A Visual Data Analysis Tool for Sport Player Performance Benchmarking, Comparison and Change Detection *

Sebastian Lühr and Mihai Lazarescu
Institute for Multi-Sensor Processing and Content Analysis
Department of Computing, Curtin University of Technology
{S.Luhr, M.Lazarescu}@curtin.edu.au

Abstract

Sports coaches today have access to a wide variety of information sources that describe the performance of their players. However, despite this great wealth of information, most techniques used to analyse performance require a significant amount of manual processing and continue to rely heavily on input from human experts. In this paper we propose an automated approach to analyse player performance. Specifically, we propose a team benchmarking and concept drift tracking based system that (1) generates adaptive baseline player performance norms, (2) interprets player performance over different time lines and (3) identifies and describes key turning points in player performance. The concept drift technique that we describe uses a combination of overlapping data windows and decision tree based learning to process the data.

1 Introduction

Fast and accurate sports data processing and interpretation is an important issue for sport coaches. Although much progress has been made in collecting detailed and reliable information on players, the analysis of such data continues to rely on largely manual approaches that involve little or no automated data processing. In this paper we describe a set of techniques that have been developed with the aim of addressing the lack of automated tools for analysing player performance data along different time lines and with minimal expert supervision.

The first contribution of this work has been the development of an automated system for generating team benchmark performance norms in order to accurately identify specific areas of player performance that invite improvement. The design of the system was motivated by a need for users to be flexible in their exploration of their data; a coach may, for example, wish to a compare player performance from one tournament or series to another using a specific mixture of teammates or previously established benchmark norms.

Our second contribution tackles the issue of identifying salient change among time series data. As the problem of analysing and interpreting player performance over time is similar to tracking concept drift [14], we employ a multiple overlapping tracking windows approach to process player data. The novelty of this approach is twofold. First, it involves a relevance based data processing technique that considers the consistency and persistency of observed features over time. Second, it addresses a real world problem in an application area where machine learning has been of very limited use.

The system that we describe has been designed to assist coaches and players such that they do not require more than a minimal understanding of statistical analysis and machine intelligence techniques. Use of the software provides sport trainers with a highly flexible yet intuitive means of exploring their data. The work presented is part of an ongoing project done in collaboration with the Australian Institute of Sport.

2 Related Work

A variety of tools are currently available to coaches wishing to analyse and improve the performance of their players. Available technologies range from inexpensive video replay systems to motion analysis and high-end computer simulation software that can assist in tactical decision making. Tools aimed at honing player reflexes and at perfecting technical skills are also available [8].

One of the more accessible tools employed by coaches is video recording of matches for post-game analysis. Such video data is commonly annotated using notational analysis [4] methods to describe the state of play. The annotated data can then be used to objectively extract performance indi-

*This work has been partially supported by a grant from the Australian Institute of Sport.

1082-3409/07 $25.00 2007 ©IEEE

cators suitable for evaluating player and team performance and for determining patterns of play that are of interest to the team [5].

An example of how annotation data has been used to examine tactical plays in basketball is given in [13]. Here, hand crafted finite state machines were employed to model the actions and state of play prior to critical game events such as shots being made or control of the ball being lost. The resulting models allowed the effectiveness of in-game tactical decisions to be systematically analysed.

An overview of research into sports analysis that considers games as dynamical systems is presented in [10]. Here it is theorised that sporting games can be modelled as cyclical rhythms and that player actions which destabilises this rhythm, especially critical destabilising actions which may result in points being scored, are likely to be of interest to team analysts. Such work seeks to identify in-game events which may indicate a momentary imbalance in the state of play that can be exploited.

A data mining approach seeking to find frequently occurring inter-related events named T-patterns was introduced in [9]. Although initially applied to behavioural psychology research, these patterns have been used to analyse soccer games [2]. Video recordings of matches were annotated with the location of players on a grid representation of the field along with observed actions and events. The resulting patterns were shown to provide valuable feedback on the style and strategy of a team's play. T-pattern mining is similar to sequential pattern mining [1, 15]. They differ, however, in that the latter is interested only in the ordering of events within a time window without considering the strict temporal nature of events or actions being observed. T-pattern mining, in contrast, seeks relationships in which the timing between events is unvarying.

