Universiteit Leiden
Opleiding Informatica

Workforce Survey Data Analysis:

Potentials and Pitfalls

Name: Timothy C. Visser

Date: 08/08/2016

Supervisor: Bas van Stein
2nd reader: Aske Plaat

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

List of Figures

Abstract

When companies go through large-scale changes it is important for them to assess the effects and perception of said changes. One way by which one can measure these effects is surveying employees and collecting information about their perception of a change directly from the source. This will provide us with a large collection of data which we can use for the purpose of better understanding the way the change is being adopted and allowing this understanding to influence decision-making in a meaningful way.

We apply clustering using several clustering algorithms to multiple data sets. Analysis of detected clusters yielded a highly positive and moderately negative cluster. Additionally, we gained valuable insight into the concept of employees shifting between clusters from one survey round to another.

1 Introduction

Modern companies tend to have a necessity of being maintained like a well-oiled machine in order to keep up with day to day operations. This oftentimes entails the implementation of changes on a large scale that affect the entirety of the employees such as a company-wide workflow adaptation. These employees will undoubtedly have an opinion about the way they experience changes and may present themselves as a valuable source of insight into the perception of changes. Requesting employees to take part in surveys about the way a change affects them may help in gathering useful information about the adoption of a change.

What if we could deduce even more valuable insights from survey results like these by approaching the problem in a smarter way? If so, we may find results embedded in our available data that might help us directly influence the successful adoption of a company-wide change. This could be advantageous with regard to picking up on potential problem areas or opportunities that haven't been used to their fullest extent. Perhaps it is even within the realm of possibility to extrapolate findings from early results. This is essentially what we will be trying to find out (chosen as this facilitates later plotting).

Data sets with an individual sample size of between 50 and 70 employees from multiple companies where surveys as described above were carried out will be analysed. For this purpose we will apply several clustering algorithms and verification methods. Additionally the application and impact of anomaly detection will be assessed and multiple ways of visualising our medium/high-dimensional data will be explored.

Essentially, the question that will be answered is as follows: "How much and what kind of useful information can we deduce from the automated analysis of employee survey data as to positively influence decision-making?"

2 Related work

The application of cluster analysis to Likert scale data is largely commonplace and is often used for the purpose of identifying segments within survey response data. An example of this is the factor analysis clustering applied to survey data acquired from the Swiss service sector by Hollenstein [1]. Rice & Slaney [2] used cluster analysis to identify adaptive and maladaptive perfectionists and nonperfectionists.

As the current research will be focusing on the application of cluster detection to Likert scale data it is important to take the peculiarities of this field of research into account. Jain [3] went into some of this with regard to K-means, e.g. the pitfalls concerning clusters of arbitrary shape. The implication that various

algorithms and approaches tend to differ in efficacy leads to the fact that the current research will explore multiple algorithms and compare their merits.

3 Research design

This chapter will outline the considerations that went into the current research's preparation and design. Several of the main building blocks of the eventual implementation will be discussed here.

3.1 Available data and goals

Available survey data primarily consists of the answers to a series of questions answered on a Likert scale from 1 (strongly disagree) to 5 (strongly agree). Additionally each series of answers is accompanied by a unique user ID. Although available data differs somewhat between data sets, there is a base set of 15 questions divided up into three categories (must, want, can). This results in a 15-dimensional data set available for analysis. As this will allow for the greatest number of comparable samples, the primary focus will lie on these specific questions. For each of these sets, multiple survey rounds are carried out at different points in time and as such at different points in the process of change adoption.

Broadly speaking, the goal is to analyse the available data in order to find valuable patterns within. All survey data relates to a specific change within a company and questions focus on how employees within that company are influenced by said change. Ultimately, we would like to be able to use this data as effectively as possible. This entails a decision making oriented approach. We would like to be able to assist in the interpretation and extrapolation of available data. This includes but is not limited to finding useful patterns and generalizations and even the prediction of trends based on early surveys. The ability to draw conclusions from a first survey would be tremendously useful as a way of influencing the successful adoption of a company-wide change. Potential pitfalls could be avoided by quickly getting the right information from the right people by casting a spotlight on employees of particular interest.

3.2 Clustering

The questions we are interested in answering are particularly suitable for an unsupervised learning approach (clustering specifically) as we do not yet know what we're looking for exactly. This makes it unfeasible for us to characterize data points. Additionally, sample size is quite limited. This will negatively influence the initial reliability of any supervised learning approaches.

Multiple significantly different methods of clustering will be explored to verify which of these methods manages to deal well with the problem at hand. In order to test the quality of resulting clusters we will use a way of calculating

a silhouette coefficient which we can then use to assess the merits of a certain clustering. Outliers will be detected using the LOF algorithm.

3.2.1 K-means

The input for *K-means* is a set of data points $x_1 \dots x_n$ of an arbitrary dimension. The user selects a number k that will decide how many centroids $c_1 \dots c_k$ will initially be placed at the start of the algorithm. Then, iteratively, for each point the nearest centroid to that point is determined by computing the distance between said point and each cluster. The point is subsequently assigned to the nearest cluster.

