Categorical Sequence Analysis with Optimal Matching: An Application with Data from the 'Women 40+ Healthy Aging Study'

A Thesis

Presented to

The Division of

University of Bern

In Partial Fulfillment

of the Requirements for the Degree

Master in Statistics and Data Science

Adriana Clavijo Daza

June 2023

Approved for the Division
(Statistics)

_____ _____

Serena

Acknowledgements

I want to thank a few people.

Preface

This is an example of a thesis setup to use the reed thesis document class (for LaTeX) and the R bookdown package, in general.

Table of Contents

List of Tables

List of Figures

Abstract

The preface pretty much says it all.

Second paragraph of abstract starts here.

Dedication

You can have a dedication here if you wish.

Introduction

Chapter 1

Optimal Matching

Optimal Matching (OM) is a technique used in social sciences for the comparison of sequences of categorical states indexed by time. this method has applications in different areas of social sciences, for instance, life course or career path analysis. OM uses the Needleman-Wunsch algorithm, that was developed to compare biological sequences. This algorithm is an application of dynamic programming, an iterative method that simplifies an optimization problem by breaking it into a recursion of smaller problems that are simpler to solve.

1.1 The OM algorithm

Consider a set of n categorical states, say, $S = \{s_1, \ldots, s_n\}$. A sequence of (discrete) length $t > 0$ can be denoted as $X = (x_1, \ldots, x_t)$, where $x_i \in S$ for $i = 1, \ldots, t$. Additionally, the set of all possible sequences with states belonging to S is denoted by $\mathbf{S}$.

Now, let $X, Y \in \mathbf{S}$ be two sequences of size t_X and t_Y, respectively. In order to assess the similarities between the sequences X and Y, and to obtain a numerical value associated with it, we define an empty array, F, of size $(t_X + 1) \times (t_Y + 1)$.

The algorithm below shows the initialization and recursion to fill the array F.

1: $F(1,1) \leftarrow 0$
2: **for** $j \leftarrow 2, t_Y + 1$ **do**
3: $\quad F(1,j) \leftarrow F(1,j-1) + d$
4: **end for**
5: **for** $i \leftarrow 2, t_X + 1$ **do**
6: $\quad F(i,1) \leftarrow F(i-1,1) + d$

7: **end for**
8: **for** $i \leftarrow 2, t_X + 1$ **do**
9: **for** $j \leftarrow 2, t_Y + 1$ **do**
10: $F(i,j) \leftarrow min\{F(i-1,j)+d, F(i,j-1)+d, F(i-1,j-1)+k(y_{i-1}, x_{j-1})\}$
11: **end for**
12: **end for**

Here, the value d is the cost of inserting a gap in one of the sequences, also known as *indel* cost, and $k(y_{i-1}, x_{j-1})$ is the cost associated to change from the state y_{i-1} to x_{j-1}, which is defined in a matrix K of size $n \times n$, commonly known as the cost matrix.

Lines 1-7 of the OM algorithm correspond to initialization. Starting with a cost of 0 in $F(1,1)$, the first row and column of F represent cumulative costs of successively adding gaps. The remaining lines of the algorithm correspond to the row-wise recursion to fill the array F according to the content of the sequences to be compared: at any step of the recursion, the algorithm is looking at a specific pair of indexes (location) and calculating if substitution or insertion/deletion is the cheapest operation. Successively adding the costs of the cheapest operations results in the overall optimal cost for aligning (arrange to detect similarities) the sequences X and Y.

In fact, when F is completely filled, the value in the last cell, i.e. $F(t_X + 1, t_Y + 1)$ corresponds to the optimal cost of aligning the sequences X and Y. It is possible to recover the steps that conduced to this alignment with a traceback from the last cell. However, this is not necessary to obtain the dissimilarities matrix.

The `R` package `TraMineR` provides several functions to analyze and visualize sequential data. In particular, the package implements OM and offers several methods for computing the cost matrix K and the normalization of the dissimilarity matrix.

