Player Classification Using a Meta-Clustering Approach

Daniel Ramirez-Cano, Simon Colton, Robin Baumgarten
Department of Computing, Imperial College London
180 Queen's Gate, London
SW7 2AZ, UK
+44 (0)20 7594 8287
d.ramirez / sgc / rb1006@doc.ic.ac.uk

ABSTRACT

Player classification has recently become a key aspect of game design in areas such as adaptive game systems, player behaviour prediction, player tutoring and non-player character design. Past research has focused on the design of hierarchical, preference-based and probabilistic models aimed at modelling players' behaviour. We propose a meta-classification approach that breaks the clustering of gameplay mixed data into three levels of analysis. The first level uses dimensionality reduction and partitional clustering of aggregate game data in an action/skill-based classification. The second level applies similarity-based clustering of action sequences to group players according to their preferences. For this we propose a new approach which uses Rubner's Earth Mover's Distance (EMD) as a similarity metric to compare histograms of players' game world explorations. The third level applies a combination of social network analysis metrics, such as shortest path length, to social data to find clusters in the players' social network. We test our approach in a gameplay dataset from a freely available first-person social hunting game.

Keywords

Player classification, game design, social gaming, clustering.

1. INTRODUCTION

Player classification is quickly becoming a key area in the development of games. It allows learning from player's actions in order to deliver personalized content back to the player. A traditional example, not originated in computer games, is targeted marketing where purchase suggestions are delivered to players based on a customer profile built from the player's previous purchases. A recent example of targeted marketing is Apple's 'Genius Recommendations for Apps' which recommends new applications based on those already installed on an iPhone/iPod touch [14]. A more recent and developing area which relies on player classification is adaptive games where the game adapts to the player's actions and decisions in order to provide a more enjoyable and challenging experience [3].

The field of machine learning offers a wide range of well-tested algorithms which have been used in player classification. Some of these algorithms include decision trees, artificial neural networks and reinforcement learning. A characteristic of these algorithms is that they rely on the prior definition of classes and a reliable training set in order to classify players in a supervised approach. However, it is not always clear which player classes are most appropriate for a game. Moreover, classes might be previously inexistent and exclusive to one game and classes might change and evolve over time.

In this paper, we focus on the problem of player classification where there is no previously defined classes. Thus, we adopt an unsupervised player classification approach, i.e. player clustering. Many clustering algorithms exist in the literature but many present practical challenges when applied to gameplay datasets. Some of these challenges are:

- Player's actions can generate dataset variables or features of very different natures. While some games rely on simple metrics such as number of enemies killed, others might have more complex datasets including, for example, variables used to build a player's psychological profile.
- Datasets change as the players' expertise evolves. New games start with a dataset composed of a very large number of beginners and a few experts. Beginners are more difficult to classify because their actions and choices are very similar due to their lack of experience. As players learn, they build a playing style making it easier to identify classes.
- Overlapping classes, i.e. the playing styles can fall into one or more classes. For example, Bartle's [1] four class player classification suggests grouping players in *Killers*, *Socialisers*, *Explorers* and *Achievers*. In practice, however, it is often found that expert players fall into all four of these categories.

The above practical challenges make it very difficult to find a single clustering algorithm which is able to find a complete and accurate player classification. We propose a partitioned specification and classification of the game features where it is possible to abstract the description of player classes according to different levels of granularity by using actions and states as the main descriptive blocks. We propose a three layer meta-classification approach consisting of the following stages:

1. Action/Skill based clustering: The first level applies dimensionality reduction and partitional clustering techniques to aggregated game data.
2. Preference based clustering: The second level applies similarity-based clustering of action sequences to group players according to their preferences.
3. Socially based clustering: This level applies a combination of social network analysis metrics such as shortest path length to social data to find clusters in the players' social network.

We worked towards solving some of the practical clustering concerns outlined above. For this we tried a new meta-classification approach. We first split the variables into four classes: *actions*, *skills*, *exploration* and *social*. Then we applied different classification algorithms to each class. The underlying idea is to first classify players according to their skills. For

example, player *A* could be a highly skilled shooter while player *B* is a new player who has widely explored the game but has not yet developed an efficient shooting strategy. Then we classify the players based on their exploration variables. We use these variables as an example to extract players' exploration preferences, e.g. player *A* prefers to stay hidden in a small area of the game world waiting for enemies while player *B* prefers to explore the game world as much as possible. Lastly, the social classification should give us an indication of how socially active a player is and consequently his/her predisposition to participate in social interactions.

