Noname manuscript No.
(will be inserted by the editor)

Bayesian Score-Based Skill Learning

Shengbo Guo · Scott Sanner
Thore Graepel · Wray Buntine

Received: date / Accepted: date

Abstract In this paper, we extend the Bayesian skill rating system of TrueSkill to accommodate score-based match outcomes. TrueSkill has proven to be a very effective algorithm for matchmaking — the process of pairing competitors based on similar skill-level — in competitive online gaming. However, for the case of two teams/players, TrueSkill only learns from win, lose, or draw outcomes and cannot use additional match outcome information such as scores. To address this deficiency, we propose novel Bayesian graphical models based on the Gaussian and Poisson likelihood as extensions of TrueSkill that (1) model player's offence and defence skills separately and (2) model how these offence and defence skills interact to generate score-based match outcomes. We derive efficient Bayesian inference methods for inferring latent skills in the Gaussian models, and fast fixed-point iteration method for the variational inference in the Poisson model, which is more than 20 times faster than a sampling method. We evaluate them on three real data sets including Halo 2 XBox Live matches. Empirical evaluations demonstrate that the new score-based models (a) provide more accurate win/loss probability estimates than TrueSkill when training data is limited, (b) provide competitive and often better win/loss (binary) and win/loss/draw (multi-class) classification perfor-

Shengbo Guo
Xerox Research Centre Europe
E-mail: shengbo.guo@xrce.xerox.com

Scott Sanner
NICTA and ANU
E-mail: scott.sanner@nicta.com.au

Thore Graepel
Microsoft Research Cambridge
E-mail: thore.graepel@microsoft.com

Wray Buntine
NICTA and ANU
E-mail: wray.buntine@nicta.com.au

mance than TrueSkill, and (c) provide reasonable score outcome predictions with an appropriate choice of likelihood — prediction for which TrueSkill was not designed, but which can be useful in many applications.

Keywords variational inference, matchmaking, graphical models

1 Introduction

In online gaming, it is important to pair players or teams of players so as to optimize their gaming experience. Game players often expect competitors with comparable skills for the most enjoyable experience; match experience can be compromised if one side consistently outperforms the other. *Matchmaking* attempts to pair players such that match results are close to being even or a draw. Hence, a prerequisite for good matchmaking is the ability to predict future match results correctly from historical match outcomes — a task that is often cast in terms of latent skill learning.

TrueSkill [10] is a state-of-the-art Bayesian skill learning system: it has been deployed in the Microsoft Xbox 360 online gaming system for both matchmaking and player ranking. For the case of two teams/players, TrueSkill, like Elo [9], is restricted to learn skills from match outcomes in terms of win, lose, or draw (WLD). While we conjecture that TrueSkill discards potentially valuable skill information carried by score-based outcomes, there are at least two arguments in favour of TrueSkill's WLD-based skill learning approach:

- WLD-based systems can be applied to any game whose outcome space is WLD, no matter what the underlying scoring system is.
- In many games, the objective is not to win by the highest score differential, but rather simply to win. In this case, it can be said that TrueSkill's skill modeling and learning from WLD outcomes aligns well with the players' underlying objective.

On the other hand, we note that discarding score results ignores two important sources of information:

- High (or low) score differentials can provide insight into relative team strengths.
- Two dimensional score outcomes (i.e., a score for each side) provide a direct basis for inferring separate offense and defense strengths for each team, hence permitting finer-grained modeling of performance against future opponents.

In this work, we augment the TrueSkill model of WLD skill learning to learn from score-based outcomes. We explore single skill models as well as separate offense/defense skill models made possible via score-based modeling. We also investigate both Gaussian and Poisson score likelihood models, deriving a novel variational update for approximate Bayesian inference in the latter case. We evaluate these novel Bayesian score-based skill-learning models in comparison to TrueSkill (for WLD outcomes) on three datasets: 14 years

of match outcomes for the UK Premier League, 11 years of match outcomes for the Australian Football (Rugby) League (AFL), and three days covering 6,000+ online match outcomes in the Halo 2 XBox video game. Empirical evaluations demonstrate that the new score-based models (a) provide more accurate win/loss probability estimates than TrueSkill (in terms of information gain) with limited amounts of training data, (b) provide competitive and often better win/loss (binary) and win/loss/draw (multi-class) classification performance than TrueSkill in terms of area under the curve and the Brier score [2], and (c) provide reasonably accurate score predictions with an appropriate likelihood — prediction for which TrueSkill was not designed but important in cases such as tournaments that rank (or break ties) by points, professional sports betting and bookmaking, and game-play strategy decisions that are dependent on final score projections.

2 Skill Learning using TrueSkill

Since our score-based Bayesian skill learning contributions build on TrueSkill [10], we begin with a review of the TrueSkill Bayesian skill-learning graphical model for two single-player teams. We note that TrueSkill itself allows for matches involving more than two teams and learning team members' individual performances, but these extensions are not needed for the application domains considered in the paper.

Suppose there are n teams available for pairwise matches in a game. Let $M = \{i, j\}$ specify the two teams participating in a match and define the outcome $o \in \{\textit{team-i-win}, \textit{team-j-win}, \textit{draw}\}$. TrueSkill models the probability $p(o|\mathbf{l}, M)$ of o given the skill level vector $\mathbf{l} \in \mathbb{R}^n$ of the teams in M, and estimates posterior distributions of skill levels according to Bayes' rule

$$p(\mathbf{l}|o, M) \propto p(o|\mathbf{l}, M)p(\mathbf{l}), \tag{1}$$

where a factorising Gaussian prior is assumed:

$$p(\mathbf{l}) := \prod_{i=1}^{n} \mathcal{N}(l_i; \mu_i, \sigma_i^2). \tag{2}$$

To model the likelihood $p(o|\mathbf{l}, M)$, each team i is assumed to exhibit a stochastic performance variable $p_i \sim \mathcal{N}(p_i; l_i, \beta^2)$ in the game [1]. From this we can model the performance differential d as an indicator function $p(d|\mathbf{p}, M) = \delta(d = p_i - p_j)$ and finally the probability of each outcome o given this differential d:

$$p(o|d) = \begin{cases} o = \textit{team-i-win}: & \mathbb{I}[d > \epsilon] \\ o = \textit{team-j-win}: & \mathbb{I}[d < -\epsilon] \\ o = \textit{draw}: & \mathbb{I}[|d| \leq \epsilon], \end{cases} \tag{3}$$

[1] Note that we sometimes abuse notations on the use of p, p_i and $\mathbf{p}$. p is a probability measure; p_i and $\mathbf{p}$ represent performance variables. The meaning of them is clear from the context.

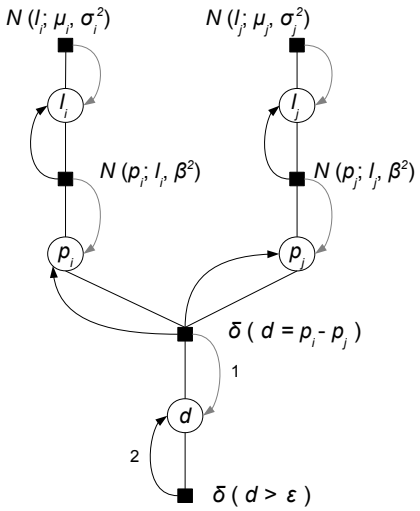

Fig. 1 TrueSkill factor graph for a match between two single-player teams with team i winning. There are three types of variables: l_i for the skills of all players, p_i for the performances of all players and d the performance difference. The first row of factors encode the (product) prior; the product of the remaining factors characterizes the likelihood for the game outcome team i winning team j. The arrows show the optimal message passing schedule: (1) messages pass along *gray* arrows from top to bottom, (2) the marginal over d is updated via message 1 followed by message 2 (which requires moment matching), (3) messages pass from bottom to top along *black* arrows.

where $\mathbb{I}[\cdot]$ is an indicator function. Then the likelihood $p(o|\mathbf{l}, M)$ in (1) can be written as

$$p(o|\mathbf{l}, M) = \int \cdots \int_{\mathbb{R}^n} \int_{-\infty}^{+\infty} p(o|d)p(d|\mathbf{p}, M) \prod_{i=1}^{n} p(p_i|l_i) \, \mathrm{d}\mathbf{p} \, \mathrm{d}d.$$

The entire TrueSkill model relevant to M is shown in the factor graph of Figure 1 with $P(o|d)$ given for the case of $o = \textit{team-i-win}$. TrueSkill uses message passing to infer the posterior distribution in (1) — note that the posterior over l_i and l_j will be updated according to the match outcome while the posterior over l_k ($k \notin \{i, j\}$) will remain unchanged from the prior. An optimal message passing schedule in the TrueSkill factor graph (Figure 1) is provided in the caption; the message along arrow 2 is a step function that leads to intractability for exact inference and thus TrueSkill uses message approximation via moment matching.

TrueSkill is an efficient and principled Bayesian skill learning system. However, due to its design goals, it discards score information and does not take into account associated domain knowledge such as offence/defence skill components. Next, we propose extensions of the TrueSkill factor graph and (ap-

proximate) inference algorithms for score-based Bayesian skill learning, which address these limitations.

3 Score-based Bayesian Skill Models

In this section, we introduce three graphical models as extensions for the TrueSkill factor graph (Figure 1) to incorporate score-based outcomes in skill learning. Our first two graphical models are motivated by modeling score-based outcomes as generated by separate offence and defence skills for each team. The first generative score model uses a Poisson, which is natural model when scores are viewed as counts of scoring events. The second generative model uses a simpler Gaussian model. Our third model is a simplified version of the Gaussian model, which like TrueSkill, only models a single skill per team (not separate offence/defence skills) and places a Gaussian likelihood on the score difference, which may be positive or negative. Next we formulate each model in detail.

3.1 Offence and Defence Skill Models

In a match between two teams i and j producing respective scores $s_i \in \mathbb{Z}$ and $s_j \in \mathbb{Z}$ for each team, it is natural to think of s_i as resulting from i's offence skill $o_i \in \mathbb{R}$ and j's defence skill $d_j \in \mathbb{R}$ (as expressed in any given match) and likewise for j's score as a result of j's offence skill $o_j \in \mathbb{R}$ and i's defence skill $d_i \in \mathbb{R}$. This is contrasted with the univariate skill estimates of team i's skill l_i and team j's skill l_j used in TrueSkill, which lump together offence and defence skills for each team.

