Clustering Ordinal Survey Data in a Highly Structured Ranking

by

Priy Werrij BSc.

Submitted to the Faculty of Sciences
in fulfilment of the Research Paper BA at the

Vrije Universiteit Amsterdam

July 2016

Author .....
Priy Werrij BSc.
Faculty of Sciences
July 14, 2016

Supervised by .....
Mark Hoogendoorn
Universitair Docent
Research Paper BA Supervisor

Accepted by .....
Internship office for Mathematics and Computer Science
Faculty of Sciences

Clustering Ordinal Survey Data in a Highly Structured Ranking

by

Priy Werrij BSc.

Submitted to the Faculty of Sciences
on July 14, 2016, in fulfilment of the Research Paper BA

Abstract

Surveys generally have complex ordinal and ranked data as output from their questions. This paper has examined the use of several clustering techniques on these complex responses. Alongside two popular clustering algorithms (K-Means and Spectral clustering) an evolutionary algorithm is introduced. All algorithms were tested on several surveys. Three dissimilarity measures were used to compute the 'distance' between two responses and/or clusters, of which one is an adapted Spearman rank-order coefficient which arguably exhibits more desirable behaviour.

Although none of the techniques could create significantly meaningful clusters, a comparison between the techniques was still possible. It seemed that K-Means even for this complex data is one of the more viable options. Combined with the adapted Spearman rank-order coefficient it performed relatively well in the least amount of time.

Supervisor: Mark Hoogendoorn

Acknowledgments

I would like to express my great appreciation to Mark Hoogendoorn, my research supervisor, for his guidance and constructive suggestions during the planning and development of this research paper. I would also like to thank Rianne Kaptein whose comments and suggestions helped improve and clarify this research paper. My grateful thanks are also extended to Dirk Jonker for giving me the opportunity to write this paper at Focus Orange and use their data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Labeling people is something we all tend to do. In surveys, this often means that respondents are grouped by some chosen attribute in order to observe or investigate certain differences in their responses. However, for businesses it might be valuable to automatically detect groups of like-minded people. In other words: performing unsupervised clustering on the responses while excluding personal (demographic) data. Detecting groups in the data can help businesses to establish the appropriate strategies to satisfy their employees and/or clients.

Often, survey data is ranked data in which respondents have to rank certain items or subjects on a certain scale. Few studies have been carried out on clustering ranked data. Moreover, most of these studies focussed on describing the structure or analysing the distribution of the ranks of all data together instead of assuming disagreement within the population [7]. While some even say that ranked (or ordinal) data is not appropriate for cluster analysis [11], Heiser (2013) reports on some studies which have succeeded to do so [7]. These, and similar studies mostly use complex probabilistic models. However, Heiser (2013) himself implements a generalized K-means method in his study and concludes that "loss-function based methods enjoy general advantages compared to methods based on probability models" [7, p. 27]. Other attempts of using classical cluster methods or loss-function based methods on clustering ranked data are not common.

The aim of this study is to investigate how we can cluster ordinal survey data in

a highly structured ranking. Specifically, we will use popular cluster techniques and, following Heiser's conclusion, a (loss-)function based evolutionary algorithm (EA) to allocate respondents into a pre-determined number of groups. Subsequently we can answer the questions whether these techniques can cluster ordinal survey data in a highly structured ranking *sufficiently* and which technique does this best. What sufficiently exactly means will be covered in the following section. All the same, it should be the case that, depending on the chosen number of clusters, the created groups should be meaningful and distinguishable. The findings should contribute to the analysis of similar future surveys.

Chapter 2

Methods

The survey responses used for clustering originates from surveys Focus Orange (an HR consultancy firm) had done in the past. They designed a survey tool in which respondents can rank certain items in a diamond shape as shown in figure 2-1.

Figure 2-1: Diamond shape in Focus Orange surveys.

The ordinal data itself has many tied ranks and is highly correlated: if one item is ranked highest (i.e. 'most important'), others cannot have this rank anymore. Although the scale unit (i.e. importance), size of the diamond and items are customisable, most conducted surveys were about importance of aspects in collective labour agreements.

To cluster and evaluate the survey responses, the following steps were taken:

1. Find rank correlation coefficients to compare responses (2.1)
2. Compare the rank coefficients in terms of desirable behaviour (2.1.1)
3. Convert the rank coefficients to dissimilarity measures (2.1.2)
4. Investigate how to evaluate different sets of clusters (2.2)
5. Choose and implement clustering algorithms (2.3)

2.1 Rank Correlation Coefficients

A few rank correlation coefficients are available: for non normally distributed data the suggested coefficients are Spearman's rank-order and Kendall's tau rank correlation [2]. Both aformentioned coefficients range from -1 to 1, in which 0 suggests no correlation, -1 indicates that the ranks of a pair of responses are correlated negatively and 1 indicates that these are correlated positively [14].

