Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Knowledge-Based Systems xxx (2006) xxx–xxx
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

Predicting football results using Bayesian nets and other machine learning techniques

A. Joseph *, N.E. Fenton, M. Neil

RADAR (Risk Assessment and Decision Analysis Research) Group, Queen Mary, University of London, UK

Received 21 April 2005; accepted 6 April 2006

Abstract

Bayesian networks (BNs) provide a means for representing, displaying, and making available in a usable form the knowledge of experts in a given field. In this paper, we look at the performance of an expert constructed BN compared with other machine learning (ML) techniques for predicting the outcome (win, lose, or draw) of matches played by Tottenham Hotspur Football Club. The period under study was 1995–1997 – the expert BN was constructed at the start of that period, based almost exclusively on subjective judgement. Our objective was to determine retrospectively the comparative accuracy of the expert BN compared to some alternative ML models that were built using data from the two-year period. The additional ML techniques considered were: MC4, a decision tree learner; Naive Bayesian learner; Data Driven Bayesian (a BN whose structure and node probability tables are learnt entirely from data); and a *K*-nearest neighbour learner. The results show that the expert BN is generally superior to the other techniques for this domain in predictive accuracy. The results are even more impressive for BNs given that, in a number of key respects, the study assumptions place them at a disadvantage. For example, we have assumed that the BN prediction is 'incorrect' if a BN predicts more than one outcome as equally most likely (whereas, in fact, such a prediction would prove valuable to somebody who could place an 'each way' bet on the outcome). Although the expert BN has now long been irrelevant (since it contains variables relating to key players who have retired or left the club) the results here tend to confirm the excellent potential of BNs when they are built by a reliable domain expert. The ability to provide accurate predictions without requiring much learning data are an obvious bonus in any domain where data are scarce. Moreover, the BN was relatively simple for the expert to build and its structure could be used again in this and similar types of problems.

Keywords: Bayesian nets; Machine learning; Football

1. Introduction

Bayesian networks [1], BNs, provide a means for capturing, displaying, and making available in a usable form knowledge, often obtained from experts in a given field. This knowledge is often obtained from experts and can be based on subjective judgements as well as (or even instead of) data. Predicting the outcome of a football match is an ideal application (although it is far removed from other applications we have been involved with such as [2,3,5]). It is in just this type of problem, with many complex interacting factors, that BNs excel. It is possible for a domain expert, in collaboration with a BN expert, to construct a network detailing the important relationships between the factors involved, and the node probability tables, (NPTs). In this paper, we look at the performance of an expert constructed BN in predicting the outcome (win, lose, or draw) of matches played by Tottenham Hotspur ('Spurs'). The BN was originally developed at the start of the 1995–96 season. Since, it involves specific players, the model was only relevant for two seasons (after which some of the key players were no longer at the club). Hence, the study is restricted to all league matches played by Spurs during the two

* Corresponding author. Tel.: +44 020 8597 9578; fax: +44 020 8980 6533.

E-mail addresses: adrianj@dcs.qmul.ac.uk (A. Joseph), norman@dcs.qmul.ac.uk (N.E. Fenton), martin@dcs.qmul.ac.uk (M. Neil).

URL: http://www.dcs.qmul.ac.uk/researchgp/radar/ (A. Joseph).

0950-7051/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2006.04.011

consecutive seasons 1995/1996 and 1996/1997. So why, almost 10 years after the expert BN was developed, have we returned to this particular problem? It is because we had a unique opportunity for a direct comparison between the expert BN and a range of alternative ML models. Such studies are relatively rare and the results and lessons learnt should be of interest to researchers outside of this particular domain (even those readers who have no interest in Spurs or football in general). The performance of the expert BN model is compared with four alternative machine learning (ML) models:

- A naive BN.
- A BN learnt from statistical relationships in the data [6].
- A *K*-nearest neighbour implementation [7].
- A decision tree [8].

The aim was to see how the expert constructed BN compares in terms of both predictive accuracy and explanatory clarity for the factors effecting the result of the matches under investigation.

Section 2 discusses the issues of model setup and how we selected the football match data to learn from. Section 3 is a brief explanation of the learning techniques used and our approach to the analysis. Section 4 provides the results of the learners for each of the data sets used, while Section 5 provides a summary of the predictive accuracy. Section 6 summarises our conclusions and looks at some possible directions of future work.

2. Selecting relevant information

There are a large number of factors which could effect the outcome of a football match from the perspective of one of the teams involved. One of the difficulties in any investigation of the relationships involved in a given effect is that to a large extent the assumption of a particular model determines the attributes to study and predetermines the possible relationships that can be found. So, the act of choosing which model and attributes to study sets a boundary on what can be discovered.

2.1. Constructing an initial model

When approaching a new problem there are two techniques which are commonly used. The first assumes we have some idea how the situation under investigation works, construct a model, and using this model select the attributes believed to contribute to the effect under investigation. An example of this approach to this type of problem is given in [9]. The second approach assumes little knowledge of the underlying mechanisms involved so we look at all the *probably* relevant attributes and try to determine those which have the most significant effect. This is still in effect the construction of an a priori model, but only a very informal one. In this paper, we take the second approach.