A methodological approach for statistical analysis of position-specific player performance attributes has recently been described [6]. Commercially available software was used to show that statistically significant differences exist in both interposition and intraposition performance metrics. This suggests that it is possible to compare the performance of players given a role-specific team benchmark. Our work extends this previous work by providing an intuitive interface that shield users from the full set of features available in commercial statistical analysis software. This has the advantage of both making the task of examining team and player performance simpler as well as protecting users unfamiliar with statistical approaches from making inadvertent errors. The software remains flexible, however, as it allows an arbitrary combination of players and matches to be used both in benchmarking and in player comparisons. Our work also seeks to identify significant change in a player's performance metrics from time series data.

The challenge we face is to develop a set of algorithms that can handle the type of sports data involved reliably and accurately. The work described in this paper is related to both data mining and concept drift tracking. There has been a large body of work done in stream data mining [3] but none that effectively deals with the type of processing required for this project. Stream data mining is a particular area where past research has focused on dealing with changing data sets. Most current algorithms require a reasonably sized data set to be effective (100 instance or more) and are designed to deal with *static* data sets. Streaming algorithms are limited, however, in several ways. First, they focus mainly on tracking current change rather than accurately detecting the starting point of observed change and estimating/predicting future change. Such algorithms also provide a very limited description of change and cannot handle recurrent change. The data in our case is *dynamic* in the sense that it undergoes change over time. It is also sparse, the rate at which international games are played throughout a year making it difficult to collect an adequate volume of data for stream mining techniques to be effective over short time spans. Furthermore, accurately pinpointing the start of change, estimating future player performance and possessing a complete description of change as it occurs along different time lines are all critical aspects of the analysis which we wish to address. For these reasons the current data mining algorithms and approaches are not well suited to this project.

3 Program Overview

Our software addresses three issues of player performance: player performance norms, the detection, estimation and description of the change in performance, and the estimation of future player performance.

The issue of player performance benchmarks and comparison is addressed in Section 4. Here, we employ statistics to find a team based performance benchmark with which to compare the performance of individual players. Section 5 addresses the latter two problem areas via an application of concept drift tracking and an ensemble of classification trees to detect rapid and subtle changes in performance from time series data.

The software takes as input the number of occurrences of both desirable and undesirable on-field events. This match data is imported from sports video data that is manually annotated using commercially available video annotation and indexing software. The time that a player spent on the field, the position that they played and the date of each match is also recorded.

4 Performance Benchmarking

Player benchmarking compares the performance of individual players against a group average represented by the mean and standard deviation of match-normalised attribute scores of each attribute a_k from the set of recorded attributes $A = \{a_1 \dots a_k \dots a_K\}$. Attribute benchmarks are calculated for each role r, or position, that is played on the field. The positions played can be ignored, however, if users wish to obtain more general benchmarks. As in [6], match-normalised attribute scores are obtained from a raw attribute count $x_{r,k}$ by: $x^{'}_{r,k} = x_{r,k}\left(\sqrt{\frac{m}{t}}\right)\left(\log_{10}\left(\frac{m}{t}\right)+1\right)$ where m is the length of a match and t is the number of minutes that a player was active within a game. Normalisation in this way reduces the likelihood of player performance being grossly overcompensated in situations where players are on the field for only a few minutes, reducing the likelihood benchmarks being skewed.

The standard deviation $s_{r,k}$ for a role r and attribute k is obtained by: $s_{r,k} = \theta\sqrt{\frac{1}{N_r}\sum_{i=1}^{N_r}\left(x^{'}_{r,k} - \bar{x}^{'}_{r,k}\right)^2}$ where $\bar{x}^{'}_{r,k}$ is the mean of each normalised score $x^{'}_{r,k}$ and where θ is a scaling factor found by using a Gaussian distribution look-up table to map a user supplied confidence level to the unit scale. A player's attribute a_k can then be said to be above or below the benchmark if their match-normalised performance is outside the benchmark norm $\bar{x}^{'}_{r,k} \pm s_{r,k}$.

