Swords, Data and Balls: Extracting Extreme Behavioural Prototypes with Kernel Minimum Enclosing Balls

Hanna Kondratiuk and Rafet Sifa
Fraunhofer IAIS
St. Augustin, Germany
name.surname@iais.fraunhofer.de

***Abstract*—Extracting behavioural prototypes plays an important role in player profiling. Understanding the type of players present in the game goes alongside improving game-play experience as well as player engagement over time. In this paper, we introduce the application of Kernel Minimum Enclosing Balls (KMEBs) as a tool to extract meaningful extreme prototypes in games and present an example use-case analyzing a behavioural dataset from a Massively Multiplayer Online Role Playing Game. Unlike the majority of the methods covered in this context, our approach allows for modelling nominal and numerical behavioural features, extending the scope and capability of the profiling methods as well as improving the interpretability of the results.**

I. Introduction

Prototype selection is a diverse field allowing for many applications: it can be used to condense the training data when further computations are expensive, to provide hidden patterns to human analysts, to explain the versatile results of classification or clustering routines [1]. The main goals of Game Analytics research include building models to understand the game dynamics and improve the game balance. Moreover, common playability problems occur if playability heuristics are violated [2]. For instance, if the difficulty ramps up too quickly, game items wear out too fast or if the game does not support different playing styles, it poses an obstacle in the game process. In this case, profiling the behaviour of players would show the overall picture, namely, whether it is possible to reach the highest level of the game using a different strategy, or if the game is bound to the usage of only one factor.

Nevertheless, the interpretability of the results and samples being "representative" of the data remains the key point of the underlying research [3]. Then providing a tool for extracting a small number of prototypes (so that domain specialists can understand the data) stays an acute research direction nowadays.

The method of Kernel Minimum Enclosing Balls (KMEBs), introduced in the context of prototyping in [4], is simple and easy to implement. It has a better run-time than commonly used in behavioural profiling in digital games Archetypal Analysis method, as can be seen on the example of multi-player shooter Destiny in [5] and UltimaIV in [6]. Archetypal Analysis extracts extreme behaviour basis vectors that are easy to interpret [1]. Opposed to the basis vectors, KMEBs offer the concept of support vectors, that are the data points corresponding to the non-zero values of an optimized vector.

978-1-6654-3886-5/21/$31.00

Having a number of hyper-parameters that affect the resulting prototypes, KMEBs allow us to introduce variation into algorithm output, paving the way to the ensemble approach. One of such parameters is the kernel, which is a powerful tool given prior knowledge about the data. Depending on the data type, be it categorical, mixed or continuous data, there exist numerous task-specific kernels for each of the described cases. All the above-mentioned advantages make KMEBs an interesting research objective in terms of analyzing the real-world player data.

II. Contribution

In this paper, we provide an ensemble of well-known Distance Substitution kernels within the context of prototype extraction using KMEBs. We augment the Divide and Conquer algorithm from [7] to return a fixed small number of prototypes (support vectors), prepared for analysis by a human analyst, and provide an ensemble approach, calculating the frequency of extracted prototypes over a variety of distance-kernel pairs. Moreover, we conclude that the kernel matrix in KMEBs is not required to be positive definite to reach substantial results on the example of this study's data.

III. Kernels in kernel minimum enclosing balls problem

Recall the *kernel MEB problem* in terms of minimization, derived in [4] and solved via the Frank-Wolfe or the Projected Gradient Descent algorithm:

$$\boldsymbol{\mu}_* = \arg\min_{\boldsymbol{\mu}} \boldsymbol{\mu}^\intercal \boldsymbol{K} \boldsymbol{\mu} - \boldsymbol{\mu}^\intercal \boldsymbol{k}$$
$$\text{s.t. } \boldsymbol{\mu} \in \Delta^{n-1},$$

where $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is a kernel matrix computed on input dataset, $\boldsymbol{k}$ contains its diagonal, Δ^{n-1} is the standard $(n-1)$-simplex, $\boldsymbol{\mu} \in \mathbb{R}^n$ is the Lagrange multipliers, $\boldsymbol{K} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is an appropriate Mercer kernel.

Kernel	Definition
Laplace	$k_{i,j} = e^{\frac{-d(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sigma}}$
Distance	$k_{i,j} = -d(\boldsymbol{x}_i, \boldsymbol{x}_j)^{\beta}, \beta \in [0, 2]$
Gaussian	$k_{i,j} = e^{\frac{-d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)}{2\sigma^2}}$
Rational Quadratic	$k_{i,j} = 1 - \frac{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)}{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)+c}$
Inverse Multiquadric	$k_{i,j} = \frac{1}{\sqrt{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)+c^2}}$

TABLE I
CONCISE OVERVIEW OF THE SELECTED KERNELS USED IN ENSEMBLE.