Spearman's rank-order coefficient

Spearman's rank-order coefficient uses the summed squared difference of item's ranks to calculate the similarity [14]. This summed squared difference is then divided by a term based on the number of total items to make sure it is -1 when this difference is maximised and 1 if the difference is 0. Because it uses the difference of ranks, even the smallest differences in rank are penalised by the Spearman's rank-order coefficient. It handles tied ranks by a correction of the divisor term. On top of that, it also sets values to the mean of the ranks of their positions in the ordered data set [14]. An example of this transformation for tied ranks will be discussed later.

Kendall's tau rank correlation

Kendall's tau rank correlation is based on the total number of discordant pairs of items in their ranking order [1]. Instead of using the exact ranks of items, it compares per item how many concordant and discordant pairs (in terms of ranking) there are and divides these by the total number of possible pairs. So although item's ranks might be different for two responses, if many items are still ranked lower than others in both, the coefficient might still suggest a high correlation. The coefficient handles tied ranks only by a correction of the divisor term.

Spearman's rank-order coefficient with inverted tie-correct

In this section an additional rank coefficient will be defined. The fact that Spearman's rank-order coefficient handles ties by setting values to the mean of the ranks might not suffice for our use-case. We can see this in a small example, presented in figure 2-2, which uses a diamond shape with 5 levels to rank items in.

Figure 2-2: Example diamond shape with ranks

When we correct these ranks for the ties, we get the values in figure 2-3a. As visible, the rank differences between more important items is in this case smaller than the difference between neutrally ranked items in the middle. One could argue that a swap of two items near the middle should not be the cause of a significant difference in coefficient. Therefore, another rank correlation coefficient based on Spearman's rank-order coefficient is defined in which the tied ranks are corrected by using the centrally inverted differences (diffs) of the regular tie-correct mechanism.

(a) Regular tie-correct (b) Inverted tie-correct

Figure 2-3: Example of two kinds of tie-corrected ranks for Spearman's rank-order coefficient

In the example: the regular tie-correct mechanism has [1.5, 2.5, 2.5, 1.5] as diffs between each of the possible ranks. Inverting this both ways from the center gives us [2.5, 1.5, 1.5, 2.5]. If we apply these diffs to our ranks, we get the tie-corrected ranks seen in figure 2-3b. By using this as basis for Spearman's rank-order coefficient, the squared differences of item's ranks will be smaller for changes in top/bottom ranks.

2.1.1 Rank coefficient criteria

Before the introduced coefficients were used in the cluster algorithms, it was valuable to evaluate them by comparing some responses. Particularly of interest was to verify whether the newly introduced Spearman's rank coefficient with inverted tie-correct did what was expected of it. In consultation with Focus Orange a set of rough criteria was defined. The criteria consist of the desirable behaviour of rank coefficients when faced with certain changes of ranks and are displayed in table 2.1. For every rank coefficient it was checked whether they were met sufficiently. As the criteria are rough guidelines, the fact whether the coefficients behaved sufficiently was decided on their relative behaviour. Besides giving insights in their reaction to differences in responses it might also help in confirming what coefficient performs best.

The criteria were checked by comparing a default response to several other responses by using the three rank coefficients. In these comparisons artificial responses for the (most common) 16 slot diamond were used. They can be grouped into four cri-

Table 2.1: Criteria for rank coefficient

Compared type	Criteria: Rank coefficient should
Similar responses	deviate very little
Inverted responses	be negative for fully inverted responses and neutral for centrally inverted responses
Shifted responses	be strongly positive for 1-rank shifts and positive for 2-rank shifts
Swapped responses	deviate more for top/bottom item swaps than middle item swaps

teria categories: similar responses, inverted responses, shifted responses and swapped responses in which a swap between top/bottom items was compared to a swap between middle (neutral) items. Response comparisons to check to what extent the criteria were met can be found in appendix A.

After having tested the three similarity measures, they all seem to have met the critera, except for the swapped responses critera. In the case of swapping top/bottom ranks both Kendall's tau and Spearman's rank coefficient reported a higher correlation compared to swapping middle ranks. Only Spearman's rank with inverted tie-correct had the oppposite behaviour. In the end it seemed that Spearman's rank coefficient with inverted tie-correct never exhibited any unwanted behaviour and sometimes even had more desirable results than the other rank coefficients. Consequently, we assume that this measure generates proper results when using it in the cluster algorithms.

2.1.2 From coefficient to dissimilarity measure

Most cluster algorithms internally have an objective to minimise some error function based on the dissimilarity of elements in a cluster. To be able to use these rank correlation coefficients more easily a transformation to (positive) dissimilarity measures was necessary. This was done by using the linear transformation, suggested by Emond & Mason (2002), of subtracting the rank correlation coefficient value from 1 [6]. After this transformation, these measures could be used to represent the dissimilarities between clusters and responses.