In Section 2 we explore some existing work on player clustering and classification. Section 3 describes our meta-clustering approach giving details of the individual clustering algorithms used in each step. In this section we propose a new approach to the clustering of exploration sequences using Rubner's Earth Mover's Distance (EMD) [9] as a similarity metric and Multidimensional Scaling (MDS) techniques as a visualization tool for exploring the data. Section 4 describes the results of applying our meta-clustering approach to a large, high-dimensional dataset from the freely available game 'The Hunter'. In this section, we explore a set of linear and non-linear dimensionality reduction algorithms. These algorithms allowed us to visualise and analyse the game data in a low-dimensional space while maintaining the local and global properties of the high dimensional dataset. We then used the K-means unsupervised classification technique to identify clusters of players with similar profiles. Finally, we extracted the adjacency matrix from the game's social network and using the geodesic distance between pairs of nodes we weighted the results from the two previous steps to identify clusters of agents which are similar in skills, preferences and social compatibility. We conclude in Section 5 by discussing future work.

2. RELATED WORK

Social data has been used as one of the main building blocks of player profiling. Lampe et al. [5] use data from social networking sites to analyse the relation between the information entered by the player in an online profile and the social connections or clusters to which the player belongs. They built their theoretical framework based on signalling theory, common ground theory and transaction cost theory. Along similar lines, Riegelsberger, et al. analysed in [8] how matching players' behavioural preferences can help reduce undesired behaviour in online interactions. Their work suggests that selecting gaming partners based on playing profiles might reduce norm-violations in online environments.

Hidden Markov Models (HMM) have been used as a tool for clustering sequences. Matsumoto and Thawonmas [7] explore the use of HMM to classify player action sequences. They exploit the time structures which can be extracted from the action sequences. They show how clustering based on action sequences outperforms clustering based on the frequency of players' actions. Smyth explored the use of HMM in a more general context [10]. While his approach was not aimed at clustering in games, it does provide two key contributions: 1) a method for choosing the initial parameters of the model and 2) a method for finding the optimal number of clusters.

An area of game design which is closely related to player classification is adaptive games. Sznita et al. [12] propose a method for building adaptive game AI engines which learn and choose different AI scripts based on an interestingness measure by using a cross-entropy measure. Spronck et al. in [11] propose a method of designing adaptive games by using dynamic scripting. Their approach consists of an adaptive rule-base system for the automatic generation of AI scripts based on the success-failure rate of the rules.

3. META-CLUSTERING APPROACH

Player classification consists of taking players' actions as an input and delivering these actions grouped into different playing styles. Simple classification methods are normally designed for one of the many types of variables, e.g. continuous, discrete, mixed, sequential, sociometric, etc. Instead, we propose a combination of clustering algorithms (a meta-classification approach) to take advantage of the different characteristics of different dataset variables.

Our meta-clustering approach consists of the sequential implementation of different clustering techniques and is summarized as follows:

1. Variable classification
2. Action/skill-based clustering
3. Preference-based clustering
4. Social-based clustering

We propose a framework of variable classification consisting of the following four classes:

- The *actions* class contains 'raw' variables, i.e. actions which describe player behaviour in terms of a player's most basic activities such as distance travelled in a session, total shots fired and total playing time7.
- The *skills* class includes those variables that can be used to analyse the player's level of expertise; for example shooting accuracy, number of enemies spotted and number of trophies received.
- The *exploration* class contains variables which log sequences of events. One of the most common examples of this type of variables is the sequence of geographical coordinates which describe the trail explored by the player.
- Finally, the *social* class incorporates those variables which measure the level of social interaction of a player in the game's social network. Examples of variables within this class are number of messages sent to other players, number of messages posted in forums and quantity of content uploaded.

3.1 Action/Skill-based clustering

The first step of our meta-clustering approach consists of applying dimensionality reduction and partitional clustering algorithms to the action and skills variable classes. Dimensionality reduction techniques allow us to identify and remove redundant information which might be contained in more than one feature due to a high correlation between the chosen variables. Also, they provide a low-dimensional visualization which allows the visual exploration of the dataset. Partitional clustering algorithms aim at assigning points in the dataset to *k* clusters by optimizing a given criterion

function and an associated distance matrix. Since partitional clustering algorithms do not provide a clear definition of the *k* extracted clusters, this level of clustering heavily relies on the visual inspection of the low-dimensional space to give meaning to the low-dimensional components in terms of the original variables. Below we describe the dimensionality reduction and partitional clustering algorithms used in our experiments.