2.2. The expert model

The expert BN (see Fig. 1) uses only a few features:

- The presence or absence of three players, Sherringham, Anderton, and Armstrong. So in each match each of these values was true or false.
- The playing position of Wilson represented by him playing in midfield or not.
- The quality of the opposing team. This particular variable was measured on a simple 3-point scale (high, medium, and low). Although based on expert judgement, it matches closely with the teams' final league positions ('top 6', 'middle 8', or 'bottom 6') and so would appear to be an accurate reflection of their average performance.
- Venue (whether the game is played at Spurs' home ground or away).

The BN shows how the expert constructed the relationships between the chosen factors and the outcome of the game. In addition to the result node (win, lose, or draw) the BN includes three other nodes to simplify the structure:

- **Attack** which represents the quality of the Spurs attacking force (low, medium, and high).
- **Spurs_quality** the overall quality of the Spurs team (low, medium, and high).
- **Performance** how well the team will perform given their own quality and that of the opposition (low, medium, and high).

2.3. The general model and its known weaknesses

We allowed the machine learners to use both the same and an alternate set of features compared to the expert BN. Specifically, the initial set of factors were the basic factors in the expert model, plus all the other registered Spurs' players (as playing or not playing) rather than just the four 'special' players in the expert BN minus the playing position of Wilson. The particular values for Opposition quality in each game were the same as those used by the expert BN.

During a game players can be injured, substituted, be sent off, or have their playing positions changed. The solution chosen to deal with these issues was to use the information about only those players who started the game. Similarly Wilson's playing position could change during the course of the match, only his initial playing position was considered.

In general terms this problem is not particularly easy from a machine learning perspective. There is not much data to go on. We have the results of two seasons' games, a total of 76 matches and for the general model a total of 30 attributes, (28 players, venue, and opponent quality). There were changes to the Spurs' squad during this period. The simple convention of a player either playing or not was chosen to avoid having missing data entries with regards to

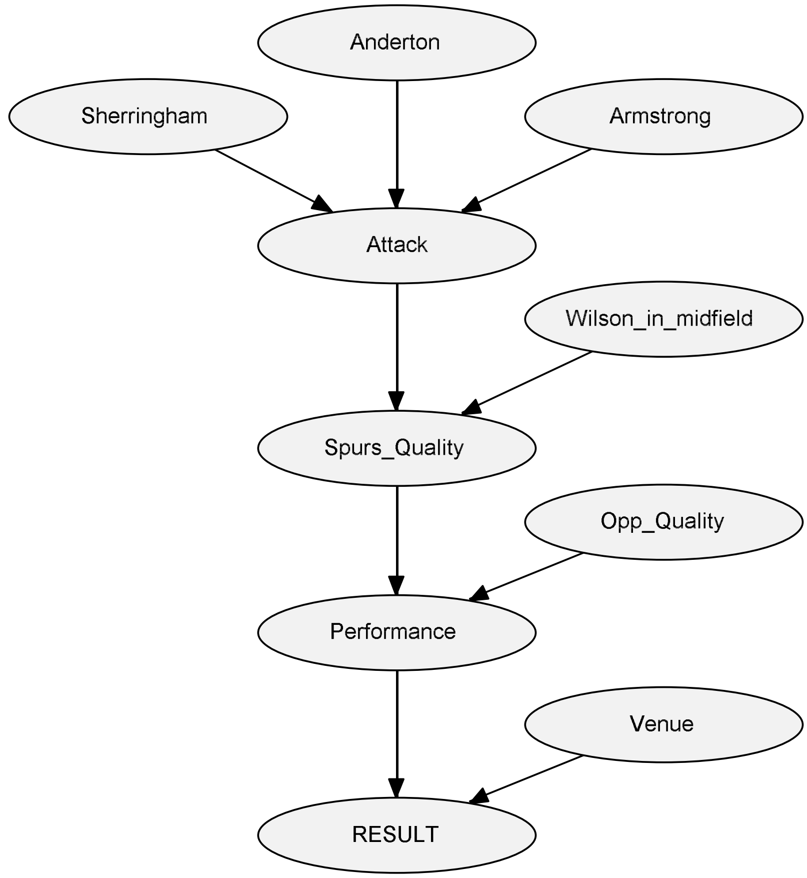

Fig. 1. Expert constructed BN for Tottenham Hotspur's performance.

squad changes. There are, of course, other external factors which effect the outcome of a game. So, even in the best case we expect to have noise in the data. Since players, except Wilson, are only considered from the point of view of playing or not playing, the effect of any player who was always present will be ignored. This is because the learners can only compare the difference in the outcome of matches with a player present or absent.

It is also worth noting that all the models (including the expert BN) are inherently asymmetric. Whereas for Spurs we consider the particular players involved in any given match to be significant, for all their opponents we only have a general rating for their overall quality.

3. Machine learning techniques and our analysis assumptions

There are a large number of ML techniques each with different strengths and weaknesses. Choosing which is the most appropriate technique often requires an understanding of both the problem domain and the different learning methods. A good introduction to many machine learning techniques can be found in [10]. The machine learners used in this analysis were:

MC4 Decision trees. Decision trees provide a visual representation of relationships which appear to effect the situation under investigation. Pruning is generally used to reduce the size of the tree. The confidence method of pruning was used.

Naive Bayesian learner. The Naive Bayesian learner makes the simplifying assumption that all the attributes are independent.

Data Driven Bayesian learner. The complex Bayesian learner as implemented by Hugin attempts to learn the structure of the network by looking at the correlation between the attributes. Once the structure has been determined data can then be used to determine the node probability tables. The strength of a correlation required to trigger the joining of two nodes can be adjusted.