2.2 Evaluation of cluster algorithms

To evaluate the sets of created clusters, it was necessary to find a metric which tells how *strong* the structure is by taking into account the intra- and inter-cluster distances in a logical way. For like-minded respondents to be grouped together it is not sufficient to only minimise the dissimilarities of responses within a cluster. Something that should also be taken into account is that dissimilarities to other (nearby) clusters should be maximised. If not, it could happen that by assigning a response to a slightly 'further' cluster, the cluster the response was previously assigned to becomes much more compact. Rousseeuw (1987) introduced the silhouette coefficient (sc) which is based on the cohesion and separation of each individual element [10] (i.e. response). For a response (i), this can be calculated by subtracting the dissimilarity to the closest cluster (a) from the second-closest cluster's dissimilarity (b) and dividing this by the largest of a and b (see formula 2.1).

$$sc(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2.1}$$

The dissimilarity to a cluster is calculated by taking the average over all dissimilarities to all responses in a cluster. A high value suggests that the response is similar to its own cluster and dissimilar to other clusters. A negative value implies a cluster the response is not assigned to is 'closer' to the response. The silhouette score is the average of all separate silhouette coefficients. Plots including the separate coefficients and the silhouette score were used to easily compare the different cluster algorithm's results.

Although it is not carved in stone, Rousseeuw (1987) presents some guidelines on how to interpret the silhouette score [10, p. 88]. These proposed interpretations are shown in table 2.2.

Table 2.2: Proposed interpretation of silhouette scores according to Rousseeuw (1987)

Silhouette score	Proposed interpretation in terms of cohesion
0.71 - 1.00	Strong structure
0.51 - 0.70	Moderate structure
0.26 - 0.50	Weak structure (needs more research)
≤ 0.25	No structure

2.3 Cluster algorithms

2.3.1 Popular Algorithms

Cluster methods come in many shapes and sizes. Two of the most popular ones are K-means and spectral clustering [3, 12, 13]. Both the spectral clustering and K-means algorithm were implemented by using the scikit-learn library [9] in Python in such a way that they could deal with any dissimilarity measure.

K-means

K-means clusters data by minimising the intra-cluster distances. It does this by continously assigning elements to the closest cluster and updating the cluster centres to the 'average' of all assigned elements. Initially, random cluster centres are created and the algorithm stops either after a certain number of iterations or whenever a certain level of convergence has been reached. There is no guarantee that the algorithm converges to an optimal solution [12] and the solution highly depends on the initial (random) state.

Spectral clustering

Spectral clustering uses a similarity graph of the elements to reformulate the clustering problem [13]. In the graph each element is connected to another element if they are similar enough. Each edge can be weighted by the dissimilarity of the two elements. Then the goal is to find a partition of the graph in such a way that the edges within a cluster have very low weights and the edges between clusters have very high weights. Although spectral clustering is able to avoid local optima, it relies strongly on the

decision of parameters for the neighborhood graph [13].

2.3.2 Evolutionary Algorithms

Many studies have shown that evolutionary algorithms for clustering problems prove to be superior compared to traditional algorithms [4, 12]. On top of that, the central ranking problem, in which finding an *average* ranking for a set of responses is the objective, can also be avoided by constructing an algorithm which has a set of clusters as representation. In other words: even if some clustering techniques create clusters with a strong structure, there are still many ways in which these can be represented as actual response.

Evolutionary algorithms work by creating a population consisting of individuals which can evolve either by mutation or recombination of multiple individuals [5]. In this case, each individual is a 'solution' to our clustering problem. The idea is that new individuals with a random valid solution are created and each generation a certain number of offspring is created by performing small mutations or combining the values of two individuals in a certain way. Each solution can be evaluated and its score (fitness) can decide whether the solution is used in a next generation. The implemented evolutionary algorithm follows the description in table 2.3 and is explained more thouroughly in the next sections.

Table 2.3: Description of the evolutionary algorithm

Representation	(k) Centroid-based permutations
Recombination	Cycle crossover
Mutation	Non-uniform rank shift with adaptive step size
Parent selection	Rank-based selection
Survival selection	$(\mu + \lambda)$ selection
Initialisation	Random

Representation

A set of clusters was the aimed representation. This is also called following a centroid-based (permutation) encoding scheme [4] in which each individual consists of multiple

cluster centers. For each individual in the population a matrix with k rows, one for each cluster, was constructed. Each row consists of a permutation of n possible ranks. In the example of a 16-slot diamond, this would mean that each individual comprises of an $k \times 16$ matrix for k clusters. There was one additional constraint on this representation that is that the number of responses assigned to each cluster should not be too imbalanced. Specifically, the largest cluster should not exceed a size which is three times as large as the smallest cluster.

Parent selection

The selection of parents was done by implementing a linear ranking (LR) selection. This mechanism is based on the rank (by fitness) of the responses in a population. Eiben (2015) states that, compared to a fitness proportionate selection, a rank-based selection "preserves a constant selection pressure" [5, p. 81] which prevents premature convergence and a weak selection pressure in a later stadium.

