Back Pain: A Spectral Clustering Approach

Joseph Fitch, Nazia Khan, and Cristina Tortora

Abstract We used a Spectral Clustering algorithm to find clusters among medical patients with lower back pain symptoms, and then we assessed the health outcomes within each cluster. First, we mapped all of the variables onto $[0,1]$ intervals. This allowed us to compute a similarity score between every pair of patients, using an adaptation of Pearson correlation. We then calculated the spectral (eigen) decomposition of this similarity matrix, and we used the first few eigenvectors to create a low-dimensional subspace. Finally, we performed *k*–means clustering in this new subspace to find four clusters. We compared the cluster means and variances for each recovery assessment variable to differentiate the health outcomes for each cluster. Lastly, we highlighted the identifying symptoms of each patient cluster by inspecting any variable whose within–cluster average is extraordinarily low or high, relative to the other clusters.

Joseph Fitch
San José State University, California, USA
✉ Joseph.Fitch@sjsu.edu

Nazia Khan
San José State University, California, USA
✉ k.nazia2010@gmail.com

Cristina Tortora
San José State University, California, USA
✉ cristina.tortora@sjsu.edu

ARCHIVES OF DATA SCIENCE (ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. -, No. -, -

ISSN 2363-9881

1 Introduction

Outside of some well-ordered data sets that are popular for basic clustering demonstrations, many situations do not provide a clear clustering structure. The back pain data set (see cite(INTRODUCTION) for details) is no different; there is a diversity of formats and scales which inhibit any straightforward clustering procedures. In contrast, literature shows that spectral clustering methods are particularly effective in this context (Von Luxburg, 2007) (Kannan et al, 2004). Specifically, similarity measures and eigen-decompositions are used to extract underlying structure from high-dimensional data (Belkin and Niyogi, 2003) (Arias-Castro et al, 2011) and to identify abstract clustering patterns (Ng et al, 2002). We rely on this flexible nature in order to handle the rougher obstacles of this data challenge.

2 Data Processing

With mixed data, it is challenging to incorporate both numerical and categorical data equally. The following procedures were designed to evenly consider both types of variables, without letting one type of data suppress the information of the other. The validation variables are only used for cluster interpretation, so we will treat them separately in section 2.3.

2.1 Variable Transformations

For any categorical variable (including binary variables), we used disjunctive coding to codify a single variable with c many categories into c distinct columns. An observation would then receive a value of 1 in the column which corresponds to its appropriate category, as shown in Table 1.

Table 1 A variable with 3 categorical values is transformed into 3 binary columns.

	Category			Category = 1	Category = 2	Category = 3
Person A	1	$\Longrightarrow$	Person A	1	0	0
Person B	2		Person B	0	1	0
Person C	3		Person C	0	0	1

Mathematically, this will inflate the similarity score between rows of data with the same category value, as they will now appear similar across three disjunctive columns instead of the original single column. To prevent this artificial inflation of covariance, we will assign weights of $1/\sqrt{c}$ to these columns, where c is the respective number of distinct categories for that variable. This will retain the total weight of each categorical variable equal to 1.

To preserve even magnitudes for numeric and categorical variables, we linearly scaled every continuous and ordinal column to the interval [0,1]. We mapped each variable's minimum value to 0, while mapping the maximum value for each variable 1. See Figure 1 for a detailed explanation. This ensured even consideration of all variables during the clustering process; uneven magnitudes between variables would otherwise allow large variables to dominate small variables during the clustering process.

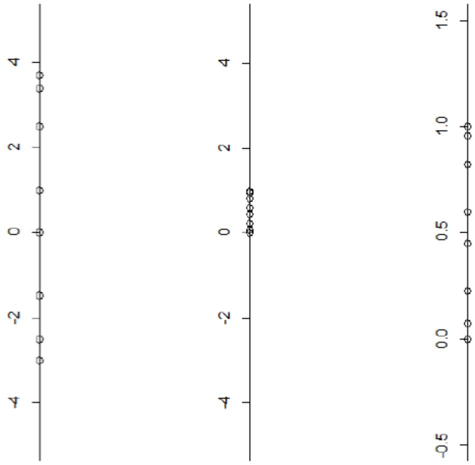


Fig. 1 Transforming a variable from its observed range of values into a set of values on [0,1].
· The left image represents data from a single continuous variable, plotted in its respective 1D space.
· The middle image represents that same data mapped to the interval [0,1], preserving the relative magnitude between the points.
· The right image zooms in on the interval [0,1], verifying that the points retained the same relative distribution.