Afterwards, we run over each cluster and for each centroid, we re-centre its position. To accomplish this we take all vectors currently assigned to its respective cluster and take the average of all these vectors. The resulting vector is the new location of our centroid. This is achieved by means of the following formula:

$$c_i = (1/k_i) \sum_{j=1}^{k_i} x_i$$

Here C is defined as the set of centroids $c_1 \dots c_k$ and X as the set of data points $x_1 \dots x_k$. k_i represents the total number of data points in cluster i.

Theoretically, the algorithm ends when the solution converges. However, practically speaking it is more feasible to set a finite number of iterations beforehand at which point the execution of the algorithm is halted and results are presented.

K-means is a relatively simple and straightforward algorithm that deals well with large volumes of data due to its relatively low complexity. It is also reasonably versatile and works on numerical data of arbitrary dimensionality.

We also explore MiniBatchKMeans, which converges more quickly than K-means.

3.2.2 Affinity Propagation

This clustering method as described by Redmond et al. [4] and Frey et al. [5] works on the basis of a similarity function and two matrices. These two matrices contain 'availability' and 'responsibility' data. The algorithm functions by considering data points as exemplars for other data points. Messages are iteratively passed between data to aid in the search of these exemplars until a final clustering presents itself. This allows for the discovery of clusters without the definition of a pre-specified parameter to represent the desired number of clusters.

3.2.3 Spectral Clustering

As described by Xing et al. [6], for this method to be used a similarity matrix and a desired number of k clusters need to be provided. Eigenvectors are then used

to reduce the dimensionality of the matrix. Subsequently, K-means is applied to the reduced matrix in $\mathbb{R}^k$. This method can be advantageous when applied to data of high dimensionality and is oftentimes applied in image processing due to the application of eigenvalues. The fact that spectral clustering is based upon the concept of connectivity and not that of proximity means it can be applicable in situations where proximity based algorithms like K-means tend to fail.

3.2.4 Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (*DBSCAN*) [7], as the name suggests, is a density based clustering algorithm. Contrary to K-means it determines the number of clusters on availability of highly dense data points in an area. The neighbourhood in which the algorithm looks for data points to cluster together is one of the parameters that can be influenced by the user.

This algorithm is especially well-equipped for dealing with outliers as it does not assign every data point to a cluster. The user is also able to state a minimal number of required points for a dense region to be detected. If this number of points in a sufficiently small area is not found the points in that area are disregarded in the resulting clustering. This leads to the fact that DBSCAN can be exceptionally useful for implementation on noisy data sets. Due to its nature it is also strong at finding any number of clusters of an arbitrary shape.

3.2.5 Silhouette Coefficient

Once we have the resulting clustering from the implementation and usage of an unsupervised learning algorithm, it would be useful for us to analyse the quality of the detected clusters. As we do not have the ability to test our results against the "true" situation, our options are somewhat limited in this regard. The silhouette coefficient method [8] is however a good method to aid us in this task.
Essentially, the silhouette coefficient allows us to measure the degree to which clusters are well-defined and contrast against other clusters. This is based on internal cohesion inside a cluster and the way in which clusters are separated from one another.
The way in which this is achieved (and as it is done by scikit-learn [9]) is fairly simple. In order to calculate the silhouette coefficient we need two things for each sample:

- Mean intra-cluster distance (a)
- Mean nearest-cluster distance (b)

Once we have these, we can calculate the silhouette coefficient as follows:

$$\frac{(b-a)}{max(a,b)}$$

At this point we can take the mean silhouette coefficient for all samples and determine how well-defined the resulting clustering is. Additionally, we can use the silhouette coefficient of individual samples for a multitude of purposes depending on how we apply them. This will be further elaborated upon later.

3.3 Local Outlier Factor

For the purpose of outlier detection we apply the *Local Outlier Factor (LOF)* algorithm [10]. This algorithm allows for the discovery of anomalous data points. More specifically it does this by assigning a value to each data point representing the degree to which said point is considered an outlier. The algorithm takes local density into account, meaning outliers can be identified more accurately in regions where they otherwise would not. A data point located at a relatively small distance from a very dense cluster may be an outlier while this same point in a region with less density might not have been an outlier.

3.4 Visualisation

For the purpose of effectively displaying results we need a method to plot our high-dimensional data points on a 2D canvas. Several different methods are used to achieve this.

3.4.1 Conventional plotting

The most conventional method we used for plotting our results was by means of a 2D scatter plot. The horizontal axis represents the different questions that make up the survey. On the vertical axis, we find an employee's response on a Likert scale ranging from 1 to 5. Using this method, we can plot each occurring combination of a question and a response on this 2D grid. In order to effectively display the frequency of each of these combinations, the size of a point on the scatter plot is defined in relation to its frequency. This allows us to quickly gauge response patterns within clusters.

3.4.2 Andrews Curves

For the purpose of cluster verification we applied the concept of Andrews curves [11]. We can use Andrews curves to represent multivariate data by generating a line for each data point in our set. Said curves are generated using the concept of Fourier series. We can plot each of the resulting curves together in order to form an Andrews plot. Although this kind of plot requires some interpretation, we can use it to verify the existence of well-defined clusters.

