4 | AllK | AlKEV | AlKLS | OL | OL2 | OL3 | EVGen | EVMxd |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Elo | Both | N | 0.645 | 0.642 | 0.636 | 0.631 | 0.642 | 0.645 | 0.633 | 0.645 | 0.645 | ***0.646*** | 0.574 | **0.647** |
| | | Y | 0.512 | 0.494 | 0.497 | 0.485 | 0.493 | 0.5 | 0.489 | 0.495 | 0.495 | 0.502 | **0.523** | ***0.521*** |
| | Tourn. | N | ***0.639*** | 0.626 | 0.607 | 0.571 | 0.628 | 0.635 | 0.592 | ***0.639*** | ***0.639*** | 0.633 | 0.572 | **0.643** |
| | | Y | 0.518 | 0.497 | 0.482 | 0.464 | 0.5 | 0.51 | 0.474 | 0.531 | 0.536 | 0.51 | **0.549** | ***0.544*** |
| | Scrim. | N | ***0.643*** | 0.639 | 0.639 | 0.631 | 0.642 | 0.64 | 0.633 | ***0.643*** | ***0.643*** | 0.64 | 0.583 | **0.644** |
| | | Y | 0.503 | 0.487 | 0.492 | 0.476 | 0.496 | 0.488 | 0.476 | 0.499 | 0.498 | 0.487 | **0.529** | ***0.512*** |
| Glicko | Both | N | 0.636 | 0.63 | 0.632 | 0.635 | ***0.64*** | ***0.64*** | 0.633 | 0.637 | 0.637 | ***0.64*** | 0.581 | **0.641** |
| | | Y | 0.522 | 0.564 | 0.562 | 0.547 | 0.569 | 0.57 | 0.548 | 0.524 | 0.552 | ***0.571*** | 0.528 | **0.573** |
| | Tourn. | N | 0.638 | 0.637 | 0.616 | 0.588 | 0.644 | ***0.647*** | 0.613 | 0.637 | 0.637 | ***0.647*** | 0.566 | **0.657** |
| | | Y | 0.484 | 0.529 | 0.531 | 0.523 | ***0.576*** | 0.57 | 0.557 | 0.526 | 0.56 | 0.57 | 0.518 | **0.62** |
| | Scrim. | N | 0.631 | 0.635 | 0.637 | 0.637 | **0.643** | 0.637 | 0.634 | 0.635 | 0.636 | 0.637 | 0.582 | ***0.638*** |
| | | Y | 0.496 | 0.559 | ***0.565*** | 0.522 | ***0.562*** | 0.551 | 0.524 | 0.531 | 0.551 | 0.551 | 0.525 | 0.554 |
| TrueSkill | Both | N | 0.635 | 0.641 | 0.636 | 0.63 | 0.638 | 0.642 | 0.632 | 0.635 | 0.636 | **0.643** | 0.572 | **0.643** |
| | | Y | 0.516 | 0.555 | 0.542 | 0.542 | 0.552 | 0.56 | 0.548 | 0.536 | 0.544 | **0.562** | 0.522 | ***0.561*** |
| | Tourn. | N | 0.64 | 0.626 | 0.601 | 0.576 | 0.626 | 0.636 | 0.601 | 0.641 | **0.644** | 0.634 | 0.569 | **0.653** |
| | | Y | 0.5 | 0.497 | 0.479 | 0.474 | 0.508 | 0.51 | 0.495 | 0.531 | ***0.547*** | 0.508 | 0.542 | **0.573** |
| | Scrim. | N | 0.636 | **0.642** | 0.639 | 0.632 | 0.636 | 0.638 | 0.634 | 0.636 | 0.637 | 0.637 | 0.581 | ***0.64*** |
| | | Y | 0.504 | **0.55** | 0.542 | 0.53 | 0.541 | 0.54 | 0.533 | 0.548 | **0.55** | 0.543 | 0.522 | 0.542 |

20% closest games for one rating system are identified and presented to the other. Because we are interested in performance beyond that of unmodified general learners (i.e., $k = 1$), the closest games from $k = 1$ were presented to the other TeamSkill approaches while EV's closest games were presented to $k = 1$ (due to its evaluated performance in [6]). The following defaults were used for Elo ($\alpha = 0.07$, $\beta = 193.4364$, $\mu_0 = 1500$, $\sigma_0^2 = \beta^2$), Glicko ($q = log(10)/400$, $\mu_0 = 1500$, $\sigma_0^2 = 100^2$), and TrueSkill ($\epsilon = 0.5$, $\mu_0 = 25$, $\sigma_0^2 = (\mu_0/3)^2$, $\beta = \sigma_0^2/2$) according to [4] and [3]. For OL1/OL2, $\beta = 1.1$, OL3, $d = 20$. For EVGen/EVMixed ($\epsilon = 0.03$), the Passive-Aggressive II algorithm [8] was used for classification ($\alpha = 0.1$, $C = 0.001$, $\eta = 0.9$). The final feature set was comprised of cumulative and windowed (10 games of history) versions of team differences in average team and player-level kill/death ratio, assist/death ratio, kills/game, and assists/game.