4.1 User Adjustable Parameters

Users are able to experiment with the benchmark and comparison by adjusting the statistical confidence level and the *game threshold*. The confidence level allows trainers to specify a level of certainty that coaches are willing to accept when examining whether players are performing above or below the team benchmark. This confidence level is used to define the value of θ that is applied as the scaling factor in Section 4.

The game threshold adjusts the percentage of games that must be played above or below the benchmark in order for a player's attribute to be of interest. This threshold is only active when multiple games are returned by the comparison filters. It serves as a means of adjusting the sensitivity of the comparison.

The matches and players used in the benchmark is constrained through a set of user definable filters. Filters allow an arbitrary selection of games, players and positions to be used to define both the benchmark and the comparison. Filters for both the benchmark and the comparison default to include all matches and all players for which data is available. It is possible to restrict the benchmark or comparison to consider groups of players or to just a single player. The active matches can similarly be constrained to a number of recent games, to the games within a specific tournament, to a specific group of games or to a single game. The positions played on field can also be used as a filter.

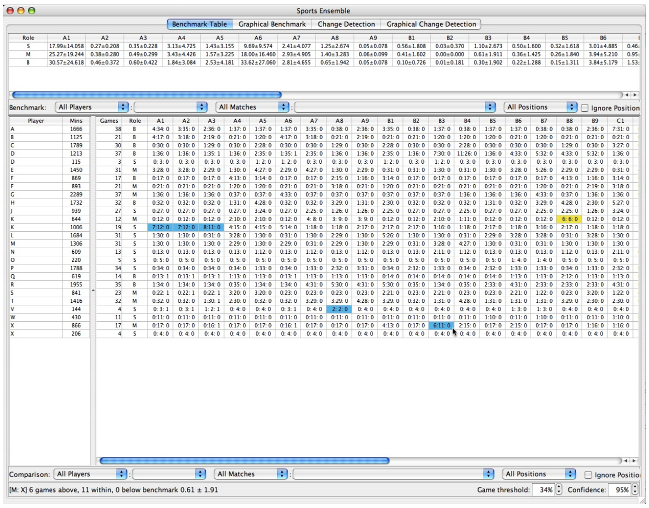

Figure 1. The tabular performance benchmark and comparison view. Attribute names have been removed by request.

4.2 Interface

The team benchmark and player comparison interface is available in two forms: a tabular view that presents all available numeric data in a compact form and a graphical view that retains the tabular benchmark component of the former while graphically charting performance comparisons.

4.2.1 Tabular Benchmark Comparison

The user interface for the tabular benchmarking component of the program is depicted in Figure 1. The top portion of the screen shows the the currently active benchmark while the lower shows a comparison of the currently selected match data against the benchmark.

The benchmark pane displays, for each role, the mean and the scaled standard deviation of each attribute given the currently active filters. The comparison pane is similar to the benchmark table with the addition of three new columns showing: the name of each player, the total number of minutes played in the selected comparison matches and the number of games that are being compared.

Two different types of views, dependent on the number of games returned by the active filter set, are used in the comparison pane. The number of games played above the benchmark, within the benchmark and below the benchmark are shown when the comparison filter returns multiple games. Cells are coloured blue when the match-normalised score of an attribute is above the benchmark for an arbitrary percentage of games specified by the game threshold. Cells

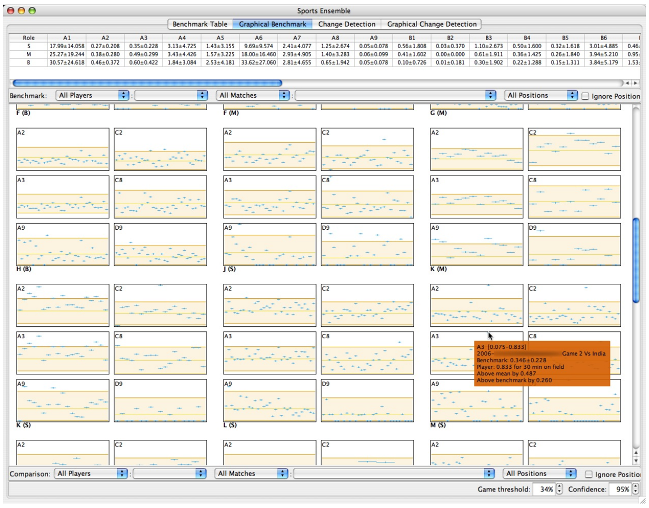

Figure 2. The graphical performance benchmark and comparison view in line graph mode. Each graph only shows abstract performance summaries. Attribute names have been removed by request.