3.4.3 RadViz

Our final visualisation method, RadViz [12], facilitates the plotting of individual multivariate data points on a conveniently interpretable 2D plot. Essentially

we evenly spread out our numbered questions on a circle. Each question is represented by a point on this circle. Imagine all of these points having a certain attraction on data points plotted into the circle. The higher the response to a specific question, the higher its attraction on the respective data point. This allows us to plot all of our data points within the circle and interpret their position. Although this method allows for the generation of very intuitive plots, its outcome is also highly dependent on the distribution of questions on the circle. This has implications for its usefulness, especially on multivariate data with a very high degree of dimensionality.

4 Implementation

During the implementation phase several choices and considerations have been made. This chapter will outline and elaborate upon these.

4.1 Preprocessing

Due to the fact that we use two separate data sets from different points in time it is possible for each data set to contain responses from employees who did not participate in the other survey. We account for this by filtering both data sets and disregarding any employees whose responses are not present in either data set. To do this, we build one list of employee ID's for each data set which we subsequently take the intersection of. Once we have a list of employees in both data sets we scrape each data set for responses from said employees and store these in two Numpy arrays, one for each data set.

As employee IDs are not an interesting metric to base our clustering upon we do not store these for usage during our clustering phase. In order to make sure we can re-identify employees, we keep a list of employee IDs available for retrieval at a later point in time.

4.2 Clustering

For clustering purposes we primarily make use of Python's *scikit-learn* and *SciPy* libraries. More specifically we will be using the K-means implementation provided by SciPy, and the DBSCAN, MiniBatchKMeans, Spectral Clustering and Affinity Propagation implementations provided by scikit-learn. All of these are fairly straightforward in their usage and easily applied to our data sets.

In the case of K-means, we apply the **kmeans2** method to our first data set and specify the desired number of clusters to determine cluster centroids. Subsequently we assign all samples to one of these centroids using SciPy's **vq** method.

DBSCAN is implemented using scitkit-learn's **dbscan** method. Once again we pass our first data set as an argument, but in this case we also pass an ϵ value

for DBSCAN to operate on. Assigned cluster labels are then gathered from the resulting output.

MiniBatchKmeans, Spectral Clustering and Affinity Propagation are implemented using the methods scikit-learn provides for this with mostly default parameters.

4.3 Cluster quality analysis

We need a way to assess the quality of the clustering resulting from our selected algorithm. The approach we use for this is the silhouette coefficient which allows us to determine how well-defined and separated from other clusters in an individual cluster is by assessing the proximity of each data point to its assigned cluster and other clusters. This approach is easily applied to our implementation of K-means.

We use the **silhouette_samples** method from scikit-learn to calculate the silhouette coefficient for each individual sample in our clustering. Additionally we use the **silhouette_score** method to determine the silhouette coefficient for the entire clustering, essentially taking the mean silhouette coefficient of all individual samples. A higher silhouette coefficient tells us that a clustering is very likely of higher quality than one with a lower silhouette coefficient. Silhouette scores range from -1 to 1 (higher implies a better and more meaningful clustering). This will primarily be used for quality analysis, but we will also apply some filtering based on the silhouette coefficient.

As an additional means of verifying our results we will make use of a *silhouette plot* as implemented by Amro [13]. In a silhouette plot, each sample is represented by a horizontal line of a length that is proportional to its silhouette coefficient. The line corresponding with every individual sample is plotted, grouped by assigned cluster. This allows us to visually interpret the quality of our clustering and gives us the ability to easily notice samples with an exceptional silhouette coefficient.

4.4 Shifter detection

Using our initial clustering based upon the first survey we are able to assign all samples from the second survey to an existing cluster as well. Given our ability to follow employees between the first and second survey, we can then determine whether employees are assigned to the same cluster for both survey rounds or not. This will give us valuable insight into employee response patterns and which employees are likely to switch between clusters. What exactly this means will depend on the characterisation of the clusters that emerge.

When an employee switches between clusters from one survey round to the other, we will define this employee as a *shifter*. Employees who don't switch between

clusters are defined as *non-shifters*.

A problem that arises with the detection of shifters as described above is a potential lack of significance. Instances may occur where an employee's responses change very minimally, yet still enough to switch clusters due to being on the outer edge of a cluster and close to another one. To eliminate this problem we will also apply the silhouette coefficient method mentioned earlier to disregard samples with an exceptionally low silhouette coefficient when detecting shifters.

4.5 Outlier detection

The **pylof** [14] implementation of LOF by D. Kužnar will be used to detect outliers. This implementation allows us the use of its readily available outlier detection functionality which we will apply to calculate the local outlier factor of potential outliers. We will consider these points outliers when the local outlier factor crosses a certain threshold.

5 Results

This chapter covers the established results of the current research. We used two separate data sets for the purpose of gathering our results. These will be individually referred to as set 1 and set 2.

5.1 Clustering

In this section, the construction of clusters and verification thereof will be discussed. Several methods of visualisation are used for this purpose.