3.1.1 Dimensionality Reduction

For dimensionality reduction, we use the widely known Principal Component Analysis (PCA) algorithm as the main dimensionality reduction technique. PCA provides a simple way of extracting the principal component from a dataset. We also consider other more complex and non-linear reduction techniques. However, as will be seen in our experimental results, more complex algorithms did not considerably outperform the result from PCA. Thus, the result from PCA was used in the succeeding clustering stages.

PCA is a linear dimensionality reduction technique based on finding a mapping between a high and a low dimensional data representation which maximises the amount of variance in the data. Other dimensionality reduction techniques we analysed are:

- ProbPCA: probabilistic formulation of PCA presented as an expectation-maximization algorithm which offers a more efficient approach to processing large high-dimensional datasets.
- MVU (and its adaptations FastMVU and Landmark MVU): MVU (Maximum Variance Unfolding) is a technique which learns a maximum-variance kernel function which is constructed under a nearest neighbour pairwise distance constraint. The optimisation problem is solved using semidefinite programming.
- Laplacian: aims at building a low-dimensional data representation which preserves the local properties of the data by building a weighted neighbourhood graph and minimising a distance cost function.
- LLC: Local Linear Coordination is a method which first computes and combines a set of local linear models and then aligns the local models by solving a generalised eigenproblem.
- LTSA: Local Tangent Space Alignment is a technique which learns the local geometry of the manifold by using the tangent spaces of individual data points and then constructs the global coordinate system for the underlying manifold by aligning those tangent spaces.

3.1.2 Partitional Clustering

We explore K-means [2, 6] as our method for partitional clustering. K-means is a method for unsupervised classification and one of the most popular clustering algorithms so we will only briefly describe the most basic algorithm here:

1. Initialise the partition by choosing *k* as the number of clusters and arbitrarily choose one cluster centre (mean) for each cluster
2. Partition the data into *k* clusters by iteratively assigning points to the cluster with the nearest mean
3. Recalculate the mean for each cluster
4. Repeat 2 and 3 until there is no change

3.2 Preference-based clustering

In the second step of our meta-clustering approach, we are interested in the clustering of action sequences as a way to classify player's actions. More specifically, we look at the data generated from the movement of players across the game world, i.e. the sequence of location coordinates. For simplicity, we call this sequence of points the player's trail. Figure 1 (left) illustrates a trail left by a player in the game 'The Hunter'.

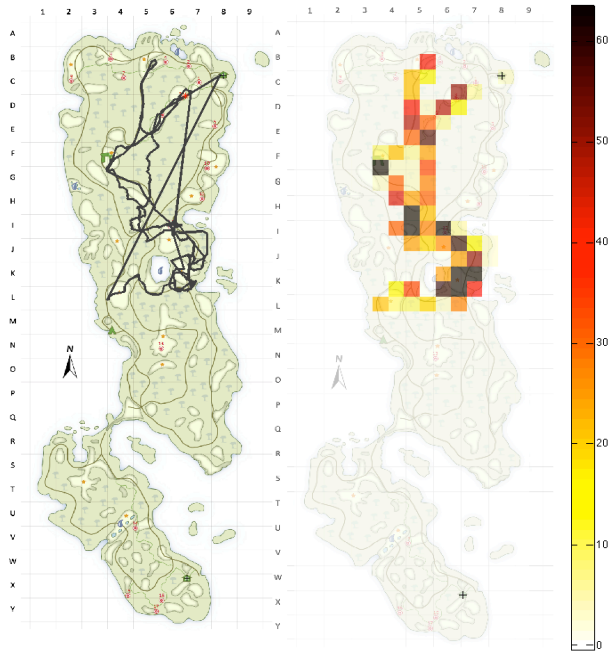

Figure 1: Trail left by a player when exploring the game world (left) and its corresponding histogram (right).