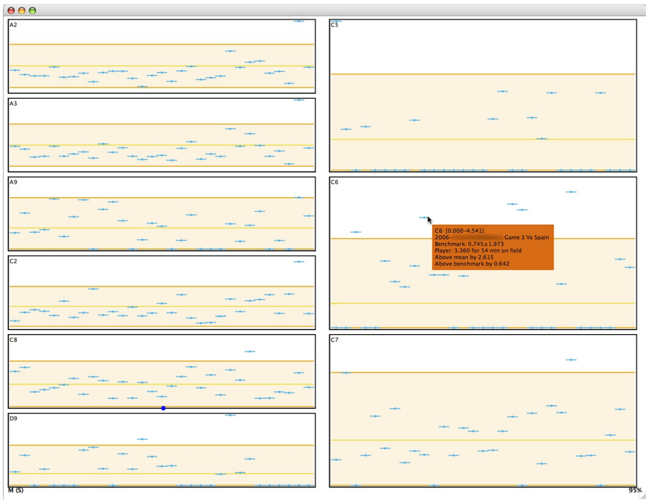

Figure 3. An interactive drill-down view in line graph mode. Attribute names have been removed by request.

are similarly coloured yellow when player attributes are performing below the benchmark. Cells in which a player's attribute lies both above and below the benchmark are assigned an orange colour. The blue and yellow cell colouring is inverted if a cell represents an attribute that is considered to have a negative impact on game performance.

The comparison table view changes to display each player's raw attribute score and the match-normalised score when a single match is selected for comparison. Blue coloured cells represent performance above the benchmark while yellow cells signify performance below the benchmark. This colouring is again inverted for negative attributes.

Mousing over a cell in the comparison table provides a brief textual description of that cell in an information bar at the bottom left of the program window. The number of games played above, within and below the benchmark is shown when multiple games are compared while the raw and match-normalised attribute scores for an attribute is shown for single games. The player name and the benchmark average and standard deviation used in the comparison for that cell is also shown.

Although not shown in Figure 1, screens allowing users to define arbitrary player groupings, match groupings and tournaments are available via a menu selection. Group membership in these windows is defined using simple toggle check boxes. A command for importing match data has also been implemented.

4.2.2 Graphical Benchmark Comparison

A graphical benchmark and comparison view is also available. This view, depicted in Figure 2, features an array of interactive line graphs that display an abstracted view of player performance. Abstract metrics distil different aspects of player performance to provide trainers with an overview of different areas of performance.

Each graph shows, for a given metric, a shaded area representative of the currently active benchmark as defined by the mean and standard deviation. Individual player scores are plotted within the graph as lines and points. Detailed information about performance in a single match is displayed in an overlay box when the user positions the mouse cursor over a data point. The overlay displays a textual comparison of the performance along with the date of the match, its description and the tournament in which it was played.

Coaches and trainers are able to visually analyse the entire set of performance data by double clicking a player's chart. This activates a drill down view that aids exploration of individual player data. An example of the drill down view in use is shown in Figure 3. The drill down view arranges abstract performance summaries on the left side of the display. Clicking inside any of these graphs activates the drill down for that summary. The low level features from which the currently selected abstract performance overview is derived are displayed on the right side of the display.

An alternative display mode that plots data points radially is shown in Figure 4. Here, attributes are plotted within segments of the circle such that a single graph can display results from an entire set of performance summaries or fea-