5.1.1 Algorithm selection

For the purpose of algorithm selection several well-known and extensively studied clustering algorithms were compared, namely K-means, MiniBatchKmeans, Affinity Propagation, Spectral Clustering and DBSCAN. In order to objectively compare these algorithms we applied them to our data sets and used the silhouette coefficient method where applicable to measure the resulting cluster quality. In order to allow for optimal results, we varied some of the input parameters to the selected algorithms In the case of K-means we varied the number of requested clusters. For DBSCAN the supplied ϵ value was varied and two different distance metrics (Euclidean and Cityblock) were used.

DBSCAN did not perform very well on our data. For most supplied ϵ values, DBSCAN only constructed a single cluster. Although results varied slightly depending on the distance metric used, the situations in which DBSCAN identified more than 1 cluster were scarce and seemingly arbitrary.

Iterating over several options for number of clusters as a parameter of K-means, the following table clearly demonstrates that we get an optimal result for two clusters based on the silhouette coefficient for each of these iterations.

Additionally Affinity Propagation, Spectral Clustering and MiniBatchKmeans were compared. Resulting clusters from these were all very comparable as can be seen below. Mean Shift was also considered in this phase, but failed to provide more than a single cluster.

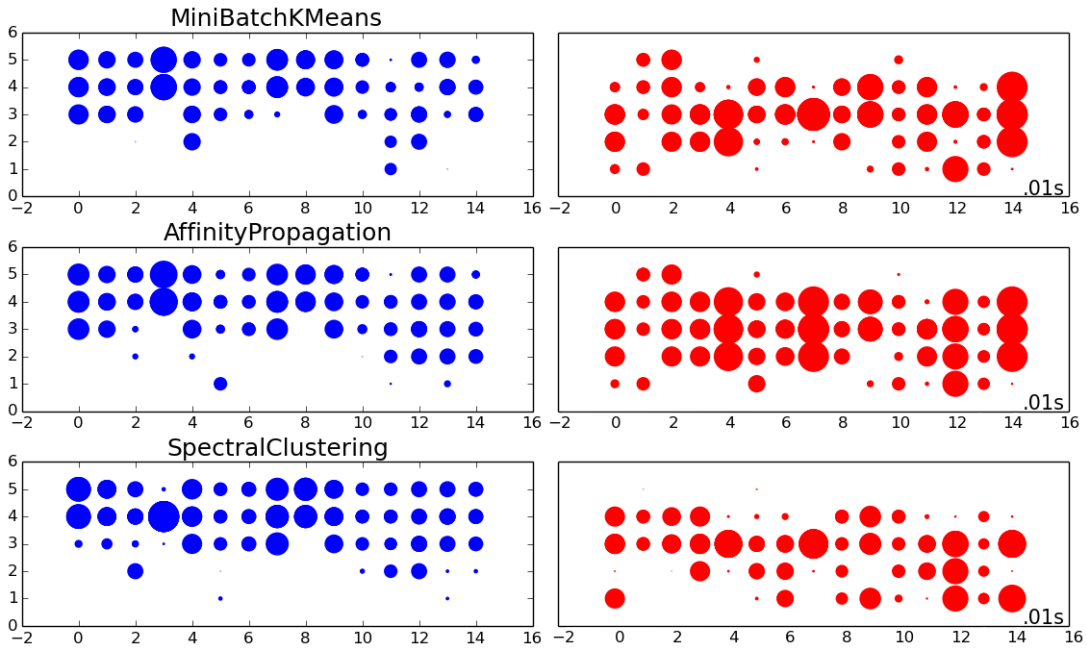

Figure 1: Clusters detected by several different clustering algorithms

Due to the fact that no significant differences seem to be present in the characterisation of resulting clusters (with the exception of Affinity Propagation resulting in slightly more overlap between clusters), we will choose to use K-means for the remainder of our analysis in the favour of the other algorithms that have been discussed. This is based on the fact that K-means is the least complicated of these options and the other algorithms do not seem to provide significantly better results overall.

5.1.2 Resulting clusters

Applying K-means to our data with a supplied number of clusters of two results in the formation of two clusters. Averaging responses from all samples in each of these clusters tells us that one of these clusters has a significantly higher average response than the other cluster. The figures below further confirm this assumption.

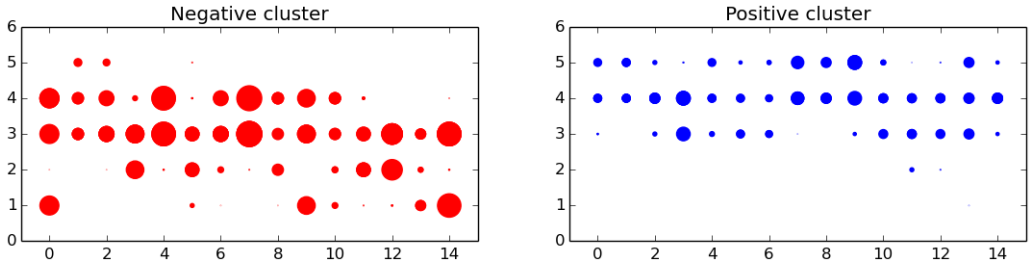

Figure 2: Detected clusters (Data set 1)

