Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Competing with Humans at Fantasy Football: Team Formation in Large Partially-Observable Domains

Tim Matthews and **Sarvapali D. Ramchurn**
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
{*tm1e10,sdr*}*@ecs.soton.ac.uk*

Georgios Chalkiadakis
Dept. of Electronic and Computer Engineering
Technical University of Crete
73100 Chania, Crete, Greece
gehalk@intelligence.tuc.gr

Abstract

We present the first real-world benchmark for sequentially-optimal team formation, working within the framework of a class of online football prediction games known as Fantasy Football. We model the problem as a Bayesian reinforcement learning one, where the action space is exponential in the number of players and where the decision maker's beliefs are over multiple characteristics of each footballer. We then exploit domain knowledge to construct computationally tractable solution techniques in order to build a competitive automated Fantasy Football manager. Thus, we are able to establish the baseline performance in this domain, even without complete information on footballers' performances (accessible to human managers), showing that our agent is able to rank at around the top percentile when pitched against 2.5M human players.

1 Introduction

In many real-world domains, a number of actors, each with their own abilities or characteristics, need to be teamed up to serve a task in order to achieve some common objective (e.g., maximising rewards or reducing inefficiencies). Especially when there is uncertainty over these characteristics, forming the best possible team is often a lengthy process involving replacing certain members with others. This fact naturally gives rise to the problem of identifying the sequence of team formation decisions with maximal value over time, for example in choosing the best sensors to surveil an area (Dang et al. 2006), dispatching of optimal teams of emergency responders (Ramchurn et al. 2010), or optimal relaying in ad hoc networks. To date, however, the lack of datasets and the ability to test sequential team formation algorithms in such domains means that there is no real-world validation of such algorithms. In this paper, we introduce and solve the sequential team formation problem posed by a popular online Fantasy Football game known as Fantasy Premier League (FPL), where a participant's task (as manager) is to repeatedly select highly-constrained sets of players in order to maximise a score reflecting the real-world performances of those selected players in the English Premier League. Given the uncertainty in each player's performance (e.g., due to injury, morale loss, or facing a stronger team) and the cost of exchanging players with previously unselected ones, it is important to properly consider future events in order to maximise the final score at the end of the season. The task is particularly challenging from a computational perspective as there are more than 500 possible footballers, selectable in over 10^{25} ways, and competitors must make 38 such selections over the season.

This problem is reminiscent of work within the multi-agent systems literature on determining in a sequentially optimal manner a team of service providing agents (Teacy et al. 2008), or the appropriate set of agents to work with in a coalition formation problem (Chalkiadakis and Boutilier 2010). Both of these approaches employ *Bayesian reinforcement learning* techniques to identify the most rewarding decisions over time. Bayesian agents maintain a *prior* over their uncertainty, representing their *beliefs* about the world, and are able to *explore optimally* (Bellman 1961). By being Bayesian, the approaches of (Teacy et al. 2008; Chalkiadakis and Boutilier 2010) are thus able to make optimal team formation decisions over time. However, they both operate on (essentially) synthetic problems, of a relatively small size. Naturally, it is fundamental to assess the usefulness of such techniques in large real-world problems.

Against this background, in this paper we develop an automated FPL manager by modelling the FPL game dynamics and building principled solutions to the sequential team formation problem it poses. More specifically, we model the manager's decision problem as a belief-state Markov decision process[1] and attempt to efficiently approximate its solution. This paper makes the following contributions. First, we provide the first real-world benchmark for the Fantasy Football problem which allows us to pitch an automated player against human players. We consider the fact that our manager achieves around the top percentile when facing 2.5M human players to be particularly encouraging. Second, we present progressively more principled methods in terms of their handling of uncertainty and demonstrate how exploiting model uncertainty can guide the search over the space of selectable teams. Finally, we compare the performance of different solution approaches and draw conclusions as to the applicability of such techniques in large real-world domains.

[1] Often known as a partially-observable Markov decision process.

The rest of the paper is structured as follows. Section 2 gives a brief high-level outline of the dynamics of FPL and Section 3 goes on to model this environment formally in terms of a belief-state Markov decision process. Section 4 then outlines techniques to solve the problem and these are empirically evaluated in Section 5. Section 6 concludes.

2 Background on Fantasy Football

Our automated player operates according to the rules and datasets of the official English Premier League (EPL) Fantasy Football game available at fantasy.premierleague.com (FPL). This is primarily due to the large number of competitors it attracts (around 2.5M) and the availability of relevant data. FPL operates as follows: before the football season commences, its 380 fixtures are split into a set of 38 chronological *gameweeks*, each gameweek typically consisting of 10 matches and featuring each of the EPL's twenty teams once. All matches within a gameweek are usually contested within a period of three to four days. Furthermore, the FPL organisers appraise each of the footballers in the EPL with a numerical 'purchase price' chosen to reflect his point-scoring potential, and assign each footballer to one of four positional categories depending on his real-world playing position.

