Fuzzy Multiset Clustering for Metagame Analysis

Alexander Dockhorn, Tony Schwensfeier, and Rudolf Kruse
Computational Intelligence Group, Computer Science Department,
Otto-von-Guericke University, Universitätsplatz 2, 39106 Magdeburg,
{alexander.dockhorn, tony.schwensfeier, rudolf.kruse}@ovgu.de

Abstract

Developing agents for automated game playing is a demanding task in the general game production cycle. Especially the involvement of frequent balance changes after the release, which for example often occur in collectible card games, require constant updates of the developed agent. The game's developers need to continuously analyze and understand the current meta-game for adjusting the agent's parameters, making balance changes to the game, and, thereby, sustaining the satisfaction of its player base. The underlying analysis largely depends on evaluating players' play traces. Necessary adjustments to the agent's and the game's parameters are taken care of by the game's developers. This paper proposes a first step in automatically observing the current state of a collectible card game, which will assist the developers in their understanding of established deck archetypes and, therefore, speed up the update cycle. Fuzzy multisets are used for modeling decks and frequently occurring subsets of cards. We propose the definition of a (fuzzy) multiset centroid to uniquely represent the cluster and its contained decks and show that it is better able to match the deck archetype than the often reported deck core. The proposed clustering procedure identifies deck archetypes and keeps track of its common variants in the current meta-game. We evaluate the approach by comparing the result of our clustering procedure with a hand-labeled data set and show that it is able to reproduce clusters of similar quality to a labeling provided by experts.

Keywords: Fuzzy multisets, Clustering, Meta-game analysis, Hearthstone

1 Introduction

Automated game playing poses many interesting challenges to the development of artificial intelligence agents. While many studies presented good results on full information games, the agent's performance is often restricted by partial information on the current game state. The online game Hearthstone: Heroes of Warcraft (in short Hearthstone) [3] is such a partial information game, which is currently very popular among players [18] and motivated many interesting works in the field of computational intelligence in games.

Hearthstone is a collectible card game in which players create their own decks to play against each other. During a game, players do not know about their opponent's deck and hand cards. While this paper will only discuss rules and characteristics of the game that influence the deck building, the interested reader is referred to some excellent resources on the web [3, 9] which offer comprehensive reviews of the game's mechanics.

When building a deck, users often combine cards and their effects with a certain strategy in mind. Players of the game often refer to deck archetypes, when they discuss decks with a common theme and similar card sets. Such a deck archetype may develop due to the popularity of a certain strategy and its accompanied deck. However, many players do not own all the necessary cards of a specific deck they are trying to build, therefore, many variants of these deck archetypes exist.

In regard to creating an autonomous agent, estimating the opponent's upcoming actions is crucial in choosing their own actions. However, due to the opponent's cards being hidden, agents are limited in their capability of predicting these moves. Algorithms like Information Set Monte Carlo Tree Search (MCTS) [12, 19] try to handle this problem by randomly sampling the opponent's hand cards to create a determinization of the game state. Based on the generated determinization the search process simulates its own and its opponent's actions and chooses the most promising action.

Sievers and Helmert [17] developed an extended version of Information Set MCTS, which does not sample a single determinization, but multiple determinizations of the game state. For each of these determinizations, the algorithm performs a separate run of Information Set MCTS. The results of each run are later aggregated to determine the agent's next action. Sievers and Helmert evaluated their approach on the game Doppelkopf, in which player's need to guess their teammates during the first turns of every game. This critical guess can have a high impact on the following actions. Their approach showed to be valuable in estimating the risk of the next action and improved the agent's overall performance. A study by Dockhorn et al. [6] further extended this approach by using neural networks for guiding the simulation, therefore, improving the quality of simulations during the search process and its result.

In a recent paper, we introduced an autonomous agent for Hearthstone [7], which implements a similar approach to the one presented in [6]. We observed that the prediction accuracy and the agent's playing performance is still limited, which seems to be due to the enormous number of possible game state determinizations. We reduced the game state's sample space by using information of previously played cards to predict likely cards on the opponent's hand. Using card co-occurrences of previously seen cards and the opponent's hand cards we were able to create sample's game state determinizations with a higher likelihood. We observed an improvement of the agent's playing performance in comparison to a uniform sampling. However, agents which were given complete information still outperformed the proposed agent. From this, we infer that further increasing the accuracy of the card sampling may in turn further improve the agent's performance.