1.1.1 Cost matrix

The cost matrix K is a symmetric matrix of size $n \times n$. The value in the i-th row and j-th column $K(s_i, s_j)$ indicates the cost of moving from state s_i in time $t > 0$ to state s_j in $t + 1$.

The following are the methods available in `TraMineR`.

Transition rates (`TRATE`):

The substitution cost between states s_i and s_j, $1 \leq i, j \leq n$, is calculated as:

$$K(s_i, s_j) = c - P(s_i|s_j) - P(s_j|s_i), \tag{1.1}$$

where $P(s_i|s_j)$ is the probability of transition from state s_j in time t to s_i in time $t+1$ and c is a constant, set to a value such that $0 \leq K(s_i, s_j) \leq 2$).

Chi-squared distance (`FUTURE`):

$$K(s_i, s_j) = d_{\chi^2}(\mathbf{P_i}, \mathbf{P_j}), \tag{1.2}$$

where $\mathbf{P}_{\cdot} = (P(s_1|s_{\cdot}), \ldots, P(s_n|s_{\cdot}))'$

Relative frequencies (`INDELS` and `INDELSLOG`):

$$K(s_i, s_j) = d_i + d_j, \tag{1.3}$$

where the *indel* cost d_i depends on the state and takes values:

$$g_i = \frac{1}{f_i}, \qquad \text{for method 'INDEL'}, \tag{1.4}$$

$$g_i = \log(\frac{2}{1+f_i}), \qquad \text{for method 'INDELSLOG'} \tag{1.5}$$

and f_i is the relative frequency of the state s_i for $i = 1, \ldots, n$.

Remarks: - For methods `TRATE` and `FUTURE`, the unique *indel* value is $d = max_{1 \leq i,j \leq n} K(i,j)/2$, so that the cost of any transition is always lower or equal than deleting and inserting an element (or vice versa). - The Needleman-Wunsch algorithm with constant costs for mismatch is known as Levenshtein distance.

1.1.2 Example

Let us suppose that S is the alphabet, let $X = \{S, E, N, D\}$ and $Y = \{A, N, D\}$ be two sequences in $\mathbf{S}$. Supposing that $d = 2$ and

$$K(i,j) = \begin{cases} 0 & \text{if } i = j, \\ 3 & \text{otherwise} \end{cases}$$

The array F is initialized as follows:

		S	E	N	D
	0	2	4	6	8
A	2				
N	4				
D	6				

To fill the second row of F we proceed as follows:

- $F(2,2) = min\{F(1,2)+d, F(2,1)+d, F(1,1)+k(y_1,x_1)\} = min\{2+2, 2+2,0+3\} = 3$
- $F(2,3) = min\{F(1,3)+d, F(2,2)+d, F(1,2)+k(y_1,x_2)\} = min\{4+2, 3+2,2+3\} = 5$
- $F(2,4) = min\{F(1,4)+d, F(2,3)+d, F(1,3)+k(y_1,x_3)\} = min\{6+2, 5+2,4+3\} = 7$
- $F(2,5) = min\{F(1,5)+d, F(2,4)+d, F(1,4)+k(y_1,x_4)\} = min\{8+2, 7+2,6+3\} = 9$
- $F(3,2) = min\{F(2,2)+d, F(3,1)+d, F(2,1)+k(y_2,x_1)\} = min\{3+2, 4+2,2+3\} = 5$
- $F(3,3) = min\{F(2,3)+d, F(3,2)+d, F(2,2)+k(y_2,x_2)\} = min\{5+2, 5+2,3+3\} = 6$
- $F(3,4) = min\{F(2,4)+d, F(3,3)+d, F(2,3)+k(y_2,x_3)\} = min\{5+2, 5+2,5+0\} = 5$
- $F(3,5) = min\{F(2,4)+d, F(3,4)+d, F(2,4)+k(y_2,x_4)\} = min\{7+2, 5+2,5+3\} = 7$
- $F(4,2) = min\{F(3,2)+d, F(4,1)+d, F(3,1)+k(y_3,x_1)\} = min\{5+2, 6+2,4+3\} = 7$
- $F(4,3) = min\{F(3,3)+d, F(4,2)+d, F(3,2)+k(y_3,x_2)\} = min\{6+2, 7+2,5+3\} = 8$
- $F(4,4) = min\{F(3,4)+d, F(4,3)+d, F(3,3)+k(y_3,x_3)\} = min\{5+2, 8+2,5+3\} = 7$
- $F(4,5) = min\{F(3,5)+d, F(4,4)+d, F(3,4)+k(y_3,x_4)\} = min\{7+2, 7+2,5+0\} = 5$