The data points of data matrix $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$ corresponding to the non-zero values of Lagrange multiplier $\boldsymbol{\mu}$ are called support vectors. Support vectors computed by KMEBs showed themselves finding central as well as varying prototypes of the data within the example of an online multiplayer shooter game [4], thus we use support vectors as prototypes for player data.

The choice of the kernel $\boldsymbol{K}$ is a way to inject prior knowledge about the data into the model. For instance, choosing the special distance measure (a city block or a hamming distance) for categorical data reflects its structure. One of the most popular kernel choices is the Distance Substitution (DS) Gaussian kernel, that is positive definite as stated in [8]:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(\frac{d(\boldsymbol{x_i}, \boldsymbol{x_j})^2}{2\sigma^2}\right),$$

and the respective Gaussian kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, where $K_{i,j} = K(\boldsymbol{x_i}, \boldsymbol{x_j})$, $d(\boldsymbol{x_i}, \boldsymbol{x_j})$ is the distance function between points $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. The distance measure is defined on the Hilbert space. It also has to be symmetric, have zero diagonal and be non negative.

The distance kernel itself is also commonly used and is conditionally positive definite given that $\beta \in [0, 2]$:

$$K = -d(\boldsymbol{x_i}, \boldsymbol{x_j})^{\beta},$$

while the Gaussian kernel possesses a nice property of being positive definite for $\sigma \in \mathbb{R}^{+}$. In practice, problem-specific distance measures often lead to DS-kernels that are not positive definite. As mentioned in [8]–[12], kernels not being positive definite for support vector machine may still lead to impressive results, hinting at this possibility in KMEBs given the close problem formulation. Even though the distance kernel is only conditionally positive definite, from the practical application of the prototype extraction in the current study it shows good results, as we may observe in Section VI.

The distance itself occurs as a hyper-parameter in DS-kernels, allowing for comparative evaluation of categorical data for a number of distances. Moreover, k_0 kernel is widely used as a superposition of the Dirac kernel, also known as the overlap kernel [13]:

$$k_{i,j} = f_p(\frac{1}{n}\sum_{k=1}^{n} f_a([x_{ik} = x_{jk}])).$$

Feature	Mean	Std	Unique	Min	Max
Friends	4.50	3.86	30	0	33
Quests completed	265.01	38.84	250	79	442
Achievements	57.26	7.09	49	39	90
Mining	39.22	44.82	176	0	185
Plants	19.61	28.14	146	0	170
Kills Monsters	6423.68	1777.38	2813	2494	15027
Loot Total Items	974.59	326.20	1216	235	2519
Death Monsters	13.70	10.49	70	0	99
Auctions	13.44	22.76	137	0	212

TABLE II
THE STATISTICS BEFORE USING MIN MAX SCALER OF THE DATA FOR LEVEL 32 OF GAME TERA.

We notice, that evaluating the Dirac kernel is equivalent to computing the hamming similarity, which is defined as $s_{hamming} = 1 - d_{hamming}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Hence, it is possible to apply any of the selected Distance Substitution kernels described in Table I as k_0 kernel using the hamming distance. We can show it as k_0 kernel can take form:

$$k_{0(i,j)} = f_{DS}(\frac{1}{n}\sum_{k=1}^{n} f_a([x_{ik} = x_{jk}])) =$$

$$f_{DS}(s_{hamming}) = f_{DS}(1 - d_{hamming}).$$

IV. DATASET

The telemetry data is provided from a massively multiplayer online role-playing game TERA, used for behaviour clustering in [6]. TERA has typical MMORPG features such as questing, crafting, and player versus player action. It also enables in-game purchases. Characters may be of different race and different class, whereas each race has a set of unique skills and class induces class-specific skills and abilities. Moreover, certain set of skills, i.e. Mining, Planting, Auctions are not essential to be completed in the game course, being optional resource gathering skills the player can choose to develop.

The data comprises of 10 features, that are Level (level of character in the game), Friends (number of friends in the game), Quests Completed (number of quests completed), Achievements (number of achievements earned), Mining and Plants (level in skill respectively), Kills Monsters (number of AI-controlled enemies killed by character), Loot Total Items (total number of items that character picked up during the game), Death Monsters (the number of times the character had been killed by AI-controlled enemies) and Auctions (number of times that character created or purchased something from an auction).