In the pursuit of creating a better agent for Hearthstone, we want to enhance the agent's prediction capabilities by correctly modeling deck archetypes. In Section 2 we review restrictions of the deck building process and provide a short overview of the theory of (fuzzy) multisets and various clustering approaches. In the subsequent Section 3, these methods will be used to develop a theoretical model of deck archetypes and how to mine them from a database of recent games. Especially the advantages of using fuzzy multisets instead of crisp multisets are highlighted based on some explanatory examples. We further present a case study, which is based on extracting deck archetypes and their centroid representation from actual playing data of the game Hearthstone (Section 4). We compare our result with a hand-labeled data set and show that the developed approach is able to identify deck archetypes of similar quality. The paper concludes with a short analysis of the proposed approach and its possible application to next card prediction.

2 Preliminaries

We begin this section with a short overview of deck archetypes and how they are defined in Hearthstone. We further provide detailed explanations on (fuzzy) multisets and hierarchical agglomerative clustering algorithms, which will be used to model and mine deck archetypes in Section 3.

2.1 Deck Building and Deck Archetypes

In Hearthstone a deck is a set of 30 cards, which can be chosen out of the 1000 cards currently available in the game. Each card offers certain effects which can be used to affect the current game state. Some of these effects create useful synergies that players try to exploit during the game, e.g. attack with a minion card, which will get damaged during the fight and follow up by healing this minion using a spell card. The choice of cards to be put in the deck is restricted by a small set of rules:

- players can only include cards they currently own (are unlocked on their player's account)
- a deck belongs to one out of 9 hero classes whom are limited to a subset of about 400 cards each
- a deck can only include cards that are either neutral or specific to the chosen hero class
- depending on its rarity, a card can be included either once or twice
- a deck can be made for a specific game mode that adds additional restrictions, e.g. standard, or arena

While players can create a large number of different decks not all of them are equally successful. The most successful decks define the meta and get known as meta-decks. Often these meta-decks will spawn multiple variants in which players replace just a few cards without changing the main theme of the deck.

A deck archetype describes the resulting cluster of decks with common card subsets. In this work, we will distinguish included cards into two groups, namely core and variant cards. While the core of a deck archetype contains cards that are included in all instances of this archetype, the inclusion of variant cards depends on the given instance. Core cards often define the main building block of the archetype and its accompanying strategy. In contrast, the variant cards are often chosen by the player, reflecting of personal preferences or restrictions in the deck building process. The following Subsection will introduce (fuzzy) multisets, which will later be used to model deck archetypes.

2.2 (Fuzzy) Multisets/Bags

Multisets (also called bags) are collections of objects in which an object can be represented multiple times. In this paper, we will closely follow the notation introduced by Miyamoto [13] and Yager [20]. We denote a multiset by:

$$M = \{C_M(x)/x \ : \ x \in X\} \tag{1}$$

in which X is the set of elements that can be included and C_M a function that maps each object x_i to its number of copies n_i in M:

$$C_M : X \to \mathbb{N} \qquad C_M(x_i) = n_i \tag{2}$$

The collection of all possible multisets of a universal set X is denoted by $\mathcal{M}(X)$.

For comparing two multisets L and M inclusion is defined by

$$L \subseteq M \text{ iff } C_L(x) \leq C_M(x) \text{ holds } \forall x \in X \tag{3}$$

and (as a consequence) equality is given by:

$$L = M \text{ iff } C_L(x) = C_M(x) \text{ holds } \forall x \in X \tag{4}$$

Union, intersection, and addition are defined pointwise for all $x \in X$ by:

$$C_{L \cup M}(x) = C_L(x) \vee C_M(x) \tag{5}$$

$$C_{L \cap M}(x) = C_L(x) \wedge C_M(x) \tag{6}$$

$$C_{L \oplus M}(x) = C_L(x) + C_M(x) \tag{7}$$

where $\vee$ and $\wedge$ imply the max and min operators.

A fuzzy extension of multisets was first introduced by Yager (using the term fuzzy bags) [20]. Here, the sample fuzzy multiset

$$A = \{(x, 0.5),\ (x, 0.3),\ (y, 1),\ (y, 0.5),\ (y, 0.2)\} \tag{8}$$

denotes the occurrence of each object and its membership degree. For simplicity we group objects of the same kind and their membership degrees, such as in:

$$A = \{(0.5, 0.3)/x,\ (1, 0.5, 0.2)/y\}$$

in which the memberships $\{0.5, 0.3\}$ and $\{1, 0.5, 0.2\}$ correspond to the objects x and y, respectively. Therefore, in fuzzy multisets $C_a(x)$ is a finite multiset of the unit interval [20].