Given scores s_i and s_j for teams i and j, we model the generation of scores from skills using a conditional probability $p(s_i, s_j|o_i, o_j, d_i, d_j)$. We assume that team i's score s_i depends only on o_i and d_j and likewise that team j's score s_j depends only on o_j and d_i:

$$p(s_i, s_j|o_i, o_j, d_i, d_j) = p(s_i|o_i, d_j)p(s_j|o_j, d_i). \tag{4}$$

Like TrueSkill, we assume that the joint marginal over skill priors independently factorises:

$$p(o_i, o_j, d_i, d_j) = p(o_i)p(d_j)p(o_j)p(d_i). \tag{5}$$

Given an observation of scores s_i for team i and s_j for team j, the problem is to update the posterior distributions over participating teams' offence and defence skills. According to Bayes rule and the previous assumptions, the posterior distribution over (o_i, o_j, d_i, d_j) is given by

$$\begin{aligned} p(o_i, d_i, o_j, d_j|s_i, s_j) &\propto p(s_i, s_j|o_i, d_i, o_j, d_j)p(o_i, d_i, o_j, d_j) \\ &\propto [p(s_i|o_i, d_j)p(o_i)p(d_j)]\ [p(s_j|o_j, d_i)p(o_j)p(d_i)]. \end{aligned} \tag{6}$$

Here we observe that estimating $p(o_i, d_i, o_j, d_j|s_i, s_j)$ factorises into the two independent inference problems:

$$p(o_i, d_j|s_i) \propto p(s_i|o_i, d_j)p(o_i)p(d_j), \text{and} \tag{7}$$
$$p(o_j, d_i|s_j) \propto p(s_j|o_j, d_i)p(o_j)p(d_i). \tag{8}$$

All models considered in this paper (including TrueSkill) assume Gaussian priors on team i's offence and defence skills, i.e., $p(o_i) := \mathcal{N}(o_i; \mu_{oi}, \sigma_{oi}^2)$ and $p(d_i) := \mathcal{N}(d_i; \mu_{di}, \sigma_{di}^2)$. Our objective then is to estimate the means and variances for the posterior distributions of $p(o_i, d_j|s_i)$ and $p(o_j, d_i|s_j)$. So far, the only missing pieces in this skill posterior update are the likelihoods $p(s_i|o_i, d_j)$ and $p(s_j|o_j, d_i)$ that specify how team i and j's offence and defence skills probabilistically generate observed scores. For this we discuss two possible models in the following subsections.

3.1.1 Poisson Offence/Defence Skill Model

Following TrueSkill, we model the generation of match outcomes (in our case, team scores) based on stochastic offence and defence *performances* that account for day-to-day performance fluctuations. Formally, we assume that team i exhibits offence performance $p_{oi} := \mathcal{N}(p_{oi}; o_i, \beta_o^2)$ and defence performance $p_{di} := \mathcal{N}(p_{di}; d_i, \beta_d^2)$. With these performances, we model team i's score s_i as generated from the following process: team i's offence performance p_{oi} promotes the scoring rate while the defence performance p_{dj} inhibits this scoring rate, the difference $p_{oi} - p_{dj}$ being the effective scoring rate of the offence against the defence.

Finally, we model the score by $s_i \sim \text{Poisson}(\lambda)$, where a requirement of a positive rate λ for the Poisson distribution requires the use of $\lambda = \exp(p_{oi} - p_{dj})$ since $p_{oi} - p_{dj}$ may be negative.[2] Likewise, one can model s_j by applying the same strategy when given $\lambda = \exp(p_{oj} - p_{di})$. We represent the resulting *Poisson-OD* model in Figure 2 where the joint posterior is

$$p(o_i, d_j, p_{oi}, p_{dj}|s_i) \propto p(s_i|p_{oi}, p_{dj})p(p_{oi}|o_i)p(p_{dj}|d_j)p(o_i)p(d_j),$$
$$p(o_j, d_i, p_{oj}, p_{di}|s_j) \propto p(s_j|p_{oj}, p_{di})p(p_{oj}|o_j)p(p_{di}|d_i)p(o_j)p(d_i).$$

We are only interested in the posterior distributions of o_i, d_j and o_j, d_i given s_i and s_j, respectively. Thus, we integrate out the latent performance variables to obtain the desired posteriors

$$p(o_i, d_j|s_i) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(o_i, d_j, p_{oi}, p_{dj}|s_i) \mathrm{d}p_{oi} \mathrm{d}p_{dj},$$
$$p(o_j, d_i|s_j) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(o_j, d_i, p_{oj}, p_{di}|s_j) \mathrm{d}p_{oj} \mathrm{d}p_{di}.$$

[2] This exponentiation of $p_{oi} - p_{dj}$ may seem to be made only to ensure model correctness, but we show experimentally that it has the benefit of allowing the Poisson model to accurately predict scores in high-scoring games even when team skills are very close (and hence $p_{oi} - p_{dj} \approx 0$).

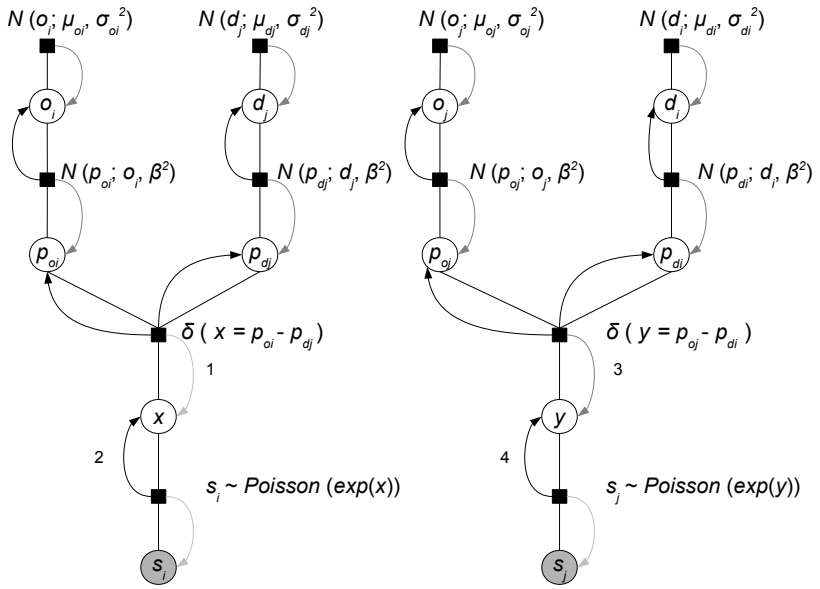

Fig. 2 The Poisson-OD variants of TrueSkill factor graph for skill update of two teams based on the match score outcome (Left: modeling s_i; Right: modeling s_j). Note that the score observation factors use the Poisson distribution for the Poisson-OD model. Shaded nodes are observed variables. For each team i, it is characterized by offence skill o_i (the offence skill of team i) and defence skill d_i (the defence skill of team i). Given s_j for team j, the posterior distributions over (o_i, d_j) are inferred via message passing.

Like TrueSkill, we use Bayesian updating to update beliefs in the skill levels of both teams in a pairwise match based on the score outcome, thus leading to an online learning scheme. Posterior distributions are approximated to be Gaussian and used as the priors in order to learn each team's skill for the next match. Approximate belief updates via variational Bayesian inference in this model will be covered in Section 4.2.

3.1.2 Gaussian Offence/Defence Skill Model

An alternative to the previous Poisson model is to model $s_i \in \mathbb{R}$ and assume it is generated as $s_i \sim \mathcal{N}(\mu, \gamma^2)$, where $\mu = p_{oi} - p_{dj}$. One can similarly model s_j by applying the same strategy when given $\mu = p_{oj} - p_{di}$. We note that unlike the Poisson model, μ can be negative here so we need not exponentiate it. While this allows us to directly model match outcomes that allow negative team scores (c.f., Halo2 as discussed in Section 5.1), it is problematic for other match outcomes that only allow non-negative team scores. One workaround would be to introduce a truncated Gaussian model to avoid the problem of assigning non-zero probability to negative scores, but we avoid this complication in exchange for the simple and exact updates offered by a purely Gaussian model.

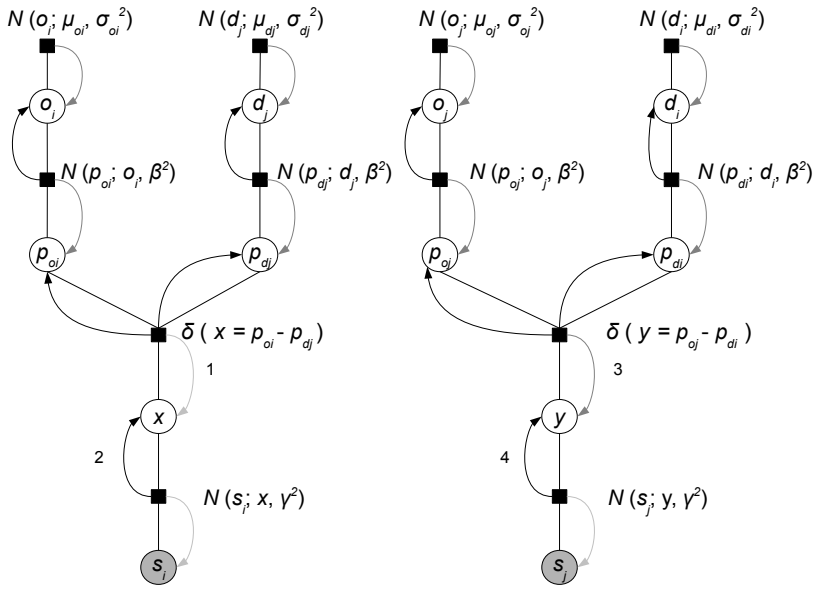

Fig. 3 The Gaussian-OD variant of the TrueSkill factor graph for skill update of two teams based on the match score outcome (Left: modeling s_i; Right: modeling s_j). Note that the score observation factors use the Gaussian distribution for the Gaussian-OD model. Shaded nodes are observed variables. For each team i, it is characterized by offence skill o_i (the offence skill of team i) and defence skill d_i (the defence skill of team i). Given s_j for team j, the posterior distributions over (o_i, d_j) are inferred via message passing.