		S	E	N	D
	0	2	4	6	8
A	2	3	5	7	9

		S	E	N	D
N	4	5	6	5	7
D	6	7	8	7	5

In this simple example, we can easily obtain two optimal (equivalent) alignments:
S E N D with
A - N D or
- A N D

In both cases we have two matches (cost 0), one mismatch (cost 3) and one gap (cost 2), giving a total cost 5 that is exactly what we obtained in the last cell of F.

The cost of inserting a gap (d) is also known as *indel* (insert or delete) cost. In this example we can observe that, in order to obtain sequence X from Y we have to **insert** a term (i.e. insert a gap and then change its value to a specific state). Equivalently, to obtain sequence Y starting from X we have to **delete** one term.

1.1.3 Normalization

In cases when the lengths of the sequences differ, it can be useful to account for this differences with a normalization factor.

Given a set two sequences $X, Y \in \mathbf{S}$ of length t_1 and t_2, respectively. Let $d(X, Y)$ be the distance between the sequences X and Y, t_{max} the length of the longest sequence in $\mathbf{S}$ and d_{max} the maximum distance between any pair of sequences in $\mathbf{S}$.

`TraMineR` offers the following options to normalize the distances between sequences:

- `maxlength`:
$$\frac{d(X,Y)}{t_{max}}$$
- `gmean`:
$$1 - \frac{d_{max} - d(X,Y)}{\sqrt{t_1 * t_2}}$$
- `maxdist`:
$$\frac{d(X,Y)}{d_{max}}$$

Chapter 2

Data from the 40+ Healthy Aging Study

2.1 About the data

As part of the Women 40+ Healthy Aging Study, a large study that was conducted by the Department of Clinical Psychology and Psychotherapy of the University of Zurich, a psychometric instrument was developed in order to obtain information about the history of romantic relationships of women. The study was conducted between June 2017 and February 2018 with women between 40 and 75 years who (self-)reported good, very good or excellent health condition and the absence of acute or chronic somatic disease or mental disorder. The participants who reported psychotherapy or psychopharmacological treatment in the previous 6 months were excluded as well as habitual drinkers. Other exclusion criteria were pregnancy in the last 6 months, premature menopause, surgical menopause, intake of hormonal treatment (including contraceptives), shift-work and recent long-distance flight. The participants were recruited from the general population using online advertisement and flyers.

The questionnaire asked the participants to provide information about relationship phases starting from the age of 15 years until the current age at the time of the data collection. The phases were defined by the start and end age and for each phase and information about civil status, relationship status, living situation, children and quality of the relationship was collected. Before including the data corresponding to their own history, the participants were prompted to answer some of the questions based on an example. Some of the participants were excluded when the example entries were not correctly filled. After data cleaning and revisions for consistency the

total number of individuals considered is 239.