For each object x we further define the membership sequence to be the decreasingly-ordered sequence of elements in $C_A(x)$. We will make use of the standard form introduced by Miyamoto [13] :

$$(\mu_A^1(x), \ldots, \mu_A^p(x)), \qquad \mu_A^1(x) \geq \cdots \geq \mu_A^p(x) \tag{9}$$

Let $L(x; A)$ be the length of the membership sequence $(\mu_A^1(x), \ldots, \mu_A^p(x))$ of multiset A be denoted by:

$$L(x; A) = \begin{cases} \max\{j \ : \ \mu_A^j(x) \neq 0\} & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Any operation between two multisets A and B requires the membership sequences of each object to be of equal length. We define the length $L(x; A, B)$ of the resulting membership sequence to be:

$$L(x; A, B) = \max\{L(x; A),\ L(x; B)\} \tag{11}$$

For the sake of simplicity we assume a membership degree of:

$$\mu_A^i(x) = 0; \quad \forall i \text{ with } L(x; A) < i \leq L(x; A, B) \tag{12}$$

in case the object x is included less than $L(x; A, B)$ times in the multiset A (likewise for B).

Similar to crisp multisets we can define inclusion, equality, union, and intersection based on the membership sequences of each element. Let A and B be two fuzzy multisets.

$$A \subseteq B \text{ iff } \mu_A^j(x) \leq \mu_B^j(x) \text{ holds for } j = 1, 2, \ldots, L(x; A, B),\ \forall x \in X \tag{13}$$

$$A = B \text{ iff } \mu_A^j(x) = \mu_B^j(x) \text{ holds for } j = 1, 2, \ldots, L(x; A, B),\ \forall x \in X \tag{14}$$

Similarly, union and intersection are defined pointwise for all $x \in X$ by :

$$\mu_{A \cup B}^j = \mu_A^j(x) \vee \mu_B^j(x) \quad j = 1, 2, \ldots, L(x; A, B) \tag{15}$$

$$\mu_{A \cap B}^j = \mu_A^j(x) \wedge \mu_B^j(x) \quad j = 1, 2, \ldots, L(x; A, B) \tag{16}$$

To clarify the notation we provide the following short example. Consider the two fuzzy multisets A and B over the set of objects $\{x, y, z\}$:

$$A = \{(0.5,\ 0.2)/x,\ (1.0,\ 0.5,\ 0.2)/y\}$$
$$B = \{(1.0)/x,\ (0.7,\ 0.6)/y,\ (0.9, 0.5)/z\}$$

The length per object is:

$$L(x; A, B) = 2; \quad L(y; A, B) = 3; \quad L(z; A, B) = 2$$

For simplicity we extend the membership sequences for both multisets according to the maximal observed length:

$$A = \{(0.5,\ 0.2)/x,\ (1.0,\ 0.5,\ 0.2)/y,\ (0.0, 0.0)/z\}$$
$$B = \{(1.0,\ 0.0)/x,\ (0.7,\ 0.6,\ 0.0)/y,\ (0.9, 0.5)/z\}$$

Based on the extended membership sequences we can determine union and intersection of both multisets:

$$A \cup B = \{(1.0,\ 0.2)/x,\ (1.0,\ 0.6,\ 0.2)/y,\ (0.9, 0.5)/z\}$$
$$A \cap B = \{(0.5)/x,\ (0.7,\ 0.5)/y\}$$

2.3 Hierarchical Agglomerative Clustering

In this work, we are going to present the results of our deck archetype clustering process. Since the clustering algorithm itself is not the focus of this paper we only explain the approach that was most successful in our experiments. The interested reader is referred to specialized literature on the topics of data mining and (fuzzy) cluster analysis, which provide a more comprehensive review of alternative approaches than this paper could offer [1, 2, 4, 10, 16].

Hierarchical agglomerative clustering is a class of bottom-up clustering algorithms, in which each data point is assigned to a unique cluster during initialization. These clusters are iteratively merged according to a linkage criterion. The merge process is repeated until a minimum number of clusters is reached or all data points belong to a common cluster.