We show the resulting *Gaussian-OD* model in Figure 3, which differs from our proposed Poisson model only in modeling the observed score s_i (s_j) for team i (j) given the univariate performance difference variable x (y). In this model, all messages passed during inference are Gaussian, allowing for efficient and exact belief updates.

3.2 Gaussian Score Difference (SD) Model

Again assuming $s_i \in \mathbb{R}$ and $s_j \in \mathbb{R}$, algebra for the performance means in Figure 3 gives:

$$s_i = p_{oi} - p_{dj}, \qquad s_j = p_{oj} - p_{di}. \tag{9}$$

This implies

$$\begin{aligned} s_i - s_j &= (p_{oi} - p_{dj}) - (p_{oj} - p_{di}) \\ &= \underbrace{(p_{oi} + p_{di})}_{p_{li}} - \underbrace{(p_{oj} + p_{dj})}_{p_{lj}}, \end{aligned} \tag{10}$$

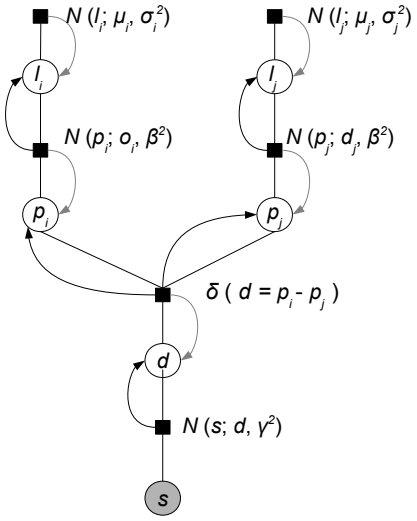

Fig. 4 The Gaussian-SD variant of the TrueSkill factor graph model for skill update of two teams based on the score difference. Both team i and team j are characterized by skill level l_i and l_j, respectively. The shaded node s ($s = s_i - s_j$) denotes the score difference between s_i and s_j. Bayesian inference for the posterior skill level distributions has a closed-form solution.

which is like modeling the score difference with performance expressions p_{li} and p_{lj} of respective univariate skill levels, l_i and l_j. Motivated by (9), we propose a score difference (SD) Gaussian model that uses a likelihood model for the observed difference $s := s_i - s_j$ specified as $s \sim \mathcal{N}(p_{li} - p_{lj}, \gamma^2)$ as shown in Figure 4.

4 Skill and Win Probability Inference

We infer skill distributions in all proposed models via online Bayesian updating. While exact inference in the purely Gaussian models can be achieved by solving linear systems, Bayesian updating provides an efficient (also exact) incremental learning alternative. Equations for Bayesian updates and the probability of three possible match outcomes (e.g., winning, losing and drawing) are model-dependent and presented below.

4.1 Inference in TrueSkill

4.1.1 Bayesian update

The Bayesian update equations in the TrueSkill model (Figure 1) are presented in [10], and we omit the details in the present paper and refer the interesting readers to Table 1 in [10].

4.1.2 Win/Lose/Draw Probability

Given skill levels of team i and j, $l_i \sim \mathcal{N}(l_i; \mu_i, \sigma_i^2)$ and $l_j \sim \mathcal{N}(l_j; \mu_j, \sigma_j^2)$, we first compute the distribution over performance difference variable d, and get $d \sim \mathcal{N}(d; \mu_d, \sigma_d^2)$ with $\mu_d = \mu_i - \mu_j$ and $\sigma_d^2 = \sigma_i^2 + \sigma_j^2 + 2\beta^2$. The winning probability of team i is given by the probability $p(d > 0)$ defined as

$$p(d > 0) = 1 - \Phi\left(\frac{-\mu_d}{\sigma_d}\right), \tag{11}$$

where $\Phi(\cdot)$ is the normal CDF. Likewise, we define the draw probability as the probability distribution funciton of the variable d, evaluated at 0, and the lose probability of team i as $\Phi\left(\frac{-\mu_d}{\sigma_d}\right)$.

4.2 Inference in Poisson-OD Model

4.2.1 Bayes Update

Some of the update equations in the Poisson-OD model (Figure 2) have been presented in [10], with the exception of the marginal distribution over x and the message passing from the Poisson factor to x. Given a prior Gaussian distribution over x, $\mathcal{N}(x; \mu, \sigma^2)$, we next demonstrate how to update the belief on x when observing team i's score s_i.

By the sum-product algorithm [12], the marginal distribution of x is given by a product of messages

$$p(x|s_i) = m_{\delta \to x}(x) m_{s_i \to x}(x). \tag{12}$$

To avoid cluttered notation, let us use $m_1(x)$ to represent $m_{\delta \to x}(x) = \mathcal{N}(x; \mu, \sigma^2)$, i.e., the message passing from the factor $\delta(\cdot)$ to x, and $m_2(x)$ for $m_{s_i \to x}(x) = \mathit{Poisson}(s_i; \exp(x))$, i.e., the message passing from the Poisson factor to x (c.f., messages labeled 1 and 2 in Figure 2). Due to the multiplication of $m_1(x)$ and $m_2(x)$, the exact marginal distribution of $p(x|s_i)$ is not Gaussian, which makes exact inference intractable. To maintain a compact representation of offence and defence skills, one can approximate $p(x|s_i)$ with a variational Bayes framework or a sampling-based approach considering its being a univariate distribution.

Bayesian update with VB In a variational Bayes framework, the problem is to choose a Gaussian distribution $q(x)^* : \mathcal{N}(x; \mu_{\text{new}}, \sigma^2_{\text{new}})$ that minimizes the KL divergence between $p(x|s_i)$ and $q(x)$, i.e.,

$$q(x)^* = \arg\min_{q(x)} \text{KL}\left[q(x)||p(x|s_i)\right]. \tag{13}$$

We derive a fixed-point approach for optimizing $q(x)$ [17] and describe this approach below.

Minimizer $q(x)$ for KL$(q(x)||p(x|s_i))$: We first expand the KL-divergence into its definition:

$$\begin{aligned}\text{KL}\left(q(x)||p(x|s_i)\right) &= \int q(x) \log\left(\frac{q(x)}{p(x|s_i)}\right) dx \\ &= -\log\sqrt{2\pi e \sigma^2_{new}} - E_{x\sim q(x)} \log\left(p(x|s_i)\right), \end{aligned} \tag{14}$$

where $p(x|s_i)$ is the posterior probability of x when observing the score s_i. Since $q(x)$ is Gaussian and the posterior has convenient Gaussian parts, manipulation of this yields an equation for μ_{new} and σ^2_{new} that can be solved using an iterative fixed-point approach:

Lemma 1 *Values for μ_{new} and σ^2_{new} minimizing KL$\left(q(x)||p(x|s_i)\right)$ satisfy*

$$\begin{aligned}\mu_{new} &= \sigma^2\left(s_i - e^{\kappa}\right) + \mu, \\ \sigma^2_{new} &= \frac{\sigma^2}{1 + \sigma^2 e^{\kappa}}, \end{aligned} \tag{15}$$

where

$$\kappa = \log\left(\frac{\mu + s_i\sigma^2 - 1 - \kappa + \sqrt{(\kappa - \mu - s_i\sigma^2 - 1)^2 + 2\sigma^2}}{2\sigma^2}\right). \tag{16}$$

Proof The second term in (14) is evaluated using Bayes Theorem, $p(x|s_i) = p(s_i|x)p(x)/p(s_i)$. The term in $\log p(s_i)$ can be dropped because it is constant with respect to μ_{new} and σ^2_{new}. The term $E_{x\sim q(x)}[\log p(s_i|x)]$ is found by expanding the Poisson distribution and noting $E_{x\sim p(x)}[\exp(x)] = \exp(\mu + \sigma^2/2)$ (Appendix A for derivation). Thus it becomes

$$s_i\mu_{\text{new}} - \exp(\mu_{\text{new}} + \sigma^2_{\text{new}}/2) - \log(s_i!)\ . \tag{17}$$

The term $E_{x\sim q(x)}[\log p(x)]$ according to the derivation in Appendix B becomes

$$-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(\sigma^2_{new} + \mu^2_{new} - 2\mu\mu_{new} + \mu^2\right)\ . \tag{18}$$

Plugging (17) and (18) into (14) gives

$$\arg\min_{q(x)} \mathrm{KL}\left(q(x)||p(x|s_i)\right) \equiv \arg\min_{q(x)} -\log\sqrt{2\pi e\sigma^2_{new}} - \Bigg(\underbrace{s_i\mu_{\text{new}} - \exp(\mu_{\text{new}} + \sigma^2_{\text{new}}/2) - \log(s_i!)}_{E_{x\sim q(x)}(\log p(s_i|x))} \underbrace{-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(\sigma^2_{new} + \mu^2_{new} - 2\mu\mu_{new} + \mu^2\right)}_{E_{x\sim q(x)}(\log p(x))} \Bigg).$$

To find the minimizer $q(x)$, we calculate the partial derivatives of $\mathrm{KL}\left(q(x)||p(x|s_i)\right)$ w.r.t. μ_{new} and σ_{new}, and set them to zero, leading to

$$\mu_{\text{new}} = \sigma^2\left(s_i - \exp\left(\mu_{\text{new}} + \frac{\sigma^2_{\text{new}}}{2}\right)\right) + \mu,$$

$$\sigma^2_{\text{new}} = \frac{\sigma^2}{1 + \sigma^2\exp(\mu_{\text{new}} + \frac{\sigma^2_{new}}{2})}\ .$$

Summing the first plus half the second of these equations, and defining $\kappa = \mu_{\text{new}} + \sigma^2_{\text{new}}/2$ yields the equation for κ of

$$\kappa = \mu + \sigma^2(s_i - \exp(\kappa)) + \frac{\sigma^2}{2(1 + \sigma^2\exp(\kappa))}, \tag{19}$$

and one gets (15) in terms of κ.

We convert (19) by solving for $\exp(\kappa)$ as it appears on the right-hand side. This yields a quadratic equation, and we take the positive solution since $\exp(\kappa)$ must be non-negative (see the Supplemental material). The result gives us (16).

We can use (16) as a fixed-point rewrite rule. For a given μ and σ^2 together with an initial value of κ, one iterates (16) until convergence. Empirically, this happens within 2-3 iterations. With convergence, we substitute the fixed-point solution into (15) to get the optimal mean and variance for $q(x)^*$.

