Array 14 (2022) 100186

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Selecting a clustering algorithm: A semi-automated hyperparameter tuning framework for effective persona development ☆

Elizabeth Ditton [a,*], Anne Swinbourne [a], Trina Myers [b]

[a] James Cook University, 1 James Cook Drive, Douglas, 4811, QLD, Australia
[b] Queensland University of Technology, 2 George St, Brisbane City, 4000, QLD, Australia

ARTICLE INFO

Keywords:
Automated persona development
Machine learning
Clustering algorithms
Hyperparameter tuning
Internal evaluation metrics

ABSTRACT

When approaching a clustering problem, such as during persona development, selecting the most appropriate algorithm and parameter combination is essential. The hyperparameter tuning process required to determine the best combination is often tedious and thus automated through evaluation metrics. However, there are no ground truth values available for the empirical evaluation of clustering algorithms and existing internal metrics cannot comment on the quality of a set of clusters for their proposed use case. This paper presents a semi-automated framework for the hyperparameter tuning of clustering algorithms for persona development, HyPersona, which minimises the manual intervention required through simple evaluation and the production of informative graphs and early-stage personas. Within HyPersona, an internal metric focused on aspects necessary to developing quality persons, average feature significance (AFS), is proposed to assist in the evaluation of results. HyPersona was validated through application to a real-world persona development problem, evaluating the three most widely used clustering algorithms for persona development. HyPersona was compared to existing hyperparameter tuning and persona development methods and developed personas of a comparable quality whilst reducing manual intervention. The proposed internal metric, AFS, was found to provide a unique insight into the performance of cluster sets for persona development.

1. Introduction

Clustering is an area of unsupervised machine learning that attempts to find structure in unstructured data by creating groups of similar values [1,2]. There are numerous approaches to clustering, and each is proficient at finding clusters of a particular nature. One of the primary challenges of clustering is that algorithm selection can have a drastic impact on the clusters developed and the performance of a particular algorithm is often dependent on the nature of the clusters in the data [1]. Even two similar algorithms may find completely different sets of clusters in the same data set [1]. Furthermore, clustering algorithms are notoriously difficult to evaluate as, generally, there is no ground truth available and multiple sets of clusters created from the same data set could be equally valid [1].

One prominent example of a clustering problem is persona development. A persona is a description of a fictitious person used to describe analytical data and customer segments in a manner that emphasises human attributes and empathy [3,4]. Traditionally, personas are used during design or marketing to represent a particular type of target user [3]. More recently, personas have moved from consumer marketing into a wider variety of industries, with personas commonly playing an integral role in the design and development of human–computer interfaces [3]. The strength of personas comes from their ability to humanise data and communicate information without the need for domain-specific jargon [3]

Manual persona development is both time and resource intensive, and often requires a high level of specialisation to perform [5,6]. Not only does the high cost of persona development act as a barrier to persona use, but the cost also makes maintaining and updating personas difficult [5,6]. There has been a push towards more automated persona development methods to address the weaknesses of manual persona development [5,6]. However, one of the major criticisms of automated and semi-automated persona development methods is that they cannot capture the complex concepts and nuance that are key aspects of manually developed personas [5,6]. Current semi-automated approaches often rely heavily on manual guidance before a clustering algorithm is applied to the data and further input through feedback from a

☆ This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.
* Corresponding author.
E-mail addresses: elizabeth.forest@my.jcu.edu.au (E. Ditton), anne.swibourne@jcu.edu.au (A. Swinbourne), trina.myers@qut.edu.au (T. Myers).

https://doi.org/10.1016/j.array.2022.100186
Received 3 April 2022; Received in revised form 8 May 2022; Accepted 11 May 2022
Available online 18 May 2022

subject-matter expert (SME) or interviews with audience segment representatives are sought to capture the depth expected of personas [5–7]. Most of these automated and semi-automated approaches rely on a small, popular selection of clustering algorithms with little documented analysis performed before selecting a clustering algorithm, meaning the differing qualities of clustering algorithms are not being employed [6,7].

The selection of a clustering algorithm and the algorithm parameters, a process known as hyperparameter tuning, is a considerable challenge when applying a clustering solution to real-world problems. Multiple iterations and considerable domain knowledge are often required to find an optimal algorithm configuration, and the process is often long and tedious [8,9]. Hyperparameter tuning is often automated in supervised problems, where the ground truth values are available. However, automated hyperparameter tuning requires accurate and objective evaluation metrics which are not available for clustering problems due to their unsupervised nature. Evaluation of both clustering algorithms and persona sets poses considerable problems. Existing methods of automated hyperparameter tuning for clustering algorithms often rely upon internal evaluation metrics [8,10,11] or having some ground truth labels available for external evaluation metrics [9,12]. However, internal evaluation metrics are often biased and do not give a comprehensive view of cluster quality [13]. External metrics require having some ground truth labels available, which moves the problem into the semi-supervised space, and means the results cannot be directly applied to unsupervised problems. Von Luxburg et al. [13] asserted that clustering algorithms cannot be evaluated independently of the context in which they will be used.

This study proposes a semi-automated clustering algorithm hyperparameter tuning framework designed for persona development, HyPersona, to begin to resolve the challenges with the tuning of clustering algorithms for automated persona development. The HyPersona framework performs an exhaustive grid search over a range of clustering algorithms and parameter combinations, determining and outputting relevant information, graphs, and primitive personas for each combination, which can then be used to identify the most appropriate algorithm. The HyPersona framework also applies simple evaluation, comparing the internal metrics of the resulting clusters to a set of predefined thresholds and dropping any results that do not meet the threshold to narrow down the valid algorithm–parameter combinations and reduce the amount of manual intervention required. To further assist in relevant cluster evaluation an additional internal metric, Average Feature Significance (AFS), was developed based on the premise that quality personas should have unique attributes and be significantly different from each other.

HyPersona was demonstrated and evaluated through application to data on cyclone preparatory behaviour, with the aim of the personas to facilitate targeted messaging around cyclone preparation behaviours in North Queensland (NQ), Australia. Domain-specific evaluation of these personas requires that the personas align with the behavioural theory around the performance of damage mitigation behaviours and represent the residents' perceptions and attitudes. The three most prominent algorithms in automated and semi-automated persona development were compared by HyPersona; (1) k-means [14–17]; (2) Agglomerative Hierarchical Clustering (AHC); and (3) Non-negative Matrix Factorization (NMF) [18]. The results of HyPersona were compared with the results of a fully automated hyperparameter tuning framework based on internal evaluation metrics.

A total of 12 algorithm–parameter combinations based on the three algorithms were tested. Five (41.6%) of the 12 algorithm–parameter combinations tested were ruled out during simple evaluation, which greatly reduced the amount of manual evaluation required. All the algorithm–parameter combinations that were ruled out were confirmed to have been invalid choices for the use case. Through manual domain-specific evaluation k-means with random initialisation was the algorithm–parameter combination selected as the best performer and a set of personas was developed from the results.

The internal metric introduced in this paper, AFS, was found to be a useful indicator of the quality of a cluster set for persona development and gave alternate insights into cluster quality to existing internal metrics. The HyPersona framework was found to develop a better set of personas for the use case than a completely automated hyperparameter tuning framework based on a singular internal metric. HyPersona was also compared to an existing semi-automated hyperparameter tuning framework for clustering algorithm, Hypercluster [10], and was found to facilitate a more efficient evaluation and algorithm selection process.

Although targeted towards persona development, the HyPersona framework has implications for the broader scope of clustering algorithm applications, primarily through providing a model that facilitates domain-specific evaluation and proposing a new internal metric. The HyPersona also demonstrates that clustering algorithms can identify clusters that reflect complex factors, such as perceptions and attitudes, with minimal manual intervention. In summary, this paper presents several key contributions:

- HyPersona, a framework for use-case focused semi-automated hyperparameter tuning of clustering algorithms that extends on existing ideas and approaches.
- A methodology for semi-automated persona development that focuses on the impact of algorithm choice.
- An internal evaluation metric for clustering algorithms that rewards cluster centroids that are significantly different to each other, an element important for quality persona development amongst other use-cases.
- The HyPersona framework is applied to a real-world use case and evaluated in comparison to existing methods of hyperparameter tuning and persona development.