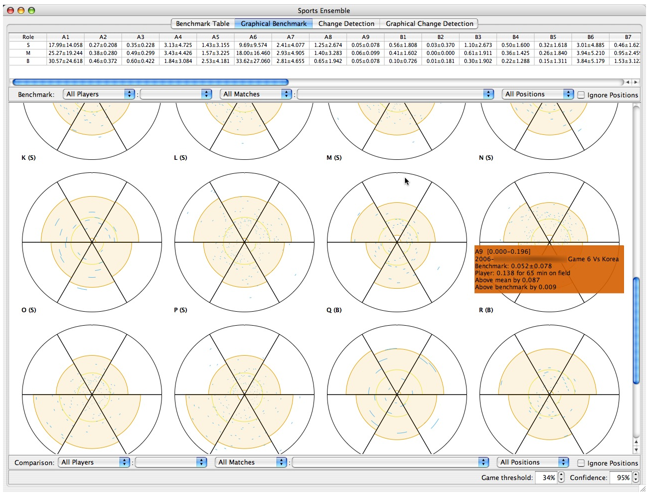

Figure 4. The graphical performance benchmark and comparison view in wheel mode. Each graph only shows abstract performance summaries. Attribute names have been removed by request.

tures. This alternative display mode, along with its corresponding drill down view, is shown in Figures 4 and 5.

4.3 Design Rationale

An important motivation behind this work was the provision of an intuitive interface that would allow flexible player and match selection for both benchmarking and player comparison. The interface should enable coaches and their assistants to explore their data in a meaningful way. These needs lead to a logical formulation of the filtering interface: a selection bar in which an arbitrary combination of players and matches can be chosen. Two such filter bars were required; each was placed directly underneath the view which they influence.

Design of the benchmark view was logical given that the mean and standard deviation for each recorded attribute is best viewed in table form. The tabular comparison pane was similarly designed, requiring only minor modifications to assist with the interpretation of comparison results. The left side of the comparison table displays the names of players and the cumulative time spent on the field for the currently active filter selection. This portion of the display is static and will not be removed from view should the comparison table be scrolled. Both the benchmark and attribute portion of the comparison table are otherwise scrollable if the table is unable to be completely displayed on screen. This was a necessary design decision to cater for users of any display size, the number of attributes being tracked being too numerous to fit comfortably even on large displays.

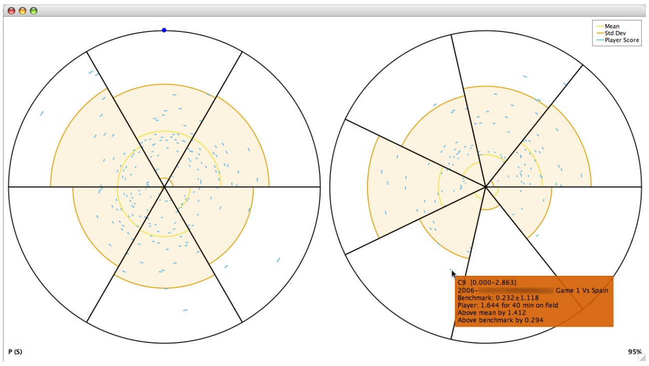

Figure 5. An interactive drill-down view in wheel mode. Attribute names have been removed by request.

Allowing filters to return multiple games necessitated the use of the two different comparison table views. Displaying single game information when large numbers of games are selected without users quickly losing oversight of how the attributes of players compare over numerous games was otherwise unfeasible. This decision resulted in the implementation of a summarised view which shows how many games were played below, within and above the benchmark threshold. Trainers are still able to view detailed attribute information, however, by drilling the comparison down to a single game.

The inclusion of colour to highlight good performance and under achievement was an obvious design decision as this allows users to identify and focus their attention on players and their attributes which they are most likely to find of interest. Adjustable parameters for the confidence level and the game threshold were likewise needed to enable users to explore various interpretations of the data. Colour mixtures used to highlight interesting comparison table cells were selected to cater for users with limited colour vision [11]. Although the comparison table employs the blue and yellow colours when results from a single match comparison are shown, a third colour was required for multiple game comparisons to cater for cases where players have performed both above and below the team benchmark in some attribute.

Inclusion of the graphical comparison views was motivated by a reluctance of some users to explore only numeric data. User feedback highlighted a need to provide coaches with a more intuitive means of analysing player performance without limiting their ability to explore and experiment.

5 Change Detection and Performance Prediction

The performance change detection and tracking component of the program employs a multiresolution sliding window approach to identify and track change in player performance. Two types of change are sought.

First, we wish to identify variations in performance attributes that manifest as concept drift. These changes present themselves as permanent or temporary shifts in a player's achievement and may indicate areas in which a player has begun to deteriorate or is showing signs of improvement. Identifying the nature of such change is important as it allows coaches to consider the rate and extent of change being observed. The tracking of this type of change is examined in Section 5.1.