In this work we will make use of the following linkage criteria, which both determine the distance of two clusters based on the distances of points contained in differing clusters:

- **single linkage** reports the minimal distance between two points of different clusters

$$d_{single}(C_i, C_j) = \min_{a \in C_i,\ b \in C_j} d(a, b) \tag{17}$$

- **complete linkage** reports the maximal distance between two points of different clusters

$$d_{complete}(C_i, C_j) = \max_{a \in C_i,\ b \in C_j} d(a, b) \tag{18}$$

3 (Fuzzy) Multiset Analysis of Deck Archetypes

In the previous section, we discussed various building components for the deck archetype mining algorithm we are proposing in this section. We will first define how a deck archetype can be represented in terms of fuzzy multisets and further describe a mining routine based on the hierarchical agglomerative clustering algorithm.

A natural representation of a deck is a multiset of cards.

$$D = \{C_D(x)/x \mid x \in X\} \tag{19}$$

where $C_D(x) \in \mathbb{N}$ is the number of inclusions of card x in deck D. Due to the restrictions of Hearthstone's deck building process, any card can be included twice or less $C_D(x) \leq 2$. It is also known that each deck has exactly 30 cards, which is equal to the sum of object counts or membership degrees in the deck.

3.1 Modelling Deck Archetypes

The Hearthstone community defines a deck archetype to be a collection of decks with a common set of cards. In the following, we are going to model a deck archetype to be the representative of a cluster of decks.

Lets consider two crisp decks D_1 and D_2 over the set of elements $X = \{a, b, c, d, e, f\}$ of the form:

$$D_1 = \{1/a,\ 1/b,\ 2/c,\ 1/d,\ 0/e,\ 2/f\}$$
$$D_2 = \{1/a,\ 1/b,\ 2/c,\ 0/d,\ 2/e,\ 1/f\}$$

The intersection $M_{D_1 \cap D_2}$ of these two decks is the multiset:

$$M_{D_1 \cap D_2} = \{1/a,\ 1/b,\ 2/c,\ 0/d,\ 0/e,\ 1/f\}$$

While the resulting set describes the core of these two decks, the information of possible variants is lost during the generation of the common subset. A similar problem occurs if we generate the union $M_{D_1 \cup D_2}$ of both decks:

$$M_{D_1 \cup D_2} = \{1/a,\ 1/b,\ 2/c,\ 1/d,\ 2/e,\ 2/f\}$$

While the union operator preserves information on the inclusion of d and e we misleadingly represent these variants, i.e. based on its count in $M_{D_1 \cup D_2}$ variant object d is indistinguishable from the core objects a and b (similar observations can be made for the objects c and e). Hence, objects with different inclusion patterns in D_1 and D_2 are equally represented in the merged multiset. Replacing the union with the addition operation would yield similar problems and also increase the cardinality of the resulting multiset.

For the crisp multiset we define the average multiset $M_{\langle L,M \rangle}$ of two multisets L and M to include the average number of occurrences per object in these multisets and denote it by:

$$C_{\langle L,M \rangle}(x) = \frac{C_L(x) + C_M(x)}{2}, \quad \forall x \in X \tag{20}$$

Hence, the average of clusters D_1 and D_2 is:

$$M_{\langle D_1, D_2 \rangle} = \{1/a,\ 1/b,\ 2/c,\ 0.5/d,\ 1/e,\ 1.5/f\}$$

While the average operator already clearly distinguishes the inclusion patterns for a, b and d, we can still observe problems with varying numbers of inclusion, e.g. a and e. However, extending the representation to fuzzy multisets can help to solve this problem.

For this purpose, we transfer the average operator for crisp multisets to fuzzy multisets by calculating the average of every element of an object's grade sequence.

Thus, the average operator for two fuzzy multisets A and B can be denoted by:

$$\mu^i_{\langle A,B\rangle}(x) = \frac{\mu^i_A(x) + \mu^i_B(x)}{2}, \quad i = 1, \ldots, p, \ \forall x \in X \tag{21}$$

Representing both decks as fuzzy multisets results in the following centroid:

$$D_1 = \{(1)/a,\ (1)/b,\ (1,1)/c,\ (1)/d,\ (0)/e,\ (1,1)/f\}$$
$$D_2 = \{(1)/a,\ (1)/b,\ (1,1)/c,\ (0)/d,\ (1,1)/e,\ (1)/f\}$$
$$M_{\langle D_1,D_2\rangle} = \{(1)/a,\ (1)/b,\ (1,1)/c,\ (0.5)/d,\ (0.5,0.5)/e,\ (1.0,0.5)/f\}$$