The remainder of this paper is structured as follows. Section 2 discusses the related work and motivation behind HyPersona. Section 3 details the HyPersona framework. Section 4 describes the case study and parameters used. Section 5 gives the results of the case study. Section 6 discusses the results in terms of the research questions. Then Section 7 concludes the paper.

2. Related work

2.1. Clustering algorithms

This study focuses on the three algorithms identified as most widely used within the persona development field [19].

K-means K-means is one of the most widely used and well-known clustering algorithms [1]. The premise of k-means is to partition the data set to identify the optimal centroids. The optimal centroids are identified through an iterative process of assigning each data point to its closest centroid, creating the clusters, and then updating the centroid to the cluster mean until the centroids no longer change [14–17]. There are many popular iterations of k-means, one of the most popular is k-means++ [20], which selects the initial centroid values based on their distance from existing centroids rather than randomly, as in the original algorithm.

Agglomerative Hierarchical Clustering (AHC) AHC algorithms are depicted as a binary tree, which can be split at any point to create the desired number of clusters, and is either agglomerative or divisive [1,2,21,22]. AHC starts with each data point as its own cluster and recursively combines the two most similar clusters until all the data points are in a single cluster [1,2,21,22]. AHC is often defined in terms of the linkage used, which is the metric used to determine the similarity of two clusters.

The linkages used in this paper are: (1) Ward's linkage [23], which measures the within-cluster variance; (2) complete linkage, which measures the maximum distance between points in a pair of clusters; (3) average linkage, which measures the average distance between each value in each cluster; and (4) single linkage, which measures the minimum distance between points in a pair of clusters.

Non-negative Matrix Factorisation (NMF) NMF finds a pair of non-negative matrices, W and H, whose product approximates the non-negative matrix of the data set [18]. Each row in W represents a data-point in terms of its importance to component c, and each row in H gives the importance of a feature for component c [18]. By setting the number of components to the number of clusters desired, clusters can be created by determining the component each data-point has the strongest affinity for Lee and Seung [18]. There are two primary solvers used for NMF: the CD solver [24], and the MU solver [25].

2.2. Cluster evaluation

There are two general categories of cluster evaluation metrics, internal and external [2,21]. Additionally, there are also meta-criteria that can be used to evaluate the quality of a clustering algorithm [13].

2.2.1. Internal evaluation metrics

Internal methods measure the cluster quality with similarity metrics, usually measuring: the inter-cluster separability; the intra-cluster homogeneity; or a combination of both [21]. Internal evaluation metrics can be useful but tell very little about the clusters developed and whether one algorithm is better than another is [13]. Most internal metrics favour particular types of clustering algorithms making them quite biased [13]. For example, a metric measuring inter-cluster separability will prefer an algorithm with a similar basis, such as k-means, while a metric measuring the intra-cluster homogeneity will prefer a density-based algorithm.

During this study, three popular internal metrics are used as part of the HyPersona framework:

Silhouette Coefficient (SC) [26] The SC is based on how well defined the clusters are, taking the intra-cluster distances and the distances between a given cluster and the next closest cluster into account. SC scores are bounded between −1 and 1, where −1 represents incorrect or overlapping clusters, and 1 represents dense, well-separated clusters.

Calinski–Harabasz Index (CHI) [27] The CHI also attempts to score a set of clusters based on cluster definition, using the ratio of the sum of between-cluster dispersion and the within-cluster dispersion. Higher CHI scores relate to better defined clusters, and the values are not bounded.

Davies–Bouldin Index (DBI) [28] The DBI evaluates a set of clusters on how well separated they are, taking both the distance between clusters and cluster size into account. Lower DBI scores represent more distinct cluster partitions, with 0 being the best possible score.

2.2.2. External evaluation metrics

External cluster evaluation metrics compare the results of a given clustering algorithm to a set of ‘‘correct" clusters. However, there are several problems with testing accuracy. A set of classes may be based on theoretical differences that are not sufficiently represented in the data, or do not reflect the “best" or most ‘‘natural" clusters [13,29]. Within one data set there may be multiple correct sets of clusters, meaning just because an algorithm does not find the expected clustering, does not necessarily mean that the algorithm did not find a valid set of clusters [29]. As there are no correct answers available in real-world data sets, external evaluation is not representative of real-world problem areas [13].

2.2.3. Meta-criteria

Meta-criteria can be useful in determining the quality of a clustering algorithm rather than the quality of the cluster set developed. Stability is a popular method of evaluation. If a clustering algorithm re-run on the same data consistently develops the same clusters the algorithm is considered stable [13]. Unstable algorithms are considered unreliable, and generally unsuitable for further use [13]. Using statistical tests to determine if the clusters developed differ significantly from each other can also be useful. If a test finds a pair of clusters do not deviate from each other, the algorithm has likely identified overlapping clusters rather than the desired, well-separated clusters [13]. A meta-criterion used in HyPersona is cluster size, as small clusters cannot be representative of a significant portion of the population, instead, they are likely to represent outliers, which is not often the desired result of persona development.

2.3. Hyperparameter tuning for clustering algorithms

The selection of an algorithm and parameters, a process known as hyperparameter tuning, is a considerable challenge when applying clustering to a real-world problem as the selection of algorithm and parameters has a significant impact on the clusters developed. Due to the difficulties surrounding the evaluation of clustering algorithms, the process of hyperparameter tuning for clustering algorithms is often a tedious [8,9]. The effect of hyperparameters on clustering results cannot be described through a convex function, meaning inferences about the effect of the hyperparameters cannot be drawn, exacerbating the tedium of hyperparameter tuning as an exhaustive grid search is thus required [10].

Existing methods, such as the Hypercluster package [10], rely on internal metrics and user interpretation to determine the best performing algorithm. Hypercluster uses the SC, DBI, and CHI as well as cluster sizes as internal metrics when no ground truth values are available, then visualisation tools, such as heat maps, can be employed to determine which set of hyperparameters performed better overall [10]. Other methods for hyperparameter tuning of clustering algorithms are similar, either relying on an existing internal metric or moving the problem into the supervised or semi-supervised space by using ground truth values and external metrics.

2.4. Automated persona development

Persona development approaches vary along a scale from completely manual to almost completely automated. Historically most persona development approaches have been manual, based on rich qualitative data, such as data from interviews or in-depth case studies, and utilising the deep interpretation and extrapolation able to be performed during the manual persona creation process [5,6]. An automated approach uses a clustering algorithm or similar as part of a framework to develop fully realised personas, which results in a quicker and less resource-intensive persona development process [5,6]. However, the automated approach has been criticised as unable to capture the complex concepts and opinions that make personas so valuable [5,6]. Semi-automated approaches are everything between the manual and automated methods, from almost completely manual methods with the addition of statistical insights to almost fully automated methods that only rely on manual intervention to polish the final personas [5,6]. Semi-automated persona development methods can benefit from the strengths of both manual and automated methods, however, often fall into the pitfalls associated with each, such as a time and resource-intensive process and more shallow personas.

The trends within persona development literature are beginning to favour automated or semi-automated approaches [7]. Though approaches are rarely near fully automated, instead semi-automated approaches rely on manual creation of personas and/or prior data manipulation to mitigate the shallowness of automated results [7]. To

develop deeper personas with automated methods Salminen et al. [7] suggest the use of more complex computational techniques, such as using multiple techniques to identify different elements of a persona. However, to treat the different elements of an individual's behaviours or perceptions as distinct would be a flawed approach, as psychological theory and findings report that elements of an individual do influence each other and cannot be easily separated.