Prior to each gameweek, a competing FPL manager is required to select a team of fifteen players from the more than 500 available. The total purchase price of the team must not exceed a given budget (equal for all managers), and must feature exactly two goalkeepers, five defenders, five midfielders, and three strikers, with no more than three players permitted from any one club. Eleven of these fifteen players must be designated as constituting the team's 'starting line-up'. These eleven players earn *points* for the team depending on their contributions during the gameweek[2] – if they do not play they are replaced by one of the four players not in the starting line-up. Figure 2 depicts (part of) the view that managers use to pick players for their team (or squad) on the FPL website.

Crucially, managers are penalised for selecting too many players who they did not select in the previous gameweek — typically only one unpenalised exchange is permitted, with extra exchanges subject to a four point penalty. This requires managers to select players who will perform well over *multiple* forthcoming gameweeks rather than just the next one. The overall aim is thus to maximise the total points obtained over the 38 gameweeks by selecting players likely to make key contributions during matches, in the face of numerous selection constraints, uncertainty in player and club abilities, and the unpredictability of the dynamic football environment. In the next section we formalise the framework given above and set out the design of an agent able to perform effectively within it.

[2]For more details on the FPL rules see http://fantasy.premierleague.com/rules/. Other (trivial) caveats exist within the FPL rules which, for simplicity, have been omitted from the above description but are handled in our model.

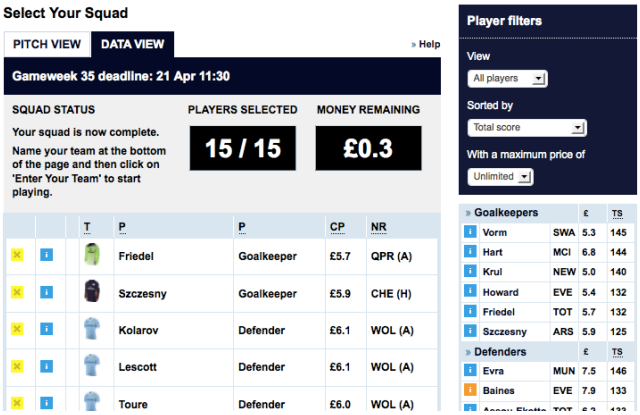

Figure 1: Partial snapshot of the team selection view for gameweek 35 in the 2011-2012 season. Note the costs for the players and the total score (TS) they have achieved so far in the right hand column.

3 Modelling FPL

Here we develop a model of the FPL as a sequential team formation problem. We first formalise the problem as a Markov decision process (MDP) and then adapt it to incorporate uncertainty by phrasing it in terms of a belief-state MDP using a Bayesian belief model of player abilities.

3.1 Basic Definitions

For each forthcoming gameweek a manager must select a team of players that obeys all the constraints imposed by the FPL rules. Formally, for the ith gameweek, the manager is aware of the set of matches to be played $\mathcal{M}_i$, the set of players available for selection $\mathcal{P}_i$, and the set of performable actions $\mathcal{A}_i$, where an action is defined as the selection of a valid team such that each $a \in \mathcal{A}_i$ is a subset of $\mathcal{P}_i$ and obeys all team selection constraints. Each player $p \in \mathcal{P}_i$ is associated with its FPL-designated position and purchase price (both the subject of team selection constraints) and $\tau_p \in \boldsymbol{\tau}$, a system of distributions representing their influence on matchplay. The set of possible outcomes of $\mathcal{M}_i$ is denoted as $\mathcal{O}_i$, with each outcome $o \in \mathcal{O}_i$ taken to consist of the result of the matches in $\mathcal{M}_i$ as well as player-specific contributions that are influenced by $\boldsymbol{\tau}$ (such as goals scored). As these contributions (and the match result) are related to the player characteristics, the probability of each $o \in \mathcal{O}_i$ is dependent in some way on $\boldsymbol{\tau}$, $\Pr(o|\boldsymbol{\tau})$. From this we may also define our immediate reward function $R(o, a_{\text{prev}}, a)$ that, given an outcome o, a selected team $a \in \mathcal{A}_i$, and the previously selected team $a_{\text{prev}} \in \mathcal{A}_{i-1}$, returns the point score of a (as defined by the FPL rules) according to what events occurred in o. a_{prev} is supplied so that the selection may be penalised for any player exchanges beyond those permitted.