We extract the prototypes based on level 32 (the highest level in the dataset) to understand the behavioural patterns of players that have achieved considerable results. The final data results in 3817 characters. For the analysis we use nine above-mentioned gameplay features (related to how characters are played), depicted in Table III alongside the statistical summary.

The data was pre-processed using Min Max scaler, to avoid the problems of mixing of data types, as advised in [6]:

$$\boldsymbol{F}' = \frac{\boldsymbol{F} - \min \boldsymbol{F}}{\max \boldsymbol{F} - \min \boldsymbol{F}}$$

Kernel	Distance					
	sqeuclidean	**euclidean**	**minkowski (p=4)**	**city block**	**cos**	**hamming**
Gaussian	6	6	6	6	6	6
Distance	5/8*	4*/8*	6	6	6	19*
Laplace	6	6	6	6	6	6
Inverse Multiquadric	6	6	2*/7	6	6	6
Rational Quadratic	6	5/7	6	6	6	6

TABLE III
IN THIS TABLE WE OBSERVE THE NUMBER OF SUPPORT VECTORS EXTRACTED BY DIVIDE AND CONQUER ALGORITHM FOR EACH RESPECTIVE KERNEL AND DISTANCE. AS THE NUMBER OF SUPPORT VECTORS IS NOT MONOTONE, IT POSES A CHALLENGE TO FIND A CORRESPONDING HYPERPARAMETER AND EXTRACT THE EXACT NUMBER OF PROTOTYPES. HENCE, WE PROVIDE THE APPROXIMATION OF THE ALGORITHM, THAT RETURNS THE 'CLOSEST' BIGGEST AND SMALLEST NUMBER OF SUPPORT VECTORS. WE NOTE, THAT FOR THE CASE OF HAMMING DISTANCE THE MINIMUM NUMBER OF SUPPORT VECTORS IS 19 FOR ALL HYPERPARAMETER INTERVAL, HENCE, IT IS NOT POSSIBLE TO RUN HYPERPARAMETER OPTIMIZATION AT THAT SETTING. THE VALUES MARKED BY STAR ARE NOT INCLUDED IN THE ENSEMBLE AS, BY OUR EMPIRICAL OBSERVATION, THE GREATER THE DIFFERENCE IN SUPPORT VECTORS NUMBER, THE MORE DIFFERENT WOULD EXTRACTED PROTOTYPES PROTOTYPES BE.

Distance	Definition
Minkowski	$(\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p)^{\frac{1}{p}}$
Euclidean	$(\sum_{i=1}^{n} (x_i - y_i)^2)^{\frac{1}{2}}$
Sq. Euclidean	$\sum_{i=1}^{n} (x_i - y_i)^2$
Manhattan	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Hamming	$1 - \sum_{i=1}^{n} [x_i = y_i]$
Cosine	$1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$

TABLE IV
CONCISE OVERVIEW OF THE SELECTED DISTANCES USED IN ENSEMBLE. FOR HAMMING DISTANCE $[x_i = y_i] = 1$ IF $x_i = y_i$, 0 IF $x_i \neq y_i$

for each feature $\boldsymbol{F}$ of the dataset. We note that before using KMEBs we do not use bins for ranges of categorical values, rather use the data only processed by Min Max scaler to map intervals to $[0, 1]$. It shows the ability of the algorithm to process the categorical data even given the different number of unique values in each feature.

Categorical variables are frequently handled using one-of-k encoding or one-hot-encoding and then use of discrete kernels. However, as the number of values that is possible to take is over 4 thousand, it becomes computationally inefficient when it comes to distance calculation, as the dimensionality of data rises.

V. ENSEMBLE APPROACH

For Ensemble approach we set up a goal for every distance-kernel pair to extract $p^* = 6$ prototypes. It makes a total of 30 distance-kernel pairs with 5 kernels (Table I) and 6 distances (Table IV). We define Kernel Minimum Enclosing Balls training with $t_{max} = 100$ iterations of Frank-Wolfe on the column data matrix $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$, $n = 3817$ as for the number of training points and $m = 9$ the number of features described in Table III.

Controlling the number of support vectors is possible using the kernel parameters: σ for Laplace and Gaussian kernel, c for Rational Quadratic and Inverse Multiquadric, and finally β for Distance kernel. Considering the non-monotonous dependency of support vectors number on the respective parameter, we apply an algorithm called Divide and Conquer [7] that allows us to control the number of support vectors in the resulting solution of KMEBs. Namely given the fixed number p^* of support vectors, it attempts to find such parameter χ that for fixed number of iterations inside of Frank-Wolfe algorithm t^*_{max} and fixed distance function d^* for DS-kernel, $p(\chi, t^*_{max}, d^*) = p^*$ holds. The algorithm returns the value of parameter χ and corresponding support vector matrix $S_{m \times p^*}$, where support vector matrix is a matrix of datapoints that correspond to the non-zero values of the Lagrange multiplier. In case of the algorithm's non-convergence, it provides the support vector matrix such that the number of support vectors in it is the closest to 6. The example of it can be seen at Table III, showing the number of support vectors extracted for each distance-kernel pair.