Recombination

With the assumption that the absolute position of items matters most for our dissimilarity measures (see 2.1.1), a cycle crossover was implemented. This crossover type preserves the absolute position of items from both parents to the best of its ability [5, 8].

Mutation

The mutation step is executed after the crossover step. A non-uniform rank shift mutation with a gaussian distribution was implemented. This mechanism mutates a cluster by randomly picking one item's rank and swapping it with another item's rank shifted by a (rounded) gaussian random sample. Each newly created individual in the previous step had his clusters mutated with probability $p = \frac{1}{k} \sum_{i=1}^{k} \frac{i}{k}$. This probability can be derived by uniformly sampling the number of clusters to mutate $m \in \{1..k\}$ and then choosing m random cluster indices.

The mutation step size was made adaptive, by using Rechenberg's 1/5 success rule, to avoid premature convergence (by having larger mutation) and to exploit promising regions later on (by doing smaller modifications) [5].

Fitness function

As covered in 2.2 the aim was to evaluate the clusters by using the silhouette score. Not surprisingly a similar function to the silhouette score was used as fitness function. To make the algorithm more efficient, the distance from a response (i) to a cluster was calculated not by averaging the dissimilaritiess of all its responses, but by taking the dissimilarity (d) to the real-valued representation of the cluster. Consider a to be the currently assigned cluster and b the 'nearest' other cluster, then the fitness function can be defined as:

$$fitness(i) = \frac{d(b,i) - d(a,i)}{max\{d(a,i),\ d(b,i)\}} \tag{2.2}$$

To check whether this derivation of the silhouette score would still lead to optimal results, 100 individuals were created (for the second-largest survey of 2451 respondents) and compared on their silhouette and fitness scores. This was done by using the pearson correlation coefficient. The corresponding assumption that the data comes from a normal distribution was checked by performing Shapiro-Wilk tests (p-value $> \alpha$ for both scores for a significance level of $a = 0.05$). The result of the pearson coefficient ($r = 0.53$) suggested a moderate positive correlation between the silhouette and fitness scores.

The time complexity of the two possible fitness functions is quite similar (depending on the implementation). However, the silhouette score needs an initialisation step of dissimilarities between all responses which has a complexity of $O(n! * d)$ for n responses and d dimensions (number of slots in the diamond). Considering the positive correlation and a significantly faster benchmark runs, the function as defined in 2.2 seemed to be the appropriate choice.

Survivor selection

The last step of the algorithm follows the phenomenon of 'survival of the fittest': responses that *fit* the environment best are selected for the next generation. The $(\mu + \lambda)$ selection method was used to implement this selection procedure. This works by adding all offspring to the current population and selecting only the best responses based on their fitness for the next generation [5].

Chapter 3

Experiments

3.1 Data

Data used to test the different techniques on is survey data from 6 companies having 113, 149, 350, 493, 2451 and 3426 respondents. All surveys held were using the 16-slot diamond shape. 5 out of 6 were about the importance of aspects for a collective labour agreement and 1 was about the competenence which should be present in a certain workfield.

3.2 Set-up

The data of the available surveys were put in a PostreSQL database. Extraction and analyses of the data was done in Python including extensive use of libraries numpy, scipy and a wide variety of efficiently implemented machine learning techniques in scikit-learn [9]. The parameters used for the evolutionary algorihtm can be found in appendix C.

To compare different kinds of cluster techniques, dissimilarity measures, cluster sizes and surveys, a script was created which would run any kind of a combination of these. Each combination was run three times and the script output the average calculation time, average (silhouette) score and a silhouette coefficient plot for the last run. In total there were 3 (cluster sizes) * 6 (surveys) * 3 (dissimilarity measures)

* 3 (algorithms) = 162 averaged test cases.

As the business requirement usually is to create up to a maximum of a handful of clusters, only benchmarks were run to compare 2, 3 and 4 number of clusters. Although the cohesion of clusters might be better with a larger number of clusters, from the business perspective it was more valuable to invest time on comparing the different techniques on a low number of clusters.

3.3 Results

All results generated seem to lead to clusters with no significant cohesion according to Rousseeuw's proposed interpretations of the silhouette score (see section 2.2). Derived from the 162 test cases, there are findings relating to the dissimilarity measure, clustering technique and calculation time which can be useful to draw conclusions from. All findings in the next sections are aggregated results based on the test case results presented in appendix B.

3.3.1 Dissimilarity measure

In all cases, using Kendall's tau coefficient scored lowest, while in 61% of the 54 cases, using Spearman's rank-order coefficient with inverted tie-correct scored highest. If we only look at the K-Means method cases, then in almost 80% the adapted Spearman's rank-order coefficient scored best.