3.2 Formulation as an MDP

We now formulate the above as an MDP with a set of states, set of actions, transition dynamics, and reward function. The MDP state for gameweek i encapsulates $\mathcal{M}_{i,\dots,38}$, the

set of upcoming fixtures, $\mathcal{P}_i$, the set of selectable players, $o \in \mathcal{O}_{i-1}$, the outcome of the previous gameweek, and $\boldsymbol{\tau}$, representing player abilities. The MDP action set is $\mathcal{A}_i$, the set of teams selectable at gameweek i, with R corresponding to the MDP reward function.

However, the state transition function is dependent on the distribution $\Pr(o|\boldsymbol{\tau})$ (where $o \in \mathcal{O}_i$), which is unknown due to our uncertainty of both the player abilities represented by $\boldsymbol{\tau}$ and the dynamics influencing the conditional distribution of o. We may instead adopt a *reinforcement learning* approach, operating under uncertainty regarding the underlying MDP dynamics and learning a Markovian policy which maximises performance based on the results of interactions with the environment. To this end, in the next section we formalise our uncertainty over $\boldsymbol{\tau}$ by defining a statistical model representing our beliefs, where those beliefs are refined and updated in response to gameweek outcome observations. We then use this model as the basis for a belief-state MDP formulation in Section 3.4.

3.3 Belief model

Here we introduce a generative belief model allowing us to represent our uncertainty over player abilities and, in turn, to generate $\boldsymbol{\tau}$ samples from the distribution $\Pr(\boldsymbol{\tau}|b)$.

Previous statistical models of football have mainly focused at a level of resolution necessary to model full-time match scorelines rather than modelling the individual player contributions required in our case. As statistical modelling is not the focus of our work, we choose to build a simple player-based model based upon an existing team-based framework (Dixon and Robinson 1998). We use this as the basis of our belief model because of its flexible treatment of football matchplay as a dynamic situation-dependent process that has been shown to return good results in football betting applications. The model works by estimating each club's attacking and defending abilities from past results and then using these estimates to derive the probabilities of either side scoring a goal at any point within a given match.

There are a number of different point-scoring categories defined in the FPL rules but for simplicity we focus on the most significant ones: appearances, goal scoring, and goal creating. Furthermore, a player's propensity to concede goals and to keep clean sheets may be derived entirely from the scoreline distributions produced by the underlying team-focused model and so requires no special player-specific attention. To this end, we define each player p's τ_p as consisting of three distributions:

- A three-state categorical distribution, ρ_p which can take values $start$, sub, or $unused$, describing player p's probability of starting a match, being substituted into the match, and being completely unused respectively.
- A Bernoulli distribution (or, equivalently, a Binomial distribution over a single trial), ω_p, describing player p's probability of scoring a goal given that he was playing at the time.
- Another Bernoulli distribution, ψ_p, describing player p's probability of creating a goal for a teammate given that he was playing at the time.

To account for uncertainty in these quantities we define prior distributions over the parameters, updating these priors as observations arrive in order to obtain new posterior distributions incorporating the new knowledge. For categorical and Bernoulli distributions such as those above, the updates can be performed via simple closed-form equations using Beta and Dirichlet (a generalisation of the Beta) *conjugate priors* (Gelman 2004). Sampling from these conjugate distributions thus allows us to obtain instantiations of τ_p. We define the hyperparameters uniformly across all players such that $\omega_p \sim \mathrm{Beta}(0,5)$, $\psi_p \sim \mathrm{Beta}(0,5)$, and $\rho_p \sim \mathrm{Dir}(\frac{1}{4},\frac{1}{4},\frac{1}{2})$. However, for many players we may also use performance data from previous seasons to define the priors, and in Section 5 we evaluate the effect of using this more informative approach.

We also define four global multinomial distributions (one for each of the four FPL-defined playing positions) S_{pos} that describe the distribution of minutes players who occupy position pos are observed to leave the match, given that they started it. A value of 90 in any of these distributions corresponds to instances of players finishing a match without being substituted. In using these distributions we adopt the simplifying assumption that all players occupying the same position have the same patterns of minutes in which they leave the match.

Now, players may also be suddenly unavailable to play for temporary, well-reported reasons, such as injury, disciplinary suspension, or international duty. For this reason we encode a list of roughly one thousand player absences recorded in media publications over the 2009/10 and 2010/11 seasons. During a period of absence for player i, we enforce that $\Pr(\rho_i = start)$ and $\Pr(\rho_i = sub)$ equal zero, and suppress updates to ρ_i. Finally, we introduce φ to describe the proportion of goals that are associated with an assist. On the datasets at our disposal we calculate $\varphi = 0.866$.