Bayes Update with Slice Sampling One caveat associated with variational Bayes is that it may yield local minimum; and thus one tends to use sampling-based appoaches to obtain exact solutions. One widely used sampling-based approaches is based on Markov chain Monte Carlo (MCMC). To approximate this univariate distribution $p(x|s_i$, we can use one type of MCMC called slice sampling [4]. The idea behind slice sampling is to sample uniformly from the region under the plot of the density function to be approximated; by doing so, one can adatp to the characteristics of the distribution. One can construct this Markov chain that converges to this uniform distribution by alternating uniform sampling in the vertical direction with uniform sampling from the horizontal "slice" defined by the current vertical position. This method can be easily implemented for univariate distributions, and we use the Matlab embedded function slicesample for approximating $p(x|s_i)$.

4.2.2 Win/Lose/Draw Probability

Suppose we are given the offence and defence skills for team i and j, we can pass the messages down in Figure 2 to estimate the distributions for the performance variables $p_{oi}, p_{dj}, p_{di}, p_{oj}$. To estimate the win/lose/draw probability, we simply construct a variable $d = p_{oi} + p_{di} - (p_{oj} + p_{dj})$ to represent the performance difference for team i and j. Since the variable d is a Gaussian distributed variable, the win/lose/draw probability $p(s > 0)$for the Poisson-OD model defined below is simliar with the TrueSkill.

$$p(d > 0) = 1 - \Phi\left(\frac{-\mu_d}{\sigma_d}\right), \tag{20}$$

where μ_d and σ_d are the mean and the standard deviation of the performance difference variable d. We can omit the definitions of μ_d and σ_d since they fall out of the linear operations of Gaussian variables. Given this Gaussian distribution performance difference variable, we define the draw and lose probability following that of TrueSkill defined in Section 4.1.2.

We pause here to note that the above definitions for win/lose/draw probability slightly deviates from the Poisson-OD model (Figure 2) in the sense that we do not compute these probabilities based on the score difference that the model predicts. To compute the winning probability of team i, i.e., $s_i > s_j$, based on score difference, we first construct a new variable $s = s_i - s_j$, the difference variable between two Poisson distributions, which proves to be a Skellam distribution in [15]. Thus, we can compute the win probability of $P(s > 0)$ of team i, according to the probability mass function for the Skellam distribution

$$P(s = k; \lambda_i, \lambda_j) = e^{-(\lambda_i + \lambda_j)} \left(\frac{\lambda_i}{\lambda_j}\right)^{k/2} I_{|k|}\left(2\sqrt{\lambda_i \lambda_j}\right),$$

where $I_k(z)$ is the modified Bessel function of the first kind given in [6].

$$I_k(z) = \left(\frac{z}{2}\right)^k \sum_{i=0}^{+\infty} \frac{(z^2/4)^i}{i! \Gamma(k + i + 1)}. \tag{21}$$

Unfortunately, we need approximation to compute the win probability $P(s > 0)$. To avoid the approximmation, we choose the Equation 20 for computing the winning probability in the present paper. The lose and draw probability are also derived from Equation 20 as described above.

4.3 Inference in Gaussian-OD Model

4.3.1 Bayesian update

In the Gaussian-OD model (Figure 3), all messages are Gaussian so one can compute the belief update in closed-form as follows

$$\begin{aligned}
\pi_{o_i} &= \frac{1}{\sigma_{o_i}^2} + \frac{1}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{d_j}^2}, \\
\tau_{o_i} &= \frac{\mu_{o_i}}{\sigma_{o_i}^2} + \frac{s_i + \mu_{d_j}}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{d_j}^2}, \\
\pi_{d_j} &= \frac{1}{\sigma_{d_j}^2} + \frac{1}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{o_i}^2}, \\
\tau_{d_j} &= \frac{\mu_{d_j}}{\sigma_{d_j}^2} + \frac{\mu_{o_i} - s_i}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{o_i}^2},
\end{aligned} \tag{22}$$

where μ_{o_i} and σ_{o_i} are the mean and standard deviation of the prior offence skill distribution of team i, $\pi_{o_i}(\pi_{d_j}) = \frac{1}{\sigma_{post}^2}$ and $\tau_{o_i}(\tau_{d_j}) = \frac{\mu_{post}}{\sigma_{post}^2}$ are the precision and precision-adjusted mean for the posterior offence (defence) skill distribution of team i (j). Likewise, one can derive the update equations for team j's offence skill o_j and team i's defence skill d_i.

4.3.2 Win/Lose/Draw Probability

To compute the probability of team i winning vs team j, we first use message passing to estimate the normally distributed distributions for score variables s_i and s_j, and then compute the probability that $s_i - s_j > 0$, i.e., team i's score is larger than team j's. Given $s_i \sim \mathcal{N}(s_i; \mu_{si}, \sigma_{si}^2)$ and $s_j \sim \mathcal{N}(s_j; \mu_{sj}, \sigma_{sj}^2)$, we can compute the winning probability of team i by

$$p(s > 0) = 1 - \Phi\left(\frac{-(\mu_{si} - \mu_{sj})}{\sigma_{si}^2 + \sigma_{sj}^2}\right). \tag{23}$$

where s is the Gaussian distributed variable representing the difference $s_i - s_j$. The draw probability is the probaiblity density function of the Gaussian distributed variable s, evaluated at 0, and the lose probability of team i is simply the probability that $s < 0$, which is the normal cdf $\Phi\left(\frac{-(\mu_{si} - \mu_{sj})}{\sigma_{si}^2 + \sigma_{sj}^2}\right)$.

4.4 Inference in Gaussian-SD Model

4.4.1 Bayes Update

In the Gaussian-SD model (Figure 4), all messages are Gaussian so we can again derive the update for the single team skills l_i and l_j in closed-form as

follows:

$$\begin{aligned}
\pi_{l_i} &= \frac{1}{\sigma_{l_i}^2} + \frac{1}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{l_j}^2}, \\
\tau_{l_i} &= \frac{\mu_{l_i}}{\sigma_{l_i}^2} + \frac{(s_i - s_j) + \mu_{l_j}}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{l_j}^2}, \\
\pi_{l_j} &= \frac{1}{\sigma_{l_j}^2} + \frac{1}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{l_i}^2}, \\
\tau_{l_j} &= \frac{\mu_{l_j}}{\sigma_{l_j}^2} + \frac{\mu_{l_i} - (s_i - s_j)}{\beta_1^2 + \beta_2^2 + \gamma^2 + \sigma_{l_i}^2},
\end{aligned} \tag{24}$$

where μ_{l_i} (μ_{l_j}) and σ_{l_i} (σ_{l_j}) are the mean and standard deviation of team i's (team j's) prior skill distribution, π_{l_i} (π_{l_j}) and τ_{l_i} (τ_{l_j}) are the precision and precision adjusted mean for team i's (team j's) posterior skill distribution.

4.4.2 Win/Lose/Draw Probability

To estimate the winning probability of team i for a match with team j, one can first use message passing to estimate the normally distributed score difference variable s, and then compute the winning probability of team i by

$$p(s > 0) = 1 - \Phi\left(\frac{l_i - l_j}{\sigma_i^2 + \sigma_j^2 + 2\beta^2}\right), \tag{25}$$

where l_i and σ_i are the mean and standard deviation for team i's skill level, and β the standard deviation of the performance variable.

5 Empirical Evaluation

5.1 Data Sets

Experimental evaluations are conducted on three data sets: Halo 2 XBox Live matches, Australian Football (Rugby) League (AFL)and UK Premier League (UK-PL)[3]. The Halo 2 data consists of a set of match outcomes comprising 6227 games for 1672 players. We note there are negative scores for this data, so we add the absolute value of the minimal score to all scores to use the data with all proposed models.

The training and testing settings are described as follows. For Halo 2 [4], the last 10% of matches are used for testing, and we use different proportions of the first 90% of data for training. There are 8 proportions used for training, ranging from 10% to 80% with an increment of 10%, and 90% is not used for

[3] `http://www.football-data.co.uk/englandm.php`

[4] Credit for the use of the Halo 2 Beta Data set is given to Microsoft Research Ltd. and Bungie.

training due to cross validation. To cross validate, we randomly sample the data and run the learning 10 times at each proportion level to get standard error bars. Note that there are some players in the testing games who are not involved in any training data sets, particularly when small proportion of training data set is selected (e.g., the first 10 percent games); we remove these games in the testing set when reporting performances for all models.

For both UK-PL and AFL datasets, cross validation is performed by training and testing for each year separately (14 years for UK-PL, and 11 years for AFL). For these two datasets, we test the last 20% percent of matches in each year, with the training data increasing incrementally from 10% to 80% of the initial matches.

5.2 Evaluation Criteria

We evaluate performances using three criteria: *information gain* of predicting winning probability of a team (Section 5.2.1), *win/lose prediction accuracy* (Section 5.2.2), *win/lose/draw prediction accuracy* (Section 5.2.3), and *score prediction errors* (Section 5.2.4). While the first three criteria focus on predicting win/lose or win/lose/draw, the fourth criterion measures how good a model is at predicting scores, for which TrueSkill does not apply since it is restricted to WLD only. Let us introduce each criterion in detail.

5.2.1 Information Gain

The first criterion we use to evaluate different approaches is *information gain*, which is proposed in the *Probabilistic Footy Tipping Competition*[5]: if a predictor assigns probability p to team i winning, then the score (in "bits") gained is $1+\log_2(p)$ if team i wins, $1+\log_2(1-p)$ if team i loses, $1+(1/2)\log_2(p(1-p))$ if draw happens. This evaluation metric can be viewed as an information gain interpretable variant of a log likelihood score where an uninformed prediction of $p = 0.5$ leads to a score of 0 and a definite prediction of $p = 1$ ($p = 0$) leads to a score of $-\infty$ if predicting incorrectly and 1 if predicting correctly. In Section 4, we showed how to compute the win probability p for each model.

5.2.2 Win/no-Win Prediction Accuracy

While information gain provides a sense of how well the models fit the data, it is also interesting to see how accurate the models were at predicting match outcomes in terms of win/no-win (e.g., loss/draw). To compare classification performance of each model, we report the win/not winning prediction accuracy in terms of area under the curve (AUC) for the games with a win or loss outcome.