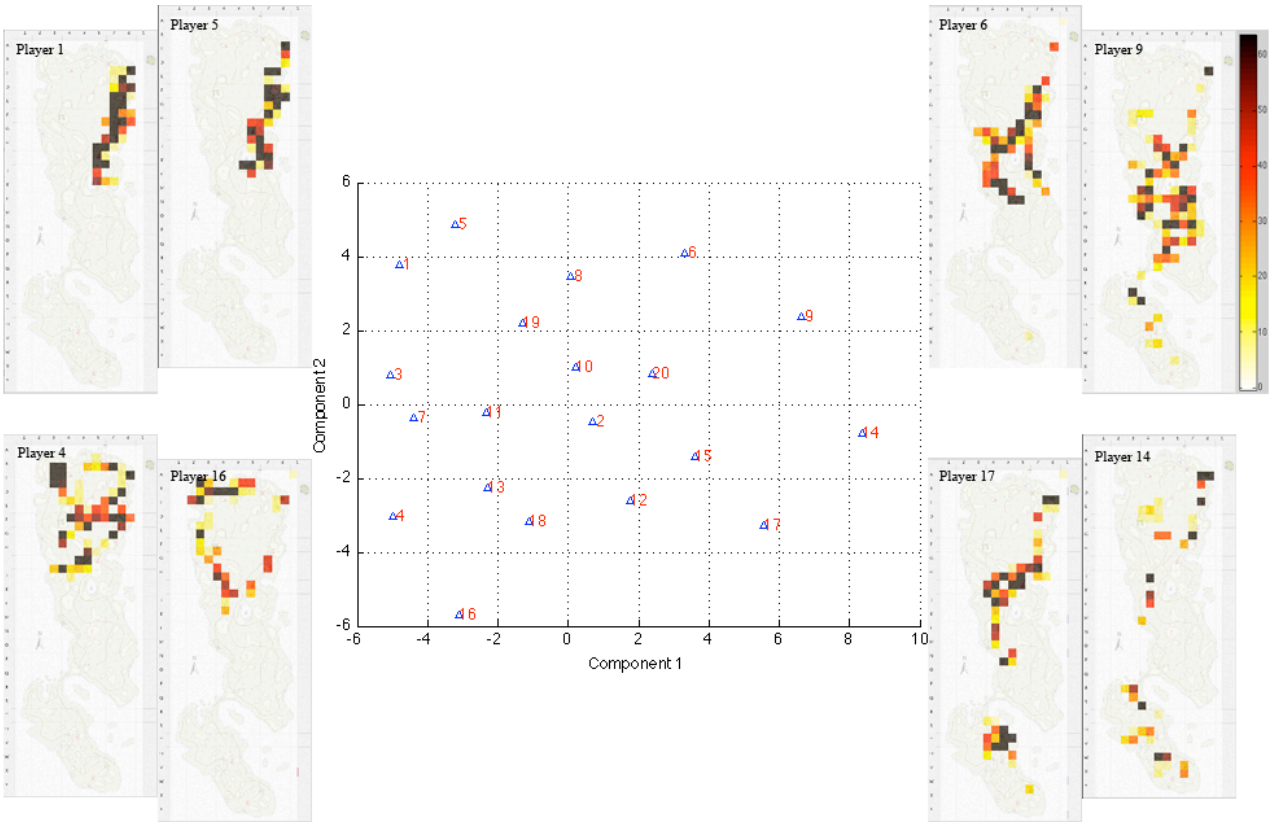

Figure 2: 2-D representation of similarities between the players' trail of 20 random players. The four pairs of histogram show the similarity between close points.

As mentioned in Section 2, Hidden Markov Models have been extensively used as a tool for clustering action sequences. Here we propose a new approach based on clustering of histograms extracted from a players' trails. Figure 1-right shows the histogram generated from the player trail illustrated by figure 1-left. Clustering is performed by defining a metric which provides a similarity distance between histograms. The different distances between histograms can then be used to build a dissimilarity matrix which, together with multidimensional scaling provide a N-dimensional representation of the similarities between histograms.

This approach is based on Rubner's algorithm and is summarized as follows:

1. Extract the players' set of trails from different playing sessions.
2. For each player, calculate the aggregated histogram generated from his set of trails.
3. Build a dissimilarity matrix by calculating the distance between each pair of histograms.
4. Apply multidimensional scaling to represent the distances between histograms as points in an N-dimensional space.