The majority of automated or semi-automated persona development approaches rely on one of a small set of clustering algorithms with limited prior analysis towards algorithm choice [7,19]. Assessing the performance of a wider range of clustering algorithms prior to automated or semi-automated persona development may assist in developing deeper and more nuanced personas by taking advantage of the differing nature of clustering algorithms. With a wider range of clustering algorithms and informed selection of the most appropriate algorithm, the need for more complicated approaches to persona development can be avoided.

Similar to cluster evaluation, persona evaluation is difficult due to there being no 'correct' answer. As a result, the evaluation and validation of automated and semi-automated persona development approaches tend to be informal and limited [7]. Other evaluation methods include further interviews or case studies with the participants or purely quantitative methods, such as the average Euclidean distance between personas [7]. In the current study, the personas will be evaluated based on how well they reflect behavioural theory and hold up to domain-specific evaluation.

3. Hypersona framework overview

The core of the HyPersona framework is to perform an exhaustive grid search over a range of algorithm and parameter combinations, calculate relevant metrics to be used for simple evaluation, and then output information on each combination that can be used to identify the most appropriate algorithm. HyPersona extends upon the semi-automated clustering algorithm hyperparameter tuning framework previously presented in [30] through the application of thresholds to rule out invalid cluster sets for persona development, the introduction of AFS, and the creation of early-stage personas. Fig. 1 gives a graphical overview of the automated portion of HyPersona.

HyPersona takes a dictionary that details each of the algorithms and parameters to be tested, which is expanded into a list of all possible algorithm and parameter combinations. The internal metrics, including the AFS, are calculated for each algorithm–parameter combination and then used to test the algorithm–parameter combination for validity based on whether their internal metrics meet certain thresholds, with any algorithm–parameter combinations that do not meet the threshold being dropped. The internal metrics and results of the simple evaluation are then outputted to a running CSV file which can be used to evaluate the performance of individual algorithm–parameter combinations and the overall algorithm performance. Graphs representing the key features of the clusters and early-stage personas are developed for each remaining algorithm–parameter combination to facilitate efficient domain-specific evaluation. The current iteration of HyPersona has been written in Python 3.8, utilising the many computer-science and scientific libraries available.

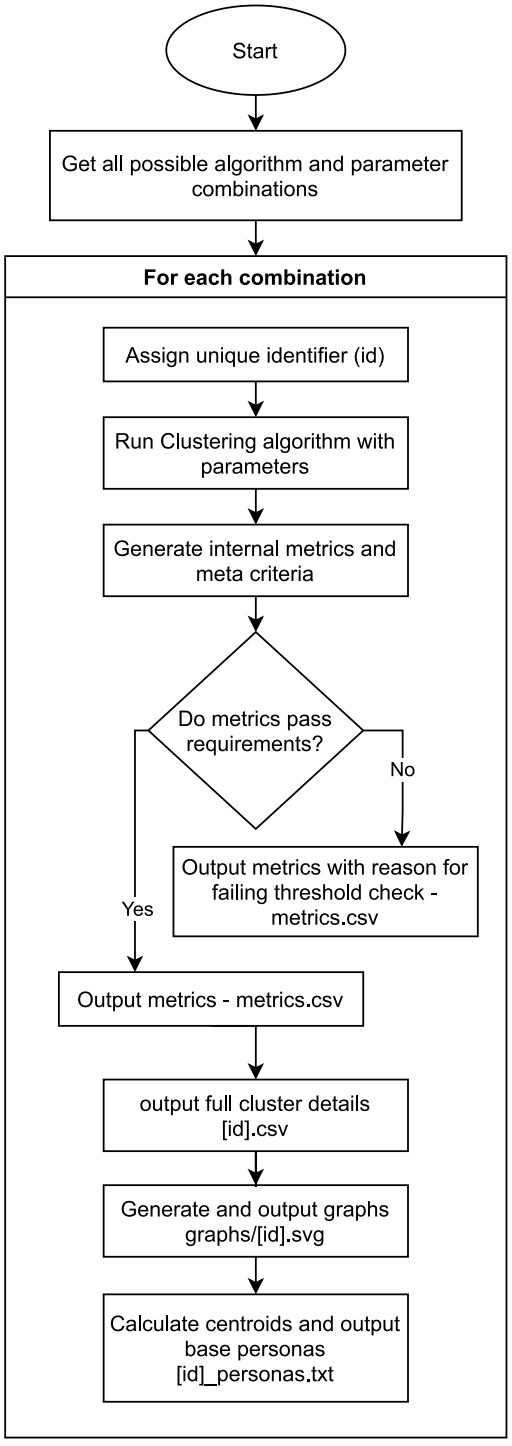

Fig. 1. A graphical representation of the automated portion of the HyPersona framework.

3.1. Inputs and data prerequisites

Three main inputs can be passed to the HyPersona framework: (1) the data to be clustered; (2) a dictionary of algorithms and parameters to be tested; and, optionally, (3) a range of domain-specific information to be used in outputs. Before running the HyPersona some initial testing is required to configure the internal metric thresholds, as the values considered acceptable can vary based on the data set used. The data passed into HyPersona is expected to be clean, numeric data, free of nulls. This is largely due to the requirements of many clustering algorithms not handling non-numeric or null data. Domain-specific information can be passed into HyPersona to determine which features are included in the graphs and acronyms can be provided to simplify the graphs. Key features that should always be included in the early-stage personas developed, such as demographic factors, can also be defined. Finally, aggregate features can be set, where outputs should also include the average of a selection of features. The aggregate features do not affect the clustering process, they are only to give more

```
{
[identifier]: {
    "algorithm": [algorithm class/function reference],
    "type": [one of: class | function | ensemble],
    "params": {
        [parameter name]: [list of potential values],
        ...
     }
...
}
```

Fig. 2. The schema for the algorithm dictionary.

concise outputs where there are multiple features relating to a single factor.

The algorithm dictionary details every clustering algorithm and parameter to be considered and assigns an identifier to each set. HyPersona uses the identifier alongside a number to represent each parameter combination to act as a unique identifier of an algorithm–parameter combination. The schema for the algorithm map is given in Fig. 2.

Each entry in the map applies to one clustering algorithm that can be provided as a class, which is the standard for the sklearn library, or a function, which is the standard for the pyclustering library. The `type` parameter defines what type of algorithm definition is given. When the type is ensemble value instead of including the algorithm and parameters directly, the map will include a set of dictionaries for each of the algorithms and parameters to be used as part of the ensemble. The `params` value contains another map of each parameter and the potential values to be used.

3.2. The framework core

The core of the HyPersona framework is an exhaustive grid search that runs each algorithm–parameter combination and outputs information about the results that can then be used to select the most appropriate algorithm. HyPersona first gets each possible algorithm–parameter combination from the algorithm dictionary and assigns it a unique identifier (id). The next step is to run each algorithm–parameter combination on the data and calculate the internal metrics of the cluster set results. The internal metrics are then compared to a set of predefined thresholds, with the cluster sets that do not meet the thresholds being dropped.

The internal metrics used are SC, CHI, and DBI, as well as the proposed, purpose-specific internal metric, AFS. The internal metrics were selected as they all primarily measure the separability and definition of the clusters, with poor values usually indicating overlapping or indistinct clusters which are factors that are also important to persona development. As well as the internal metrics, the cluster size is considered. As the desired personas would reflect the common, significant attitudes and beliefs within the population, rather than outlying opinions, the clusters should each contain a significant portion of the population.

The internal metrics of all algorithm–parameter combinations are outputted to a running CSV file (metrics.csv), for algorithm–parameter combinations that were dropped, the details of the threshold they did not meet are also included. For the algorithm–parameter combinations that were not dropped, graphs representing the cluster centroids and early-stage personas are developed. The graphs display the number of standard deviations each feature of the cluster centroid is from the population mean. When set, only key features are included in the graph and acronyms are used when available. A separate graph is given for each cluster centroid, and an SVG file containing the graphs is saved for the algorithm–parameter combination using the id ([id].csv). Similarly, the early-stage personas list all the values found to significantly differ from the population mean or between cluster centroids. For each feature the mean value for the cluster, the population mean, and the number of standard deviations the cluster mean differs from the population mean is listed. The early-stage personas are saved to a text file using the id ([id]_personas.txt).