Second, we wish to identify player performances that do not match position profiles derived from other players on the team. We tackle this problem by training an ensemble of C4.5 [12] classification trees to discriminate between various on-field roles. The expert classifiers are thus able to make position-centric performance comparisons using a more expressive set of rules than is possible by the single attribute comparison approach described in Section 4. The use of ensemble learning for role conformity change detection is described in Section 5.2.

5.1 Concept Drift

We define a concept to be the mean and standard deviation of a single attribute over time for the purpose of tracking basic variations in performance. Concept drift, then, seeks to monitor change in the mean attribute values in order to track change and to provide a descriptive analysis of the change. Determining the behaviour of change offers trainers an objective supporting tool to further their examination of how players performance varies over time. Knowing the nature of any change may also assist in the prediction a player's future performance.

The performance change detection and tracking component of the program employs a multiple overlapping windows concept drift tracking approach. The reason for using multiple windows is twofold. First, the performance of the player needs to be analysed over different time intervals and, second, it has to be able to handle all types of changes as well as accurately identify the starting point of any change in the player performance. The approach uses a set of dynamically sized windows with a static sized base window. The static window size is provided by coaches and is used to provide an interpretation of the performance data over a predefined length of time. The dynamic window size varies based on the consistency and persistence of the change observed in player performance and is used to provide an interpretation that is more focused on the recent data rather than a predefined time interval. The windows provide competing interpretations of the player performance. The data contained in the windows can range from time intervals covering from as few as 5 games to intervals covering one or more tournaments. Some parameters of the software, such as what constitutes significant change, are subjective and as such have been selected by coaches who are experts at analysing performance in their sport.

The Competing Windows Algorithm (CWA) [7] forms the framework of our concept drift implementation. CWA employs multiresolution windows to scan incoming data, dynamically adjusting the size of windows to track change while remaining resistant to noise. It does so by maintaining three sliding windows w_1, w_2 and w_3 of size $|w_1|$, $|w_2|$ and $|w_3|$ respectively. The size of w_1 is static and is selected by the user; the size of windows w_2 and w_3 is bounded by $\min(|w_2|) \leq w_2 \leq \max(|w_2|)$ and $\min(|w_3|) \leq w_3 \leq \max(|w_3|)$ respectively. Our implementation of CWA initialises the window bounds such that: $\min(|w_2|) = 2 \cdot |w_1|$, $\min(|w_3|) = 4 \cdot |w_1|$, $\max(|w_2|) = 4 \cdot |w_1|$ and $\max(|w_3|) = 8 \cdot |w_1|$.

Windows increase beyond their minimum size only when a lack of consistency in the data has been detected. Consistency, here, refers to the amount of change that a concept undergoes when a new data point arrives in a window. A concept's new value is said to be consistent with its previous value if the amount of change experienced is greater than some threshold δ_T. In our implementation the change $\delta(c_i)$ in a concept c_i is measured by the angle of change in the mean such that $\delta(c_i) = \arctan(c_{i,t} - c_{i,t-1})$ where $c_{i,t}$ and $c_{i,t-1}$ are the means of the concept at time t and at time $t-1$ respectively. A concept is therefore said to be consistent if $|\delta(c_i)| \leq \delta_T$. Experimentation has shown that default values of $|w_1| = 8$ and $\delta_T = 10°$ are reasonable.

Windows w_2 and w_3 can be safely reduced to their minimum sizes if evidence of consistency in the data has been found. CWA achieves this by tracking the number of consistent data points within a window via a persistency counter. Window w_2 is shrunk to its minimum size if the persistency of window w_1 reaches $\min(w_2)$. Similarly, window w_3 is shrunk to its minimum size if the persistency of window w_2 reaches $\min(w_3)$. The persistency count of a window is reset whenever a data point that causes an inconsistent change in a concept is introduced to the window.

We say that significant levels of change are detected when at least 25% of data points within a sub-window region have triggered an inconsistent concept change. This change is tracked as it progresses through the window until the level of change present in another region of the window falls below 25%. Textual descriptions of the rate of change, its span and its permanency can be generated by examining how the change is tracked across the windows.