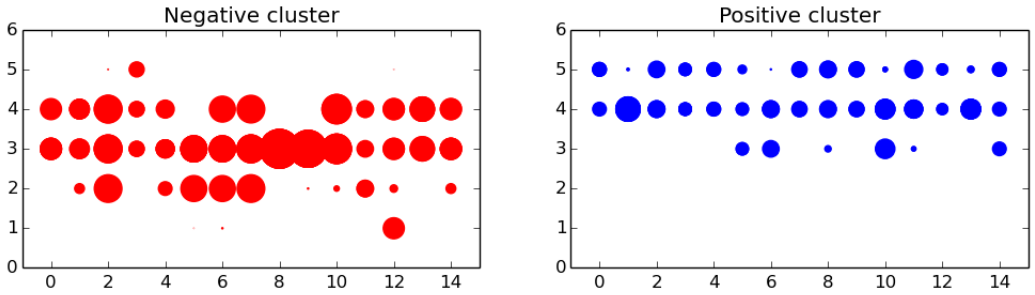

Figure 3: Detected clusters (Data set 2)

From the above figures we can conclude that one of the clusters can be characterised as highly positive whilst the other cluster is moderately negative. The results based upon our data sets seem to be very comparable, suggesting that our results might be generalizable. However, we can't draw this conclusion with certainty due to our limited sample size.

5.2 Cluster validation

We will be using several methods to validate the quality of our resulting clusters. All of these will help us to determine the significance of our clustering.

5.2.1 Silhouette Plots

The silhouette plots based on our clustered data from the first data set is displayed below.

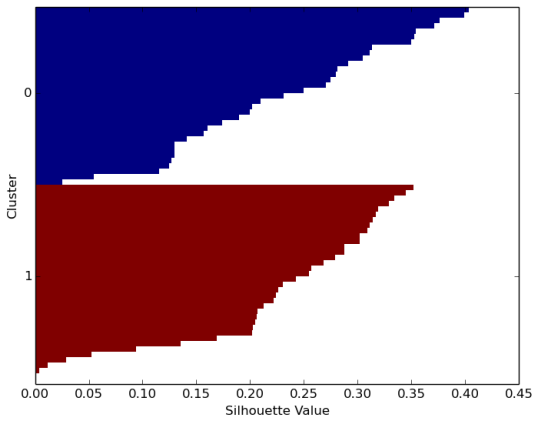

Figure 4: Unfiltered silhouette plot

We can clearly see that for the largest part the silhouette score of our data points implies well-defined clusters where data points tend to show a much larger degree of equivalence towards their assigned cluster than the other cluster. However, we can also see a number of data points with a reasonably lower silhouette coefficient. This implies that these data points are not necessarily strongly grouped with the other data points in their cluster and/or show a significant degree of similarity to data points from the other cluster.

As described in section 4.3, the problem that we might expect to run into is the fact that these data points are highly likely to switch between cluster assignments with minor adjustments to their values. As this might skew our results on the topic of shifter detection, we decide to filter these data points from our data set. We apply a cut-off threshold based on individual data points' silhouette coefficients in relation to the mean silhouette coefficient. This allows us to counter the effect of a minor change in a data point showing up when we explore the concept of shifter detection. Figures showing the results of this filtering may be seen below.

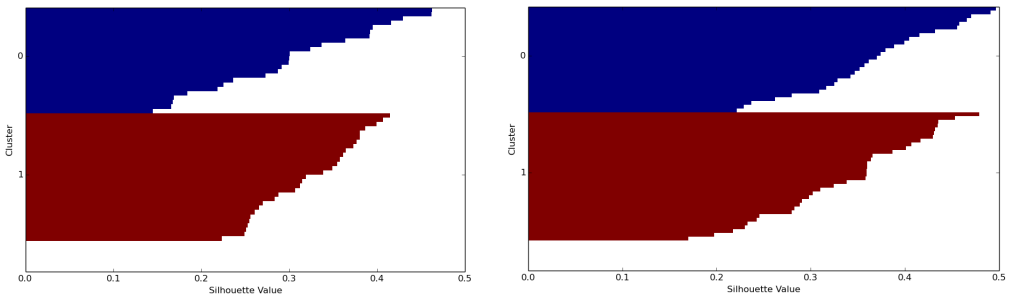

Figure 5: Filtered silhouette plots (left: set 1, right: set 2)

Filtering using our silhouette coefficient threshold has the desired effect of allow-

ing us to disregard the data points that will not be representative of employees significantly shifting between clusters. This is apparent from the fact that data points with lower silhouette coefficients, which are likely to be of little significance as potential shifters are no longer included in our plot. It is important to remain careful not to cut off too large of a number of data points as to not disregard too much valuable information we can use for the purpose of shifter analysis.

5.2.2 Andrews Curves

The Andrews plots based on our clustered data are displayed below. The legend refers to the individual clusters as 0.0 and 1.0 while outliers are referred to as 2.0.