3.3.2 Clustering techniques

Of the three techniques, the K-means method scored highest in half of the cases while the spectral clustering was usually surpassed by a substantial difference by both the evolutionary algorithm and K-means. K-means also seems to produce the best results in terms of number of wrongly assigned responses. An example of this is presented in figure 3-1. This was one of the test cases on the largest dataset, using the Spearman rank-order coefficient with inverted tie-correct. In this figure,

for every created cluster (i.e. 4) the silhouette coefficient (see 2.2) per response is plotted descendingly: the longer the line ranges to the right, the higher the silhouette coefficient of that particular response is. The red dotted line displays the average score of all these coefficients. Compared to the spectral (figure 3-2) and evolutionary clustering (figure 3-3) the K-means has little negative silhouette coefficients.

Not only in cases in which the K-means scored best (highest silhouette score) was the number of negative silhouette coefficients low, but also in some test cases where other techniques scored better. In those cases, the other algorithms made the 'sacrifice' of having a cluster with some negative silhouette coefficients in order to reach a higher (average) silhouette score.

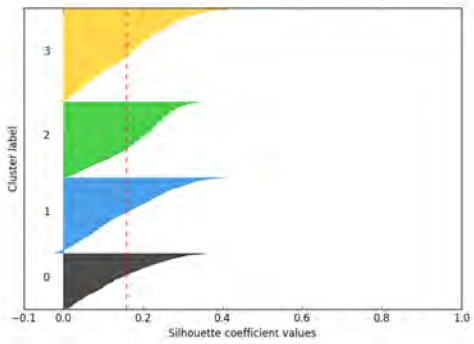

Figure 3-1: Silhouette coefficients for K-means clustering on largest dataset.

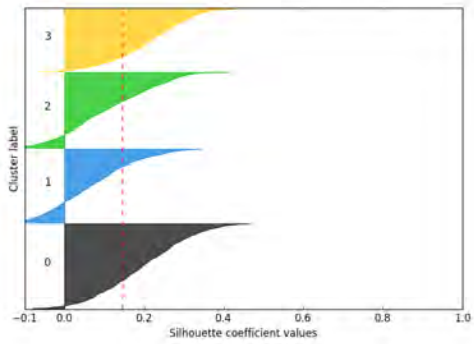

Figure 3-2: Silhouette coefficients for Spectral clustering on largest dataset.

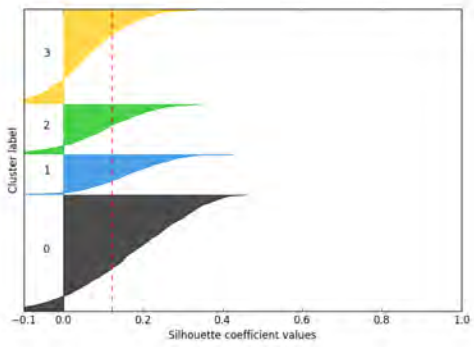

Figure 3-3: Silhouette coefficients for Evolutionary clustering on largest dataset.

Another observation is that the evolutionary algorithm often presented solutions in which the cluster sizes were not as equal as the other techniques. It seems that the

algorithm often converged to solutions near the constraint boundary that cluster sizes for a solution should not be too imbalanced. An example of this difference can be found in one of the better results obtained with the Spearman rank-order coefficient: in figure 3-4 it seems the cluster sizes are not balanced, while the K-Means's output (figure 3-5) is a lot more balanced.

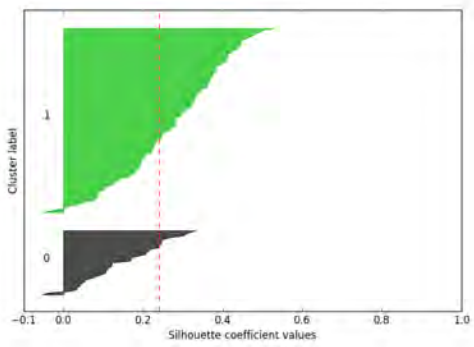

Figure 3-4: Silhouette coefficients for Evolutionary clustering on dataset about competences (149 respondents).

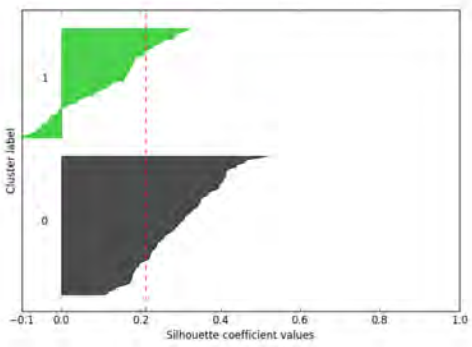

Figure 3-5: Silhouette coefficients for K-Means clustering on dataset about competences (149 respondents).

3.3.3 Calculation time

The test cases of the benchmark were run on multiple computers, so exact calculation time varies. Therefore a relative comparison between cluster techniques and dissimilarity measures is more appropriate. All computation times were averaged by the number of responses of the survey. In the case of creating 2 clusters, if the K-means method's average duration is 1, Spectral clustering is on average 2.5 times as slow and the evolutionary algorithm's duration was on average 20 times longer. For smaller surveys, the difference between the evolutionary algorithm and the others becomes even more. In practice this meant that it could still take more than an hour to cluster a small survey!