The accuracy of our clustering algorithm heavily relies on the selection of a suitable metric for measuring the distance between histograms. For this, we have employed the Earth Mover's Distance (EMD) as metric. EMD is a well-known metric which is used to measure distance between two distributions. It is widely used in image processing algorithms as a way to measure similarity between two images. EMD is based on the idea of finding the minimum amount of work that it takes to transform one distribution into another. Examples of commonly used distributions are histograms, colour distributions and image intensities. Work is measured by the cost c_{ij} of moving one unit of distribution mass from one distribution i to another distribution j. Transformations are represented as a set of flows f_{ij} which show the number of units of distribution mass which are moved from i to j. The problem then becomes finding the set of flows which minimises the overall cost. Thus, the earth mover's distance is defined by the following equation:

$$EMD = \frac{\sum_{i \in I} \sum_{j \in J} c_{ij} f_{ij}}{\sum_{i \in I} \sum_{j \in J} f_{ij}} \qquad (1)$$

where the numerator is the overall cost and the denominator is a normalization factor to compensate for distributions of different sizes.

In order to find the optimal set of flows which minimizes the cost, we adopt Rubner's optimization implementation which computes the optimum EMD distance and corresponding flows [9]. The representation of these distances in a N-dimensional space is done by applying multidimensional scaling with Kruskal's normalized stress criterion [4].

Before integrating our histogram based algorithm into our meta-classification approach, we analyse the results of applying the algorithm to a random sample of 20 players from our dataset. Figure 2 shows the result of using MDS on the distances obtained by applying the EMD metric to the 20 random trails.

It can be observed from figure 2 that points which are close together (1-5, 4-26, 6-9, 14-17) present histograms with similar distributions. Moreover, even though the interpretation of the axes is not obvious from the found distances, it can be observed that points with larger values in the vertical axis show a concentration of mass in the right hand side of the map while points with larger values in the horizontal axis show a concentration of mass in the lower part of the map.

3.3 Social-based clustering

We start from the assumption that two players could be motivated to create a social link between them if they perceive that they share some common ground. Traditionally, in social networking websites, this common ground can be extracted from information explicitly expressed in the players' profiles. Some examples of this common ground are mutual friends, common expressed interests and common affiliations. We use this information to weight the distances obtained from the EMD analysis. More specifically, we use the geodesic distance between pairs of points built from the length of the shortest path between players in order to build a weight matrix. This weight matrix is incorporated into the MDS analysis to redistribute the distances between points according to their social interactions.

4. EXPERIMENTAL RESULTS

We test our approach by analysing logged game data from the freely available game 'The Hunter'. This is a free roaming, animal hunting game where the objective is to track, spot and harvest animals following the rules of ethical hunting.

We extracted a set of gameplay features from data collected from a series of real players' gaming sessions. This feature extraction and selection was based on choosing variables which can be used to describe a player's profile. 30 different features were selected; 11 from the actions class, 11 from and skills class, 1 from the exploration class and 7 from the social class. The complete dataset included multiple hunting sessions from approximately 50,000 players.

4.1 Action/Skill-based clustering

We applied the six different dimensionality reduction techniques described in Section 3.1.1 to the set of actions and skills variables. The results obtained from each technique are shown in Figure 3.

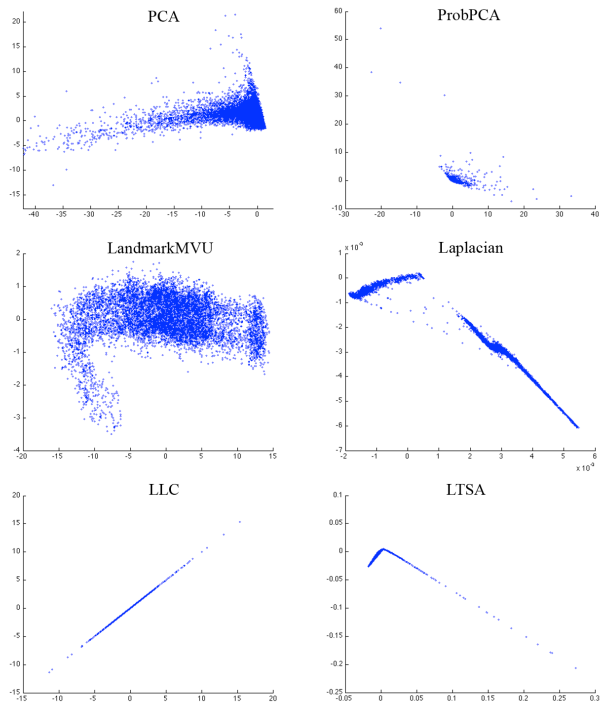

Figure 3: Result of applying dimensionality reduction techniques to actions and skills variables. From top to bottom, left to right, PCA, ProbPCA, LandmarkMVU, Laplacian, LLC, LTSA

It can be observed that in most techniques points are either concentrated in one particular area of the graph or spread around the figure. An in-in-depth analysis of the graphs revealed that most points concentrated around that particular area forming one very large and dense cluster. The rest of the players that spread around the graphs exhibit different characteristics. However, as explained and due to the practical reasons outlined in the introduction, it is not possible yet, in any of the six graphs, to identify clusters only through visual inspection.