From the results in table 1, it is clear that EVMixed performs the best overall, and in the widest array of evaluation conditions. It has the best performance in 10 of the 18 test cases and 16 of 18 in which it was at least second best, a testament to its consistency. EVGen's overall performance, however, is roughly 7-10% lower on average over all games, exceeding EVMixed's results only in 3 of the "close" game test cases.

### 4.3 Results over Time

Next we explore how these approaches perform over time by predicting the outcomes of games occuring prior to 10 tournaments which took place during 2008 and 2009, using tournament data only in order to isolate conditions in which we expect teamwork to be strongest. From figures 2 and 3, EVMixed's superior peformance is readily apparent. Of particular note, however, is how well EVMixed does when little history is available, having a roughly 64% accuracy just prior to the first tournament for all three learner cases. For close games,

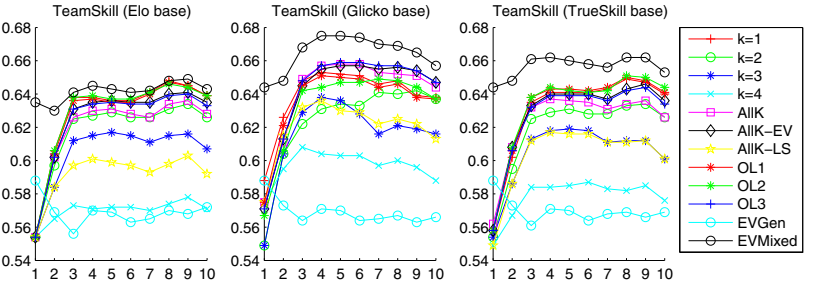


**Fig. 2.** Prediction accuracy over time for tournament games

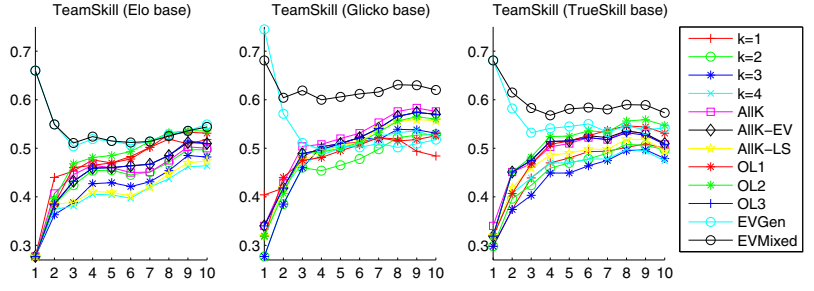


**Fig. 3.** Prediction accuracy over time for tournament games, close games only

both EVGen and EVMixed show strong results, eventually tapering off and approaching the other competing methods as more game history is observed.

### 4.4 Online Classification Variants

For EVGen and EVMixed, we investigated a number of different online classification frameworks - the perceptron [7], Passive-Aggressive algorithms [8], and Confidence-Weighted learning [9] - and evaluated them using a subset of the testbed from section 4.2. The results are shown in table 2. Though similar, the PA-II approach appears to be the most consistent overall (with CW-diag not far behind).

**Table 2.** Comparison of prediction accuracy by online classification framework using Glicko as the general learner

| | | EVGen | | | EVMixed | | |
|---|---|---|---|---|---|---|---|
| Data | Close? | Perceptron | PA-II | CW-diag | Perceptron | PA-II | CW-diag |
| Both | N | 0.575 | ***0.581*** | **0.584** | **0.641** | **0.641** | **0.641** |
| | Y | ***0.514*** | **0.528** | **0.528** | **0.573** | **0.573** | **0.573** |
| Tourn. | N | 0.543 | **0.566** | ***0.564*** | ***0.655*** | **0.657** | **0.657** |
| | Y | 0.474 | **0.518** | ***0.51*** | 0.609 | **0.62** | ***0.617*** |
| Scrim. | N | 0.575 | ***0.582*** | **0.586** | **0.638** | **0.638** | ***0.637*** |
| | Y | ***0.515*** | **0.525** | 0.512 | **0.556** | ***0.554*** | 0.551 |