[5] Refer to `http://www.csse.monash.edu.au/~footy/`

5.2.3 Multi-Class Classification Among Win, Lose, and Draw

The Win/no-Win prediction accuracy introduced above suits for binary classification. This criterion does not necessarily apply to the cases where draws can happen such as a match outcomes end up with each team obtaining 3 goals in football games. To accommodate the applications where draws can happen, we evaluate the predictive performance in terms of a multi-class classification problem where there are three classes for a game: winning, drawing, and losing for one out of the two teams participating in two-team game. We use the Brier score introduced in [2] as a performance measure for this multi-class classification setting, as in [3].

Following [3], we define the Brier Score for a set of N testing games below:

$$\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{K}\left(\mathbb{I}[y_i=k]-\hat{p}_k^i\right)^2\right), \tag{26}$$

where $\hat{p}_k^i$ is the probability estimate of the ith game being assigned to the k-th class, and $I(y_i = k)$ is an indicator function (1 if $y_i = k$ and 0 otherwise). Note that one advantage of this measurement is that it does not require the true probability of a data point belonging to a class.

5.2.4 Score Prediction Error

We evaluate the score prediction accuracy for Poisson-OD and Gaussian-OD models for *each* team in terms of the mean absolute error (MAE), defined as below:

$$\frac{1}{2N}\sum_{i=1}^{2N}\left(|\hat{s_i}-s_i|+\hat{s_j}-s_j|\right) \tag{27}$$

where $\hat{s_i}$ ($\hat{s_j}$) is the predicted score for team i (j), s_i (s_j) the ground truth, and N the number of two-team matches for with teams indexed by i and j.

Note that we must omit the Gaussian-SD model since it can only predict score differences rather than scores. To benchmark the score prediction performance of the Poisson-OD and Gaussian-OD models, we compare with an average score prediction methods. This average score methods simply use the average scores for a team computed from the training games as predictions for testing games.

5.3 Results on Four Criteria

Experimental results are reported according to the parameter configurations shown in Table 1. Parameters for the slice sampling used in the Poisson-OD model include the burn-in, thinning, and the number of samples required, which we set to 1000, 5, and 1000, respectively. Now we discuss the results against these four criteria on three real data sets below.

Table 1 Parameter settings. Priors on offence/defence skills: $\mathcal{N}(\mu_0, \sigma_0^2)$ with $\mu_0 = 25$ and $\sigma_0 = 25/3$. Performance variance: β, β_o, β_d.

Model	Parameter (ϵ, γ empirically estimated)
TrueSkill	$\beta = \sigma_0/2$, ϵ: draw probability
Poisson-OD(VB)	$\beta_o = \beta_d = \sigma_0/2$
Poisson-OD(Sampling)	$\beta_o = \beta_d = \sigma_0/2$
Gaussian-OD	$\beta_o = \beta_d = \sigma_0/2$, γ: score variance
Gaussian-SD	$\beta = \sigma_0/2$, γ: score difference variance

5.3.1 Information Gain

Information gain for four models on the UK data set is shown in Figure 5. The results indicate that the proposed model are significantly better than the TrueSkill for most of the training proportions, particularly when less than 30% data used for training. For limited data, the score-based skill learning models including the Poisson-OD[6], Gaussian-OD, and Gaussian-SD, all of which significantly outperform the TrueSkill. These results indicte that score information are indeed useful for making predictions when training data is limited.

Out of the three proposed models, the Gaussian models significantly outperform the Poisson-OD model when training with 10% and 20% of the data. This is because that the scores for the UK data set are relatively small; thus the amplification due to the exponential term in the Poisson-OD model leads to more extreme probaiblities, thus may hurt the performance if the predictions are wrong. But when the training data increases, the Poisson-OD model can refine further the belief over teams' skill levels thus making less mistakes, leading to comparable performances with the Gaussian models. The two Gaussian models achieve comparable performances for all training settings, with the Gaussian-SD model slightly edging out the Gaussian-OD model.

Empirical evaluations with the information gain criterion on the AFL data set is shown in Figure 6. We observed that all the proposed models significantly outperform TrueSkill for limited data, which further validates our hypothesis that information carried by score-based match outcomes are very helpful for skill learning when only small amount of data is available for training. When more data is available, TrueSkill improved its performance and performed the best, which indicates that updating skill levels with win/lose/draw based outcomes is more robust comparing with that based on the possibly noisy scores.

Across the three proposed models of the AFL data set, the Poisson-OD model significantly outperforms the Gaussian models, which agrees with the

[6] We note that the Poisson-OD model achieves much better performances when the win/lose/draw probabilities are defined according to Section 4.2.2 in the present paper, constrasted with the results in [22] that computes the win probability based on the score difference variable.

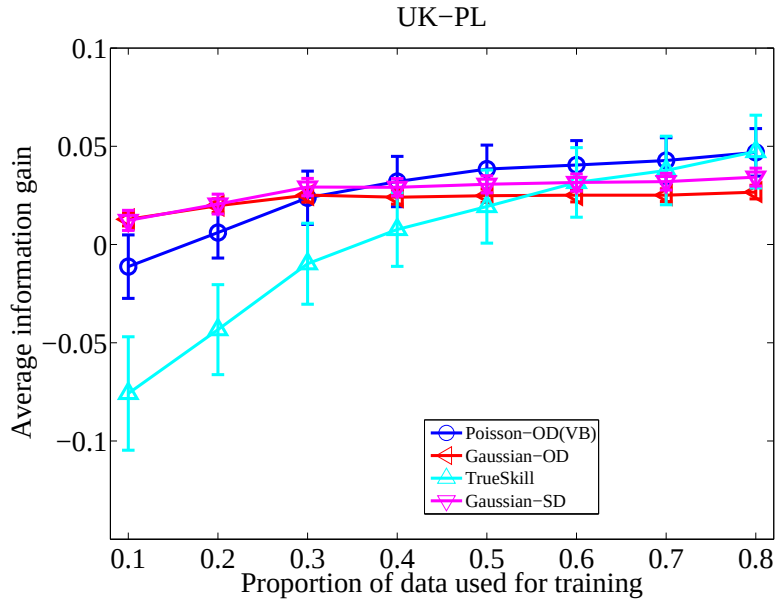

Fig. 5 Results on the UK-PL, evaluated using information gain. Error bars indicate 95% confidence intervals.

fact that the exponential term in the Poisson-OD model can effectively learn relatively large match outcomes. Note that the average score for the AFL data set is 95.4 vs 1.3 for that of the UK data set.

Results on the Halo data set are shown in Figure 7, again all the three proposed models significantly outperform TrueSkill for all training setting. Contrasted with the performances on the AFL data set, the Gaussian models significantly outperform the Poisson models, considering the fact that the scores for matches of the Halo data on average are much larger than the UK, but smaller than the AFL data set. Regarding the Gaussian-OD and Gaussian-SD model, we observed that the Gaussian-OD model slightly outperforms the Gaussian-SD model, thanks to modeling offence/defence skill levels separately. spects separately.

As a short summary for the results on the information gain, our observed that all the proposed models can outperform the TrueSkill for most of the training settings, and the differences are significant particularly for limited training data. This observation reflects our motivation of modeling the more informative score-based match outcomes in improving skill modeling. The Gaussian-OD model performs better than the Gaussian-SD model for most of the training cases, and the improvement is significant for the Halo data set,

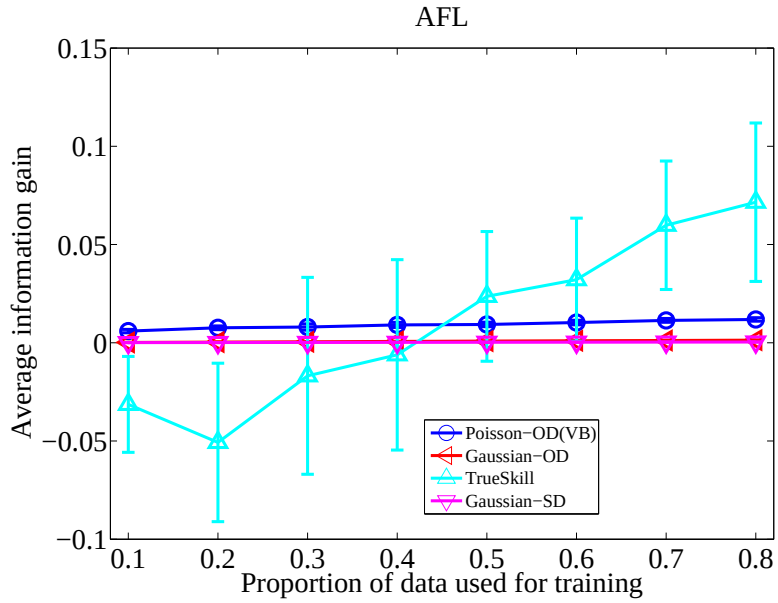

Fig. 6 Results on the AFL data set, evaluated using information gain. Error bars indicate 95% confidence intervals.

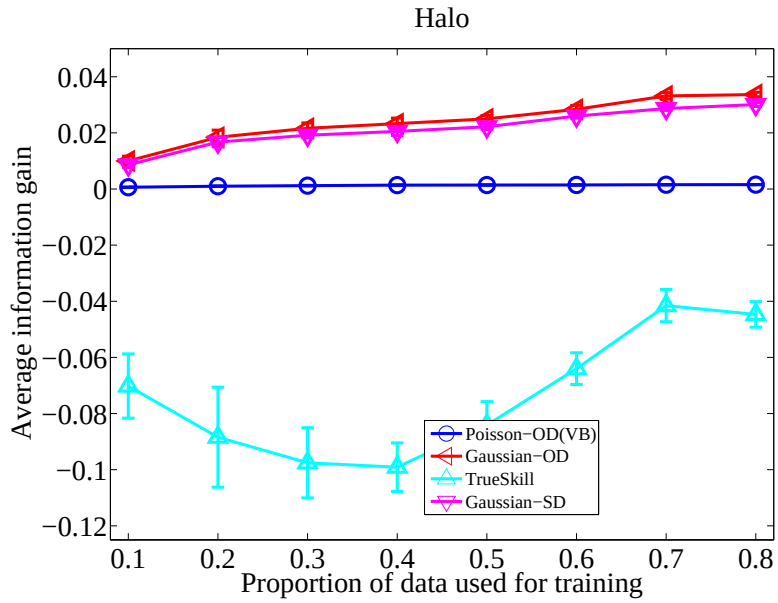

Fig. 7 Results on the Halo 2 data set, evaluated using information gain. Error bars indicate 95% confidence intervals.

indicating the benefits of modeling the offence and defence of a team's skills. While the Gaussian models are more appropriate for the data sets with small average match scores, the Poisson model edges out for the AFL data set that comes with an overall large average scores.