We show how this model may be used as the basis for a belief-state MDP in the next section.

3.4 Formulation as a Belief-state MDP

In formulating the FPL problem as a belief-state MDP we adopt an approach similar to (Teacy et al. 2008), maintaining prior distributions over the characteristics held in $\boldsymbol{\tau}$. This is done using the belief model introduced in the previous section. Our belief state at gameweek i, b_i is then an instantiation of the model updated with all outcome observations prior to gameweek i. On updating the belief state to b_{i+1} in response to an outcome $o \in \mathcal{O}_i$, the posterior player characteristics may be obtained by application of Bayes rule: $\Pr(\boldsymbol{\tau}|b_{i+1}) \propto \Pr(o|\boldsymbol{\tau}) \Pr(\boldsymbol{\tau}|b_i)$. The manager can then perform optimally, based on its current belief of player characteristics b_i, by maximising the value of the Bellman (1957) equations:

$$V(b_i) = \max_{a \in \mathcal{A}_i} Q(b_i, a) \tag{1}$$

$$Q(b_i, a) = \int_{\boldsymbol{\tau}} \Pr(\boldsymbol{\tau}|b_i) \int_{o \in \mathcal{O}_i} \Pr(o|\boldsymbol{\tau})[r_i + \gamma V_i(b_{i+1})] \,\mathrm{d}o \,\mathrm{d}\boldsymbol{\tau} \tag{2}$$

where $\gamma \in [0, 1)$ is a discount factor influencing the extent to which the manager should consider long-term effects

of team selection, and r_i represents the immediate reward yielded by $R(o, a_{\text{prev}}, a)$. Equation (2) thus returns the long-term discounted cumulative reward of performing a, a quantity known as a's *Q-value*.

In summary, a manager may perform optimally over a season by iteratively performing the following procedure for each gameweek $i = 1, \ldots, 38$:

- Receive observation tuple: $\langle \mathcal{P}_i, \mathcal{M}_{i,\cdots,38}, o \in \mathcal{O}_{i-1}, a_{i-1} \rangle$.
- Update b_{i-1} to obtain b_i and $\Pr(\boldsymbol{\tau}|b_i)$, using Bayes rule.
- Select $a \in \mathcal{A}_i$ that maximises (2).

Exact solutions to equations (1) and (2) are often in practice intractable. In our particular case this is due to the size of the outcome set $|\mathcal{O}_i|$, the size of the action set $|\mathcal{A}_i|$ (comprised of over 10^{25} actions), and the need to consider up to 38 gameweeks in order to calculate Q-values exactly. The latter issue may be solved without greatly sacrificing optimality by imposing a maximum recursion depth beyond which (1) is defined to return zero. The first two issues may be alleviated through the use of sampling procedures: in the next section we outline a simple procedure for sampling from $\mathcal{O}_i$ by simulating match outcomes, and in Section 3.6 we detail how high-quality actions can be generated from $\mathcal{A}_i$ by treating team formation as a constraint optimisation problem.

3.5 Sampling outcomes

The following routine describes a simple match process model able to sample outcomes for gameweek i from $\Pr(\mathcal{O}_i|\boldsymbol{\tau})$. We then combine this routine with the belief model described in Section 3.3 in order to sample from the joint distribution of observations and abilities, $\Pr(\mathcal{O}_i|\boldsymbol{\tau}) \Pr(\boldsymbol{\tau}|b_i)$, thus treating uncertainty in player abilities in a Bayesian manner. The routine described below focuses on simulating the outcome of a single match, but extends naturally to sampling the outcomes of a gameweek by applying the process in turn to each match within that gameweek. We use P_H and P_A to represent the set of players available for the home and away sides respectively. The routine below focuses on P_H, but applies identically to P_A.

First, we sample τ_p for each $p \in P_H$ from $\Pr(\tau_p|b_i)$. Next, eleven players from P_H are randomly selected in proportion to $\Pr(\rho_p = start)$. These players constitute L_H, the home side's starting line-up. Furthermore, the minute at which each of these players leaves the pitch is sampled from the S distribution corresponding to that player's position. All players in P_H that are not in L_H are consigned to another set U_H, representing the club's unselected players. We then proceed as per the match process of (Dixon and Robinson 1998) with two differences:

- At the start of each minute we check if any player in L_H is scheduled to leave the pitch in that minute. If so, we remove this player from L_H and randomly select a replacement $p \in U_H$ in proportion to $\Pr(\rho_p = sub)$. The replacement is added to L_H and removed from U_H. We also assume that players are never substituted off after being substituted on – a suitably rare event to not justify explicit consideration.
- If a goal is scored according to the underlying team-based model then it is allocated to a player $p \in L_H$ in proportion to $\Pr(\omega_p = 1)$ while an assist is allocated in proportion to $\Pr(\psi_p = 1)$ (with the restriction that a player may not assist his own goal).