It seems that Kendall's tau coefficient calculation takes a lot more time than the others. However, Spearman's rank-order coefficient (with and without inverted tie-correct) was implemented by ourselves in an efficient way specifically for the provided data. For Kendall's tau coefficient an existing implementation in the scipy Python

library was used. This implementation does a lot more than just calculating the coefficient including validations and transformations of the data.

Chapter 4

Discussion

The main goal was to examine whether it is possible to create meaningful clusters of ordinal survey data in a highly structured ranking, and to compare several techniques for this. A quick comparison to randomly generated clusters does indicate that some of the used techniques add some value. However, all of the results suggest that clusters without significant structure were created, confirming the claim that this kind of data is unsuitable for cluster analysis [11]. On the contrary, it could also be the case that the conducted surveys do not have any distinguishable groups in them, or that more meaningful clusters could be created with larger, for the business less useful, k.

Regardless of the exact silhouette scores on all data, there were some differences in results for the used cluster techniques and dissimilarity measures. Overall, we can say that our custom Spearman's rank-order coefficient with inverted tie-correct performs best for this specific data. Also taken into account the computation time, it seems that K-Means relatively does the best job. The combination of these two lead to the best results.

It should be noted that the evolutionary algorithm was not optimised to a large extent. Future research should at least consider optimising the parameters of the evolutionary algorithm more for the specific data. Moreover, the choice of using certain selection and mutation/recombination mechanisms was based on theory and logical reasoning, while a more test-driven approach could lead to other choices. Finally, there were some aspects of the evolutionary algorithm which can be processed in

parallel. The gained speedup might make it feasible to use the real silhouette score as fitness function and/or to simulate more generations. After future investigation of aformentioned improvements it might be the case that the evolutionary algorithm will outperform a K-Means method. Nevertheless, it will probably never be as fast as K-Means.

This study is in a way unique, because we had to work with very specific ordinal survey data. A direct comparison to other studies is therefore not possible, but we hope that this study provides new insights into the use of certain cluster techniques and (newly introduced) dissimilarity measures on the provided data. Future studies could consider using additional artificially created data with the help of experts on the subject of the survey to evaluate the techniques more extensively. Finally, it might be possible to create more meaningful clusters when some elements are left out. E.g. some aspect of a collective labour agreement might not be relevant for every respondent, so excluding it from the calculation might improve the results. However, this would require specific analyses for each survey.

Appendix A

Rank coefficient criteria results

The responses of the four categories in the next sections were at least compared on the three rank coefficients (r) to a default assignment of ranks:

$$A := [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 7]$$

A.1 Similar responses

Consider the following two, fairly similar to A, assignments of ranks to items:

$$B_{si} := [1, 2, 2, 3, 3, \mathbf{4}, \mathbf{3}, 4, 4, 4, 5, 5, 5, 6, 6, 7]$$
$$C_{si} := [1, 2, 2, \mathbf{4}, \mathbf{4}, \mathbf{4}, \mathbf{4}, \mathbf{3}, \mathbf{3}, \mathbf{3}, 5, 5, 5, 6, 6, 7]$$

Table A.1: Rank coefficients for fairly similar responses

	Kendall tau	Spearman rank	Spearman rank inverted tie-correct
$r(A, A)$	1.0	1.0	1.0
$r(A, B_{si})$	0.9340	0.9629	0.9932
$r(A, C_{si})$	0.8019	0.8886	0.9795

In table A.1 two interesting observations are presented. Firstly, all measures have a coefficient of 1 when comparing the same response. Secondly, all rank coefficients mostly deviate little when comparing responses with some small changes of ranks.

A.2 Inverted responses

Consider the following two, (centrally) inverted to A, assignments of ranks to items:

$$B_i := [7, 6, 6, 5, 5, 5, 4, 4, \mathbf{4}, \mathbf{4}, \mathbf{3}, \mathbf{3}, \mathbf{3}, \mathbf{2}, \mathbf{2}, \mathbf{1}]$$
$$C_i := [\mathbf{4}, \mathbf{4}, \mathbf{3}, \mathbf{3}, \mathbf{3}, \mathbf{2}, \mathbf{2}, \mathbf{1}, 7, 6, 6, 5, 5, 5, 4, 4]$$

Table A.2: Rank coefficients for (centrally) inverted responses

	Kendall tau	Spearman rank	Spearman rank inverted tie-correct
$r(A, B_i)$	-1.0	-1.0	-0.1515
$r(A, C_i)$	0.1698	0.4030	0.5242

The first row in table A.2 shows that the Spearman rank coefficient with inverted tie-correct is the only one which does not result in a (negative) correlation of -1. This means the domain for this rank coefficient is smaller. The second row shows that the Spearman rank coefficient *relatively* shows the highest similarity. On the contrary, the Kendall tau and Spearman rank coefficient with inverted tie-correct are relatively closer towards their central value (suggesting no correlation).