5.3.2 Win/no-Win Prediction Accuracy

In terms of win/no-win prediction accuracy on the three data sets (Figure 8, Figure 9, and Figure 10), the Gaussian-OD model generally provides the best average AUC, followed by Gaussian-SD, then TrueSkill, then Poisson-OD model. The better performance achieved by the Gaussian-OD model again indicated the benefits of separating offence/defence skill modeling in the Gaussian-OD model, achieving a performance edge over the combined skill model of the Gaussian-SD model.

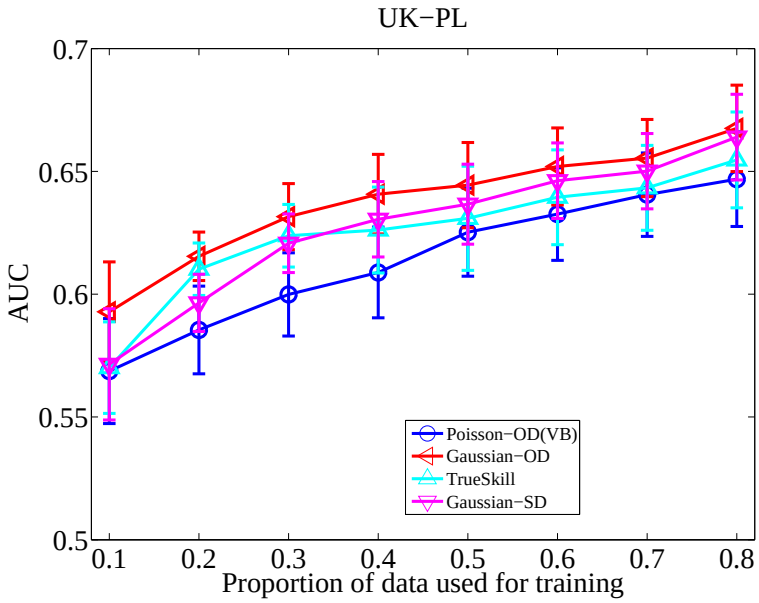

Fig. 8 Results on the UK-PL, evaluated using win/loss prediction accuracy in term of the area of the curve (AUC). Error bars indicate standard errors.

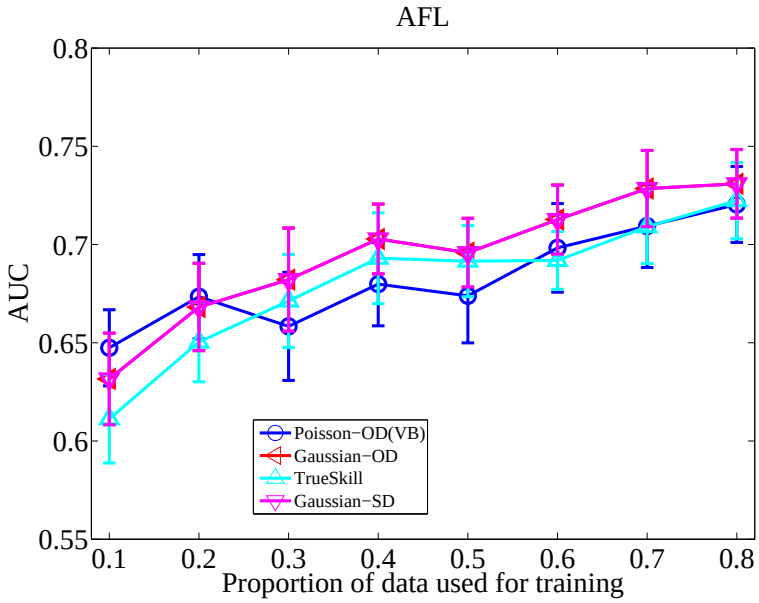

Fig. 9 Results on the AFL data set, evaluated using win/loss prediction accuracy in term of the area of the curve (AUC). Error bars indicate standard errors.

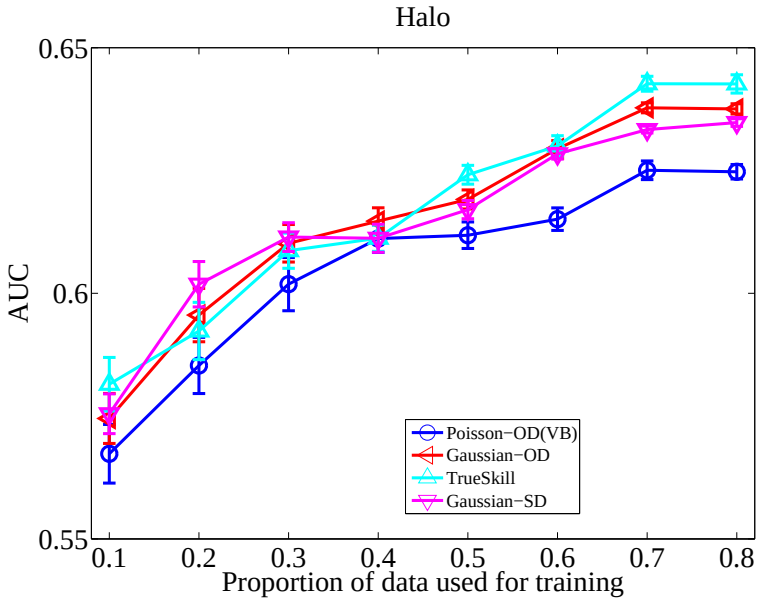

Fig. 10 Results on the Halo 2 data set, evaluated using win/loss prediction accuracy in term of the area of the curve (AUC). Error bars indicate standard errors.

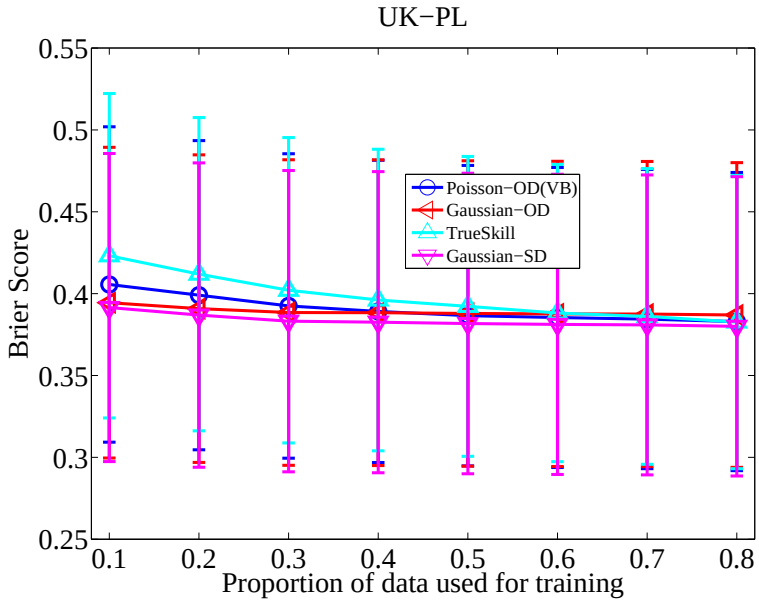

Fig. 11 Results on the UK data set, evaluated using the Brier Score for multi-class classification. Error bars indicate standard errors.

5.3.3 Win/Lose/Draw Prediction with Brier Score

We further studied the performances of various models with the Brier Score in a multi-class classification setting, where each game is an observation and the class for the observation between two teams (i, j) is in $\{i \text{ win}, i \text{ lose}, \text{ draw}\}$. Comparing with the previous reported performance on Win/no-Win, the Brier score we used for multi-class classification can handle the matches that are a draw between the participating teams.

For the results on the UK and AFL data set shown in Figure 11 and Figure 12, we observed that the the standard errors for all the models are much larger, indicating that the performance differences between these models are not significant. For the Brier score of UK-PL data set (Figure 11), the Poisson-OD and the Gaussian models seem to outperform TrueSkill for limited training data. Within the three proposed model, the Gaussian models slightly edge out the Poisson-OD model, which is again due to the scores for the UK data set are relatively small numbers.

For the results on the AFL data set (Figure 12), we observed that all the three proposed models perform much better than the TrueSkill, although TrueSkill reached the best possible performance of the Poisson-OD, Gaussian-OD, and Gaussian-SD when sufficiently large amount of data is used for training. Across the three proposed models for the AFL data set, the Poisson-OD

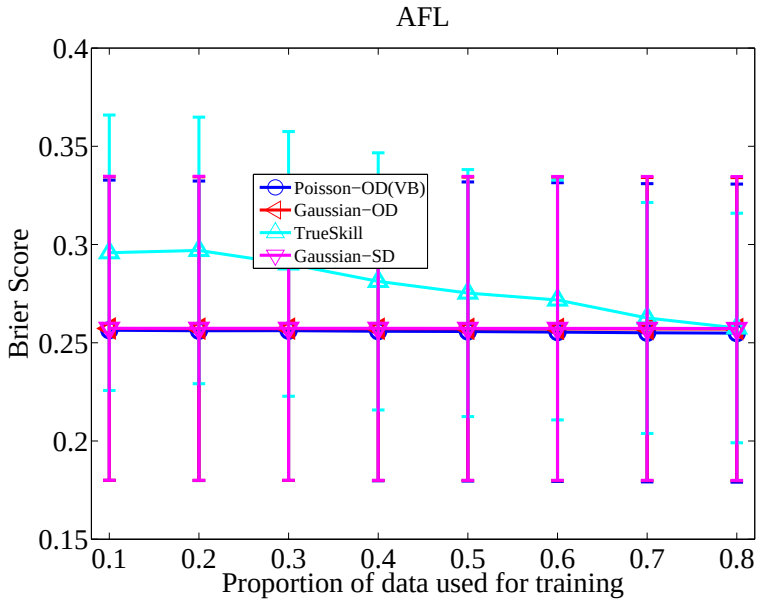

Fig. 12 Results on the AFL data set, evaluated using the Brier Score for Win/Lose/Draw prediction. Error bars indicate standard errors.

model slightly outperforms the Gaussian models, however the performance difference is not significant.