Despite the simplicity of the method above (there is no attempt to capture at a deeper level the many considerations influencing line-up and substitution selection) it provides a reasonable estimate of the point-scoring dynamics for a single match.[3] These point estimates may then be used in combination with the MDP reward function R to approximate the immediate expected reward from performing any action, as well as to guide exploration of high-quality regions of the action space, as we show in the next section.

3.6 Generating actions

Using the outcome sampling procedure defined in the previous section we are able to approximate the expected points score of each player within the dataset. By treating team selection as an optimisation problem we may use these expectations to generate high-quality actions, thus avoiding an expensive search over the vast action space. This section outlines a means of doing this by phrasing the problem of team selection in terms of a multi-dimensional knapsack packing problem (MKP). The general form for an MKP problem is given as per (Kellerer, Pferschy, and Pisinger 2004):

$$\begin{aligned} \text{maximise} \quad & \sum_{i=1}^{n} v_i x_i, \\ \text{subject to} \quad & \sum_{i=1}^{n} w_{ij} x_i \leq c_j, \quad j = 1, \ldots, m, \\ & x_i \in \{0, 1\}, \qquad\quad i = 1, \ldots, n. \end{aligned}$$

MKPs require selecting some subset of n items that attains the maximum total value across all possible subsets, where each item $i = 1, \ldots, n$ is associated with a value v_i and m costs ($\boldsymbol{w}_i$). The total for each of the m costs of the items packed must not exceed corresponding capacities $c_{1,\ldots,m}$. Applied to team selection, the 'items' in the definition above are equivalent to the players available for selection. $\boldsymbol{v}$ then corresponds to the expectation of the point total for each player derived from outcomes generated using the sampling procedure in Section 3.5. Our capacities — in accordance with the FPL rules — are as follows:

- The team must be formed of exactly fifteen players.
- The fifteen players must comprise of two goalkeepers, five defenders, five midfielders, and three strikers.
- The total purchase price of the selected players must not exceed the available budget.
- Up to three players from any one club may be selected.
- Only a restricted number of unpenalised exchanges are permitted. The ability to selectively perform extra exchanges is implemented by introducing negative-weight

[3] After training the model on data from the 2009/10 EPL season the normalised root mean square error between observed point scores and expected point scores (calculated over 5000 match simulation samples) for the 2010/11 season is 0.09.

'dummy' items with $v = -4$, allowing an extra player selection. Selecting these items permits an extra exchange at the expense of a four point penalty, as per the FPL rules.

The resulting MKP can be solved using Integer Programming solvers such as IBM ILOG's CPLEX 12.3. The resulting selection can then be formed into a team by greedily filling the starting line-up with the selected players according to $\boldsymbol{v}$ and the FPL formation criteria.

As the generated selection is dependent on $\boldsymbol{v}$ (and the number of outcome samples n_s used to approximate the expectations held in $\boldsymbol{v}$) then as $n_s \to \infty$ we will generate the selection consisting of the fifteen players with the highest summed points expectation. However, due to tenets of the FPL game not captured within the MKP formulation, such as substitution rules, this generated selection does not necessarily correspond to the team in the action space with the highest immediate reward. Furthermore the generated selection is only myopically optimal (which we evaluate in Section 5) and not necessarily the best selection across *multiple* gameweeks. For these reasons it is desirable for us to explore more of the variability of the action space so as to possibly generate better quality long-term selections; this can be done by generating teams using lower values of n_s. Hence, in the next section we outline techniques that may be used to solve the FPL MDP using the belief model and sampling procedures described.

4 Solving the Belief-state MDP

Using the techniques described in the previous section we are able to sample good-quality actions and approximate their associated immediate reward. However, solving equation (1) in Section 3.4 still presents a challenge due to the computational cost of calculating each action's *long-term* reward, i.e., its Q-value. We may consider a naive depth-first search (DFS) approach where we solve (1) by walking down the recursive structure of the equation up to some fixed depth d, generating only n teams at each step (we evaluate such an approach in Section 5). However, DFS has time complexity $O(n^d)$, and so we can expect even modest search depths to be computationally unsatisfactory. Hence, in what follows, we provide an outline of a well-known reinforcement learning technique known as Q-learning in order to remove this exponential growth in d. An improvement to better handle uncertainty is presented in Section 4.2, and we adapt the techniques to FPL in Section 4.3

4.1 Basics of Q-Learning

Q-Learning is a technique for discovering the highest-quality action by iteratively learning the Q-values of actions in the action space and focusing exploration on actions with the highest Q-value estimates (Watkins 1989). Q-learning approaches run in $O(\eta d)$, where η is the number of *episodes*. Each episode proceeds (shown in Algorithm 1) by iterating through belief-states up to the maximum depth d. In each state an action is selected from the action space based on current Q-value estimates (line 4), an outcome from performing the action is sampled (line 5), and a reward associated with that outcome is determined (line 6). The Q-value estimate is then updated using the reward (line 7) (often using a simple exponential smoothing technique), and the belief-state updated based on the outcome (line 8).