Expert constructed Bayesian network. When expert knowledge of a given domain is to be represented as a BN the usual process is for the domain expert(s) and BN expert(s) to jointly construct the BN. If sufficient data are available then the NPTs can be directly learnt and then adjusted if required. However, when there is insufficient data to learn the NPTs these must also be obtained from the expert(s).

***K*-nearest neighbour.** *K*-nearest neighbour learners use a *likeness* approach to prediction. That is, they look at the instances most like the test case and usually have some voting method by which the prediction is

Table 1
Comparison of learner accuracy with expert model data

Train period–Test period	Number of correct predictions by learner					
	Most common	MC4	Naive BN	Hugin BN	Expert BN	KNN
95/96–95/96 season	16 (42.11%)	28 (73.68%)	26 (68.42%)	21 (55.26%)	20 (52.63%)	37 (97.37%)
96/97–96/97 season	18 (47.37%)	30 (78.95%)	31 (81.58%)	26 (68.42%)	25 (65.79%)	37 (97.37%)
Average for full seasons	17 (44.74%)	29 (76.32%)	28.5 (75.00%)	23.5 (61.84%)	22.5 (59.21%)	37 (97.37%)
Period 1–period 234 95/96	12 (42.86%)	8 (28.57%)	9 (32.14%)	8 (28.57%)	14 (50.00%)	12 (42.86%)
Period 12–period 34 95/96	7 (38.89%)	6 (33.33%)	6 (33.33%)	3 (16.67%)	10 (55.56%)	7 (38.89%)
Period 123–period 4 95/96	2 (25.00%)	2 (25.00%)	2 (25.00%)	2 (25.00%)	3 (37.50%)	2 (25.00%)
Sum for 1995/1996 periods	21 (38.89%)	16 (29.63%)	17 (31.48%)	13 (24.07%)	27 (50.00%)	21 (38.89%)
Period 1–period 234 96/97	11.5 (41.07%)	10 (35.71%)	13 (46.43%)	11 (39.29%)	19 (67.86%)	11 (39.29%)
Period 12–period 34 96/97	7.5 (41.67%)	7 (38.89%)	10 (55.56%)	3 (16.67%)	10 (55.56%)	5 (27.78%)
Period 123–period 4 96/97	5 (62.50%)	2 (25.00%)	5 (62.50%)	2 (25.00%)	3 (37.50%)	1 (12.50%)
Sum for 96/97 periods	24 (44.44%)	19 (35.19%)	28 (51.85%)	16 (29.63%)	32 (59.26%)	17 (31.48%)
Period 23 95/96–period 4/1 95/97	6 (33.33%)	4 (22.22%)	6 (33.33%)	Unavailable	9 (50.00%)	7 (38.89%)
Period 234 95/96–period 1 96/97	4 (40.00%)	2 (20.00%)	4 (40.00%)	3 (30.00%)	6 (60.00%)	3 (30.00%)
Period 34 95/96–period 12 96/97	8 (40.00%)	6 (30.00%)	8 (40.00%)	11 (55.00%)	15 (75.00%)	7 (35.00%)
Period 4 95/96–period 123 96/97	6 (20.00%)	8 (26.67%)	6 (20.00%)	10 (33.33%)	22 (73.33%)	8 (26.67%)
Period 4/1 95/97–period 23 96/7	6.67 (33.33%)	7 (35.00%)	8 (40.00%)	7 (35.00%)	16 (80.00%)	7 (35.00%)
Season 95/96–season 96/97	13 (34.21%)	8 (21.05%)	13 (34.21%)	20 (52.63%)	25 (65.79%)	15 (39.47%)
Sum for cross season periods	43.67 (32.11%)	35 (25.74%)	45 (33.09%)	51 (43.22%)	93 (68.38%)	47 (34.56%)
Overall average percentage	40.05%	41.72%	47.86%	39.69%	59.21%	50.58%
Overall disjoint training/data	38.48%	30.19%	38.81%	32.31%	59.21%	34.98%

chosen. The usual measure of *likeness* is Euclidean distance as plotted on an *n*-dimensional graph where each dimension is one of the supplied attributes.

All the learners used were part of the MLC++ [11] package[1] apart from the complex Bayesian learner which was part of the Hugin tool[2], the Hugin tool was also used to *run* the expert constructed BN.

The different models do not all provide the same sort of prediction. The MC4 and KNN learners usually give a prediction in the form of an unqualified value from the possible range of values. BNs do not make predictions in the same format as the MC4 or KNN learners. Rather than supply a simple answer they supply a probability for each of the possible outcomes. This allows for a greater sensitivity of prediction; the BN not only makes a prediction, but is also able to provide some idea of confidence in the prediction. To make a direct comparison with the learners we had to interpret the BN prediction as a definite result (win, lose, or draw). Our approach was to choose the result with the highest predicted probability, irrespective of how close two or more results might be. In cases where two or more of the outcomes of the BN were equally likely we deemed that the prediction was incorrect (even if the actual result was one of the two most likely). This approach clearly treats BNs harshly in the analysis. In reality, a prediction involving equal (or nearly equal) probabilities would be useful. For example, if we were betting on the outcome of a game, and the BN predicted Win 45\% Draw 45\% Loss 10\% then this would indicate a likely win for an each way bet. However, such an analysis of the potential value of a shared highest probability prediction is beyond the scope of this paper.