For simplicity, and given that there is no obvious advantage over choosing a more complex approach, we chose the output from PCA as input to the K-means algorithm. We explored the result of choosing different numbers of clusters and it was found that k=5 showed the best result. Figure 4-top shows the result of applying the K-means algorithm. For improved clarity, figure 4-bottom shows the results of the algorithm when applied to a smaller random sample of the dataset for which it is easier to visualize the different clusters.

Initial observations showed that it is possible to identify clusters of players using unsupervised classification. The analysis from applying K-means clustering revealed that the previously identified large cluster corresponds to new players. This large cluster is the sum of a very large number of beginners combined with a large number of people who created an account but never accessed the game. We can also identify other clusters containing players with very clear identifiable skills. The identified clusters are:

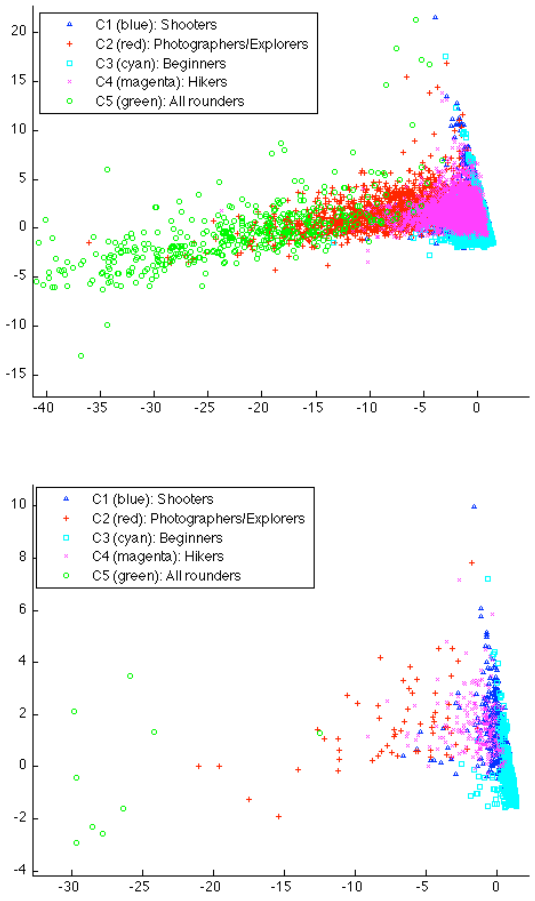

Figure 4: Result of applying K-means to the output of PCA to the complete dataset (top) and to a reduced random sample of points (bottom).

- Shooters: players with high shooting accuracy
- Photographers/Explorers: players who like to explore and photograph the game world
- Hikers: players with lengthy trails but no shooting
- All rounders: players with good tracking, spotting and shooting skills.

The combined analysis of these two techniques also allowed the identification of variables containing minimum variance and which consequently did not contribute to the differentiation between players.

4.2 Preference-based clustering

In this clustering stage, the classification of players' exploration variables was done by finding similarities between preferred hunting areas. Preferred hunting areas are identified by calculating 2D-histograms of walking paths and finding the similarities between them. We implemented the algorithm described in Section 3.2 which requires the definition of a cost function. This function measures the cost from moving one distribution mass unit from one histogram to another. We assume histograms of equal size. We define a cost function where moving mass units vertically or horizontally to adjacent bins has a unitary cost per bin travelled. The cost of moving one unit from equivalent bins is zero. A vertical plus a horizontal move compose diagonal moves between adjacent bins. Thus, diagonal moves have a cost of two. Figure 5 shows a representation of these costs. The cost function is defined by the following equation:

$$c_{ij} = \left|i_{row} - j_{row}\right| + \left|i_{col} - j_{col}\right| \qquad (2)$$

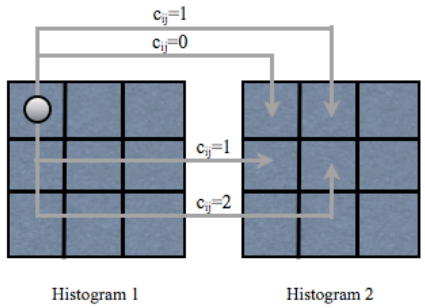

Figure 5: Graphical representation of cost function.