In order to create a sequence for each participant the information about civil status, relationship status, living situation and the maternity is taken into account. A yearly sequence is created and the states considered are the following:

- 1 = Single + no children
- 2 = Single + children
- 3 = Changing relationships + no children
- 4 = Changing rel. + children
- 5 = Relationship + living apart + no children
- 6 = Relationship + living together + no children
- 7 = Relationship + living apart + children
- 8 = Relationship + living together + children
- 9 = Married + no children
- 10 = Married + children

Additionally, personality scores for the women included in the study are available. Personality refers to the enduring characteristics and behavior that comprise the unique adjustment to life of a person, including major traits, interests, drives, values, self-concept, abilities, and emotional patterns. These scores are obtained via psychometric instruments and evaluate the main personality traits:

- Agreeableness
- Conscientiousness
- Extraversion
- Neuroticism
- Openness

2.2 Application of OM

Using the `R` package `TraMineR` the cost matrix is calculated with transition rates between states. We consider a base setup with method `TRATE` for the calculation of the cost matrix and `maxlenght` normalization for the dissimilarities matrix. The obtained cost matrix is shown below. As expected, the elements in the diagonal are equal to 0, meaning there is no cost associated to staying in the same state. By default, the constant c in 1.1 is set to 2. This, and the fact that the duration of the states is often longer that the time unit (one year), makes that all of the values outside the diagonal are close to 2 and even equal in cases where no transition between the states were

Table 2.1: Cost matrix obtained from transition probabilities.

State	1	2	3	4	5	6	7	8	9	10	NA
1	0.00	2.00	1.98	2.00	1.92	1.95	2.00	1.99	1.98	1.98	2
2	2.00	0.00	2.00	2.00	2.00	2.00	1.96	1.92	2.00	2.00	2
3	1.98	2.00	0.00	2.00	1.94	1.92	2.00	2.00	1.97	1.98	2
4	2.00	2.00	2.00	0.00	1.99	2.00	1.95	1.95	2.00	2.00	2
5	1.92	2.00	1.94	1.99	0.00	1.95	1.98	1.99	1.98	1.97	2
6	1.95	2.00	1.92	2.00	1.95	0.00	2.00	2.00	1.98	1.96	2
7	2.00	1.96	2.00	1.95	1.98	2.00	0.00	1.97	2.00	2.00	2
8	1.99	1.92	2.00	1.95	1.99	2.00	1.97	0.00	2.00	1.98	2
9	1.98	2.00	1.97	2.00	1.98	1.98	2.00	2.00	0.00	1.99	2
10	1.98	2.00	1.98	2.00	1.97	1.96	2.00	1.98	1.99	0.00	2
NA	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	0

observed in the data (e.g. from single without children to single with children and vice versa). Finally, we observe that missing value (`NA`) is considered as a separate state and, by default, the cost of changing from or to a missing value is 2, which might be too high in cases where the individuals made a mistake in the beginning or end age of a phase leaving a gap in the sequence.

From this cost matrix it is possible to calculate pairwise distances between all the sequences using the algorithm described in section XX. As stated before, a correction of the distances is done to account for the differences in size of the sequences. This is done dividing the obtained distance by the length of the longest sequence.

Having obtained the distance matrix, we apply a hierarchical agglomerative clustering method in order to explore the data and the differences captured by the distance matrix. In particular, we set the number of clusters to 4 and the following figure shows the distribution of the states.