We divided the match data into disjoint subsets so that some could be used for training and separate data used to check the accuracy of the learners. The data for each season was divided up into three groups of ten matches and one group of eight matches, organised chronologically. We maintain the ordering of games and always organise the training so that the training data set are chronologically immediately before the test data set. For comparison we also used each complete season's data for training and test set for the learners. This again prejudices the results against the expert BN because this will tend to overestimate the accuracy of all the other learners. The machine learners were tested with both our general model data and with the data used by the expert BN. Using the two data sets allows for a direct comparison with the same, expert chosen, data set and a more general comparison with a data set a non expert might choose. The results for both the general data and the expert chosen data, shown in Tables 1 and 2, are similar. Where changes in classification error are mentioned they are relative to the error obtained by choosing the most common result from the training data.

4. Results analysis

In this section, we compare the accuracy of the different models' predictions (for some general information on making comparisons between learners see [12]). We also look

[1] Version 2.01 of the MLC++ libraries was used, modified to run under the GNU/Linux operating system. All the MLC++ learners were used with their default settings except where noted otherwise.

[2] Version 6.1 of this tool was used for this paper.

Table 2
Comparison of learner accuracy with expert model data

Train period–Test period	Number of correct predictions by learner					
	Most common	MC4	Naive BN	Hugin BN	Expert BN	KNN
95/96–95/96 season	16 (42.11%)	25 (65.79%)	22 (57.89%)	23 (60.53%)	20 (52.63%)	27 (71.05%)
96/97–96/97 season	18 (47.37%)	26 (68.42%)	25 (65.79%)	26 (68.42%)	25 (65.79%)	32 (84.21%)
Average for full seasons	17 (44.74%)	25.5 (67.11%)	23.5 (61.83%)	24.5 (64.47%)	22.5 (59.21%)	29.5 (77.63%)
Period 1–period 234 95/96	12 (42.86%)	8 (28.57%)	7 (25.00%)	8 (28.57%)	14 (50.00%)	9 (32.14%)
Period 12–period 34 95/96	7 (38.89%)	5 (27.78%)	9 (50.00%)	0 (0.00%)	10 (55.56%)	8 (44.44%)
Period 123–period 4 95/96	2 (25.00%)	4 (50.00%)	3 (37.50%)	2 (25.00%)	3 (37.50%)	4 (50.00%)
Sum for 1995/1996 periods	21 (38.89%)	17 (31.48%)	19 (35.19%)	10 (18.52%)	27 (50.00%)	21 (38.89%)
Period 1–period 234 96/97	11.5 (41.07%)	11 (39.26%)	12 (42.86%)	13 (46.43%)	19 (67.86%)	7 (25.00%)
Period 12–period 34 96/97	7.5 (41.67%)	6 (33.33%)	8 (44.44%)	6 (33.33%)	10 (55.56%)	8 (44.44%)
Period 123–period 4 96/97	5 (62.50%)	4 (50.00%)	2 (25.00%)	2 (25.00%)	3 (37.50%)	3 (37.50%)
Sum for 1996/1997 periods	24 (44.44%)	21 (38.89%)	22 (40.74%)	21 (38.89%)	32 (59.26%)	18 (33.33%)
Period 23 95/96–period 4/1 95/97	6 (33.33%)	7 (38.89%)	7 (30.89%)	7 (30.89%)	9 (50.00%)	8 (44.44%)
Period 234 95/96–period 1 96/97	4 (40.00%)	7 (70.00%)	3 (30.00%)	6 (60.00%)	6 (60.00%)	5 (50.00%)
Period 34 95/96–period 12 96/97	8 (40.00%)	14 (70.00%)	9 (45.00%)	11 (55.00%)	15 (75.00%)	11 (55.00%)
Period 4 95/96–period 123 96/97	6 (20.00%)	6 (20.00%)	8 (26.67%)	4 (13.33%)	22 (73.33%)	7 (23.33%)
Period 4/1 95/97–period 23 96/97	6.67 (33.33%)	6 (30.00%)	8 (40.00%)	6 (30.00%)	16 (80.00%)	8 (40.00%)
Season 95/96–season 96/97	13 (34.21%)	22 (57.89%)	13 (34.21%)	21 (55.26%)	25 (65.79%)	14 (36.84%)
Sum for cross season periods	43.67 (32.11%)	62 (45.59%)	48 (35.29%)	55 (40.44%)	93 (68.38%)	53 (38.97%)
Overall average percentage	40.05%	45.77%	42.26%	40.58%	59.21%	47.21%
Overall disjoint training/data sets	38.48%	38.65%	35.74%	32.62%	59.21%	37.06%

at any information provided by each model about the factors effecting the outcome of the games. Note that, because of space limitations, we do not include the full set of data and models. This is, however, all available on-line here [4].

4.1. The MC4 Learner

Decision tree learners like MC4 are good at dealing with relatively static situations, that is, situations in which the relationships between the various attributes are fixed. We were not sure how true this was of the Spurs team, and its performances, over the period being examined. The overall classification error of the MC4 learner for disjoint training and test data sets in the general model was 69.81% and 61.35% for the expert chosen data.