To ensure a stable clustering process we want to adjust the definition of the (fuzzy) multiset centroid to fulfill the associative property, since the result of merging multiple multisets should be independent of their merging order, specifically we want the following properties to be fulfilled:

$$\begin{aligned} C_{\langle\langle D_1,D_2\rangle,D_3\rangle}(x) &= C_{\langle D_1,\langle D_2,D_3\rangle\rangle}(x), \quad \forall x \in X \\ M_{\langle\langle D_1,D_2\rangle,D_3\rangle} &= M_{\langle D_1,\langle D_2,D_3\rangle\rangle} \end{aligned} \tag{22}$$

Let a cluster $\mathcal{C}$ be a multiset over the set $\{M_1, \ldots, M_n\}$ of multisets over the set of objects X. The centroid $\langle\mathcal{C}\rangle$ of cluster $\mathcal{C}$, which is itself a multiset over the set of objects X, should be independent of the order of inclusion of said multisets, thus fulfilling the associative property of the order of merges. For this purpose, we generalize the average operator of two multisets (Equation 20) to take the number of inclusions per multiset into account:

$$C_{\langle\mathcal{C}\rangle}(x) = \frac{\sum_{M_i \in \mathcal{C}} C_{M_i}(x) \cdot C_{\mathcal{C}}(M_i)}{\sum_j C_{\mathcal{C}}(M_j)}, \quad \forall x \in X \tag{23}$$

The same can be done for a cluster of fuzzy multisets

$$\mu^k_{\langle\mathcal{C}\rangle}(x) = \frac{\sum_{M_i \in \mathcal{C}} \mu^k_{M_i}(x) \cdot C_{\mathcal{C}}(M_i)}{\sum_j C_{\mathcal{C}}(M_j)}, \quad k = 1, \ldots, p, \quad \forall x \in X \tag{24}$$

We will use the cluster centroid to represent the cluster and all its contained decks in a single (fuzzy) multiset.

3.2 Clustering of (Fuzzy) Multisets

For mining deck archetypes we are going to apply the hierarchical clustering algorithm using single and complete linkage to a data set of recently played decks. Both linkage methods need a suitable distance measure to group data points into clusters of similar objects.

To measure the distance of two multisets L and M we define their Euclidean distance by:

$$d_{euclid}(L, M) = \left(\sum_{x \in X} \big(C_L(x) - C_M(x)\big)^2 \right)^{\frac{1}{2}} \tag{25}$$

and transfer the definition to be applied to fuzzy multisets A and B:

$$d_{euclid}(A, B) = \left(\sum_{x \in X} \sum_{i}^{L(x;A,B)} \big(\mu^i_A(x) - \mu^i_B(x)\big)^2 \right)^{\frac{1}{2}} \tag{26}$$

In our work, we will compare results based on the Euclidean distance with results obtained from applying the Jaccard distance measure. Here we use the general Jaccard distance [11]:

$$d_{jaccard}(x, y) = 1 - \frac{|x \cap y|}{|x \cup y|} = 1 - \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)} \tag{27}$$

and apply it to two multisets L and M:

$$d_{jaccard}(L, M) = 1 - \frac{\sum_{x \in X} \min(C_L(x), C_M(x))}{\sum_{x \in X} \max(C_L(x), C_M(x))} \tag{28}$$

Similar to the Euclidean distance we transfer the equation to measure the distance of two fuzzy multisets A and B:

$$d_{jaccard}(A, B) = 1 - \frac{\sum_{x \in X} \sum_{j=1}^{L(x;A,B)} \mu^j_{A \cap B}}{\sum_{x \in X} \sum_{j=1}^{L(x;A,B)} \mu^j_{A \cup B}} \tag{29}$$

The clustering process is stopped in case only one cluster remains or the Jaccard distance of all cluster pairs is 1.0 during a single merge step. In the latter, none of the current clusters have any object in common.

4 Evaluation

All project files for the following evaluation are available at [5]. We evaluated our clustering approach using deck data of the HSReplay website [8]. The website offers easy access to a large collection of recently played games. Players can choose to use the Hearthstone Deck Tracker plugin, which automatically records played games and uploads them to the HSReplay servers. In return, players can access information on the probability of their opponent's cards while playing the game.