3.3. Average feature significance

Alongside existing internal metrics, HyPersona uses a new internal metric specifically designed to target the elements important to developing quality personas, Average Feature Significance (AFS). The AFS metric is based on the premise that personas within a set of personas should have unique attributes and differ significantly from one another. As such, the value of AFS is based on the statistical significance of the features of each cluster centroid.

AFS gives the average number of features in a cluster that significantly differ from either the population mean or the other clusters. When the list of clusters is given as $c = \{c_1, \ldots, c_n\}$ and the distinct pairs of clusters, ${}_nC_2$, are given as $p = \{p_1 \ldots, p_m\}$. Let $t_1(c_i, \mu)$ return the number of features in the cluster, c_i, that are significantly different compared to the mean μ using a one-sample t-test and $t_2(p_i)$ return the number of features that are significantly different between a pair of clusters, p_i, using a two-sample t-test. Then, AFS can be defined as:

$$AFS = \frac{\sum_{i+1}^{n} t_1(c_i, \mu) + \sum_{j+1}^{m} t_2(p_i)}{n + m} \tag{1}$$

A feature is considered statistically significant if it has a p-value less than 0.05. The AFS is not bounded but will always be greater than 0, with higher values meaning that, on average, the features of the clusters are more significantly different.

3.4. Manual evaluation and persona creation

The final aspect of the HyPersona framework is to use the outputs to facilitate the manual, domain-specific evaluation of the algorithm–parameter combinations so that the most appropriate algorithm–parameter combination can be selected. There can be multiple valid clusterings of one data set, and internal metrics can be biased towards particular clustering algorithms, rewarding algorithms based on similar premises. For example, the SC is generally higher for convex clusters, meaning algorithms, like k-means, that tend to develop convex clusters are more likely to perform well. Thus, the top performing algorithm–parameter combination according to the internal metrics should not be automatically chosen. Instead, the internal metrics are used as guides to direct which cluster sets should be considered first. As AFS was developed with the goals of persona development in mind, AFS is used as the primary indicator of the quality of a cluster set for persona development.

Some domain-specific expertise is required to evaluate the results, and if key features are being used some domain-specific expertise may also be required to identify which features qualify. The process of domain-specific evaluation will differ depending on the use case. However, HyPersona is designed to make evaluation more straightforward with graphs and simple metrics.

Identifying algorithm–parameter combinations that have developed significantly similar cluster sets is one of the first steps during domain-specific evaluation. A pair of cluster sets are significantly similar if they are identical, or the differences between the cluster sets would not affect the interpretation of the clusters during persona development. The graphs developed by HyPersona allow for efficient comparison of cluster sets to determine similarity. When two cluster sets are significantly similar, the internal metrics determine which cluster set would be used.

Once the domain-specific evaluation has been used to determine the best performing algorithm–parameter combination, the early-stage personas are then used as a base for the fully realised personas. The early-stage persona files are simple to allow for the results to be transferred into any desired persona format, with the focus on the features

Table 1
Algorithm parameter combinations and unique identifiers.

ID	Parameters
AHC based algorithm–parameter combinations	
agg_heir_v0	linkage: Ward's [23]
agg_heir_v1	linkage: complete
agg_heir_v2	linkage: average
agg_heir_v3	linkage: single
K-means based algorithm–parameter combinations	
kmeans_v0	initialization: k-means++ [20]
kmeans_v1	initialization: random
NMF based algorithm–parameter combinations	
nmf_v0	solver: cd [24], iterations: 100
nmf_v1	solver: cd [24], iterations: 500
nmf_v2	solver: cd [24], iterations: 1000
nmf_v3	solver: mu [25], iterations: 100
nmf_v4	solver: mu [25], iterations: 500
nmf_v5	solver: mu [25], iterations: 1000

that significantly differ for each cluster and the features predetermined to be important for the persona creation. The early-stage personas minimise the amount of data interpretation required during the persona creation phase.

4. Hypersona case study

To evaluate the HyPersona framework it was applied to a real-world use case for personas. The selected use case requires a set of personas to target communication around cyclone damage mitigation behaviours. The HyPersona evaluation was designed to answer a set of research questions:

RQ1 How effective is the use of thresholds based on internal metrics at ruling out algorithm–parameter combinations?
RQ2 Is AFS a useful internal metric that provides alternate insights to existing internal metrics?
RQ3 How does the selection of algorithm–parameter combination based on the HyPersona framework differ from that based on an automated framework using an internal metric?

4.1. Algorithms and parameters

The algorithms selected to be compared were the three most prominent algorithms within the persona development field [19]: k-means, AHC, and NMF. The details of the algorithms and parameters used are given in Section 2.1.

Table 1 gives the specifics of each of the algorithm–parameter combinations and the id assigned. Based on inferences from behavioural models and requirements, the only number of clusters, k, used was 3.

4.2. Case study background

Tropical Cyclones form over the warm waters close to the equator, resulting in areas such as North Queensland (NQ), Australia, frequently experiencing cyclones during the summer months. In a recent survey, nearly all NQ residents (92%) reported having experienced at least one cyclone, with almost a third (29%) having experienced more than 5 cyclones [31]. Risk mitigation strategies can help reduce cyclone damage to structures, from simple, low-cost actions such as tidying a yard or securing loose outdoor items, to more difficult and costly actions such as adding structural upgrades to homes. Understandably, people are more likely to undertake the simple low-cost options, than the more difficult, high-cost options, despite the more expensive methods being highly effective [31,32].

Table 2
Key aggregate behavioural features and acronyms used.

Acronym	Feature description
Eff	Encompasses the perceived effectiveness of cyclone shutters to reduce damage, keep their family safe, increase property value, and for other purposes.
C	Encompasses financial, time, effort, and knowledge cost of having cyclone shutters installed.
PR	Encompasses the perceived personal risk of a cyclone; how the individual's daily life, job, mental health, and physical health would be affected.
GR	Encompasses the perceived general risk of a cyclone, the likelihood of catastrophic destruction, widespread death, the financial threat, and the threat to future generations.

To ensure that those living in NQ and the surrounding regions undertake all possible measures to protect both themselves and their property effective communication and education are required. As individuals have different perceptions surrounding mitigation behaviours, segmenting the audience into personas based on underlying motivators can allow for more effective communication targeting. As such, personas developed for this purpose can be evaluated based on how well they align with behavioural models that attempt to reflect the processes that determine the intent of an individual to perform a particular behaviour.

One of the most prominent behavioural models that focus on the performance of protective actions in response to and leading up to natural disasters is the Protective Action Decision Model (PADM) [33–35]. The PADM was selected based on the model's has been successfully applied to explain motivation to perform cyclone damage mitigation behaviours, and the model's history of being used to design and target messaging around protective behaviours [36–38]. The PADM proposes that an individual's exposure to, knowledge of, and the amount of attention paid to the risk play a role in protective action motivation as part of a pre-decisional phase [34,35]. The individual's motivation to perform the protective action is then suggested to be based on three key factors: (1) their perception of the threat itself, such as likelihood and severity; (2) their perception of the protective action, such as efficacy and cost; and (3) their perception of key stakeholders [34,35].

4.3. Case study data

This study used survey responses from 519 NQ residents on cyclone preparatory behaviours, psychological characteristics, and demographics [31]. Informed consent was obtained before any data was collected and all possible steps were taken to protect the privacy of the individuals who participated. The survey covered key elements identified as part of the risk mitigation decision process, as well as the likelihood that they will perform some risk mitigation behaviours, as well as more general demographic details [31]. The data was prepared by first converting any non-numeric features either through directly mapping the values, i.e. *{None, Low, Moderate, High} = {0, 1, 2, 3}*, or one-hot encoding when the values were not ordinal. Then any null values were replaced using an iterative imputation [39].