This method is admittedly very sensitive to outliers. Within a numeric variable, if there exists a value or set of values which are much larger than the rest, then the other values will all be pushed together into a small margin as the outliers disproportionately stretch the maximum/minimum values. This may artificially reduce the variance within that specific variable, relative to other variables. Despite these concerns, this technique allowed for other useful interpretations which will be discussed in section 3.1.

2.2 NA Treatment

There were two variables in the data set with an exceedingly large amount of missing values. The "musclegroup_palp" variable had 52% missing data, and the "triggerpoint" variable had 37.4% missing data.

The "musclegroup_palp" variable mentions that roughly half of the missing values are not actually not unknown quantities, but rather represent individuals who simply reported no painful muscle groups (these are the "real" missings). However, this is characteristically different from individuals who may truly experience muscle pain but simply didn't respond to this question (the "not real" missings). Grouping these two responses together seems unreasonable, so we chose to exclude this variable.

The "triggerpoint" variable offers far less confusion. The variable summary mentions that many of the NA values here were simply recorded under the musclepalp variable instead (not to be confused with the musclegroup_palp variable we discussed earlier). We therefore choose to keep the triggerpoint variable, in the expectation that many missing values here will be reconciled by the musclepalp variable and vice-versa.

After removing the "musclegroup_palp" variable, we replaced all missing values with mean imputation by column. We chose mean imputation because our algorithm relies on centering the data for correlation calculations, and mean imputation will preserve the variability of each row (having zero effect on the correlation similarity between rows). Effectively, if either row is missing the data for that column, then that column will provide "zero information" of the similarity between those two rows. This is discussed in depth in section 3.1.

2.3 Validation Data

The first issue with the validation data was the different scales for each measurement. The "gen" score is measured on a discrete scale from 1–7; the "vasl" score is measured on a discrete scale from 0–10; and the "rmprop" score is a continuous variable ranging from 0–100. To normalize the scale between variables, we employed the standard scaling procedure of centering each column and dividing by its standard deviation.

After scaling, the main issue with the validation variables is the huge proportion of missing data. In order to ensure reliability of validation testing, our testing data only includes observations which have *at least one* value recorded for each time period (2 weeks, 3 months, and 12 months). We partitioned the 9 validation variables into those three distinct time periods, where we used the 3–nearest–neighbor average for each observation to impute any NA's. The variables within each time partition are significantly correlated (see Figure 2), so the missing variables should be well-predicted by the other variables within the same time period. This justifies the usage of the k-nearest neighbor (kNN) algorithm as a strong choice for imputation. Also note that the variables were standardized to equivalent scales, further validating the usage of kNN. For implementation of kNN, we used the R package VIM (Kowarik and Templ, 2016).

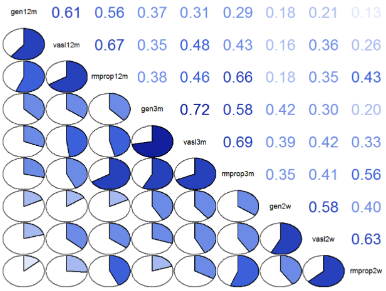


Fig. 2 Correlations appear strong within a single time period and weak between different time periods. We use observations within a single time period to predict any other missing values within that time period.

3 Spectral Clustering Algorithm

With a mixed data set and many different scales of measurement, it does not seem appropriate to run a standard clustering procedure directly within that data space. Instead, we aimed for a more well-ordered data space through construction of a similarity matrix. Groups of mutually similar observations should clump together, avoiding some common pitfalls of working with non-spherical or abstract clustering structure (see Von Luxburg, 2007).