Concept drift enables us to predict how a player may perform in the near future. We do so by selecting the concept from the longest window with the highest persistency count. The mean, standard deviation and the average amount of change within the concept is used to predict future short term performance.

5.2 Role Conformity

The role conformity portion of our work seeks to identify instances of performance where players may not be adhering to the in-game roles that they have been assigned. Role-specific classification rules are found by training an ensemble of C4.5 [12] classification trees on instances of match performance labelled with a player's position on the field. Here, an instance is defined as a vector of normalised attribute scores representing a player's entire set of performance metrics within a match. Instances from both the player under examination and of their teammates are used to train the ensemble.

Expert classifiers are considered to be concepts within the concept drift framework. The experts are hence trained only on instances of play that reside within one of the sliding windows. As with [16], a set of classification experts are kept for each window and experts are ranked according to their ability to correctly classify new match instances as they are added to a window. Final classification of an instance is made via a majority vote between the N highest ranked experts in the longest window with the highest persistency score.

Match instances that are misclassified by a majority of experts are flagged by the system. Decision rules resulting in the misclassification can then be parsed to highlight those attributes that are likely indicators role non-conformity.

5.3 Interface and Design Rationale

The change detection interface is separated into two sections. The first displays a textual summary of any changes that are found in a player's performance data while the second provides an annotated line graph view of each attribute over time. Both views share a single filter selection panel with which coaches may select their player and define the matches that are passed to the change detection algorithms. The window w_1 size and a role conformity acceptance threshold parameters are also adjustable here.

An example showing the plotting of two attributes over time with annotations of discovered change is given in Figure 6. Here, match-normalised attribute scores are plotted as blue data points. The predicted future score at each time step is represented on the graph by three solid lines depicting the change-tracked mean as well as the upper and lower bounds as given by two standard deviations from the mean.

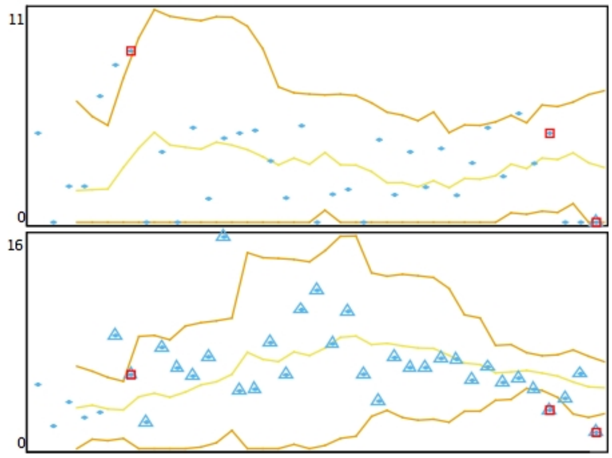

Figure 6. Change detection in two performance attributes. Normal attribute scores are plotted as blue data points while expected player scores are shown as solid lines representing the mean and standard deviation. Three matches in which the player is considered to have played outside their role have been highlighted with red squares. Temporary long term drift has been detected in the lower attribute and marked with blue triangles. Attribute names have been removed by request.

Change is shown in the graphs by glyphs that enclose the data points. Scores enclosed by triangles indicate that the score is part of an identified concept drift. The extent and permanence of the drift can be extracted by mousing over the data point. The second (lower) attribute plot in Figure 6 shows, for example, a temporary long term change in which a player's performance momentarily rises above the norm.

Role conformity changes are similarly represented, albeit with square glyphs. Users may obtain a breakdown of how and why experts classified such performances either by mousing over each highlighted data point or by referring to the textual summary. Mousing over a point activates an overlay display as used in other parts of the program.

The graphical display inherits the design decisions made for the line graphs described in Section 4.2.2. Such graphs give trainers an intuitive visual summary of how performance has changed while allowing detailed information to be interactively retrieved for games that are deemed of interest. The textual summary was included to provide a more traditional overview of detected change.