## 5 Discussion

In sum, the results show EVMixed consistently outperforming competing approaches in a multitude of scenarios, often by great margins. Initially, we found the subpar performance of EVGen somewhat surprising given that the only difference between it and EVMixed is the classifier choice according to a given $\epsilon$. Upon closer examination, the reason for this discrepency becomes clear: the game-specific data used to supplement the feature set was *not* weighted according to the strength of their opposition in each game, effectively adding "noise" in cases where the games were not considered close. Only the skill rating is a function of opposition skill, and as such, when the ratings of two teams are sufficiently divergent, the additional features are not necessary, nor desired. It follows that this is also the reason why both EVGen and EVMixed perform well in close games. Here, because the difference in skill ratings is small, the supplemental feature information tells us something about how two otherwise evenly-matched teams might perform if they competed. This is also why EVGen and EVMixed have excellent results when little game history has been observed - nearly all games are considered "close" early in the rating process.

Turning our attention back to OL1, OL2, and OL3, it's clear that little improvement was made relative to EV's results for any of these approaches. In fact, while the weights for OL1 eventually converge to the classifier $k = 1$, OL2's weights largely mimic EV's, suggesting there are more subtle group-level dynamics we need to pay attention to as this would only arise if the classifiers corresponding to $1 \leq k \leq K$ have somewhat similar ratings. OL3 also produces results similar to EV (even moreso than OL2), adding to the previous observation. While the results for OL1, OL2, and OL3 are unfortunate, the naive means by which EV weights each of the aggregated group-level skill ratings leaves the door open for improvement.

Our future work takes two directions. The first is to more fully explore what can be done to enhance the EVMixed model, perhaps by introducing a mechanism by which $\epsilon$ can vary over time or weighting player performances in-game by the strength of their opponents. The second is to derive an adaptive weighting framework which does improve on EV's results significantly, and then integrate it into EVGen and EVMixed.

## 6 Conclusions

In this paper, we extended our previous work by introducing three methods in which various strategies are used to maintain a set of weights over aggregate group-level skill rating information. Additionally, we explored the utility of incorporating game-specific data as features during the prediction process, describing two such approaches: EVGen and EVMixed. EVMixed outperformed all previous efforts in the vast majority of cases, leading to the conclusion that game-specific data is best included when teams are relatively evenly-matched, and disregarded otherwise.

**Acknowledgments.** We would like to thank members of the Data Mining Research Group for their feedback and suggestions. We would also like to thank Major League Gaming for making their 2008-2009 tournament data available.

## References

1. Elo, A.: The Rating of Chess Players, Past and Present. Arco Publishing, New York (1978)
2. Glickman, M.: Paired Comparison Model with Time-Varying Parameters. PhD thesis. Harvard University, Cambridge, Massachusetts (1993)
3. Glickman, M.: Parameter estimation in large dynamic paired comparison experiments. Applied Statistics 48, 377–394 (1999)
4. Herbrich, R., Graepel, T.: Trueskill: A bayesian skill rating system. Microsoft Research, Tech. Rep. MSR-TR-2006-80 (2006)
5. Yukelson, D.: Principles of effective team building interventions in sport: A direct services approach at penn state university. Journal of Applied Sport Psychology 9(1), 73–96 (1997)
6. DeLong, C., Pathak, N., Erickson, K., Perrino, E., Shim, K., Srivastava, J.: TeamSkill: Modeling Team Chemistry in Online Multi-player Games. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 519–531. Springer, Heidelberg (2011)
7. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review 65(6), 386–408 (1958)
8. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
9. Crammer, K., Dredze, M., Pereira, F.: Exact convex confidence-weighted learning. In: Advances in Neural Information Processing Systems, vol. 21, pp. 345–352 (2009)
10. Thurstone, L.: Psychophysical analysis. American Journal of Psychology 38, 368–389 (1927)
11. Bradley, R.A., Terry, M.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika 39(3/4), 324–345 (1952)
12. Coulom, R.: Whole-History Rating: A Bayesian Rating System for Players of Time-Varying Strength. In: van den Herik, H.J., Xu, X., Ma, Z., Winands, M.H.M. (eds.) CG 2008. LNCS, vol. 5131, pp. 113–124. Springer, Heidelberg (2008)
13. Huang, T., Lin, C., Weng, R.: Ranking individuals by group comparisons. Journal of Machine Learning Research 9, 2187–2216 (2008)
14. Menke, J.E., Reese, C.S., Martinez, T.R.: Hierarchical models for estimating individual ratings from group competitions. American Statistical Association (2007) (in preparation)
15. Birlutiu, A., Heskes, T.: Expectation Propagation for Rating Players in Sports Competitions. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 374–381. Springer, Heidelberg (2007)