A.3 Shifted responses

Consider the following two assignments of ranks to items in which every item's rank is shifted, either lower or higher, by 1 (for B_{sh}) and 2 (for C_{sh}) compared to A:

$$B_{sh} := [2, 1, 3, 2, 4, 4, 3, 3, 5, 5, 4, 4, 6, 5, 7, 6]$$
$$C_{sh} := [3, 4, 4, 1, 5, 5, 2, 2, 6, 6, 3, 3, 7, 4, 4, 5]$$

Table A.3: Rank coefficients r for shifted responses

	Kendall tau	Spearman rank	Spearman rank inverted tie-correct
$r(A, B_{sh})$	0.6792	0.8000	0.8606
$r(A, C_{sh})$	0.2075	0.1697	0.5333

As visible in table A.3, shifting all ranks seems to result in a very similar behaviour for all rank coefficients. For a rank shift of 1, the correlation is still strong, although the Kendall tau reports the lowest correlation. The 2 ranks shift is causing all correlations to be weak, but still positive.

A.4 Swap top/bottom vs middle ranks

Consider the following two assignments of ranks to some items:

$$B_{sw} := [\mathbf{2}, \mathbf{1}, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6, \mathbf{7}, \mathbf{6}]$$
$$C_{sw} := [1, 2, 2, 3, 3, \mathbf{4}, \mathbf{3}, 4, 4, \mathbf{5}, \mathbf{4}, 5, 5, 6, 6, 7]$$

Table A.4: Rank coefficients r for responses with swapped ranks

	Kendall tau	Spearman rank	Spearman rank inverted tie-correct
$r(A, B_{sw})$	0.9434	0.9864	0.9258
$r(A, C_{sw})$	0.8774	0.9258	0.9864

As seen before, the results in table A.4 show again that the all rank coefficient deviate little when swapping middle ranks. Another interesting observation is that when swapping the top/bottom ranks, the Spearman rank coefficient with inverted tie-correct decreases, while the Kendall tau and Spearman rank coefficient are reporting a stronger correlation compared to swapping middle ranks.

Appendix B

Benchmark results

B.1 Results for survey with 3426 respondents

Table B.1: Average silhouette scores for 2 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1638	0.1588	0.1154
Spectral	0.1547	0.1536	0.1205
Evolutionary	0.1873	0.1787	0.1440

Table B.2: Average silhouette scores for 3 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1316	0.1394	0.0919
Spectral	0.1214	0.0998	0.0885
Evolutionary	0.1254	0.1403	0.0875

Table B.3: Average silhouette scores for 4 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1186	0.1273	0.0839
Spectral	0.1038	0.0920	0.0734
Evolutionary	0.1185	0.1119	0.0904

Table B.4: Average runtime (in seconds) for 2 clusters on machine 1

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	765	659	2241
Spectral	1682	1684	5964
Evolutionary	16035	19090	24366

B.2 Results for survey with 2451 respondents

Table B.5: Average silhouette scores for 2 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1894	0.2024	0.1392
Spectral	0.1843	0.1992	0.1408
Evolutionary	0.1923	0.1852	0.1545

Table B.6: Average silhouette scores for 3 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1487	0.1834	0.1009
Spectral	0.1392	0.1730	0.1028
Evolutionary	0.1570	0.1481	0.1168

Table B.7: Average silhouette scores for 4 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1378	0.1574	0.0795
Spectral	0.1146	0.1460	0.0833
Evolutionary	0.1180	0.1215	0.0902

Table B.8: Average runtime (in seconds) for 2 clusters on machine 2

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	351	327	1260
Spectral	765	765	3044
Evolutionary	15796	16390	17403

B.3 Results for survey with 493 respondents

Table B.9: Average silhouette scores for 2 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1950	0.2065	0.1439
Spectral	0.1674	0.1724	0.1212
Evolutionary	0.2311	0.2299	0.1824

Table B.10: Average silhouette scores for 3 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1416	0.1821	0.0705
Spectral	0.1094	0.1275	0.0694
Evolutionary	0.1697	0.1789	0.1214

Table B.11: Average silhouette scores for 4 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1280	0.1472	0.0794
Spectral	0.0831	0.1207	0.0518
Evolutionary	0.1293	0.1429	0.1029

Table B.12: Average runtime (in seconds) for 2 clusters on machine 3

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	30	26	76
Spectral	52	62	160
Evolutionary	10328	10395	11606

B.4 Results for survey with 350 respondents

Table B.13: Average silhouette scores for 2 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1883	0.2014	0.1327
Spectral	0.1805	0.1865	0.1375
Evolutionary	0.2048	0.1947	0.1568

Table B.14: Average silhouette scores for 3 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1781	0.1717	0.1229
Spectral	0.1452	0.1510	0.1115
Evolutionary	0.1271	0.1648	0.1070