The Divide and Conquer approach [7] finds the parameter values that correspond to the highest and lowest support vector numbers in the initial parameter interval and performs a bisection. We choose for next step the left half-interval if $|p(t^*_{max}, \sigma'_{higher}, d^*) - p^*| > |p(t^*_{max}, \sigma'_{lower}, d^*) - p^*|$ and right interval otherwise, where σ'_{lower} and σ'_{higher} are two values defining the half-interval. Then we check, whether $p^* \in [p(\sigma'_{higher}), p(\sigma'_{lower})]$. If it does not, then the algorithm returns one step back and chooses the other side of the interval. After that, the current half-interval is divided into $g = 10$ equal parts and this division is passed as an interval to the next iteration.

Based on the previous step, we calculate the frequency of each appeared support vector for the kernel-distance pairs that returned exactly with weight $w = 1$ and $w = 0.5$ to both of the approximate solutions, in order to account for both returned support vector matrices. Then the frequency of the appearance of each support vector is calculated as $\nu(\boldsymbol{s} \in \boldsymbol{S}_i) = \sum_i^h w(l(\boldsymbol{S}_i))$, where $w(l(\boldsymbol{S}_i) \neq 6) = 0.5$, $w(l(\boldsymbol{S}_i) = 6) = 1$, l is number of support vectors in the support vector matrix $S_{m \times l}$, h is the number of resulting distance-kernel support matrices with $h = 29$ according to Table III.

VI. OBSERVATIONS

The resulting classes obtained from the paper [6] using Simplex Volume Maximization (SIVM) are **Planters**, **Miners** (that have respectively average scores across the performance features, but very high Planting or Mining skill respectively), **Auction Devils** (focused on using the Auction house feature of

No	Friends	Quests Completed	Achievements	Mining	Plants	Kills Monsters	Loot Total Items	Deaths Monsters	Auctions	Type
1	15.0	279.0	80.0	73.0	7.0	11130.0	1574.0	26.0	201.0	Auction Devil
2	2.0	277.0	55.0	36.0	129.0	4696.0	659.0	4.0	3.0	Planter
3	5.0	177.0	48.0	28.0	1.0	6603.0	1308.0	80.0	0.0	Straggler
4	2.0	346.0	75.0	76.0	21.0	14649.0	2444.0	15.0	13.0	Elite
5	7.0	198.0	58.0	163.0	5.0	5867.0	1119.0	11.0	37.0	Miner
6	16.0	274.0	62.0	8.0	5.0	9180.0	901.0	6.0	0.0	Friendly Pro

TABLE V
PROFILES OF SIMPLEX VOLUME MAXIMIZATION(SIVM) [14] WITH MIN MAX SCALING.

No	ν	Feature									Type
		Friends	Quests Completed	Achieve-ments	Mining	Plants	Kills Monsters	Loot Total Items	Deaths Monsters	Auctions	
1	18.0	4.0	283.0	61.0	0.0	9.0	7283.0	1162.0	13.0	11.0	Exclude (svs=1)
2	8.0	15.0	279.0	80.0	73.0	7.0	11130.0	1574.0	26.0	201.0	Auction Devil
3	7.0	2.0	346.0	75.0	76.0	21.0	14649.0	2444.0	15.0	13.0	Elite
4	7	1.0	183.0	42.0	5.0	13.0	4638.0	647.0	7.0	0.0	Straggler
5	6.5	11.0	282.0	60.0	155.0	50.0	8539.0	1345.0	8.0	15.0	Friendly Pro/Miner
6	6.5	3.0	253.0	57.0	9.0	4.0	7447.0	1026.0	12.0	0.0	Central
7	6	6.0	285.0	72.0	122.0	128.0	6587.0	1165.0	11.0	16.0	Planter

TABLE VI
TOP 7 FREQUENT RESULTS FROM THE ENSEMBLE OF KMEBS. THE PROTOTYPE NUMBER ONE IS EXCLUDED FROM RESULTS AS EXACTLY SAME SUPPORT VECTOR IS OBTAINED FOR ALL DISTANCE-KERNEL PAIRS WHEN SUPPORT VECTOR MATRIX CONSISTS OF JUST ONE SUPPORT VECTOR.

the game, gaining Achievements, Loot, strong social networks and high Mining skills), **Friendly Pros** (similar to the Auction Devils, but exhibit low Auction and Loot scores, and otherwise strong scores in the performance features), **Elite** (high scores overall, except for Mining/Plants and deaths from monsters with no Auctions created) and finally **Stragglers** with low scores overall and a high number from Death from Monsters.