3.1 Similarity Matrix

We transformed the original $n \times p$ data matrix into a $n \times n$ similarity matrix S, where S_{xy} = Sim(x,y) = Sim$(y,x) \in [0,1]$. We adapted a variant of Pearson correlation to calculate pairwise similarity between rows of the data:

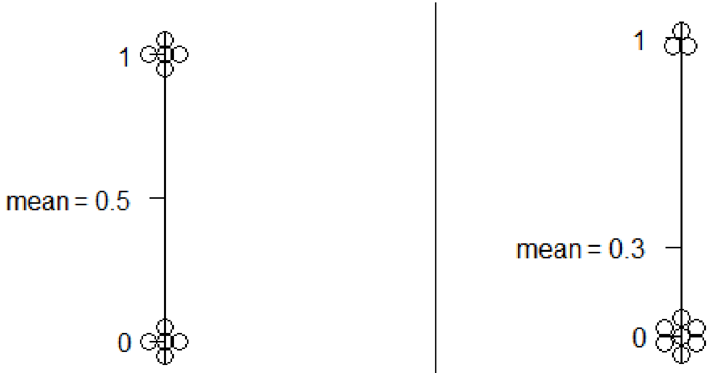
$$\mathrm{Corr}(\mathbf{x},\mathbf{y}) = \frac{\sum\limits_k w_k(x_k-\mu_k)(y_k-\mu_k)}{\sqrt{\sum\limits_k w_k(x_k-\mu_k)^2 \sum\limits_k w_k(y_k-\mu_k)^2}} = \frac{(\mathbf{x}-\mu)^T W(\mathbf{y}-\mu)}{||\mathbf{x}-\mu|| \cdot ||\mathbf{y}-\mu||} \tag{1}$$

where k iterates over the data columns and ^{T} signifies the transpose for the matrix multiplication. This calculation essentially compares rows **x** and **y** according to their deviation from the mean in each column. If these rows deviate from the mean in opposite directions, that variable will contribute negatively to their similarity score. If these rows deviate from the mean in the same direction, then that variable will contribute positively to their similarity score. If one row or both rows hover around the mean value, that variable will contribute zero similarity between the rows. The stronger the mean deviations, the stronger the similarity contribution.

For binary and disjunctive categorical data, the mean reflects the proportion of 1's for each category. If we see an even split between 0's and 1's – signifying a mean value around 0.5 for that variable – then that variable will strongly distinguish those two groups (positive similarity within a group, and negative similarity between the groups). On the other hand, if a variable has a majority of values equal to zero, then the sample mean will have a value close to zero. This will provide *weak* positive similarity to the observations in the large group, but *strong* positive similarity to the observations in the minority group (see figure 3.1). Intuitively, that variable acts as a unique identifier of that minority group: if most of the data acts one way, then there is likely some unifying factor within the minority group that causes it to act differently. It is harder to claim any strong identifiability within the majority group, as *most* of the data acts that way – it's not a good indicator of some identifying property, only a lack thereof.

A similar interpretation can be made with numeric data in regards to the mean, albeit perhaps less clear-cut than with binary variables. Also, as mentioned in section 1.3, outliers may skew the [0,1] interval to force a large clump of values near 0 or 1. Correlation similarity will assign strong similarity within the outlier group (if there is more than one outlier) while attributing negligible similarity within the clump of "regular" values.

Fig. 3 We observe how the distribution of points – specifically, the location of the mean – influences the pairwise correlation similarity.



Even Split between values of 0, 1.

Correlation similarity of (.5)(.5) = (–.5)(–.5) = .25 is given to each pair of observations with the same value. Correlation similarity of (–.5)(.5) = -.25 is given to each pair with differing values.

Uneven Split between values of 0, 1.

Correlation similarity of (–.3)(–.3) = .09 is given to observation pairs within the majority group, while (.7)(.7) = .49 similarity is assigned to members of the minority group. (.7)(–.3) = –.21 similarity is given to observations with differing values.