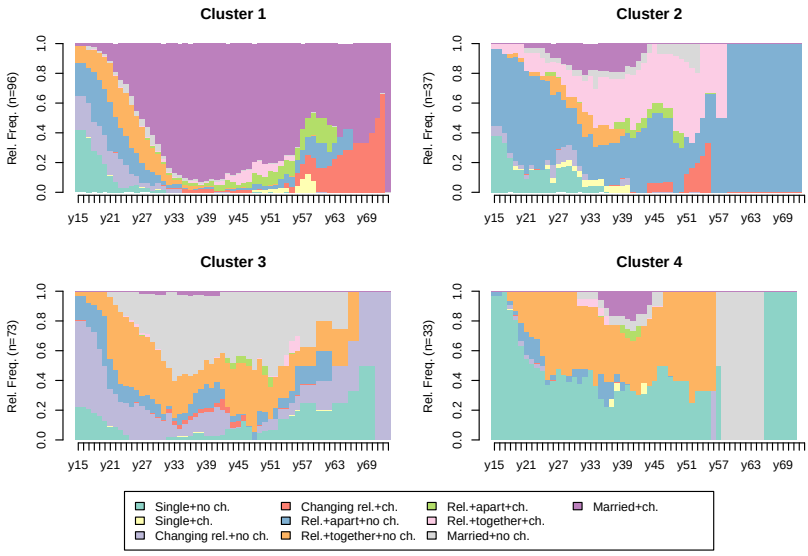

Figure 2.1: Distribution of states by cluster.

The figure below shows the transverse entropy by cluster, i.e. the cross-sectional entropy of the states distributions is calculated at each time point as follows:

$$h(f_1, \ldots, f_n) = -\sum_{i=1}^{n} f_i \log(f_i). \tag{2.1}$$

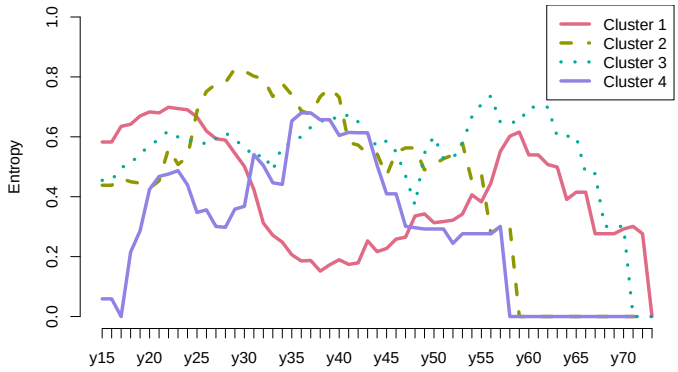

Figure 2.2: Transversal entropy by cluster.

The previous visualizations allow us to try to identify common and contrasting features of the clusters that can be useful to describe them. It is important to remember that this description is subjective and incomplete.

- Cluster 1: Married young and had children.
- Cluster 2: Often in relationships but not married.
- Cluster 3: Older, mostly married or in long relationship without children.

Table 2.2: Number of individuals by cluster.

Cluster	n
1	96
2	37
3	73
4	33

- Cluster 4: Younger, single or in a relationship without children.

On the other hand, in figure XX we can also appreciate that the conformation of some clusters seems to be highly affected by the length of the sequence and is possible that the normalization method is not achieving the expected result.

We are interested in exploring how the relationships history of the women relate to personality traits. As a first exploratory step, the following figure shows the distribution of the score for each trait by cluster.

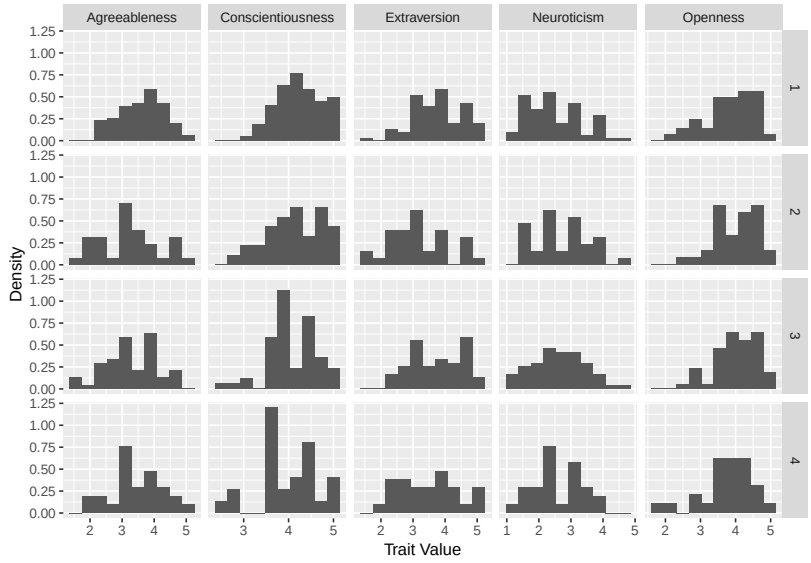

Figure 2.3: Distribution of personality scores by cluster.