Algorithm 1 Q-Learning algorithm to determine the best action performable in belief state b_0

function Q-LEARN(b_0, d)

```
1  for e = 1 → η
2      b = b_0
3      for i = 1 → d
4          a = SELECTACTION(b, i)
5          o = SAMPLEOUTCOME(b, a)
6          r = REWARD(a, o)
7          Q̂(b, a) = Q-UPDATE(b, a, r)
8          b = UPDATEBELIEF(b, o)
9      next
10 next
11 return arg max_a[Q̂(b_0, a)]
```

Exploration in such techniques is not particularly principled and Q-value convergence can be slow: it is possible for outcomes to be explored despite the fact that doing so is unlikely to reveal new information, or for promising actions to be starved out by 'unlucky' outcome sampling. The next section introduces a Bayesian variation of Q-learning designed to remedy these shortcomings.

4.2 Bayesian Q-Learning

A Bayesian approach to Q-learning incorporates uncertainty around Q-values into action selection. (Dearden, Friedman, and Russell 1998) represent the knowledge regarding each Q-value as a normal distribution updated as reward observations arrive using a normal-gamma conjugate prior. Exploration using these distributions is handled elegantly through the concept of *value of perfect information* (VPI), where the VPI of performing action a with belief b is the extent by which learning its true Q-value, q_a^*, is expected to change our knowledge of $V(b)$. For the best known action a_1, we only learn anything from performing it if $q_{a_1}^*$ is now lower than the currently estimated Q-value of a_2, $\hat{q}_{a_2}$, the second-best action. Likewise, for all $a \neq a_1$, we only learn anything from performing a if q_a^* is now *greater* than $\hat{q}_{a_1}$. The *extent* by which q_a^* is greater than $\hat{q}_{a_1}$ represents the gain in knowledge (and vice-versa for a_1). In general for any a the gain of learning q_a^* is:

$$Gain_a(q_a^*) = \begin{cases} \max[\hat{q}_{a_2} - q_a^*, 0] & \text{if } a = a_1 \\ \max[q_a^* - \hat{q}_{a_1}, 0] & \text{if } a \neq a_1 \end{cases} \tag{3}$$

VPI is then defined as the expected gain from performing a:

$$VPI(a) = \int_{-\infty}^{\infty} Gain_a(x) \Pr(q_a^* = x)\, \mathrm{d}x \tag{4}$$

which may be calculated exactly using the marginal cumulative distribution over the normal-gamma mean (Teacy et al. 2012).

Now, Bayesian Q-learning can be implemented using the same framework shown in Algorithm 1 with two adjustments: SELECTACTION is modified for Bayesian Q-learning by returning the action with the highest value of

$\hat{Q}(b,a)+VPI(a)$; and Q-UPDATE is modified to implement the moment updating procedure of (Dearden, Friedman, and Russell 1998).

4.3 Adapting Q-learning to FPL

Q-learning techniques often assume the availability of the entire action set during operation but the size of this set in FPL means this is not feasible. We instead choose to operate on only a promising subset of the available actions at any one time, denoted $\mathcal{A}_b$: a size of just three was sufficient in experimentation, with further increases not leading to any corresponding performance benefit. We then intermittently replace weak members of $\mathcal{A}_b$ with newly generated members of the unexplored action space. For traditional Q-learning this can be done simply by replacing the weakest member of A_b with a newly generated member at each decision point.

For Bayesian Q-learning we instead use VPI as an indicator of how much worth there is in continuing to explore action $a \in \mathcal{A}_b$, such that when $\hat{q}_a + VPI(a) < \hat{q}_{a_1}$ we choose to replace a with a newly-generated action. In so doing, we are able to avoid wasteful execution of actions unlikely to provide us with more information beyond that which is already known, and are able to explore more of the ungenerated action space.