Key features were identified based on the PADM and where multiple elements were required to describe a single perception or belief, aggregate features were defined. Each key or aggregate feature was assigned an acronym. The aggregate features and acronyms are available in Table 2, and the key individual behavioural features are given in Table 3. The values of each key feature reflect how strongly an individual agrees with the given statement, larger values always mean a stronger level of agreement.

Table 3
Key individual behavioural features and acronyms used.

Acronym	Feature description
1–2C	Likelihood of a category 1-2 cyclone
3–4C	Likelihood of a category 3-4 cyclone
5C	Likelihood of a category 5 cyclone
VA	How visually appealing cyclone shutters are
AO	The individual feels they could organise cyclone shutter installation
GS	The perceived level of financial support the government would give in the event of a cyclone
TF	How often the individual thinks about cyclones
IS	Whether the individual has actively looked for ways to minimise cyclone damage
The possibility of a cyclone makes the individual feel	
S	Stressed
F	Fearful
H	Helpless
D	Depressed
Perceived damage caused by a	
1–2S	Category 1–2 cyclone
3–4S	Category 3–4 cyclone
5S	Category 5 cyclone
Likelihood to perform the following next cyclone season	
TT	Trim treetops and branches
CR	Check property for rust and rotten timber
CW	Check property walls and roof are secure
CF	Check fencing is not loose or damaged
CG	Clean gutters and down-pipes
Ply	Put plywood up on glass windows/doors
SO	Secure outdoor furniture and garden items
CY	Clear yard of any loose items
Likelihood of the individual to install cyclone shutters	
XU	Extremely unlikely
MU	Moderately unlikely
SU	Slightly unlikely
N	Neither likely nor unlikely
SL	Slightly likely
ML	Moderately likely
XL	Extremely likely

4.4. Internal metric thresholds

The thresholds for each of the internal metrics and the cluster size had to be set before the HyPersona framework was run. The thresholds were designed not to be too strict, instead, to only rule out inadmissible results. Any cluster with less than 5% of the data points was considered too small, as such clusters were likely to be representing edge cases. The AFS threshold was 15, as there were more than 30 key features, and if there were, on average, less than 15 significantly different features between clusters, the personas created from them were unlikely to have significantly different behavioural features. For the other internal metrics, SC values less than 0, CHI values less than 10, and DBI values greater than 5 were all found to be indicative of poorly formed or overlapping clusters. Algorithm–parameter combinations that did not meet these thresholds were dropped by the HyPersona framework.

4.5. Domain-specific evaluation

The cluster sets that were not dropped by HyPersona or found to be significantly similar to another cluster set were manually evaluated to determine the best performer. How well each cluster aligns with the PADM was determined using the graphs developed by HyPersona. That is, whether the features that indicate the individual's perceptions and attitudes towards cyclones and cyclone preparatory behaviours explain the individual's motivation to perform preparatory behaviours. The cluster sets that did not align with PADM were ruled out and the remaining cluster sets were ranked based on how well each cluster aligns with PADM. As the personas were intended for targeted messaging, to be most effective, each persona should represent a discrete segment of the population that would require different messaging to be most effective. As such, how distinct the clusters within each set were was also taken into account during ranking. After manual evaluation, the algorithm–parameter combination that produced the cluster set that ranked highest was selected as the best performer.

Table 4
HyPersona framework results.

ID	SC	CHI	DBI	AFS
agg_heir_v0	0.0663	38.141	3.3818	**67.33**
agg_heir_v1	0.0741	36.817	2.9288	53.67
agg_heir_v2[a]	**0.1742**	3.084	*1.2825*	16.50
agg_heir_v3[a]	*0.1684*	2.098	**0.6768**	0.00
kmeans_v0	0.0875	*47.084*	2.8595	58.00
kmeans_v1	0.0889	**47.095**	2.9145	*60.67*
nmf_v0	0.0429	26.947	3.3509	55.17
nmf_v1	0.0627	31.088	3.1030	56.33
nmf_v2	0.0655	30.114	3.0520	55.33
nmf_v3[a]	0.0207	6.873	3.5346	36.00
nmf_v4[a]	0.0207	6.873	3.5346	36.00
nmf_v5[a]	0.0207	6.873	3.5346	36.00

[a]The algorithm–parameter combination was dropped by HyPersona.

5. Results

The internal metrics of the cluster sets developed by the algorithm–parameter combinations are given in Table 4. The top score of each metric is given in bold, and the second-best score is italicized. All five of the algorithm–parameter combinations that were dropped failed to meet multiple thresholds. The dropped algorithm–parameter combinations all failed to meet the CHI threshold of 10 and had created clusters that contained less than 5% of the total data points. Additionally, agg_heir_v3 also failed to meet the AFS threshold, with an average of 0 significant features.

Using the graphs and early-stage personas developed by HyPersona, the clusters developed by kmeans_v1 and kmeans_v0 were found to be functionally identical, as the minor differences between the clusters developed would not have any impact on a set of personas developed. As such only kmeans_v1, which had the higher internal metric values, was considered. The graphs developed for agg_heir_v0, agg_heir_v1, kmeans_v1, and nmf_v1 are given in Fig. 3. The clusters have been re-ordered to allow for the most similar clusters to be compared to one another. The cluster sets developed by agg_heir_v0 and kmeans_v1 were quite similar, with each of the clusters following similar overall patterns, while the cluster set developed by nmf_v1 differed most greatly.

Based on the internal metrics, the domain-specific evaluation focused on agg_heir_v0 and kmeans_v1 first, followed by agg_heir_v1, and nmf_v1. Each cluster set was evaluated based upon how well it aligned with behavioural theory, and how distinctive each prospective persona would be was also considered during the domain-specific evaluation. Through the domain-specific evaluation, kmeans_v1 was determined to be the best performer. Compared to agg_heir_v1, k-means_v1 was selected as the difference between likelihoods to install cyclone shutters was more significant, and the average risk perceptions of each cluster within the set were more distinct.

A set of three personas were developed based on the early-stage personas produced by HyPersona for kmeans_v1. As there were no significant differences in age, gender, marital status, or location between the clusters, those demographic factors were not included in the final personas. The most important demographic factor was found to be previous experience with cyclones and cyclone damage. Each