Results on the Halo data set (Figure 13) again demonstrated that the proposed models provide much better win/lose/draw predictions comparing with TrueSkill. For this data set, the Gaussian models outperform the Poisson-OD model, which is not surprising as the average score for the Halo data set is relatively small numbers. Note that the performances for the Gaussian models are comparable with each other, which suggested that it may not be beneficial to model offence/defence skills separately for applications where the notions of offence and defence are not clear, which is the case for the Halo data but not for the AFL and UK data sets.

When comparing the best performance on the AFL and UK dataset, we note that the Brier Score is much lower for the AFL and the Halo data set, close to 0.25 and 0.275 respectively, comparing with the UK data set (close to 0.38), indicating that predicting the football games may be much harder than that the rugby games for AFL and online games for the Halo. The difficulty in predicting win/lose/draw for football matches is perhaps caused by the unexpected player injuries, teams trading players, etc., which is not the case for the Halo data set. One may argue that these factors apply for the AFL

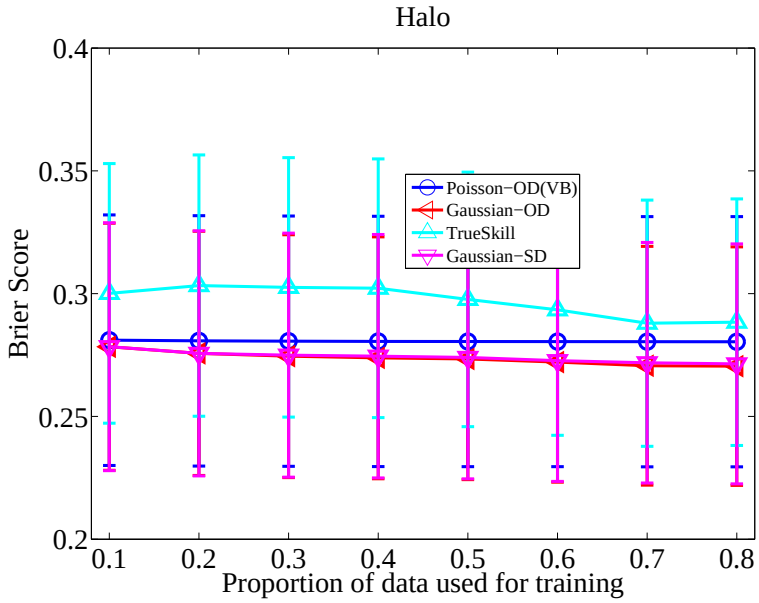

Fig. 13 Results on the Halo data set, evaluated using the Brier Score for Win/Lose/Draw prediction. Error bars indicate standard errors.

data set, but it seems that the AFL has somehow strict rules in regulating the trading of players.

5.3.4 Score Prediction Errors

We report the score prediction errors for different data sets in Figure 14, Figure 16, and Figure 17. Gaussian-OD predicts more accurate scores on the UK-PL and the Halo, while the Poisson-OD model is more accurate for the AFL dataset. For the AFL data set (Figure 14), we observed that the Gaussian-OD model clearly failed in predicting the scores of the matches, because the differences in the learned skill levels between teams are relatively small. A Gaussian likelihood function with the small difference has low probability of generating huge match scores for the AFL data set. Given the same difference, the Poisson-OD model with an exponentiated scoring rate would seem to amplify these small performance differences in learned AFL skills to make more accurate score predictions on AFL data. This amplification appears to hurt the Poisson-OD model on the lower-scoring UK-PL (the mean score for the AFL data is 95.4 vs 42.7 and 1.3 respectively for the Halo 2 and UK-PL data).

As inclusion of the Gaussian-OD model makes the comparison between the Poisson model and the baseline hard to visualize, we removed the Gaussian-OD

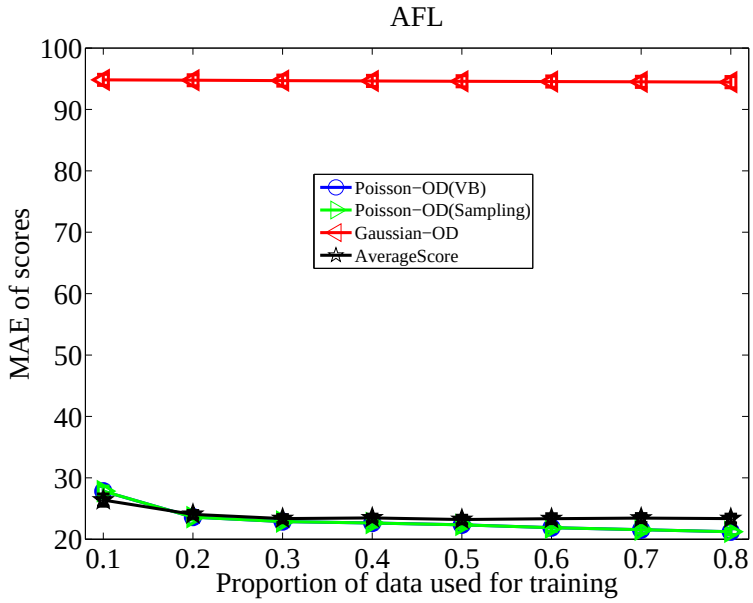

Fig. 14 Results on the AFL data set, evaluated using win/loss prediction accuracy in term of the area of the curve (AUC). Error bars indicate standard errors.

model in Figure 15 for better comparing the Poisson model and the baseline method. Interestingly, we observe that the baseline method based on average scores of each team makes better predictions than the Poisson model when only 10% of data is used for training. This is not surprising because the belief associated with the Poisson-OD model on team skills exhibits large uncertainty when only a few observations are used for training. As we can see when more data is used for training, the Poisson-OD model performs significantly better than the baseline. Note that for the Poisson-OD model, we observed that the performances are comparable when Bayes inference is conducted by variational Bayes and slice sampling.

As mentioned above for predicting scores on the UK data set (Figure 16), the Poisson-OD model clearly fails in predicting this type of matches with low match scores (average scores being 1.3 for the UK data set). But the Gaussian-OD model significantly outperforms the baseline for all the training settings.

Results on the Halo data set (Figure 17) further demonstrate that our proposed score predictors can achieve good score prediction results. Both the Gaussian-OD model and the Poisson-OD model significantly outperform the baseline. Comparing results across the three data sets, we suggest that one

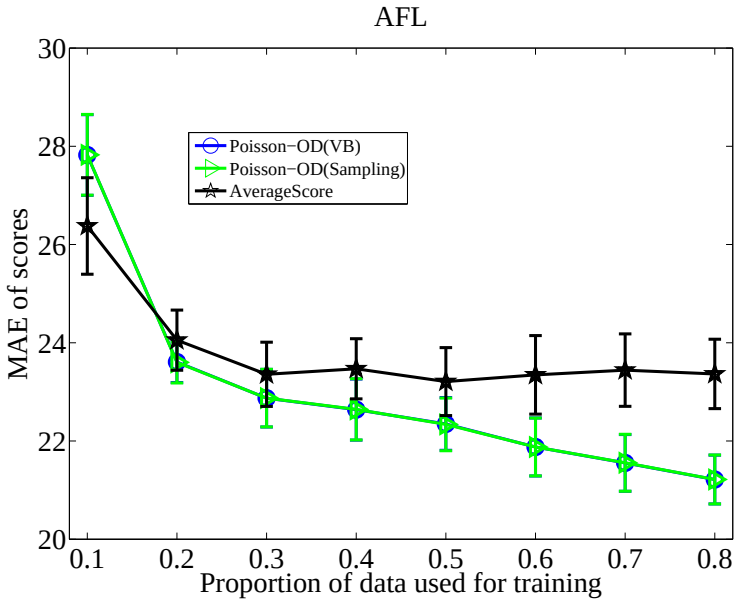

Fig. 15 Results on the AFL data set, evaluated using win/loss prediction accuracy in term of the area of the curve (AUC). Error bars indicate standard errors.

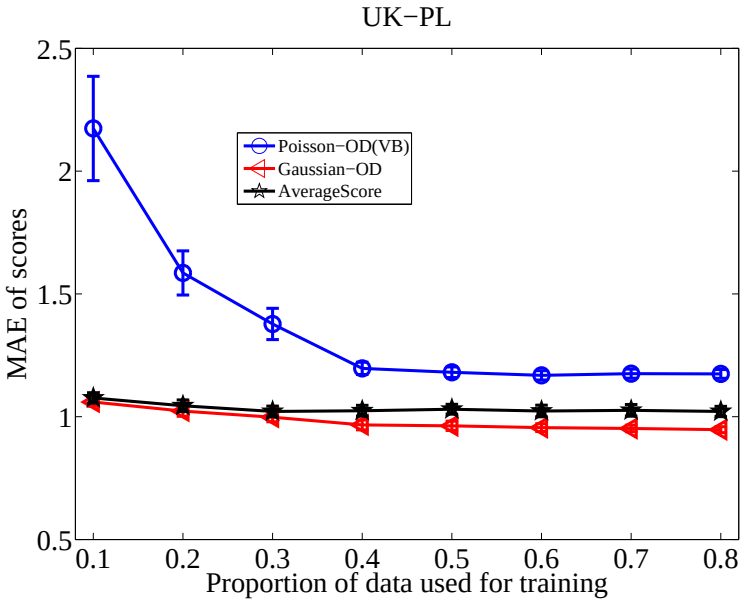

Fig. 16 Results on the UK-PL, evaluated using score prediction error (right column). Error bars indicate standard errors.