No difference is obvious at first glance. Also, the number of clusters and the fact that the personality scores are not continuous makes it difficult to appreciate differences. For that reason, we also explore with a lower number of clusters.

Furthermore, we obtain better defined clusters that are less affected by the length of the sequences as we can observe in the distribution plots of the sequences states: the majority of women in cluster 1 have children, while we mostly find women without children in cluster 2.

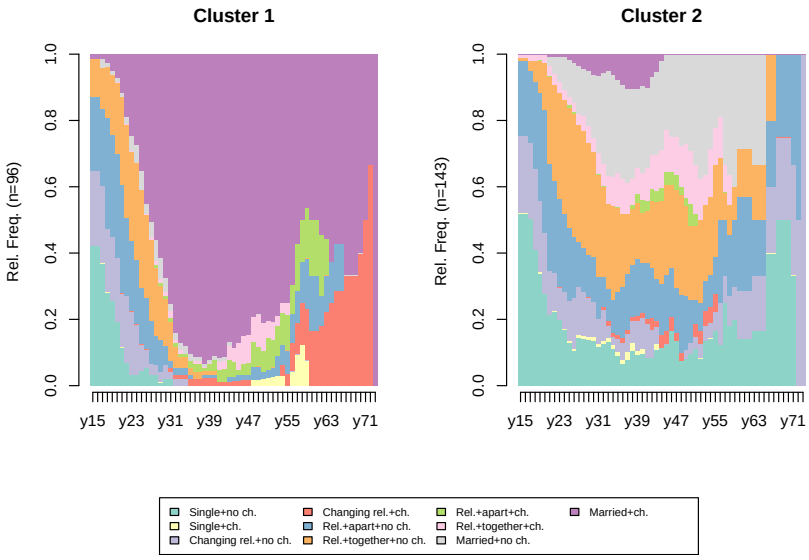

Figure 2.4: Distribution of states for two clusters.

In addition, the transversal entropy of the sequences for the two clusters is displayed in the figure below.

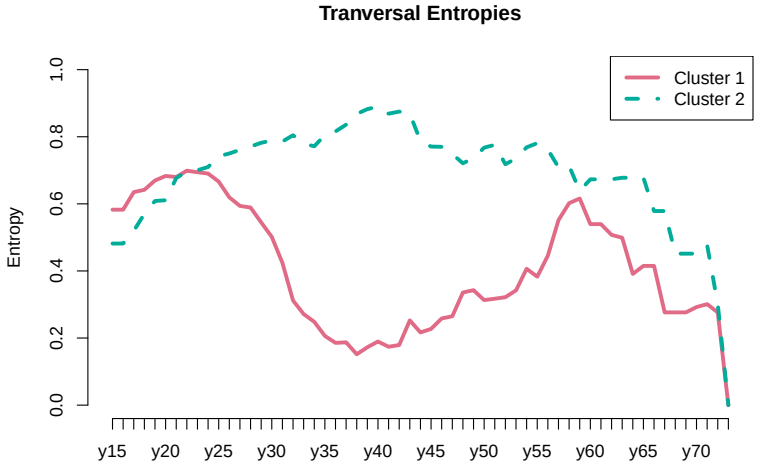

Figure 2.5: Transversal entropy for two clusters.

This figure shows that the entropy decreases significantly around mid age for the cluster of women with children as compared to women without children, which means that the variability of the states for the first group is much lower as compared to the second group. This can be interpreted as a sign of stability in the relationship status for women during the time they have children at home.