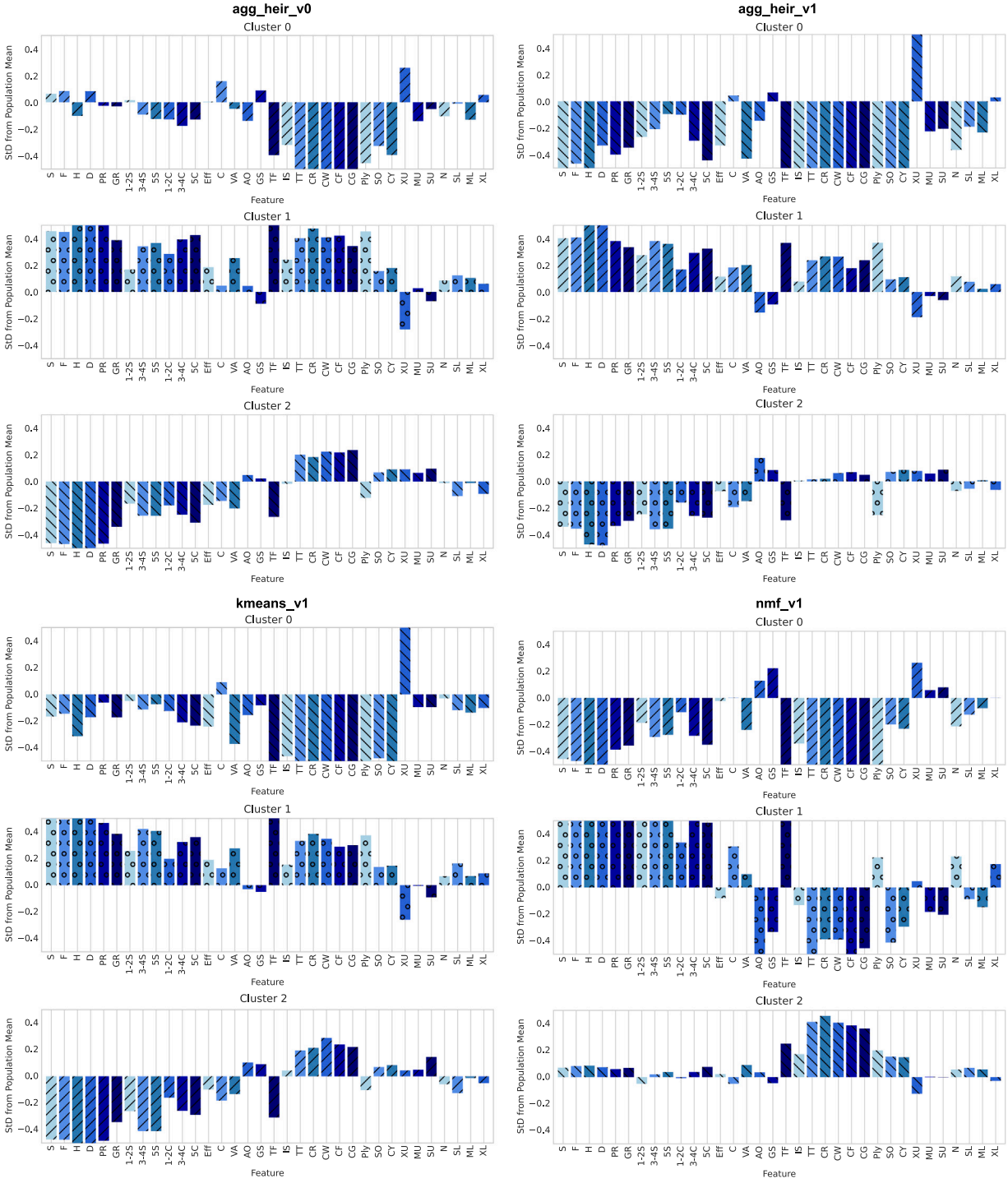

Fig. 3. The graphs developed by HyPersona for the algorithm–parameter combinations agg_heir_v0, agg_heir_v1, kmeans_v1, and nmf_v1. Each set of graphs give the number of standard deviations each of the key features are from the population mean for the centroid of each cluster.

persona was assigned an epithet to describe their attitude towards performing damage mitigation behaviours leading up to a cyclone. The three personas created are:

The Unconcerned (Cluster 0) Cyclones are not on the radar of the unconcerned persona. The unconcerned persona is the least likely to think about or discuss cyclones in their day-to-day life or to have looked for methods to help prevent cyclone damage. The unconcerned persona is the least likely to have experienced a cyclone and has a relatively low perception of the risk associated with cyclones. The unconcerned persona had the lowest self-reported likelihood to perform any of the preparatory behaviours or install structural upgrades to their property.

The Concerned (Cluster 1) The concerned persona is the most worried about a future cyclone. The concerned persona has the highest level of cyclone risk perception, with the most significant elements being the perceived impact of a cyclone on their mental and physical well-being and the feelings of helplessness and depression present when thinking about the possibility of

a cyclone. The concerned persona spends the most time thinking about and discussing cyclones and is most likely to have investigated ways to protect against a cyclone. The concerned persona self-reported as most likely to perform all available preparatory behaviours leading up to the next cyclone, and has the highest motivation to install structural upgrades, such as cyclone shutters. The concerned persona is most likely to have previously experienced cyclone damage with 70% of the individuals that make up the persona having reported receiving cyclone damage. Of those, almost half reported having received moderate or severe damage.

The Confident (Cluster 2) The confident persona has the lowest perception of the risks associated with cyclones and the severity of high category cyclones. Mirroring the concerned persona, the confident persona differs most in their feelings of helplessness and depression when thinking about a possible cyclone and the perceived impact of a cyclone on their mental and physical health. The confident persona self-reported being likely to perform simple preparatory behaviours, but are less likely to perform more difficult behaviours, such as putting up plywood or installing structural upgrades, such as cyclone shutters. The confident persona is most likely to have experienced a cyclone without receiving any damage, as approximately 46.3% of the individuals that make up the persona have experienced a cyclone and received no damage. Of those who experienced cyclone damage, they were least likely to have experienced significant damage with only 12.9% of the individuals that make up the persona having ever received moderate or severe damage.

6. Framework evaluation

HyPersona was applied to a real-world use case to demonstrate its ability to be effectively applied to persona development problems. Through the application of the HyPersona framework and the results found, HyPersona can be evaluated by how well it answers the research questions.

6.1. How effective is the use of thresholds based on internal metrics at ruling out algorithm–parameter combinations?

Five of the twelve algorithm–parameter combinations used, or just over 40%, were dropped as they did not meet the required thresholds. The dropped algorithm–parameter combinations had created heavily imbalanced clusters, with the clusters not meeting the minimum size threshold of 5% of the total data. Additionally, although not tested for, all the dropped algorithm–parameter combinations developed one cluster that contained more than 90% of the total data points. In comparison, the sizes of clusters that were developed by the algorithm–parameter combinations that were not dropped were more balanced. The size of each cluster created by each algorithm–parameter combination is given in Table 5.

As a result of the size imbalance, the clusters developed do not differ in a statistically significant way. The large clusters contain nearly all of the data points, and thus sit extremely close to the population mean. While the values of the small clusters differ greatly, due to their small sizes the differences were rarely statistically significant. This is reflected in the AFS of the dropped algorithm–parameter combinations being the lowest, with agg_heir_v3 not meeting the required threshold. The CHI also acts as an indicator of how well balanced the cluster sizes are, as none of the dropped algorithm–parameter combinations met the minimum CHI threshold.

Alternately, two of the dropped algorithm–parameter combinations performed quite well in terms of the SC and DBI, with agg_heir_v2 and agg_heir_v3 together achieving the best and second-best values for both DBI and SC. Without using the internal metrics or cluster sizes to rule out the incompatible algorithm–parameter combinations, agg_heir_v2 and agg_heir_v3 would have been considered based upon their SC and DBI scores. The dropped algorithm–parameter combinations would not have created quality personas. Thus, by automatically dropping 40% of the algorithm–parameter combinations, considerable manual time and effort was saved.

Table 5
Algorithm–parameter combination cluster sizes.

ID	Cluster 0	Cluster 1	Cluster 2
agg_heir_v0	131	186	202
agg_heir_v1	245	242	32
agg_heir_v2[a]	515	3	1
agg_heir_v3[a]	517	1	1
kmeans_v0	80	223	216
kmeans_v1	93	223	203
nmf_v0	239	35	245
nmf_v1	143	52	324
nmf_v2	124	52	343
nmf_v3[a]	479	9	31
nmf_v4[a]	479	9	31
nmf_v5[a]	479	9	31

[a]The algorithm–parameter combination was dropped by HyPersona.

Table 6
Pearson's correlation coefficient between internal metrics.

	SC	CHI	DBI	AFS
SC	1.000			
CHI	−0.112	1.000		
DBI	−0.954	0.342	1.000	
AFS	−0.526	0.864	0.733	1.000

6.2. Is AFS a useful internal metric that provides alternate insights to existing internal metrics?

The algorithm–parameter combinations that scored best in AFS differ from the best performing algorithm–parameter combinations according to the other internal metrics.

Table 6 gives the Pearson's Correlation Coefficient between each of the internal metrics based on the results of the HyPersona framework. A level of correlation was expected between the internal metrics as they were all rewarding similar traits in cluster sets. That is, all the internal metrics prefer well separated, convex clusters. The strongest correlation was between the SC and DBI, while the weakest correlation was between the SC and CHI. AFS has a moderate correlation to the SC and a strong correlation to both the CHI and DBI. However, the results of AFS were different enough to the existing internal metrics not to be redundant.