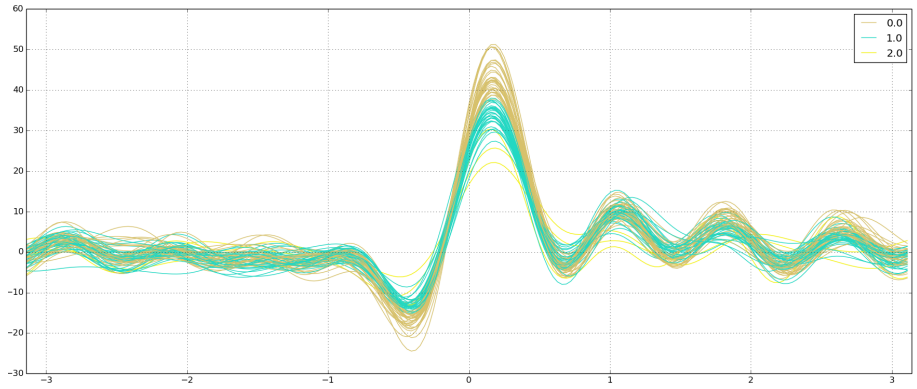

Figure 6: Andrews plot (Data set 1)

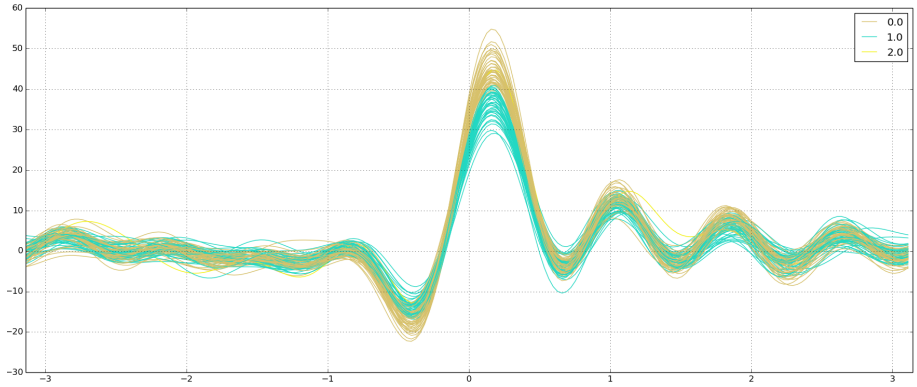

Figure 7: Andrews plot (Data set 2)

We can safely conclude that the different clusters are individually identifiable in a number of regions in our plot, mainly at the minimum and maximum values around the middle of the Andrews plots. This leads to the conclusion that judging by the data points' Andrews curves both clusters are indeed distinct and well-defined. Interestingly we can also easily spot several instances where the curve

corresponding to an outlier differs significantly from all of the other curves. This is mainly obvious to the right of the middle where yellow lines representing outliers are visible significantly above and below the lines representing instances from both clusters. This implies the validity of the characterisation of the respective data point as an outlier.

5.2.3 RadViz

The RadViz plots based on our clustered data are displayed below. As before, the legend refers to the individual clusters as 0.0 (positive) and 1.0 (negative) while outliers are referred to as 2.0.

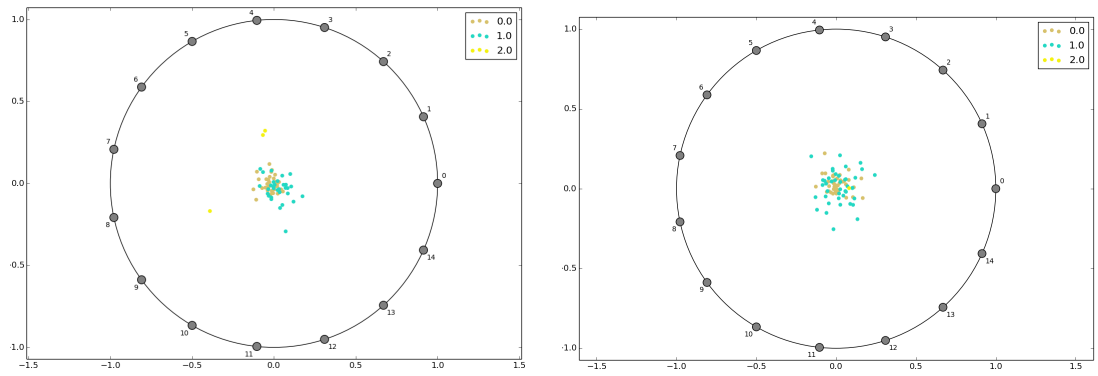

Figure 8: RadViz plots (left: set 1, right: set 2)

Although results differ slightly between the different data sets it is immediately obvious that there are some significant similarities visible between both visualisations. First and foremost, cluster 0 is more dense than cluster 1, implying a larger amount of internal variance for data points belonging to cluster 1. This is unsurprising given the results from our conventional plotting method, which display a more varied response pattern among employees assigned to cluster 1.

Additionally, this method allows us to easily identify most of the outliers we earlier characterised as such using LOF. These data points are all reasonably far away from both of the identified clusters, especially in the case of the first data set.

Furthermore we can see a reasonable degree of separation between the two clusters, especially with regard to the first data set. The brownish points representing cluster 0 are located more to the upper left whereas the green points representing cluster 1 are located more the the bottom right. Given the high dimensionality of our data however, this is highly likely to be mostly arbitrary.