4.1.1. Complete seasons

The basic tree produced by MC4 when looking at the general model data for the 1995/1996 season is a fairly simple tree using only 6 of the available 30 attributes, the players Dozzell, Campbell, and Nethercott, the venue and the opposing team ranking. The tree, Fig. 2, shows Dozzell as a key player[3]. For the 1995/1996 season the MC4 analysis give a reduction in the classification error of 34.57% and 23.68% for the general and expert models, respectively.

An analysis of the 1996/1997 seasons matches produced a slightly more complex tree (which can be seen in [4]), using 8 rather than 6 attributes. MC4 analysis gives a reduction in the classification error of 31.58% using the general model and a reduction of 21.05% using the expert chosen data.

4.1.2. Separate training and test data – single season

The performance of the MC4 learner was, as expected, less impressive when it was only given part of a season's data and used to predict the remainder. The classification error for the tests using general model data from 1995/1996 season increased by 9.26%, and the same tests for the 1996/1997 season showed an increase in the error of 9.25%. The learner faired slightly better with the expert chosen data giving an increase in error of 7.41% and 5.55% for the 1995/1996 and 1996/1997 seasons, respectively. The performance of the learner did not seem to improve with increasing amounts of training data. The trees built by MC4 with increasing data develop towards that built with the full season's data.

The performance of the learner over all five cross season periods, for the general model, was quite poor. The classification error for the general model averaged over all the cross season tests increased by 6.37%. The learnt tree for the

[3] It is interesting to note that after seeing this analysis the expert stated that while he suspected Dozzell was a key player this was not the general opinion at that time and he thus left Dozzell out of the expert BN.

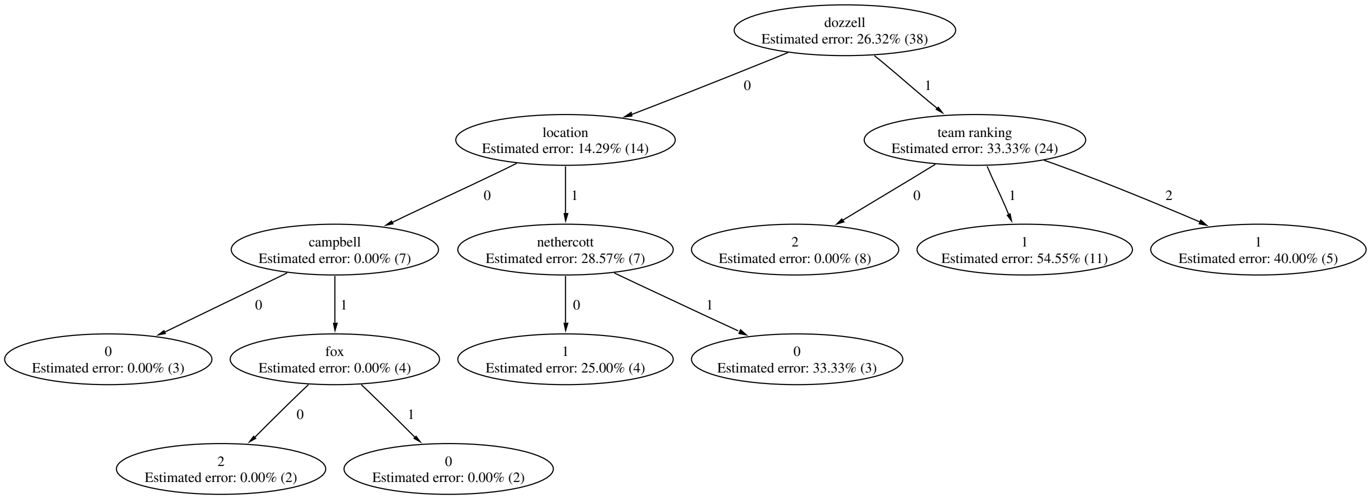

Fig. 2. Decision Tree for the general model 95/96 season with error estimates.

end of the 1995/1996 season, period 4, and the beginning of the 1996/1997 season, period 1, is the largest of the trees for any two period group. This may indicate that significant changes take place between seasons, which would not be contradicted by the slight drop in performance of cross season tests compared to similar intra-season tests. There is also a drop in the predictive ability of the most common test result which means that overall for the cross seasons tests the classification error from the MC4 learner was 6.37% worse than that from choosing the most common test result. Over the same period the expert chosen data gave a better result with an average reduction in the error of 13.48%.

4.2. Naive Bayesian learner

While the attributes of the problem do not adhere to the strict independence assumption of the naive Bayesian learner we would expect there to be a reasonable match and thus for this learner to perform relatively well. This is reflected in that for non-overlapping training and test data sets on the general model this learner came second overall with a classification error of 61.19%. Interestingly on the expert chosen data the naive Bayesian learner only came in fifth best with a classification error of 64.26%.

4.2.1. Complete seasons

For the 1995/1996 season the Naive Bayesian learner correctly predicted the result of 26 and 22 of the 38 games in the general and expert models respectively. This is a reduction in the classification error of about 26.31% and 15.78%. The naive Bayesian classifier gives no direct indication of the importance of any given attribute. However, looking at the NPT for the classifier in the general model we can see that the six most significant attributes in descending order are: Team Ranking, Dozzell, Edinburgh, Anderton, Du-mitrescu, and Calderwood. There is some, limited, agreement between MC4 and the naive Bayesian learner on the significant attributes, they agree on the two most important of the thirty attributes for the 1995/1996 season. For the 1996/1997 season the Naive Bayesian learner correctly predicted the result of 31 and 25 of the 38 games for the general and expert models, respectively. This is a reduction in the classification error of about 34.21% and 18.42%.