We initialise a given action's Bayesian Q-learning normal-gamma hyperparameters $(\mu, \lambda, \alpha, \beta)$ such that $\alpha = 2$, $\lambda = 1$, and μ is chosen to equal a sampled approximation of the reward obtained by performing the action unchanged up to the search depth. β is set to $\theta^2 \mathrm{M}_2$ where M_2 is the value of the second moment used in the moment updating procedure. This defines the initial normal-gamma variance to equal some proportion θ of its initial mean μ, where θ is selected to provide a trade-off between over-exploration and neglect of newly-generated actions.

Having described different techniques to solve the belief-state MDP posed by FPL, we next proceed to evaluate these approaches empirically to determine their performance against human players of FPL.

5 Evaluation

Here we pitch different approaches to solving the sequential decision problem presented by the FPL game against each other and against human players. Model parameters are trained on datasets covering the 2009/10 EPL season and each approach is evaluated over between 30 and 50 iterations of the 2010/11 EPL season and its average end-of-season score recorded. Where scores are shown, standard errors and the approximate corresponding rank are displayed in brackets. In order to compete with humans on a level playing field we provide each manager with the ability to play a *wild-card* in the 8th and 23rd gameweeks, a benefit available to human competitors that absolves them of exchange penalties for that gameweek (that is, they may replace their whole team unpenalised if they so wish).

5.1 Effect of player type priors

We first consider a baseline manager (`M1`) that is naive in three different respects: it acts myopically, only considering the forthcoming gameweek; its player ability distributions are uniformly defined across the dataset (as per Section 3.3); and it always selects a team according to the expectation of these distributions (approximated with $n_s = 5000$), without taking into account the uncertainty therein. `M1` achieves a score of 1981.3 (SE: 8.0, Rank: 113,921). We also create a manager `M2` which defines the player ability priors to reflect the occurrences of the previous season (i.e., 2009/2010 EPL). Although this still leaves many players who did not appear that season with uniform priors, performance is generally greatly improved, yielding a mean end-of-season score of 2021.8 (SE: 8.3, Rank: 60,633). achieving the 2.5th ranking percentile compared to 4.6 for `M1`.

5.2 Variability of the action space

We hypothesised in Section 3.6 that a further score boost may be realised by generating multiple teams sampled to capture more of the variability of the action space. Thus we define manager `M3` that generates 40 candidate teams at each gameweek with $n_s = 20$, instead of just a single team with $n_s = 5000$ as for `M1` and `M2`. In this way a score of 2034.4 (SE: 8.5, Rank: 50,076) is achieved, approximately the 2nd ranking percentile.

5.3 Long-term uncertainty

We then investigate the effect of increasing search depth on the resulting score by assessing managers that consider future effects of their actions (i.e. using DFS) as discussed in Section 4. The best discount factor for such managers is determined to be around $\gamma = 0.5$. For a `DFS` manager conducting depth-first search with $d = 2$, 40 candidates generated at each search node, and $n_s = 20$, a mean score of 2049.8 (SE: 11.1, Rank: 37,137) is obtained, near the 1.5th ranking percentile. However, further increases in search depth lead to exponential increases in computation time: a search depth of three results in the manager taking around forty minutes per decision, so we do not evaluate deeper depths for the depth-first search manager.

To combat this we use the linear complexity Q-learning (QL) algorithms detailed in Section 4. Updates are performed using the mean across 100 simulation samples and initially are limited to just one minute running time per gameweek. QL is assessed with smoothing parameter $\delta = 0.1$ and selects actions using a ϵ-Greedy selection strategy (Sutton and Barto 1998) with $\epsilon = 0.5$. The best parametrisations of both approaches were for $d = 3$ with team generation performed with $n_s = 20$. Both give similar scores: traditional QL (`QL-60`) averaging 2046.9 (SE: 12.6, Rank: 39,574), Bayesian QL (`BQL-60`) reaching 2056.7 (SE: 8.6, Rank: 31,924). Performance deteriorated for $d \geq 4$, most probably because the time constraints imposed hindered exploration. Finally, the best QL parametrisations were re-assessed with a more generous time limit of three minutes and further small increases in mean score were obtained: 2049.9 (SE: 9.5, Rank: 37,053) for QL (`QL-180`); and 2068.5 (SE: 9.0, Rank: 26,065) for Bayesian Q-learning (`BQL-180`), corresponding to percentile ranks around the 1.5th and 1.1st percentiles respectively. Scores for each of the implementations above are summarised in Table 5.3

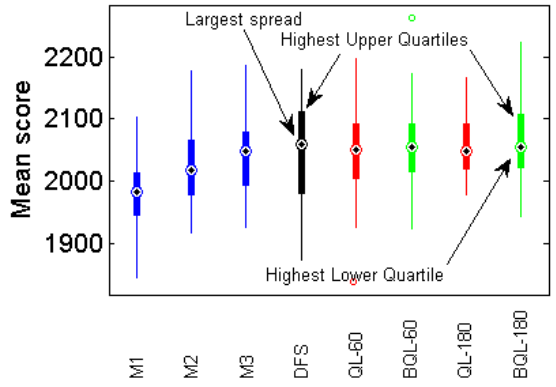

Figure 2: Boxplots for all manager types with whiskers from minimum to maximum.