5.3 Shifter detection

Shifter detection was carried out on the filtered data sets where data points with a silhouette coefficient below a certain threshold were taken out. This allows us to only take significant changes between cluster assignments into consideration as minor differences resulting in a cluster shift would not necessarily lead to the deduction of useful insights. Response patterns for shifters as well as non-shifters were plotted as shown below in order to gain an understanding of shifter behaviour.

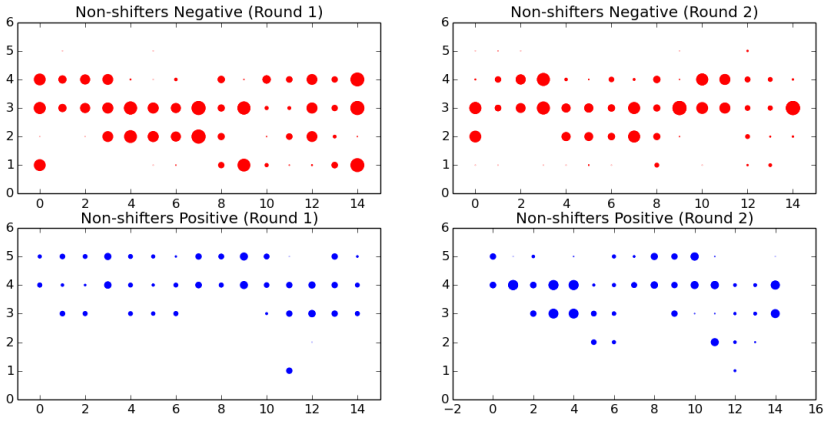

Figure 9: Non-shifters (Data set 1)

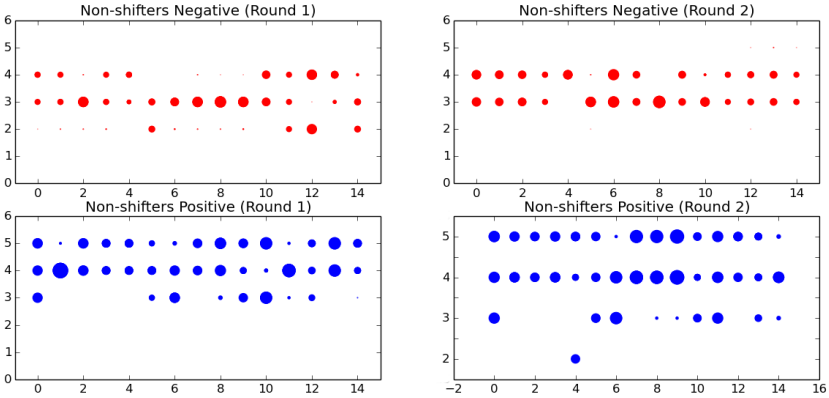

Figure 10: Non-shifters (Data set 2)

As can be deduced from these figures, non-shifters in the negative cluster primarily answer to most questions with a 2-4 response on the Likert scale. Non-shifters in the positive cluster primarily answer with a 3-5 response on the Likert scale. This is mostly representative for the clusters as a whole (as presented in section 5.1.2), where we can see comparable response patterns. This is unsurprising given the fact that non-shifters are likely characterised by their assigned cluster fairly well and as such are unlikely to shift between clusters.

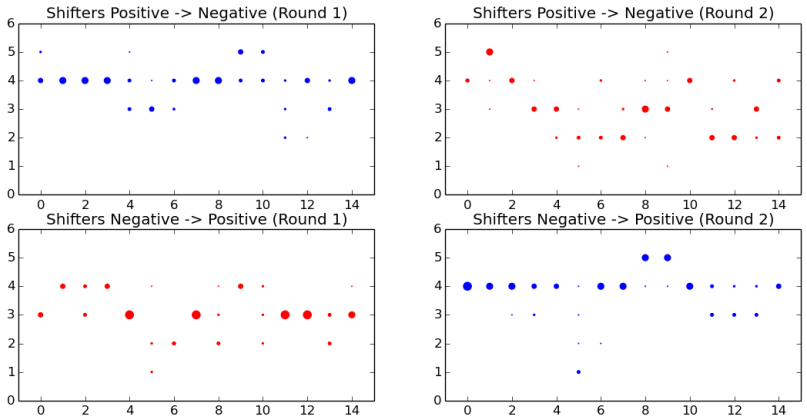

Figure 11: Shifters (Data set 1)

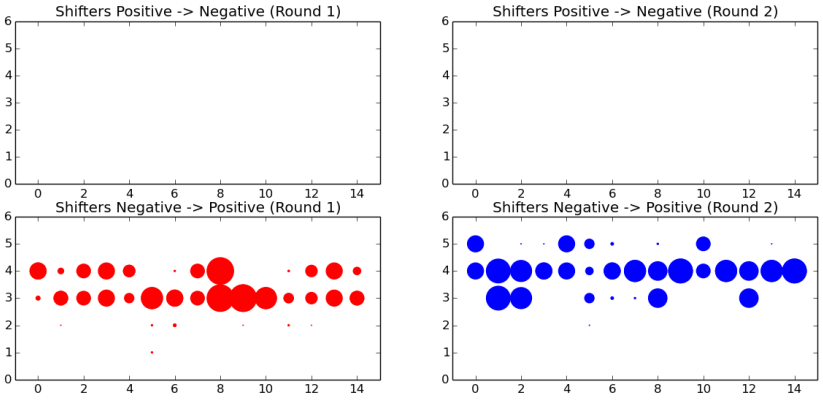

Figure 12: Shifters (Data set 2)