4.2.2. Separate training and test data – single season

The results for the 1995/1996 season showed the average classification error to be 7.41% and 3.70% higher for the general and expert data sets, respectively. However, for the 1996/1997 season the general model classification error was 7.41% lower while that for expert data set model increased by 3.70%. Most classifiers achieved better results for the 1996/1997 season than the 1995/1996 season which may indicate greater stability in the team in the later season.

4.2.3. Separate training and test data – cross seasons

The cross season results for the naive Bayesian learner were roughly comparable to its in-season results. Overall it achieved a classification accuracy of 33.09% and 35.29% for the general and expert models which only bettered the most common classifier by 0.98% and 3.18%, respectively. Ignoring the case using the same training and test data for the complete seasons, the naive Bayesian learner came out second best overall on the general model and fifth overall on the expert model.

4.3. Data driven Bayesian learner

The BNs for the data driven Bayesian learner were generated using the structural learning wizard from the Hugin Developer version 6.1 program. The process used was to run the program using an initial `Level of Significance` of 0.1. If no link directed to the **result** node was formed the process was rerun doubling the `Level of Significance` until a network with at least one link directed to the **result** node was achieved. Since, in this problem all of the nodes except the **result** node have their values specified any nodes in the network with no links directed to the **result** node were removed. The remaining network was used for the testing. The overall classification error of the various learnt networks for disjoint training and test data sets was 67.69% and 67.38% for the general and expert models, respectively.

4.3.1. Complete seasons

The learned network using the general data for 1995/1996 season is shown in Fig. 3. It is possibly significant that the two nodes with the greatest number of dependencies are **dozzell** and **wilson**. We know from our other analysis that these are two important players, but with the network as shown we are unable to usefully include them. A crucial feature of this network is the **result** node has no children and its only parent is the **team_ranking** node. Since, in this problem the data for all the nodes except **result** are specified, we can infer the outcome of the game simply by knowing the quality of the opposition, the other attributes become irrelevant if the **team ranking** is specified. See Section 6 for further comment on this issue. Using the quality of the opposing team it is possible to correctly predict the outcome of 21 of the 38 games for the 1995/1996 season. This amounts to a reduction in the classification error of 13.15%. Using the expert data for the 1995/1996 season the network obtained is that shown in Fig. 4. This network correctly predicted 23 of the 38 games for the season a reduction in error of 18.42%. The Hugin BN learnt networks for the general and expert models for the 1996/1997 season are identical, consisting of the **team_ranking** and **result** nodes. These particular networks were extracted using a `Level of Significance` of 0.1 for both models.

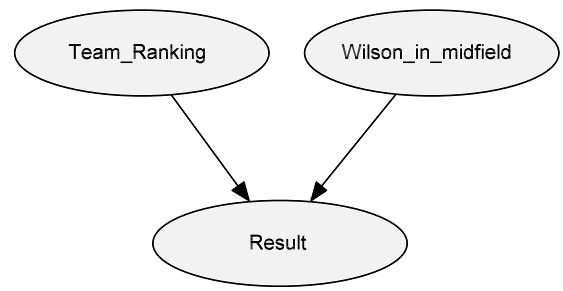

Fig. 4. Learnt BN for the expert model 95/96 season with `Level of Significance 0.1`.

4.3.2. Separate training and test data – single season

It is interesting to note that for the general model the attributes chosen by the Hugin learner for the periods in 1995/1996 season are a subset of those chosen by the MC4 learner for the same periods. There is a less strong relationship for the general model between the chosen attributes of the Hugin and MC4 learners for the 1996/1997, but still a lot of shared attributes. This is reasonable given that both learners are presumably choosing attributes with a strong correlation with the result. For both seasons the intra-season average classification error using the general data increased by 14.81%. Using the expert data set the average intra-season classification error increased by 20.37% and 5.55% for the 1995/1996 and 1996/1997 seasons, respectively Fig. 5.

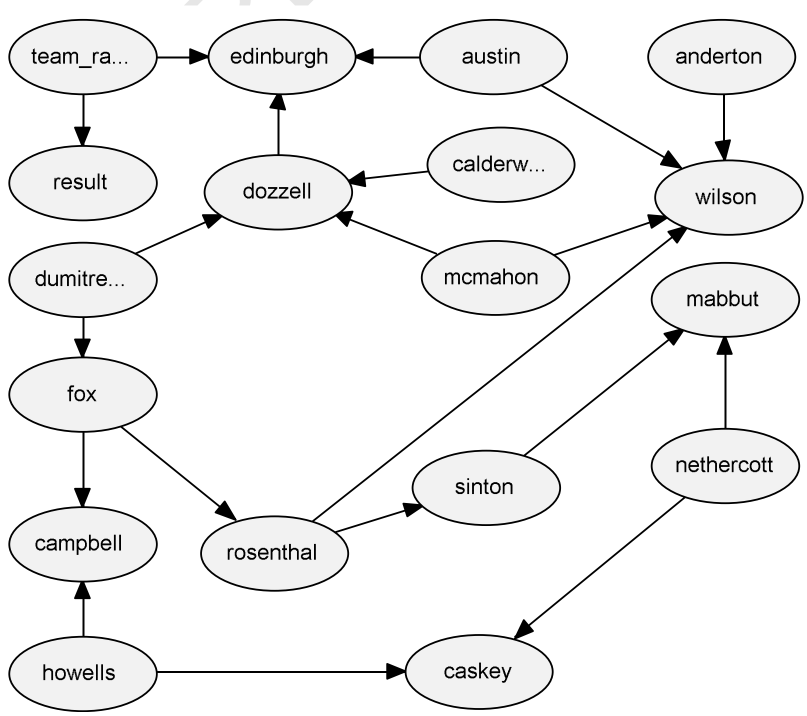

Fig. 3. Learnt BN for the general model 95/96 season with `Level of Significance 0.1`.