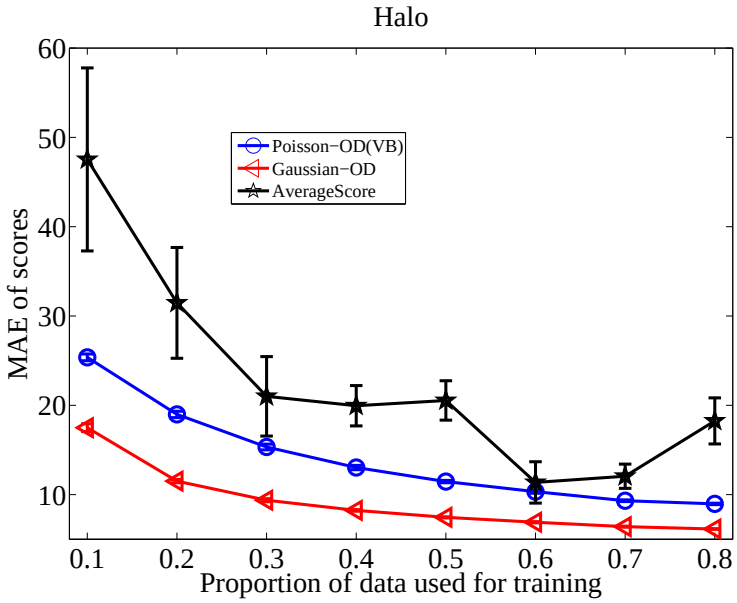

Fig. 17 Results on the Halo 2 data set, evaluated using score prediction error (right column). Error bars indicate standard errors.

can choose the models proposed in the present paper depending on the specific applications in order to achieve satisfactory results.

5.4 Poisson-OD(VB) Vs Poisson-OD(Sampling)

Finally, we studied the performance for the Poisson-OD model when approximate inference is conducted by variational Bayes and slice sampling. On the AFL data set, we show the score prediction results in Figure **??**. We observed that the differences in the performances achieved by the Poisson-OD model with VB and slice sampling are negligible; however, it is important to note that the slice sampling takes about 20 seconds; however, our proposed fixed-point solution often converges after two or three iterations that can be finished within less than 0.01 seconds. Note that all the experiments are conducted on a laptop with Intel i5 CPU, 4G memory, and codes are implemented in Matlab. Thus, our proposed variational Bayes inference achieves comparable performance with the slice sampling, but our method is much more efficient than the sampling based approach. For this reason, we can omit the report of the Poisson-OD(Sampling) on other data sets due to heavy computational requirements. Note that one can perhaps explore the advanced sampling approaches such as [23] to speed up the sampling method.

6 Related Work

Skill rating dates at least as far back as the Elo system [9], the idea of which is to model the probability of the possible game outcome as a function of the two players' skill levels. Players' skill levels are updated after each game in a way such that the observed game outcome becomes more likely and the summation of players' ratings remains unchanged.

The Elo system cannot handle the case when three or more teams participate in one match, a disadvantage addressed by TrueSkill [10]. Further extensions of TrueSkill incorporate time-dependent skill modeling for historical data [8].

In [7], the authors model and learn the correlation between all players' skills when updating skill beliefs, and develop a method called "EP-Correlated", contrasted with the independent assumption on players' skills (EP-Independent). Empirically, EP-Correlated outperforms EP-Independent on professional tennis match results; this suggests modeling correlations in extensions of the score-based learning presented here.

These skill learning methods all share a common feature that they are restricted to model WLD only and have to discard meaningful information carried with scores. While we proposed score-based extensions of TrueSkill in this work; it remains to incorporate other extensions motivated by this related work.

Score modeling has been studied since the 1950s [20] [21] [16] [19] [18]; one of the most popular score models is the Poisson model, first presented in [20], and this work continues to the present [18]. Other commonly used score models are based on normal distributions [16]. However, it appears that most score-based models do not distinguish offence and defence skills of each team and the results here indicate that such separate offence/defence skill models can perform better than univariate models with limited data.

More recently, [5] introduced a log-linear random effect model to model the number of goals for a football match, which takes into account home field advantages and distinguish teams' attack and defense skills, and proposed a Bayesian hierarchical model to generate the match outcomes in terms of scores. Inference in the model is conducted by MCMC, which can be slow as discussed in Section 5.4.

7 Conclusion

We proposed novel score-based, online Bayesian skill learning extensions of TrueSkill that modeled (1) player's offence and defence skills separately and (2) how these offence and defence skills interact to generate scores. Overall these new models — and Gaussian-OD (using a separate offence/defence skill model) in particular — show an often improved ability to model winning probability and win/loss prediction accuracy over TrueSkill, especially when the amount of training data is limited. This indicates that there is indeed

useful information in score-based outcomes that is ignored by TrueSkill and that separate offence/defence skill modeling does help (c.f. the performance of Gaussian-OD vs. Gaussian-SD). Furthermore, these new models allow the prediction of scores (unlike TrueSkill), with the Poisson-OD model and its variational Bayesian update derived in Section 4.2 performing best on the high-scoring AFL data. Altogether, these results suggest the potential advantages of score-based Bayesian skill learning over state-of-the-art WLD-based skill learning approaches like TrueSkill.

Future research could combine the proposed models with related work that models home field advantage, time-dependent skills, multi-team games, and correlated skills to utilise score-based outcomes.

Acknowledgements We thank Marconi Barbosa, Guillaume Bouchard, David Stern and Onno Zoeter for interesting discussions, and we also thank the anonymous reviwers at ECML-PKDD'12 for their constructive comments, which help to improve the paper. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. Bryan L. Boulier and H. O. Stekler *Predicting the outcomes of National Football League games.* International Journal of Forecasting, 19(2):257-270, 2003
2. G. W. Brier *Verification of forecasts expressed in probabilities.* Montly Weather Review, 78:1-3, 1950
3. T. K. Huang and R. C. Weng and C. J. Lin *Generalized Bradley-Terry Models and Multi-Class Probability Estimates.* Journal of Machine Learning Research, 7:85-115, 2006
4. R. M. Neal *Slice sampling.* The Annals of Statistics, 31(3):705-767, 2003.
5. G. Baio and M. A. Blangiardo *Bayesian hierarchical model for the prediction of football results.* JOURNAL OF APPLIED STATISTICS, 37(2):253-264, 2010.
6. M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables.* Dover Publications, New York, 1974.
7. A. Birlutiu and T. Heskes. Expectation propagation for rating players in sports competitions. In *ECML-PKDD*, volume 4702 of *LNCS*, pages 374–381. Springer, 2007.
8. P. Dangauthier, R. Herbrich, T. Minka, and T. Graepel. Trueskill through time: Revisiting the history of chess. In *NIPS*, pages 337–344. MIT Press, Cambridge, MA, 2008.
9. A. E. Elo. *The rating of chess players: past and present.* Arco Publishing, New York, 1978.
10. R. Herbrich, T. Minka, and T. Graepel. Trueskill^{TM}: A Bayesian skill rating system. In *NIPS*, pages 569–576, 2006.
11. D. Karlis and I. Ntzoufras. Bayesian modelling of football outcomes: using the skellam's distribution for the goal difference. *IMA Journal of Management Mathematics*, 20(2):133–145, 2009.
12. F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, February 2001.
13. T. Minka. Expectation propagation for approximate bayesian inference. In *UAI*, pages 362–369. Morgan Kaufmann, 2001.
14. M. J. Moroney. *Facts from figures.* Penguin Press Science, 3rd edition, 1956.
15. J. G. Skellam. The frequency distribution of the difference between two Poisson variates belonging to different populations. *Journal of the Royal Statistical Society: Series A*, 109(3):296, 1946.
16. M. E. Glickman, and H. S. Stern. A state-space model for football league scores. *Journal of the American Statistical Association*, 93(441):25–35, 1998.

17. M. J. Beal, and Z. Ghahramani. The Variational Bayesian EM Algorithm for Incomplete Data: with Application to Scoring Graphical Model Structures. *Proceedings of the Seventh Valencia International Meeting*: 453–464, 2002.
18. D. Karlis and I. Ntzoufras, Bayesian modelling of football outcomes: using the Skellam's distribution for the goal difference, *IMA Journal of Management Mathematics*, 20(2):133–145, 2009.
19. D. Karlis and I. Ntzoufras, Analysis of Sports Data by Using Bivariate Poisson Models, *Journal of the Royal Statistical Society: Series D*, 52(3):381-393, 2003.
20. M. J. Moroney Facts from figures, Penguin Press Science, 3rd, 1956.
21. M. J. Dixon and S. G. Coles, Modelling Association Football Scores and Inefficiencies in the Football Betting Market, *Journal of the Royal Statistical Society: Series C*, 46(2):265–280, 1997.
22. S. Guo and S. Sanner and T. Graepel and W. Buntine *Score-based Bayesian Skill Learning*. Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 106-121, 2012
23. I. Murray and R. P. Adams and D. J.C. MacKay *Elliptical slice sampling*. Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 541-548, 2010

Appendix

A Exponential Integral

Suppose that x is a random variable with Gaussian distribution, i.e., $p(x) := \mathcal{N}(x; \mu, \sigma^2)$, we present the derivations of the expectation for the $\exp(x)$ w.r.t. x as follows:

$$
\begin{aligned}
&E_{x\sim p(x)}(\exp(x)) = \int_x \frac{\exp(x)}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \mathrm{d}x \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\mu^2}{2\sigma^2}\right) \int_x \exp\left(-\frac{x^2 - 2x(\mu+\sigma^2)}{2\sigma^2}\right) \mathrm{d}x \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\mu + \frac{\sigma^2}{2}\right) \int_x \exp\left(-\frac{(x-(\mu+\sigma^2))^2}{2\sigma^2}\right) \mathrm{d}x \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\mu + \frac{\sigma^2}{2}\right) \sqrt{2\pi\sigma^2} \\
&= \exp(\mu + \sigma^2/2).
\end{aligned}
$$

B Log Gaussian Integral

Suppose x is a random variable with Gaussian distribution $p(x) : \mathcal{N} \sim (\mu, \sigma^2)$ and $q(x)$ is a Gaussian, $\mathcal{N} \sim (\mu_1, \sigma_1^2)$, let us show how to derive the expectation of $\log q(x)$ w.r.t. x as follows:

$$
\begin{aligned}
&E_{x\sim p(x)}(\log q(x)) = E_{x\sim p(x)}\left(\log\left(\frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)\right)\right) \\
&= -\frac{1}{2}\log(2\pi\sigma_1^2) - \frac{1}{2\sigma_1^2} E_{x\sim p(x)}(x-\mu_1)^2 \\
&= -\frac{1}{2}\log(2\pi\sigma_1^2) - \frac{1}{2\sigma_1^2}\left(E_{x\sim p(x)}(x^2) - 2\mu_1\mu + \mu_1^2\right) \\
&= -\frac{1}{2}\log(2\pi\sigma_1^2) - \frac{1}{2\sigma_1^2}\left(\sigma^2 + \mu^2 - 2\mu_1\mu + \mu_1^2\right).
\end{aligned}
$$