One drawback of correlation similarity, with regard to numeric data, is that it relies on extreme values to identify similarity. This process can identify only 2 clusters within a single variable (assigning positive similarity to observations near the extrema), even though it is reasonable to believe that there could exist a third group of observations which contains *neutral* values clustered around the mean. However, given that correlation is calculated according to distance from the mean, this method cannot identify any "neutral"-valued cluster within a single variable. It relies on the assumption that each cluster will uniquely display extreme values (very large or very small) in at least one variable. We proceeded under the assumption that clusters are identified by extreme values, rather than middling values (in regards to numeric data). If there are any observations that naturally cluster together with extreme values across multiple data values, correlation similarity should identify that trend and assign those observations a large overall similarity score (close to 1).

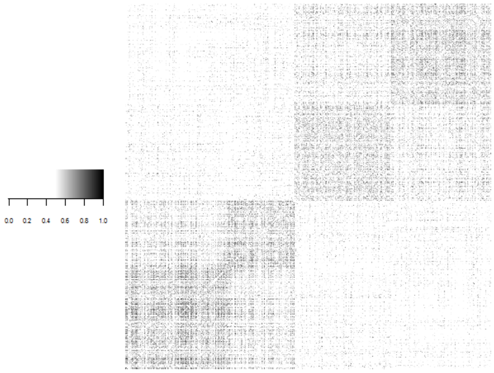


Fig. 4 A heatmap of the data using correlation similarity described above, with the index reordered for visual clarity. Notice that there is a strong case for two, three, or four clusters, based on this heatmap.

3.2 Diffusion Map Spectral Clustering

After constructing the similarity matrix, we needed to extract the clustering structure. We adapted the popular Diffusion Map algorithm, which treats the similarity values as a probability distribution in a Markov process.

First, we mapped each column of the similarity matrix (excluding the diagonal) to a value between [0,1], using the same method described in Figure 1. We then set the diagonal elements all equal to zero, which we will explain in the next paragraph. From there, we divided each row by its respective row sum, converting each row into a probability distribution with nonnegative entries that sum to 1.

Given a matrix of probability distributions, a Markov chain lets you take any number of "steps" forward and calculate the probability of traveling from observation A to observation B. For our purposes, we postulated that this probability is large if A and B are similar to the same set of observations, while this Markov probability is conversely low if A and B do not share mutual similarities. We previously set the diagonal elements equal to zero because the diagonal probabilities correspond to self-loops, which are not interesting or useful for this algorithm.

A large number of steps (e.g. $t \geq 3$) tends to erase the finer details and focus on the large-scale structure. A smaller number of steps (e.g. $0 \leq t \leq 2$) will often let the smaller clusters show themselves, allowing for more clusters.

Because there is not clear resolution in the similarity structure, we are very cautious toward erasing any finer detail. We chose to take $t = 0.5$ steps forward, which should ideally eliminate some noise between weakly-connected observations while preserving the connections within smaller clusters.

We used the eigen-decomposition of this probability matrix to reduce the dimensions while retaining the overall similarity structure (see Nadler et al, 2006). We chose the number of dimensions equal to the number of clusters, allowing enough space to separate the clusters but keeping the dimensionality low for strong clustering. For k clusters and t steps, we constructed the Diffusion Map space Ψ according to:

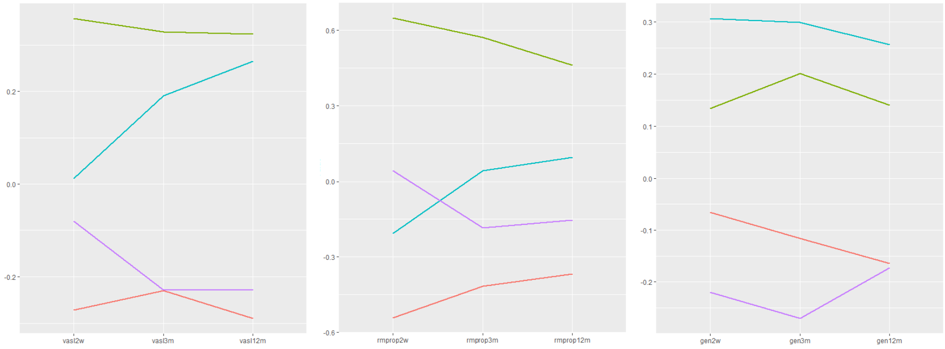
$$\Psi_t(x) = \left[\lambda_2^t \, \psi_2(x),\ \lambda_3^t \, \psi_3(x),\ \ldots,\ \lambda_{k+1}^t \, \psi_{k+1}(x)\right] \tag{2}$$

where λ_p is the p-largest eigenvalue, ψ_p is the corresponding eigenvector, and x is any row of the data. Notice that we specifically chose to exclude the first eigenpair. As a result of mapping all similarity values to [0,1], the first eigenvector tends to point directly through the "first quadrant", yielding a simple weighted mean of the data. This is not useful for separating clusters within the data. Moreover, the first eigenvalue is always much larger than the following eigenvalues, and hence the first dimension would dominate the clustering structure. Eigenvalues 2 through k should theoretically correspond to the dimensions *separating* clusters, which is where we want to focus. Lastly, we normalized each row within this Diffusion Matrix to a unit vector, such that each vector is mapped to the unit sphere. This guarantees that we are only considering the "direction" of a vector and ignoring the magnitudes (equalizing strongly connected and weakly connected rows). For efficient eigen-decomposition, we use the R package RSpectra (Qiu and et. al., 2016).

4 Results & Insights

We focused on the 4-cluster structure, as this clustering gave very clear interpretations when analyzed using the validation variables.

For a baseline interpretation of the clusters, we constructed a parallel coordinates plot for each set of validation variables. Recall that we centered each variable and scaled by the standard deviations; hence, in the following plot, the vertical axis represents a cluster's standardized performance relative to each column's mean and standard deviation.



Vasl RMprop Gen

Fig. 5 Each plot reads temporally from left to right. A low y-value indicates good health (according to the respective test measure) while a high y-value indicates poor health. Generated with R package GGally (Schloerke and et. al., 2016)

These plots suggest certain insights to the behavior of the clustering structure. The red cluster appears to score consistently healthy, while the green cluster scores consistently poor over time. According to the Vasl and RMprop variables, the purple cluster starts in bad shape but heals over time, while the blue cluster conversely starts out relatively well but gets worse over time.

Interestingly, the Gen variable agrees that the red and purple clusters do better compared to the green and blue clusters, yet it seems to switch the behaviors of red/purple and green/blue. We hypothesize that the Gen variable gives less reliable interpretations because of the narrow scoring scale: values place in one of 7 categories, while Vasl allows for 11 categories and RMprop is measured on a continuous scale from 0–100. Further investigation is certainly warranted.

For completeness' sake, we also inspected each cluster on the original unnormalized validation variables. We still imputed the missing values using kNN on the scaled data, to equalize the distance magnitudes (otherwise the kNN algorithm would be dominated by the variable which operates on the largest scale, i.e. RMprop). We subsequently un-scaled the data by multiplying by the original standard deviations and adding the original column means, producing the exact same data space that we had initially. Results in Figure 6.

To determine the effectiveness of the clustering, we must also consider the within-cluster variance of each validation variable, instead of only measuring