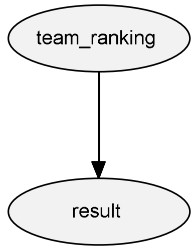

Fig. 5. Learnt BN for the general model 96/97 season with a `Level of Significance 0.1`.

4.3.3. Separate training and test data – cross seasons

Similar to the intra-season networks there is a striking similarity between the attributes chosen by the Hugin learner and the MC4 algorithm for the general model. We encountered a problem with the network produced by the Hugin learner for the period 2 and 3 general model data in the 1995/1996 season. This network crashed when we tried to run it so no results could be obtained for this training period. The classification error for the cross season data showed reductions of 11.11% and 8.33% for the general and expert data sets, respectively.

4.4. K-nearest neighbour

The IB classifier from the MLC++ library is a version of the *K*-nearest neighbour algorithm. In effect the KNN algorithm constructs a graph with as many dimensions as we have attributes. We are not aware of an easy to interpret representation for graphs of high dimension so we provide no visual representation of the model constructed by this learner. We chose to use 3 neighbours for the voting comparison in this paper. Overall for the disjoint training and test data sets KNN proved to be an average performer with a classification error of 65.02% and 62.94% for the general and expert models, respectively. However, as expected with the same training and test data provided KNN performs exceptionally.

4.4.1. Complete seasons

For the 1995/1996 season KNN correctly predicts the result of 37 of the 38 games for the general model and 27 games for the expert model data. This amounts to an error reductions of 55.26% and 28.94%. For the 1996/1997 season the KNN algorithm again correctly predicts the result of 37 of the 38 games for the general model and 32 for the expert model giving error reductions of 50.00% and 36.84%, respectively.

4.4.2. Separate training and data – single season

With separate training and test data sets the performance of the KNN learner dropped dramatically, and interestingly providing more training data did not seem to improve its performance. The overall classification error for the 1995/1996 season for both general and expert models was 61.11% and for the 1996/1997 season it was 68.52% and 66.67% for the general and expert models, respectively.

4.4.3. Separate training and data – cross seasons

Cross season performance was generally a bit weak for the KNN learner. This might be because of an inability to filter out unimportant attributes involved in cross season changes. KNN produced an overall classification error for the cross season test periods of 65.44% for the general and 61.03% for the expert models respectively.

4.5. Validation and overfitting

In this problem we would not expect to get a completely accurate classification for the outcome of a given game. We have only a small sample of data a situation that will tend to cause a strong bias towards the specific data set. However, what we are interested here is in the relative performance of each learner and, since each learner could be expected to generate the same data set bias, the comparisons should be valid. We also have a situation in which the underlying mechanisms that determine the performance of the football team, the members of the team, their playing positions, fitness and tactics can all change. We would not expect our chosen attributes to account for all of the likely variations so its difficult to determine what is a reasonable level of predictive accuracy to expect.

4.6. Expert constructed Bayesian network

We already noted that the expert BN (Fig. 1) contained 3 nodes **Attack**, **Spurs_Quality**, and **Performance**, which do not directly represent any of the supplied attributes or the result. These nodes are a result of the model the expert has built to capture more detailed relationships between the attributes and the result than those provided by the other learners. Another difference with the expert BN is that is does not use the supplied training data for any of the tests. The structure of the BN and the value of the NPTs have all been fixed by the expert. This means it is unable to take into account any change that may occur outside of the expert chosen attributes. Despite these limitations, and the inherent analysis bias against the BN already discussed, the expert BN was the most accurate predictor of the outcome of the Spurs games with a classification error over the disjoint training and test data sets of 40.79%. Since, the expert BN only used the expert data set only one set of accuracy figures are given.

4.6.1. Complete seasons

The expert BN is the only learner we would not expect to appear overly accurate when looking at a complete season's data for both training and testing as it does not use training data. The expert BN did better than the most common value predictions for both the 1995/1996 and 1996/1997 seasons with a classification error of 40.79%.

4.6.2. Separate training and test data – single season

The expert BN had its poorest performance on the data for the 1995/1996 season. This is not difficult to understand given that: Sherringham played in every match for Spurs during that season; Anderton played only 6 matches in the season; Armstrong played in all bar one game of the season; Wilson only played in midfield in 3 games in the season. Thus given its chosen set of attributes there was little variation the expert BN could produce over the 1995/1996 season. However, it is worth noting that with classification errors of 50.00% and 40.74% for the 1995/1996 and 1996/1997 seasons, respectively, it was still the best classifier for the intra-season data.