A number of interesting findings can be deduced from the figures displayed above. With regard to the first data set, as is to be expected, there are very few instances of shifters from positive to negative where questions were answered with a '5' response on the Likert scale. We also see most shifters from the negative to the positive cluster can be found within the upper regions of the negative cluster. This implies the fact that employees who are highly positive or highly negative about a certain situation are unlikely to change their opinion over the course of time. This should be taken into consideration when trying to influence change adoption over the course of its implementation. Employees who did in fact shift from the negative to the positive cluster became significantly more positive. It could be very interesting from a business perspective to find out what triggered this.

In the second data set we can see something very interesting happening, namely the fact that there are no instances of employees shifting from the negative to the positive cluster. Previous findings do still apply to this data set as employees shifting from the negative cluster to the positive cluster tend to be located around the neutral range in the first round and become significantly more positive overall.

The fact that there are no shifters from positive to negative reinforce the idea that from a business perspective, this type of analysis can be a powerful tool to measure and analyse change adoption. After all, when a significantly larger amount of shifters from the negative to the positive cluster are detected than the other way around, we can safely assume that the success of a change's adoption is increasing.

6 Conclusions

The current research suggests that it is highly viable to analyse high-dimensional workforce survey data using data clustering methods. This is based on the fact that distinct and well-defined clusters arise from analysis by the methods we used and these clusters can be used to investigate the degree in which a change is currently being adopted as discussed in the section 5.3. Highly positive and moderately negative clusters are consistently found in multiple data sets. The verification of clusters established from these data sets carries some challenges with it. However, several visualisation methods including Andrews curves and more conventional plotting methods have proven to be successful tools for the verification and characterisation of detected clusters.

The analysis of employees shifting between opposing clusters from one survey round to another allows for valuable insights into the perception of large-scale change adoption through the interpretation of these results. This is very interesting from a decision-making oriented perspective as these findings provide a wealth of knowledge upon which decisions can be based. Additionally, the success of the implementation of a certain change can be judged fairly intuitively using the established clusters and the demonstrated concept of shifter detection. Taking all these factors into consideration, the presented findings imply the fact that this type of analysis can be hugely beneficial in a variety of ways.

6.1 Future work

Several choices have been made in the current research regarding algorithm selection, methods of visualisation and interpretation of results. Further research may look to expand upon these choices, possibly applying optimisations to selected algorithms in order to increase specificity and potentially achieving greater results.

Additionally it would be interesting to research the scalability of the current research by applying our methods to data sets with larger sample sizes. Perhaps this could even include collective data sets combining comparable surveys and studying the resulting sets as a whole. Although this would decrease specificity, the increased sample size might lead to valuable new insights and provide a chance to increase the degree of generalisability. Examples of this might include

insights into intra-response patterns as to gain a greater understanding of how to potentially influence employee perception to large-scale changes.

References

[1] H. Hollenstein, "Innovation modes in the swiss service sector: a cluster analysis based on firm-level data," *Research Policy*, vol. 32, no. 5, pp. 845–863, 2003.

[2] K. G. Rice and R. B. Slaney, "Clusters of perfectionists: Two studies of emotional adjustment and academic achievement," *Measurement and Evaluation in Counseling and Development*, vol. 35, no. 1, p. 35, 2002.

[3] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[4] P. Redmond, J. A. Trono, and D. Kronenberg, "Affinity propagation, and other data clustering techniques." `http://academics.smcvt.edu/jtrono/Papers/SMCClustering%20Paper_PatrickRedmond.pdf`. Accessed: 2016-07-18.

[5] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[6] E. Xing, M. Hein, U. V. Luxburg, and A. Singh, "Spectral clustering." `https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf`, 2010. Accessed: 2016-07-21.

[7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96-34, pp. 226–231, 1996.

[8] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[9] R. Layton, "Unsupervised evaluation metrics (scikit-learn github repository)." `https://github.com/scikit-learn/scikit-learn/blob/51a765acfa4c5d1ec05fc4b406968ad233c75162/sklearn/metrics/cluster/unsupervised.py`, 2015. Accessed: 2016-06-08.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29-2, pp. 93–104, ACM, 2000.

[11] R. E. Moustafa, "Andrews curves," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 4, pp. 373–382, 2011.

[12] J. Sharko, G. Grinstein, and K. A. Marx, "Vectorized radviz and its application to multiple cluster datasets," *IEEE transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1444–1427, 2008.

[13] Amro, "Silhouette plot." `http://stackoverflow.com/a/6725320`, 2014. Accessed: 2016-06-12.

[14] D. Kužnar, "Pylof." `https://github.com/damjankuznar/pylof`, 2013. Accessed: 2016-07-28.