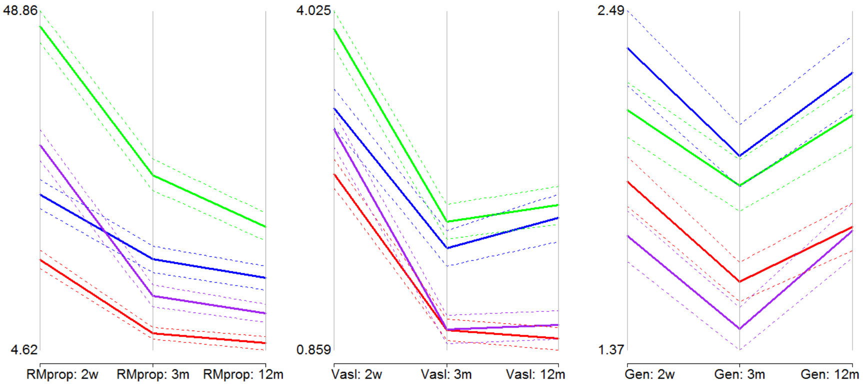


Fig. 6 Each plot reads temporally from left to right. A low y-value indicates good health (according to the respective test measure) while a high y-value indicates poor health. A solid line represents the cluster mean, and the dotted lines represent ± 1 standard error of the mean. Notice that Vasl and RMprop show a general recovery from the 2 week measurement to the 12 month measurement, while the Gen variable is hard to interpret in terms of recovery.

the cluster means. In an ideal case, we would hope to see the mean values separated by significant magnitude of standard deviations. The within–cluster standard deviations are shown in Figure 7.

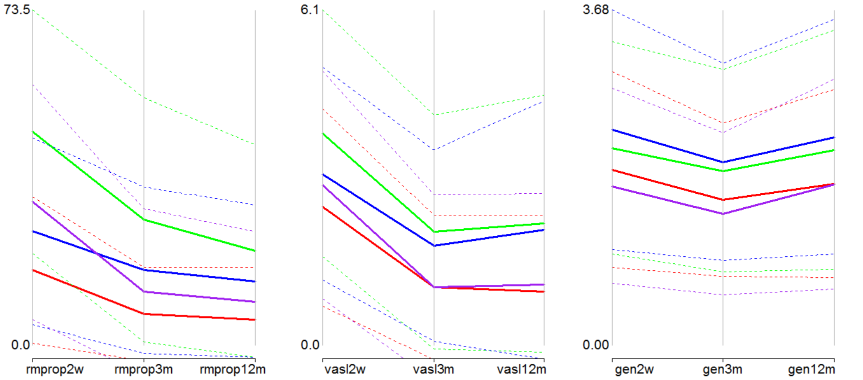


Fig. 7 Each plot reads temporally from left to right. A low y-value indicates good health (according to the respective test measure) while a high y-value indicates poor health. The solid lines represent the cluster means, and the dotted lines represent ± 1 standard deviation (measured separately within each cluster).

Unfortunately, this condition is poorly reflected in all measurements except the RMprop score, which shows good separation between the green and red clusters (the “consistently healthy” and the “consistently unhealthy” clusters) but still shows strong overlap between the two middling clusters. Overall, Figure 6 suggests that we did find some real information, as the means show a significant and interpretable pattern. Unfortunately, Figure 7 shows that the clusters’ separation in the validation data may not be distinct enough to classify the patients in a reliable way.

To highlight the identifying factors of each cluster, we analyze the mean of each cluster within those centered columns. We searched for any component of a cluster centroid which is near the extremes, given that all variables have been mapped to the [0,1] interval. This standardization also validates the comparison of means across different variables. Extreme averages will point to any value in which the cluster scores consistently high or consistently low. As with correlation similarity, we note that this method cannot identify variables in which a cluster scores consistently in the middle; an analysis of column variances would be required for such identification. See tables 2 through 5 for the results of the cluster analysis.

Table 2 Cluster 1 demonstrates the best recovery outcomes, according to the validation variables. In the clustering data, Cluster 1 is characterized by low scores in the "physical pain & trouble" variables, indicating good overall health (a high score would indicate high physical pain/trouble).

Mean Score	Description
Low	"Because of my back problem, I go upstairs more slowly than usual."
Low	"I only walk short distances because of my back problem or leg pain (sciatica)."
Low	"Because of my back problem, I use a handrail to get upstairs."
High	Sex = male.
Low	"Because of my back problem, I am doing less of the daily work around the house than I would usually do."