The algorithm–parameter combination that achieved the best AFS score, agg_heir_v0, achieved more mediocre scores in the other internal metrics, however, the domain-specific evaluation found agg_heir_v0 to be a serious contender. While, the algorithm–parameter combination that was determined to be the best performer, kmeans_v1, had the second best AFS score. The AFS score was also a primary reason why nmf_v1 was considered, which demonstrated interesting differences from the other algorithm–parameter combinations.

AFS was found to provide insight and information on cluster quality not otherwise present in existing internal metrics that were important to selecting relevant cluster sets for persona development. As such, AFS proved to be a useful addition to the hyperparameter tuning framework. Other problem areas where having distinct cluster centroids are important, may also benefit from applying AFS.

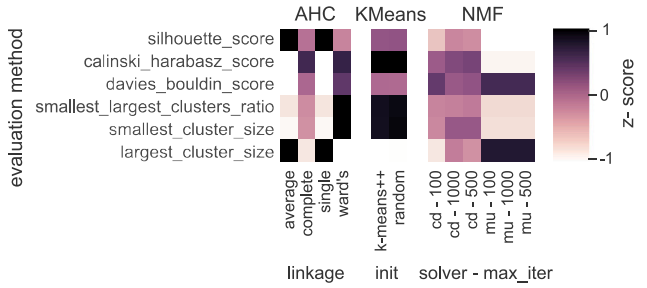

Fig. 4. Results of the Hypercluster [10] framework: heat map of the internal metrics.

6.3. How does the selection of algorithm–parameter combination based on the hypersona framework differ from that based on an automated framework using an internal metric?

Without any ground-truth values available, a fully automated hyperparameter tuning framework relies on internal metrics to determine the best performing algorithm. Based purely on an individual internal metric, an automated hyperparameter tuning method using the SC would select agg_heir_v2, a method based on the DBI would select agg_heir_v3, and a method based on the CHI would select kmeans_v1. As such, a framework based on the SC or DBI would give poor results, as both algorithm–parameter combinations selected by these metrics were ruled out by the proposed framework. While acting as a very useful guide, the algorithm with the best AFS, agg_heir_v0, also was not chosen as the best performing algorithm although agg_heir_v0 did produce an acceptable set of clusters for persona development.

Alternately, the algorithm–parameter combination that performed the best according to the CHI was selected by the proposed framework as the best performer. Additionally, once minimum thresholds were applied the next best SC score achieved is by kmeans_v1, and the best DBI score was achieved by kmeans_v0 which was found to be almost identical to kmeans_v1. This suggests that by using a combination of internal metrics and ruling out algorithm–parameter combinations that do not meet minimum thresholds the internal metrics may be reliable predictors of persona quality.

Hyperparameter tuning was also performed on the data set using the Hypercluster [10] framework with the same algorithm and parameter combinations. Hypercluster develops a heat map to graphically display the quality of a range of internal metrics, which has been given in Fig. 4. One important note in interpreting the heat map developed by Hypercluster is that it does not adjust for the fact that the closer the DBI score is to 0, the better quality the clusters are, which is the opposite of the other internal metrics. As such, in only the DBI row, the lower z-score is the better result.

Based on the heat map, when considering all the evaluation methods either k-means initialization appears to be the best performer, followed by AHC with single or average linkage. This was quite similar to the results found by the HyPersona framework, which was expected as a similar set of internal metrics were applied to the data in both cases. Other than providing an in-built tool for visualisation of the internal metrics, Hypercluster does not provide any additional information to the proposed semi-automated framework and still relies on the manual identification and selection of the best performing algorithm–parameter combination. As such, the Hypercluster framework does not give any insights into the nature and content of the clusters.

As there were no minimum thresholds applied or other information given by Hypercluster, determining the best performer was more difficult. For example, AHC with Ward's linkage, agg_heir_v0, and AHC with single linkage, agg_heir_v3, performed well in different internal metrics but overall appear to have performed similarly. However, AHC with single linkage, agg_heir_v3, was dropped by HyPersona due to cluster size, the CHI score, and the AFS score. As such, when using Hypercluster manual evaluation of AHC with single linkage, agg_heir_v3, would be required. Additionally, the heat map developed by Hypercluster only shows minimal differences between AHC with wards linkage, agg_heir_v0, and AHC with complete linkage, agg_heir_v1. By comparing the graphs developed for agg_heir_v0 and agg_heir_v1, both of which can be found in Fig. 3, it is apparent that the difference in linkage used had a significant impact on the clusters developed and thus the personas that would be developed.

Internal metrics cannot be solely used to identify the quality of a set of clusters for a specific purpose, such as persona development. Applying the Hypercluster framework for hyperparameter tuning of clustering algorithms requires significantly more manual intervention than applying HyPersona. The graphs developed by HyPersona simplify the manual evaluation process, saving considerable time compared to methods that only provide insights into the internal metrics.

Existing methods such as Hypercluster [10] still require manual, domain-specific evaluation for their effective application however do not facilitate this evaluation. HyPersona extends upon this approach by outputting relevant information and visualisations to assist in the efficient domain-specific evaluation and streamline the evaluation process by eliminating cluster sets that were not appropriate for the use case. As such, the algorithm–parameter combination chosen by HyPersona is assured to be useful and relevant to the use case.

7. Conclusion

The clustering algorithm and parameters used to solve a clustering problem greatly affects the solution reached and clusters developed. As such hyperparameter tuning is essential when applying a clustering algorithm to a problem. However, the subjective nature of cluster evaluation makes hyperparameter tuning difficult to automate, resulting in a time consuming, tedious process. Previous approaches to automated hyperparameter tuning for clustering have relied on having some ground truth labels available, moving the problem out of the unsupervised space, or on internal metrics, which are known to be biased and unreliable.

This paper proposed and tested a semi-automated framework for the hyperparameter tuning of clustering algorithms for persona development, the HyPersona framework. HyPersona uses an exhaustive grid search to verify all possible algorithm–parameter combinations against a set of naive evaluation thresholds. Easy-to-use graphs and metrics are then outputted for each valid algorithm–parameter combination which can then be used for effective comparison and domain-specific evaluation. As part of HyPersona, a new internal metric, AFS, was proposed. The AFS was found to provide insights into the cluster quality that were not present with existing internal metrics and acted as an indicator of cluster quality for persona development. Although targeted towards persona development, the HyPersona framework and the AFS metric could both be applied to a wide range of use cases.

HyPersona was tested through application to the real-world use case of developing personas to facilitate the targeting of information around cyclone preparatory behaviours. The three most prominent clustering algorithms in the persona development field were applied to this problem: AHC, k-means, and NMF. K-means with random initialisation was found to be the most effective solution and a set of 3 personas representing prominent attitudes towards cyclones and risk mitigation behaviours was developed. The comparison of algorithm results from the various algorithm–parameter combination demonstrated the importance of hyperparameter tuning in persona development methodologies. The HyPersona framework and personas developed were validated against an existing hyperparameter tuning framework for clustering, Hypercluster. HyPersona was found to facilitate the development of relevant, deep personas while minimising the amount of manual intervention required.

CRediT authorship contribution statement

Elizabeth Ditton: Writing – original draft, Conceptualization, Methodology, Software, Validation, Formal analysis. **Anne Swinbourne:** Supervision, Conceptualization, Writing – review & editing. **Trina Myers:** Supervision, Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Jain AK. Data clustering: 50 years beyond K-means. In: Award winning papers from the 19th International Conference on Pattern Recognition (ICPR), Pattern Recognit Lett In: Award winning papers from the 19th International Conference on Pattern Recognition (ICPR), 2010;31(8):651–66. http://dx.doi.org/10.1016/j.patrec.2009.09.011.URL http://www.sciencedirect.com/science/article/pii/S0167865509002323,

[2] Xu D, Tian Y. A comprehensive survey of clustering algorithms. Ann Data Sci 2015;2(2):165–93. http://dx.doi.org/10.1007/s40745-015-0040-1.