We have extracted a deck data set which contains data from February 5th to 20th 2019. Each deck entry stores the deck's cards, the total amount of games recorded during the two weeks, its average win-rate, average game-length, and average turn count. Additionally, each deck entry provides information on the suggested deck archetype, which was labeled by expert players.

We will compare the result of our clustering method with the labeling provided in the data set. For this purpose, we use the external validation measures homogeneity, completeness, and v-measure [15]. While homogeneity is satisfied in case the clusters contain only data points which are members of a single class (as labeled in the ground truth), completeness is satisfied if all the data points that are members of a given class are elements of the same cluster. The v-measure is the harmonic mean of homogeneity and completeness. All three measures provide values between 0.0 and 1.0, where larger values are desirable.

We first calculated the distance matrix of decks of the same hero class. Figure 1a shows the heat map plot of the Jaccard distance of all Druid decks contained in the data set encoded as fuzzy multisets. Clusters of similar groups are clearly distinguishable. Euclidean distance looks similar but is not limited to a range of 0.0 to 1.0, which makes it harder to define a cutoff threshold for stopping the clustering process.

For this reason, we decided to compare level-wise extraction of cluster labels and report the evaluation measures grouped by the number of clusters and the applied linkage method in Figure 2a-c. Changing the distance to Euclidean distance yielded similar results. The plots show that the best clustering result in regard to the v-measure of the true archetype labels was achieved by complete linkage and reporting a total of 13 clusters. The other methods performed worse in regard to homogeneity and completeness for a small number of reported clusters. Single linkage performed better in regard to the reported measures for a high number of clusters. However, reporting higher numbers of clusters reduces the usefulness of defining a deck archetype by being overly specific in regard to the player's strategy with this deck. Even when the card sets vary a bit more than in the larger deck archetypes the main theme of the deck still stays the same.

We plotted the clustering result of the best performing algorithm in Figure 1b using multi-dimensional scaling for projecting the multisets into 2-dimensional space [14]. We also added cluster centroids and cluster cores to the scaling process to compare their suitability for describing a deck archetype. While small clusters can be satisfyingly represented using both methods, decks of larger clusters seem to have a higher average distance to their cluster core than to their cluster centroid (compare dark blue and bright green clusters). We also calculated the sum of squared errors (SSE) of all points to their respective cluster centroids and cores and show the result in Figure 1c. Centroids are on average much closer to the decks contained in the same cluster than cores are, which highlights their suitability in encoding deck archetypes.

5 Conclusion

In this work, we proposed an automatic clustering process for deck archetypes and evaluated it in the context of the online collectible card game Hearthstone: Heroes of World of Warcraft. We chose to represent decks in the form of (fuzzy) multisets and define a centroid of clusters of such multisets. A clustering on player data was done using the hierarchical agglomerative clustering algorithm. We applied complete as well as single linkage and compared their results when using Euclidean distance or Jaccard distance. The evaluation of the clustering result shows that the outcome is comparable to a labeling by expert players. Experiments on the average distance to cluster representatives show that the cluster centroid performs best in describing the deck archetype, while not losing expressiveness on the deck archetype's core and its variants.

We plan on extending this work by using found deck archetypes to predict the opponent's deck at run-time and use this to better sample a likely determinization of the game state. We further want to explore the applicability of the proposed clustering approach in a stream mining context. From a developers point of view, it could be interesting to quickly recognize changes in the meta-game.

References

[1] C. C. Aggarwal, C. K. Reddy, Data clustering: algorithms and applications, CRC Press, Boca Raton, FL, 2013.

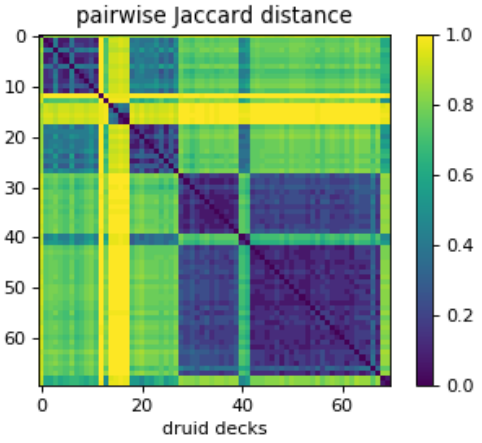
[2] M. R. Berthold, C. Borgelt, F. Höppner, F. Klawonn, Guide to Intelligent Data Analysis, Texts in Computer Science, Springer London, London, 2010.