	d	n_t	n_s	Score	Rank	Time (s)
M1	1	1	5000	1981.3	113,921	3
M2	1	1	5000	2021.8	60,633	3
M3	1	40	20	2034.4	50,076	7
DFS	2	40	20	2049.8	37,137	61
QL-60	3	-	20	2046.9	39,574	60
BQL-60	3	-	20	2056.7	31,924	60
QL-180	3	-	20	2049.9	37,053	180
BQL-180	3	-	20	2068.5	26,065	180

Table 1: Summary of mean end-of-season score, corresponding rank, and deliberation time per decision point for managers. d: search depth, n_t: teams generated per search node, n_s: number of samples per generated team.

and a boxplot illustrating point spread is shown in Figure 5.3. This provides some insight into the only modest performance increase for QL-180 and BQL-180 over QL-60 and BQL-60 despite being permitted three times the previous deliberation time: whilst the central tendency of the scores is not particularly influenced, there appears to be a reduced chance of performing poorly as evidenced by the position of the lower quartiles. This effect is further exaggerated in the score spread of DFS which also obtains a similar median score, but is far more erratic in its spread, achieving low scores fairly often.

6 Conclusion

In this paper, we developed a competitive and fully-automated agent for the FPL. Specifically, we modelled the FPL sequential team formation problem as a belief-state MDP which captures the uncertainty in player contributions. Moreover, given the complexity of the domain, we provide a computationally tractable and principled approach to handling such uncertainty in this domain with a Bayesian Q-learning (BQL) algorithm. Our evaluation of BQL against other uncertainty-agnostic approaches on a dataset covering the 2010/11 season of the FPL, shows that BQL outperforms other approaches in terms of mean final score, reaching around the top percentile on average, and in its best case where 2222 points were obtained, within the top 500 players. When taken together, our results establish the first benchmarks for the FPL and more importantly, the first real-world benchmarks for sequential team formation algorithms in general. Future work will look at developing other algorithms and improving parameter selection to improve scores and computation time.

Acknowledgements

Tim Matthews was supported by a EPSRC Doctoral Training Grant. Sarvapali D. Ramchurn was supported by the ORCHID project (EP/I011587/1). Georgios Chalkiadakis was partially supported by the European Commission FP7-ICT Cognitive Systems, Interaction, and Robotics under the contract #270180 (NOPTILUS).

References

Bellman, R. 1957. *Dynamic Programming*. Princeton University Press.

Bellman, R. 1961. *Adaptive Control Processes: A guided tour*. Princeton Uni. Press.

Chalkiadakis, G., and Boutilier, C. 2010. Sequentially optimal repeated coalition formation under uncertainty. *Autonomous Agents and Multi-Agent Systems* 24(3).

Dang, V. D.; Dash, R. K.; Rogers, A.; and Jennings, N. R. 2006. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proc. of the 21st National Conference on Artificial Intelligence (AAAI-2006)*, 635–640.

Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian q-learning. In *Proc. of the National Conference on Artificial Intelligence*, 761–768.

Dixon, M., and Robinson, M. 1998. A birth process model for association football matches. *Journal of the Royal Statistical Society: Series D (The Statistician)* 47(3):523–538.

Gelman, A. 2004. *Bayesian data analysis*. Chapman & Hall/CRC.

Kellerer, H.; Pferschy, U.; and Pisinger, D. 2004. *Knapsack problems*. Springer Verlag.

Ramchurn, S. D.; Polukarov, M.; Farinelli, A.; Jennings, N.; and Truong, C. 2010. Coalition formation with spatial and temporal constraints. In *Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, 1181–1188.

Sutton, R., and Barto, A. 1998. *Reinforcement learning: An introduction*, volume 116. Cambridge Univ Press.

Teacy, W.; Chalkiadakis, G.; Rogers, A.; and Jennings, N. 2008. Sequential decision making with untrustworthy service providers. In *Proc. of the 7th Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, 755–762.

Teacy, W.; Chalkiadakis, G.; Farinelli, A.; Rogers, A.; Jennings, N. R.; McClean, S.; and Parr, G. 2012. Decentralised bayesian reinforcement learning for online agent collaboration. In *Proc. 11th Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems*.

Watkins, C. 1989. *Learning from delayed rewards*. Ph.D. Dissertation, King's College, Cambridge.