4.6.3. Separate training and test data – cross seasons

The expert BN produced the best results of any of the classifiers for every one of the cross season test periods. Since, it does not use the training data, any changes that occur between season not involving its key attributes are ignored. This is really a case of the expert being able to select the key features, and thus remove any other features which could adversely effect its predictions. However, in the case of something like a football team where over the course of a few seasons all the players may change it does potentially limit the useful lifetime of any given expert constructed BN. The classification error averaged 33.62% for the cross season data.

5. Predictive accuracy

Tables 1 and 2 show the relative accuracy of the different learners in predicting the outcome of the games using the general and expert model data, respectively. When using the same training and test data for the complete seasons all of the learners perform significantly better than the most common assumption with KNN as the best performer. When disjoint training and test data sets were used the performance of the KNN learner dropped significantly and the expert BN outperformed all the other learners. The learners generally performed similarly with both the general and expert chosen data sets.

6. Conclusions and way forward

The process of machine learning, and learning in general, provides us with two tangible benefits, understanding and prediction. While it is true that the better our understanding the better we should be able to make predictions, it is possible to make accurate predictions with limited understanding. We can treat these as qualitative and quantitative results from the learning process. The understanding we gain from the learning process allows us to construct models which reflect what we have learned about the relationships between the attributes and the relative importance of each attribute. In terms of the football matches it lets us see which of the selected attributes are the crucial factors effecting the outcome of a game, and gives some clues as to the relationships between some of those factors.

The different learning techniques vary in what they provide in terms of understanding of the interrelationships between the attributes and the outcome of a game. The MC4 learner identifies those attributes which have the largest effect on the outcome of the game. It shows their relationships to each other in terms of their effect on the outcome of the game. This is a very simplified model of the game itself. The naive Bayesian learner does not construct a model as such, its model is predefined. The learning process for the naive Bayesian learner is then simply one of discovering the relative strength, and polarity, of the effect of each attribute with respect to the result. The learnt BN looks for correlations between the values of the attributes including the result. Once a BN is constructed using the correlations that lie within the required sensitivity, then the NPTs can be learnt from the available data. KNN does not construct a model as such, it simply uses the existing data and provides a *likeness* comparison with any test data. Thus KNN does not significantly enhance our understanding. The expert constructed BN represents the knowledge of the expert, that is, it is a model is the expert's belief of the interrelationships between the attributes and their relative importance. One of the limitations of all the non expert methods used here is that they only use the supplied attributes. This is particularly limiting in its effect on the learnt BNs. In a problem where most of the supplied attributes have defined values the possible network structures for a learnt BN are very restricted and, in effect, become just reduced versions of the naive Bayesian model. While they are not observed the nodes **Attack**, **Spurs_Quality**, and **Performance** in the expert BN help build a model of the games Spurs played. This model gives us some additional insight into how the observed attributes effect the outcome of the game.

Given the inherent analysis bias against the BN model, its performance is genuinely impressive. Although the model has now long been irrelevant (since it contains variables relating to key players who have retired or left the club) the results here tend to confirm the excellent potential of BNs when they are built by a reliable domain expert. The ability to provide accurate predictions without requiring much learning data are an obvious bonus in any domain where data are scarce. Moreover, the BN was relatively simple for the expert to build and its basic structure could be used again in this and similar types of problems.

There are a number of directions in which future work could be done. As pointed out this method of prediction is inherently asymmetric. It should be possible to construct a more symmetrical model using similar data for all the teams in the league. However, this would involve at least multiplying the amount of computational work by the number of additional teams in the league. Another obvious potential improvement would be to qualify the inherent quality of each player who plays – a simple 3-point scale based on objective criteria like international performances could be feasible. This approach would provide much greater *longevity* to the model. Also, learning from the expert BN here, we could use abstract nodes like 'attack quality' and 'defence quality' to both improve the model and ensure its longevity.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan-Kaufmann, San Mateo, CA, 1988.

[2] N.E. Fenton, P. Krause, M. Neil, Software measurement: uncertainty and causal modelling, IEEE Software 10 (4) (2002) 116–122.

[3] N.E. Fenton, M. Neil, The jury observation fallacy and the use of Bayesian networks to present probabilistic legal arguments, Mathematics Today (Bulletin of the IMA) 36 (6) (2000) 180–187.

[4] A. Joseph, N.E. Fenton, M. Neil, Predicting football results using Bayesian nets and other machine learning techniques (version with full set of models and data), (2005), http://www.dcs.qmw.ac.uk/~norman/papers/Spurs-2.pdf.

[5] M. Neil, N. Fenton, S. Forey, R. Harris, Using Bayesian belief networks to predict the reliability of military vehicles, IEE Computing and Control Engineering Journal 12 (1) (2001) 11–20.

[6] D. Heckerman, D. Geiger, D. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, Machine Learning 20 (1995) 197–243.

[7] T. Cover, P. Heart, Nearest neighbour pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27.

[8] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.

[9] Håvard Rue, Øyvind Salvesen, Prediction and retrospective analysis of soccer matches in a league, preprint Statistics 10 (1997).

[10] Tom M. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.

[11] Ronny Kohavi, Dan Sommerfield, MLC++ machine learning library in C++, SGI http://www.sgi.com/tech/mlc/ (1996).

[12] Andrew Bradley, Brian Lovell, Michael Ray, Geoffrey Hawson, On the methodology for comparing learning algorithms: a case study, Australian and New Zealand Conference on Intelligent Information Systems, 37–41 (1994).