[3] Salminen J, Jansen BJ, An J, Kwak H, Jung S-g. Are personas done? Evaluating their usefulness in the age of digital analytics. Persona Stud 2018;4(2):47–65. http://dx.doi.org/10.21153/psj2018vol4no2art737, URL https://ojs.deakin.edu.au/index.php/ps/article/view/737.

[4] Thoma V, Williams B. Developing and validating personas in e-commerce: A heuristic approach. In: Gross T, Gulliksen J, Kotzé P, Oestreicher L, Palanque P, Prates RO, et al., editors. Human-computer interaction – INTERACT 2009, vol. 5727. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009, p. 524–7. http://dx.doi.org/10.1007/978-3-642-03658-3_56, URL http://link.springer.com/10.1007/978-3-642-03658-3_56.

[5] Mesgari M, Okoli C, Guinea AOd. Affordance-based user personas : A mixed-method approach to persona development. In: AMCIS 2015 Proceedings. Puerto Rico; 2015, URL https://aisel.aisnet.org/amcis2015/HCI/GeneralPresentations/1.

[6] Brickey J, Walczak S, Burgess T. Comparing semi-automated clustering methods for persona development. IEEE Trans Softw Eng 2012;38(3):537–46. http://dx.doi.org/10.1109/TSE.2011.60.

[7] Salminen J, Guan K, Jung S-G, Jansen BJ. A survey of 15 years of data-driven persona development. Int J Hum–Comput Interact 2021;1–24. http://dx.doi.org/10.1080/10447318.2021.1908670.

[8] Fan X, Yue Y, Sarkar P, Wang YXR. On hyperparameter tuning in general clustering problems. In: Proceedings of the 37th International conference on machine learning. Proceedings of machine learning research, vol. 119, PLMR; 2020, p. 2996–3007, URL http://proceedings.mlr.press/v119/fan20b.html.

[9] Van Craenendonck T, Blockeel H. Constraint-based clustering selection. Mach Learn 2017;106(9):1497–521. http://dx.doi.org/10.1007/s10994-017-5643-7.

[10] Blumenberg L, Ruggles KV. Hypercluster: a flexible tool for parallelized unsupervised clustering optimization. BMC Bioinformatics 2020;21(1):428. http://dx.doi.org/10.1186/s12859-020-03774-1.

[11] Shalamov V, Efimova V, Muravyov S, Filchenkov A. Reinforcement-based method for simultaneous clustering algorithm selection and its hyperparameters optimization. Procedia Comput Sci 2018;136:144–53. http://dx.doi.org/10.1016/j.procs.2018.08.247, URL http://www.sciencedirect.com/science/article/pii/S1877050918315527 Publisher: Elsevier.

[12] Minku LL. A novel online supervised hyperparameter tuning procedure applied to cross-company software effort estimation. Empir Softw Eng 2019;24(5):3153–204. http://dx.doi.org/10.1007/s10664-019-09686-w.

[13] Von Luxburg U, Williamson RC, Guyon I. Clustering: Science or art?. 2012, p. 65–79.

[14] Ball GH, Hall DJ. ISODATA, a novel method of data analysis and pattern classification. Technical Report, Stanford research inst Menlo Park CA; 1965.

[15] Lloyd S. Least squares quantization in PCM. IEEE Trans Inform Theory 1982;28(2):129–37.

[16] MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1. Oakland, CA, USA; 1967, p. 281–97.

[17] Steinhaus H. Sur la division des corp materiels en parties. Bull Acad Polon Sci 1956;1(804):801.

[18] Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature 1999;401(6755):788–91, Publisher: Nature Publishing Group.

[19] Salminen J, Guan K, Jung S-G, Chowdhury SA, Jansen BJ. A literature review of quantitative persona creation. In: Proceedings of the 2020 CHI Conference on human factors in computing systems. Honolulu HI USA: ACM; 2020, p. 1–14. http://dx.doi.org/10.1145/3313831.3376502, URL https://dl.acm.org/doi/10.1145/3313831.3376502.

[20] Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics; 2007, p. 1027–35.

[21] Saxena A, Prasad M, Gupta A, Bharill N, Patel OP, Tiwari A, et al. A review of clustering techniques and developments. Neurocomputing 2017;267:664–81. http://dx.doi.org/10.1016/j.neucom.2017.06.053, URL http://www.sciencedirect.com/science/article/pii/S0925231217311815.

[22] Xu R, Wunsch D. Survey of clustering algorithms. IEEE Trans Neural Netw 2005;16(3):645–78. http://dx.doi.org/10.1109/TNN.2005.845141.

[23] Szekely GJ, Rizzo ML. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. J Classification 2005;22(2):151–83. http://dx.doi.org/10.1007/s00357-005-0012-9.

[24] Cichocki A, Phan A-H. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans Fundam Electron Commun Comput Sci 2009;92(3):708–21, Publisher: The Institute of Electronics, Information and Communication Engineers.

[25] Févotte C, Idier J. Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput 2011;23(9):2421–56. http://dx.doi.org/10.1162/NECO_a_00168, Publisher: MIT Press.

[26] Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 1987;20:53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7, URL http://www.sciencedirect.com/science/article/pii/0377042787901257 Publisher: North-Holland.

[27] Caliński T, Harabasz J. A dendrite method for cluster analysis. Comm Statist Theory Methods 1974;3:1–27. http://dx.doi.org/10.1080/03610927408827101.

[28] Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 1979;PAMI-1(2):224–7. http://dx.doi.org/10.1109/TPAMI.1979.4766909, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[29] Färber I, Günnemann S, Kriegel H-P, Kröger P, Müller E, Schubert E, et al. On using class-labels in evaluation of clusterings. 2010, p. 1.

[30] Ditton E, Swinbourne A, Myers T, Scovell M. Applying semi-automated hyperparameter tuning for clustering algorithms. 2021, http://dx.doi.org/10.48550/ARXIV.2108.11053, arXiv URL https://arxiv.org/abs/2108.11053.

[31] Scovell M, McShane C, Swinbourne A, Smith D. North queenslanders' perceptions of cyclone risk and structural mitigation intentions. part I: psychological and demographic factors. 2018.

[32] Scovell M, McShane C, Swinbourne A, Smith D. Investigating factors that influence cyclone mitigation behaviour: a pilot study. Gold Coast, QLD, Australia: Australian Institute of Emergency Services; 2017, URL https://researchonline.jcu.edu.au/49758/.

[33] Lindell MK, Perry RW. Behavioral foundations of community emergency planning. Behavioral foundations of community emergency planning., Washington, DC, US: Hemisphere Publishing Corp; 1992.

[34] Lindell MK, Perry RW. The protective action decision model: theoretical modifications and additional evidence. Risk Anal Int J 2012;32(4):616–32.

[35] Terpstra T, Lindell MK. Citizens' perceptions of flood hazard adjustments: an application of the protective action decision model. Environ Behav 2013;45(8):993–1018.

[36] Scovell M, McShane C, Smith D, Swinbourne A. Personalising the message: Promoting cyclone protection in north queensland. Austr J Emerg Manag 2019;34(4):48–53.

[37] Scovell M, McShane C, Swinbourne A, Smith D. Applying the protective action decision model to explain cyclone shutter installation behavior. Nat Hazards Rev 2021;22(1):04020043. http://dx.doi.org/10.1061/(ASCE)NH.1527-6996.0000417.

[38] Doermann JL, Kuligowski ED, Milke J. From social science research to engineering practice: Development of a short message creation tool for wildfire emergencies. Fire Technol 2021;57(2):815–37. http://dx.doi.org/10.1007/s10694-020-01008-7.

[39] van Buuren S, Groothuis-Oudshoorn K. Mice: Multivariate imputation by chained equations in R. J Stat Softw Articles 2011;45(3):1–67. http://dx.doi.org/10.18637/jss.v045.i03, URL https://www.jstatsoft.org/v045/i03.