[3] Blizzard Entertainment, Hearthstone webpage, accessed on 13.04.2019.
URL `https://playhearthstone.com`

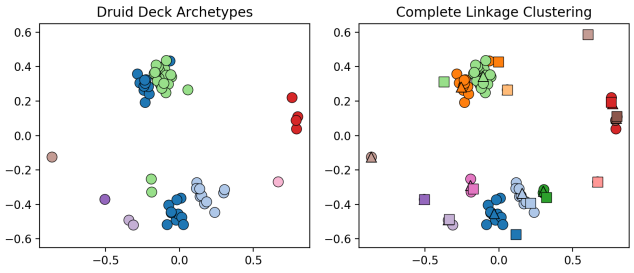
[4] C. Borgelt, R. Kruse, Agglomerative fuzzy clustering, in: International Conference on Soft Methods in Probability and Statistics, Springer, 2016, pp. 69–77.

[5] A. Dockhorn, Project files and supporting material, accessed on 16.05.2019.
URL `https://github.com/ADockhorn/FuzzyDeckClustering`

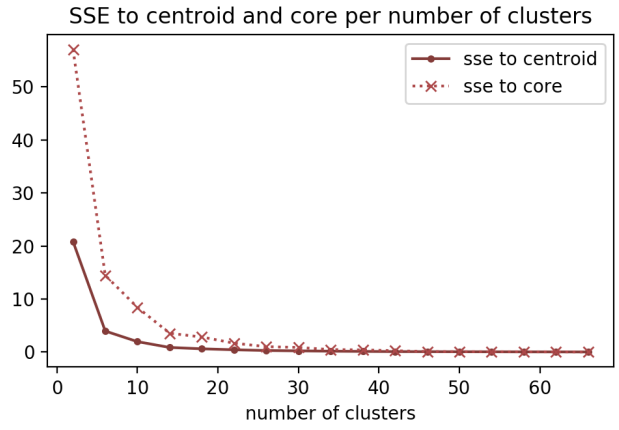
[6] A. Dockhorn, C. Doell, M. Hewelt, R. Kruse, A decision heuristic for Monte Carlo tree search doppelkopf agents, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–8.



(a) distance matrix of Druid decks

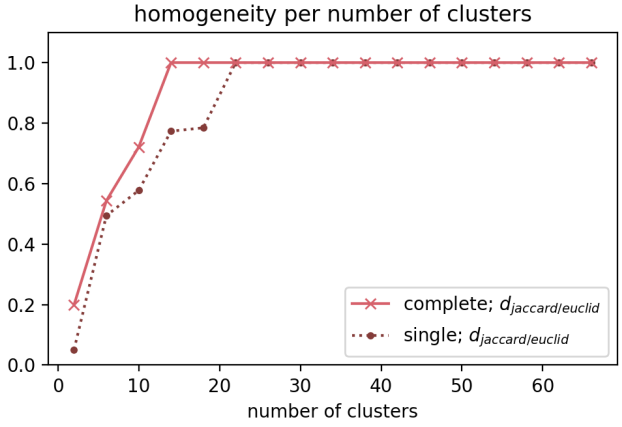


(b) 2d-projection of druid decks (●) their archetypes (color), and their cores (▲) and centroids (■)

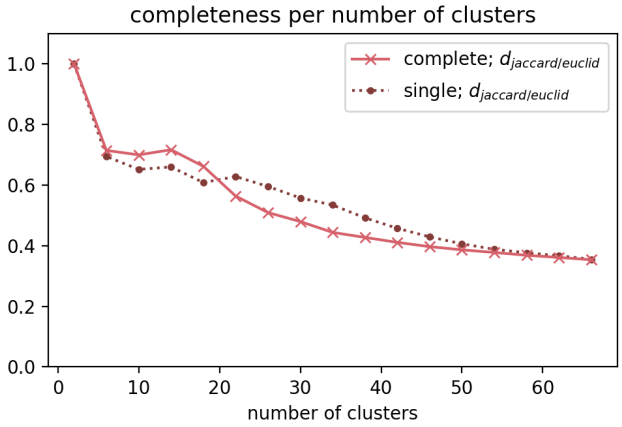


(c) comparison of the sum of squared errors of decks to their related deck archetype's centroid and core