Table B.15: Average silhouette scores for 4 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1572	0.1542	0.1031
Spectral	0.1327	0.1366	0.1075
Evolutionary	0.1247	0.1306	0.0965

Table B.16: Average runtime (in seconds) for 2 clusters on machine 3

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	28	27	60
Spectral	39	41	88
Evolutionary	10214	7002	7314

B.5 Results for survey with 149 respondents

Table B.17: Average silhouette scores for 2 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.2128	0.2204	0.1630
Spectral	0.1816	0.1711	0.1411
Evolutionary	0.2406	0.2256	0.1843

Table B.18: Average silhouette scores for 3 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1579	0.1621	0.1135
Spectral	0.1403	0.1374	0.1040
Evolutionary	0.1569	0.1561	0.1428

Table B.19: Average silhouette scores for 4 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1607	0.1397	0.0998
Spectral	0.1242	0.1066	0.0908
Evolutionary	0.1370	0.1241	0.1036

Table B.20: Average runtime (in seconds) for 2 clusters on machine 3

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	4	5	12
Spectral	6	41	10
Evolutionary	9130	9050	10170

B.6 Results for survey with 113 respondents

Table B.21: Average silhouette scores for 2 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1906	0.1977	0.1341
Spectral	0.1640	0.1843	0.1283
Evolutionary	0.1739	0.1743	0.1293

Table B.22: Average silhouette scores for 3 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1724	0.1785	0.1161
Spectral	0.1741	0.1704	0.1313
Evolutionary	0.1612	0.1767	0.1207

Table B.23: Average silhouette scores for 4 clusters

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	0.1699	0.1885	0.1209
Spectral	0.1636	0.1772	0.1147
Evolutionary	0.1626	0.1600	0.1023

Table B.24: Average runtime (in seconds) for 2 clusters on machine 3

Cluster technique	Spearman rank	Spearman rank inverted tie-correct	Kendall tau
K-Means	3	3	8
Spectral	2	2	6
Evolutionary	6591	6730	5781

Appendix C

Evolutionary Algorithm parameters

The settings displayed in table C.1 were used in the Evolutionary Algorithm as described in section 2.3.2.

Table C.1: Parameters used in Evolutionary Algorithm

#Evaluations	5000
Population size (μ)	100
#Offspring per generation	60
Max. cluster size difference	3
Min. cluster dissimilarity within individual	0.4
Initial mutation step size	2.5
Min. mutation step size	0.1
Max. mutation step size	7
Mutation step size adjustment rate	0.8
Selection probability s (linear ranking selection)	0.5

Bibliography

[1] H. Abdi. Kendall rank correlation. In Neil Salkind, editor, *Encyclopedia of Measurement and Statistics*, pages 508–510. SAGE, Thousand Oaks (CA), 2007.

[2] Nian Shong Chok. PEARSON'S VERSUS SPEARMAN'S AND KENDALL'S CORRELATION COEFFICIENTS FOR CONTINUOUS DATA. Master thesis, University of Pittsburgh, Graduate School of Public Health, 2008.

[3] Chunhui Zhu, Fang Wen, Jian Sun. A Rank-Order Distance based Clustering Algorithm for Face Tagging. In *Computer Vision and Pattern Recognition (CVPR)*, pages 481–488, Providence (RI), June 2011. IEEE.

[4] Eduardo R. Hruschka and Ricardo J. G. B. Campello and Alex A. Freitas and André C. P. L. F. de Varvalho. A Survey of Evolutionary Algorithms for Clustering. *IEEE Transaction on System, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155, 2009.

[5] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*, volume 2 of *Natural Computing Series*. Springer, second edition, 2015.

[6] Edward J. Emond and David W. Mason. A New Rank Correlation Coefficient with Application to the Consensus Ranking Problem. *Journal of Multi-Criteria Decision Analysis*, 11:17–28, 2002.

[7] Willem J. Heiser and Antonio D'Ambrosio. Clustering and Prediction of Rankings Within a Kemeny Distance Framework. In Berthold Lausen, Dirk van den Poel, and Alfred Ultrsch, editors, *Algorithms from and for Nature and Life*, pages 19–31. Springer, 2013.

[8] Alberto Moraglio and Ricardo Poli. Geometric Crossover for the Permutation Representation. *Journal of Ambient Intelligence and Smart Environments*, (1), 2011.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Peter J. Rousseeuw. Silhouettes: a graphical aid to interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[11] SAS Institute Inc., Cary, North Carolina. *SAS/STAT® 9.22 User's Guide*, 2010.

[12] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition*, 33:1455–1465, 2000.

[13] Ulrike von Luxburg. A Tutorial on Spectral Clustering. Technical Report TR-149, Max Planck Institue for Biological Cybernetics, March 2007.

[14] Jerrold H. Zar. Spearman Rank Correlation. In *Encyclopedia of Biostatistics*. John Wiley & Sons, Ltd, 2005.