From the clusters obtained from our previous PCA/K-means clustering stage, we randomly selected 5 players from each cluster. For each of these 25 players, we extracted their set of trails, calculated their histograms and applied the EMD/MDS clustering algorithm. The result of this algorithm is shown in Figure 6. From this figure it can be observed that points within the same PCA/K-means cluster also seem to fall within the same concentric circle around the zero point of origin. This is particularly clear for clusters containing all rounders, photographers/explorers and shooters. It is not as clear for the beginners class, but this can be easily explained by the fact that some beginners might walk the same areas as those preferred by the shooters. However, they not necessarily share the same shooting skills. An exception to this double clustering seems to be players belonging to the hikers class.

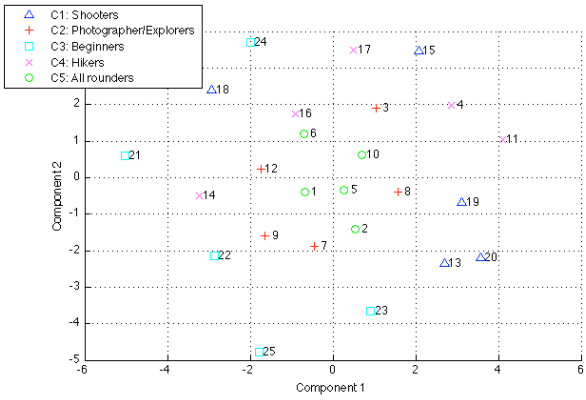

Figure 6: Result of applying EMD/MDS algorithm to the trails of 25 randomly selected players, 5 from each class defined by the previous PCA/K-means clustering stage.

4.3 Social-based clustering

This type of clustering is characterised by adding information concerning social interactions among players to the previous clustering stages. Figure 7 shows the social network of the 25

previously selected players. This network is formed by friendship relations between players.

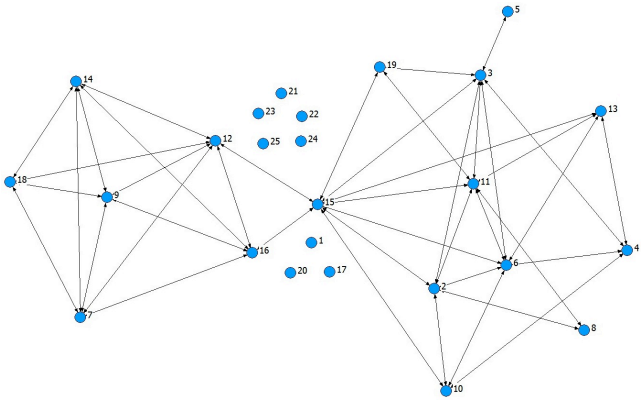

Figure 7: Social network based on friendship relations of the 25 previously selected players.

Based on these friendship relations, we extract a matrix of geodesic distances to calculate how far is one player from another in terms of mutual friends. We insert this information as weights into the MDS algorithm such that players who are already friends are brought closer together.

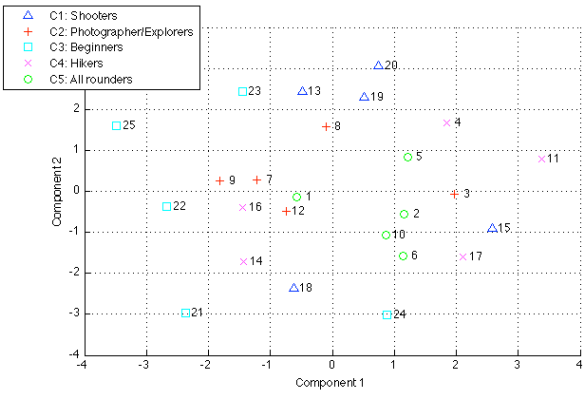

Figure 8: End result of applying the three layers of the meta-clustering approach

Figure 8 shows the end result of inserting the friendship-based weights into the EMD/MDS clustering algorithm. From figure 7, it can be observed that player 15 is a cutpoint that joins two subgraphs. Figure 8 shows how points belonging to the same subgraph are pulled closer to each other. It is particularly clear for points belonging to the left hand side subgraph (players 7, 9, 12, 14, 16 and 18). Points which in figure 6 where already close to each other and which belong to the same subgraph, are brought even closer (e.g. players 8, 13, 19 and 20). Points representing players in the beginners cluster remain in the periphery and do not form part of any of the newly formed clusters. This should be expected as one characteristic of new players is their lack of social connections.