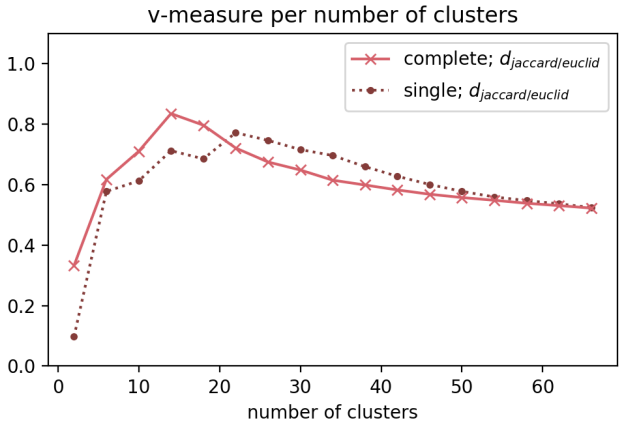
Figure 1: analysis of internal cluster metrics, archetypes, and their cores and centroids



(a) homogeneity



(b) completeness



(c) v-measure

Figure 2: comparison of clustering results using external validation measures on druid deck archetypes

[7] A. Dockhorn, M. Frick, Ü. Akkaya, R. Kruse, Predicting Opponent Moves for Improving Hearthstone AI, in: J. Medina, M. Ojeda-Aciego, J. L. Verdegay, D. A. Pelta, I. P. Cabrera, B. Bouchon-Meunier, R. R. Yager (Eds.), 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2018, Springer International Publishing, 2018, pp. 621–632.

[8] HearthSim, HSReplay.net, accessed on 13.04.2019.
URL https://hsreplay.net/decks/

[9] Hearthstone Top Decks, Hearthstone beginners guide 2018 guides, tips, and tricks for new players!, accessed on 13.04.2019.
URL https://www.hearthstonetopdecks.com/hearthstone-beginners-guide-2017-guides-tips-tricks-new-players/

[10] R. Kruse, C. Borgelt, C. Braune, S. Mostaghim, M. Steinbrecher, Computational Intelligence, 2nd Edition, Texts in Computer Science, Springer London, London, 2016.
URL http://link.springer.com/10.1007/978-1-4471-5013-8

[11] M. Levandowsky, D. Winter, Distance between sets, Nature 234 (5323) (1971) 34–35.

[12] J. Long, N. Sturtevant, M. Buro, T. Furtak, Understanding the success of perfect information monte carlo sampling in game tree search, AAAI Conference on Artificial Intelligence (2010) 134–140.

[13] S. Miyamoto, Fuzzy multisets and their generalizations, in: C. Calude, G. Puaun, G. Rozenberg, A. Salomaa (Eds.), Multiset Processing, Mathematical, Computer Science, and Molecular Computing Points of View [Workshop on Multiset Processing, WMP 2000, Curtea de Arges, Romania, August 21-25, 2000], Vol. 2235 of Lecture Notes in Computer Science, Springer, 2000, pp. 225–236.

[14] A. A. O'Connell, I. Borg, P. Groenen, Modern Multidimensional Scaling, Vol. 94 of Springer Series in Statistics, Springer New York, New York, NY, 2005.

[15] A. Rosenberg, J. Hirschberg, V-measure: A conditional entropy-based external cluster evaluation measure, Computational Linguistics 1 (June) (2007) 410–420.
URL http://acl.ldc.upenn.edu/D/D07/D07-1043.pdf

[16] K. H. a. Sadaaki Miyamoto, Hidetomo Ichihashi, Algorithms for Fuzzy Clustering: Methods in C-Means Clustering with Applications, 1st Edition, Studies in Fuzziness and Soft Computing 229, Springer-Verlag Berlin Heidelberg, 2008.
URL http://gen.lib.rus.ec/book/index.php?md5=9EA3A9597668B70D3AE134FEBFA94B1F

[17] S. Sievers, M. Helmert, A doppelkopf player based on UCT, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 9324, 2015, pp. 151–165.

[18] Statista, Inc., Number of Hearthstone: Heroes of Warcraft players worldwide as of November 2018 (in millions), accessed on 13.04.2019.
URL https://www.statista.com/statistics/323239/number-gamers-hearthstone-heroes-warcraft-worldwide/

[19] D. Whitehouse, E. Powley, P. Cowling, Determinization and information set monte carlo tree search for the card game dou di zhu, in: Proceedings of the 2011 IEEE Conference on Computational Intelligence and Games (CIG'11), IEEE, 2011, pp. 87–94.

[20] R. R. Yager, On the Theory of Bags, International Journal of General Systems 13 (1) (1986) 23–37.