Indeed, reproducing the results of SiVM is illustrated in Table V. The first prototype is clearly distinguished as the *Auction Devil*, second and fifth prototypes as respectively *Planter* and *Miner*, majoring in the respective skill and having the other skills averaged. The third prototype is exposing the *Straggler* behaviour but rather according to the k-means definition of the class with the death of monsters peaking. The fourth prototype is identified as *Elite* with a high number of Quests completed. The sixth prototype exhibits the behaviour of *Friendly Pro* with the high number of Friends.

Results in Table VI obtained from the ensemble show the similar narrative with two prototypes matching (highlighted in **bold**). Prototype number five exposes the behavior of *Friendly Pro*, yet due to high Mining skill, it can be posed as a *Miner*. With more extreme *Straggler* prototype four, the ensemble also returns *Central* prototype six acknowledging not only for the extreme data. The *Planter* at place seven exhibits high Mining revealing the complementary skills in the player's behavior.

VII. CONCLUSION

In this work, we have presented an ensemble approach for extracting prototypes from heterogeneous datasets with nominal and numerical behavioural data. Our approach showed itself successful with respect to determining all the player classes mentioned in [6] and moreover, provided us with more central prototypes. The ensemble showed another outlook on the data, exposing new dependencies and complementing the data analysis. At last, we have discussed the importance of Distance Substitution Kernels in terms of using the hamming distance and proved that for KMEBs the use of positive definite, or even conditional positive definite kernels is not necessary to acquire tangible results.

REFERENCES

[1] R. Sifa, A. Drachen, and C. Bauckhage, "Profiling in games: Understanding behavior from telemetry," *Social interactions in virtual worlds: An interdisciplinary perspective*, 2018.

[2] J. Paavilainen, H. Korhonen, E. Koskinen, and K. Alha, "Heuristic evaluation of playability: Examples from social games research and free-to-play heuristics," 2018.

[3] J. Bien and R. Tibshirani, "Prototype selection for interpretable classification," *The Annals of Applied Statistics*, pp. 2403–2424, 2011.

[4] C. Bauckhage and R. Sifa, "Joint selection of central and extremal prototypes based on kernel minimum enclosing balls," in *Proc. DSAA*, 2019.

[5] A. Drachen, J. Green, C. Gray, E. Harik, P. Lu, R. Sifa, and D. Klabjan, "Guns and guardians: Comparative cluster analysis and behavioral profiling in destiny," in *CIG 2016*, pp. 1–8, IEEE, 2016.

[6] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, "Guns, swords and data: Clustering of player behavior in computer games in the wild," in *CIG 2012*, pp. 163–170, IEEE, 2012.

[7] H. Kondratiuk and R. Sifa, "Towards structural hyperparameter search in kernel minimum enclosing balls," in *Proc. LION (accepted and to be published)*, Springer, 2021.

[8] B. Haasdonk and C. Bahlmann, "Learning with distance substitution kernels," in *Pattern Recognition, 26th DAGM Symposium Proceedings*, vol. 3175 of *Lecture Notes in Computer Science*, pp. 220–227, Springer, 2004.

[9] C. Bahlmann, B. Haasdonk, and H. Burkhardt, "Online handwriting recognition with support vector machines - a kernel approach," in *IWFHR 2002*, pp. 49–54, IEEE, 2002.

[10] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[11] B. Haasdonk and D. Keysers, "Tangent distance kernels for support vector machines," in *16th ICPR*, pp. 864–868, IEEE Computer Society, 2002.

[12] P. J. Moreno, P. Ho, and N. Vasconcelos, "A kullback-leibler divergence based kernel for SVM classification in multimedia applications," in *Advances in NIPS*, pp. 1385–1392, MIT Press, 2003.

[13] C. Garcia Marquez, "Multivariate kernel functions for categorical variables," 2014.

[14] C. Thurau, K. Kersting, and C. Bauckhage, "Yes we can: simplex volume maximization for descriptive web-scale matrix factorization," in *Proceedings of the 19th CIKM*, pp. 1785–1788, ACM, 2010.