This end result of applying the three layers of meta-clustering gives us enough information to design mechanisms which take advantage of the different levels of similarity between players. Some examples of these mechanisms aimed at boosting social interaction among players are:

- Matchmaking: player 19 and 20 can be matched as potential friends. They both have the same hunting skills and they both prefer hunting in the same areas of the game world.
- Team building: players 3, 15 and 17 or 1, 12 and 16 can be matched together to form a team to participate in hunting competitions which require multiple skills. Players 13, 19, 20 can be teamed together for a shooting competition.
- Tutoring: Players 21 to 25 can be assigned tutors who help them develop different skills. For example, player 13 and 23 walk the same areas of the world so player 13 could tutor 23 in shooting skills.

5. CONCLUSIONS AND FUTURE WORK

We proposed a meta-clustering approach for player classification which divides the players' game variables into different classes and then sequentially applies different clustering algorithms to subsets of these classes. It has been shown how this meta-clustering approach leverages the particular characteristics of the individual clustering algorithms to solve some of the problems faced when analyzing real game datasets such as overlapping classes and mixed-variables datasets. Also, we have given some examples of how the end result of the meta-clustering allows for the designing of mechanisms which take advantage of a combined clustering analysis.

Future work will focus on replacing the K-means clustering algorithm in the action/skill-based clustering stage in order to tackle the problem of identifying clusters within highly dense areas. One example of such highly dense areas is the cluster formed by beginners. The DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm provides a good starting point.

A second avenue of future research will look at using the output from this meta-clustering to define classes and training sets which can be used as input to supervised machine learning algorithms such as decision tree learning or a genetic algorithms approach.

6. ACKNOWLEDGMENTS

This work has been supported by the joint EPSRC (TS/G002886/1) / TSB (TP/AJ366J) AI Social Agents project. Thanks for support to Felipe Orihuela-Espina, Matt Samsan and Tom Scutt. Finally, thanks to the members of 'The Hunter' forums and particularly to Crain who created and published the game's map used in this document.

7. REFERENCES

[1] Bartle, R. Hearts, clubs, diamonds, spades: Players who suit MUDs. http://www.mud.co.uk/richard/hcds.htm; Last accessed: February, 2010.

[2] Forgy, E.W., Cluster analysis of multivariate data: efficiency vs interpretability of classifications. Biometrics, 21, 768-769.

[3] Charles, D. et al. 2005. Player-centred game design: Player modelling and adaptive digital games. In Proceedings of DiGRA 2005 Conference: Changing Views - Worlds in Play.

[4] Kruskal, J. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29(1), 1-27.

[5] Lampe, C., N. Ellison, and C. Steinfield. 2007. A familiar Face(book): profile elements as signals in an online social network. In Proceedings of the SIGCHI conference on Human factors in computing systems.

[6] MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 281-297.

[7] Matsumoto, Y. and R. Thawonmas. 2004. MMOG player classification using Hidden Markov Models. Entertainment Computing - ICEC 2004, Volume 3166/2004.

[8] Riegelsberger, J., et al. 2007. Personality matters: incorporating detailed user attributes and preferences into the matchmaking process. Hawaii International Conference on System Sciences.

[9] Rubner, Y., C. Tomasi, and L.J. Guibas. 1998. A metric for distributions with applications to image databases. In Proceedings of the Sixth International Conference on Computer Vision, 59-66.

[10] Smyth, P. 1997. Clustering sequences with Hidden Markov Models. Advances in Neural Information Processing Systems. MIT Press, 648-654.

[11] Spronck, P., et al. 2006. Adaptive game AI with dynamic scripting. Machine Learning. 63(3), 217-248.

[12] Szita, I., M. Ponsen, and P. Spronck. 2008. Keeping adaptive game AI interesting. In Proceedings of CGAMES 2008 (eds. Quasim Mehdi, Robert Moreton, and Stuart Slater), 70-74.

[13] http://www.thehunter.com/pub/; last accessed: February 2010.

[14] http://www.apple.com/ipodtouch/features/app-store.html; last accessed: February 2010

CGAT Conference 2010, April 6–7, 2010, Singapore.