5 Conclusions

We used a Spectral Clustering algorithm to find clusters among lower back pain symptoms in medical patients. We processed the different types of data separately such that all types of data are mapped to comparable dimensions. We computed a similarity score between every pair of patients using an adap-

Table 3 Cluster 2 demonstrates the worst recovery outcomes, according to the validation variables. In the clustering data, Cluster 2 is characterized by high scores in the "psychological distress" variables. This agrees with known literature that commonly shows how psychological health is highly correlated with physical well-being.

Mean Score	Description
High	"In general I have not enjoyed all the things I used to enjoy."
High	"Worrying thoughts have been going through my mind a lot of the time."
High	"Because of my back problem, I am more irritable and bad tempered with people than usual."
High	"My back pain has spread down my leg(s) at some time in the last 2 weeks."
High	"Because of my back problem, I am doing less of the daily work around the house than I would usually do."

Table 4 Cluster 3 starts out with mediocre recovery symptoms but converges toward Cluster 2 over time. Interestingly, we found that Cluster 3 assigns importance to many of the same variables as cluster 1, but *in the opposite extremity*. Cluster 1 demonstrated very healthy scores in the "physical trouble" categories, while cluster 3 clearly captures all the individuals who have poor initial physical symptoms.

Mean Score	Description
High	"I only walk short distances because of my back problem or leg pain (sciatica)."
High	"Because of my back problem, I have to hold on to something to get out of an easy chair."
High	"I find it difficult to get out of a chair because of my back problem or leg pain (sciatica)."
High	"Because of my back problem, I go upstairs more slowly than usual."
High	"Because of my back problem, I use a handrail to get upstairs."

Table 5 Cluster 4 is characterized by healthy physical symptoms, as with Cluster 1. In the validation variables, Cluster 4 converges toward cluster 1, and the difference between them mostly disappears by the 12-month follow-up. It seems that one of the major distinctions (most notably, male vs. female) may catalyze different recovery patterns in the short term, but converge to similar prospects for long term health and recovery.

Mean Score	Description
Low	"I have trouble putting on my socks (or stockings) because of the pain in my back or leg."
Low	"I get dressed more slowly than usual because of my back problem or leg pain (sciatica)."
High	Sex = female.
Low	"I find it difficult to get out of a chair because of my back problem or leg pain (sciatica)."
Low	"Because of my back problem, I have to hold on to something to get out of an easy chair."

tation of Pearson correlation and then we calculated the spectral (eigen) decomposition of this similarity matrix, reducing the dimensionality; we finally performed k-means clustering in this new subspace.

We believe that there is truly a 4-cluster structure within this data, but the partition is hard to produce reliably. At the very least, we can yield a significant 2-cluster structure (combining Cluster 1 with Cluster 4, and Cluster 2 with Cluster 3) that produces meaningful separation between positive vs. negative long-term outcomes. In the absence of finer resolution, this 2-cluster structure would be useful for identifying patients in danger of long term health risks.

Acknowledgements This work was supported by the Woodward Student Travel Award. The authors would like to thank the IFCS committee, specifically Christian Hennig and Iven Van Michelen, for the opportunity to participate in the challenge.

References

Arias-Castro E, Chen G, Lerman G (2011) Spectral clustering based on local linear approximations. Electronic Journal of Statistics

Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6):1373–1396

Kannan R, Vempala S, Vetta A (2004) On clusterings: Good, bad and spectral. Journal of the ACM 51(3):497–515

Kowarik A, Templ M (2016) Imputation with the R package VIM. Journal of Statistical Software 74(7):1–16

Nadler B, Lafon S, Coifman R, Kevrekidis I (2006) Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. Applied and Computational Harmonic Analysis 21(1):113–127.

Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems (NIPS) 14:849–856

Qiu Y, et al (2016) Rspectra: Solvers for large scale eigenvalue and svd problems. R Package Version 0120

Schloerke B, et al (2016) Ggally: Extension to ’ggplot2’. R Package Version 132

Von Luxburg U (2007) A tutorial on spectral clustering. Max Planck Institute for Biological Cybernetics TR-149.