6 Coaches' Evaluation

Our performance analysis system has been evaluated by users coaching at an elite level. Their feedback has confirmed that the software provides coaching staff with a tool that assists in the objective interpretation of player performance data and in the monitoring of change. This is a critical advantage as the automated analysis inbuilt in the program combined with the ability to visualise the results has given the coaching team a tool which is now regularly used as part of player briefings to discuss past games and possible areas of improvement. The software is also regularly being used in player group analysis.

The system has also proven to be valuable in helping particular aspects of game analysis. Specifically, coaches were able to analyse the effectiveness of players in terms of ball possession by objectively identifying those players that were most effective at keeping possession and retrieving the ball from the opponents. Users also stated that the software helped assist to identify those defenders that were more effective at stopping the opposing team penetrating into the defensive third of the ground. Likewise, the system helped to objectively identify those defenders that were most able to intercept their opponent's passes. Further examples of the system's utility include the identification of how well defenders fared against different opposition, which defenders were more aggressive and thus more likely to miss tackles, which players were more effective at delivering penetrating passes and, finally, how well the players performed against opponents using different play styles and strategies.

7 Conclusion

This paper has introduced a visual data analysis tool aimed at assisting coaches monitoring the performance of their players. The two main components for team benchmarking and individual player analysis have been described. The rationale behind design decisions made over the course of the software development period have also been examined.

The software has been trialled with users coaching at an elite level. Although the feedback that we have received from them has been positive, we have identified several areas of future work that will further improve the flexibility and ease of use of the system.

Acknowledgements

The authors wish to acknowledge the assistance of the coaches who wish to remain anonymous. Their involvement and feedback on the direction of this project has been invaluable.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. L. P. Chen, editors, *Proc. Eleventh Int'l Conf. Data Engineering*, pages 3–14, Taipei, Taiwan, March 1995. IEEE Computer Society.

[2] A. Borrie, G. K. Jonsson, and M. S. Magnusson. Temporal pattern analysis and its applicability in sport: An explanation and exemplar data. *Journal of Sports Sciences*, 20(10):845–852, October 2002.

[3] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: A review. *SIGMOD Rec.*, 34(2):18–26, 2005.

[4] M. Hughes and I. Franks. *Notational Analysis of Sport*. E & FN Spon, London, 2nd edition, 2004.

[5] M. D. Hughes and R. M. Bartlett. The use of performance indicators in performance analysis. *Journal of Sport Statistics*, 20(10):739–754, October 2002.

[6] N. James, S. D. Mellalieu, and N. M. P. Jones. The development of position-specific performance indicators in professional rugby-union. *Journal of Sport Sciences*, 23(1):63–72, January 2005.

[7] M. M. Lazarescu, S. Venkatesh, and H. H. Bui. Using multiple windows to track concept drift. *Intelligent Data Analysis*, 8(1):29–59, 2004.

[8] D. G. Liebermann, L. Katz, M. D. Hughes, R. M. Bartlett, J. McClements, and I. M. Franks. Advances in the application of information technology to sport performance. *Journal of Sports Sciences*, 20(10):755–769, October 2002.

[9] M. S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments and Computers*, 23(1):415–429, 2000.

[10] T. McGarry, D. I. Anderson, S. A. Wallace, M. D. Hughes, and I. M. Franks. Sport competition as a dynamical self-organizing system. *Journal of Sports Sciences*, 20(10):771–781, October 2002.

[11] M. Okabe and K. Ito. How to make figures and presentations that are friendly to color blind people, November 2002. Retrieved January 2007 from http://jfly.iam.u-tokyo.ac.jp/color/.

[12] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc, 1993.

[13] H. Remmert. Analysis of group-tactical offensive behavior in elite basketball on the basis of a process orientated model. *European Journal of Sport Science*, 3(3):1–12, June 2003.

[14] J. C. Schlimmer and R. H. Granger. Beyond incremental processing: Tracking concept drift. In *Proc. 5th National Conf. Artificial Intelligence*, pages 502–507, Philadelphia, Pennsylvania, USA, August 1986. Morgan Kaufmann.

[15] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, editors, *Proc. 5th Int'l Conf. Extending Database Technology: Advances in Database Technology*, volume 1057 of *Lecture Notes In Computer Science*, pages 3–17. Springer, 1996.

[16] W. N. Street and Y. S. Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In *Proc. 7th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 377–382, San Francisco, California, USA, 2001.