As before, we want to explore possible links between the information from the sequences and personality scores. The following figure shows the distribution of the personality traits for the two clusters.

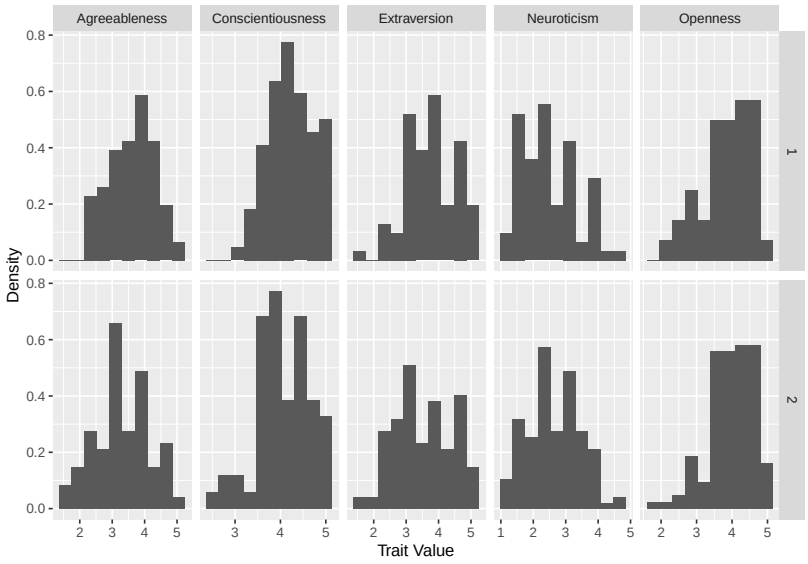

Figure 2.6: Distribution of personality scores for two clusters.

There seems to be differences in the distributions of some personality scores: the scores of agreeableness are concentrated in larger values for women with children; women without children have greater frequency in lower values of conscientiousness than women with children; and women with children exhibit lower scores of neuroticism.

Even though, the distribution of personality scores by cluster does not reveal significant differences, the obtention of a distance matrix also provide us a numerical expression of the categorical sequences that allows us to use it for other purposes. In particular, we explore the predictive capability of this data with a non-parametric prediction method in the next section.

Chapter 3

Personality Scores Prediction with k-Nearest Neighbors

Given a training set $\mathcal{D} = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ of n labeled data points, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathcal{Y}$ (a finite set of class labels for classification or a continuous range of values for regression). k-NN provides a way to predict the label or value for a new, data point x_{n+1} (for which Y is unknown) by finding the k training data points closest to x_{n+1} and taking a majority vote of their labels (for classification) or averaging the values of Y (for regression).

There are different ways of calculating the distance between the new data point x_{n+1} and the points in $\mathcal{D}$. For instance, the Euclidean o Mahalanobis distances are usually used. In our case we already count with a matrix distance obtained with OM.

The choice of k is a hyperparameter that can be tuned to optimize the performance of the k-NN algorithm. A larger k reduces the effect of noise and outliers, but can also lead to overfitting. A smaller k is more sensitive to noise and outliers, but can better capture local structure.

To compare the performance of different values of k, we use the mean squared error (MSE).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.1}$$

where y_i is the observed value and $\hat{Y}_i$ is the predicted value via kNN.

In this part of the analysis we only consider the individuals who have available personality scores, that leaves us with a sample size of 200 individuals. We also split the data into two subsets: train (70%) and test (30%). We evaluate the MSE of the predictions for the individuals in the test set but only using the data from the nearest neighbors available in the train set.

The following figure shows for every personality trait and different values of k the MSE, i.e. for $k = 1, \ldots, 80$ we predict values of Y and compare them with the observed values using the MSE. As a reference, a red line for every personality trait is added to indicate the MSE of the trivial prediction, i.e. the prediction considering all the sample points in the train set.

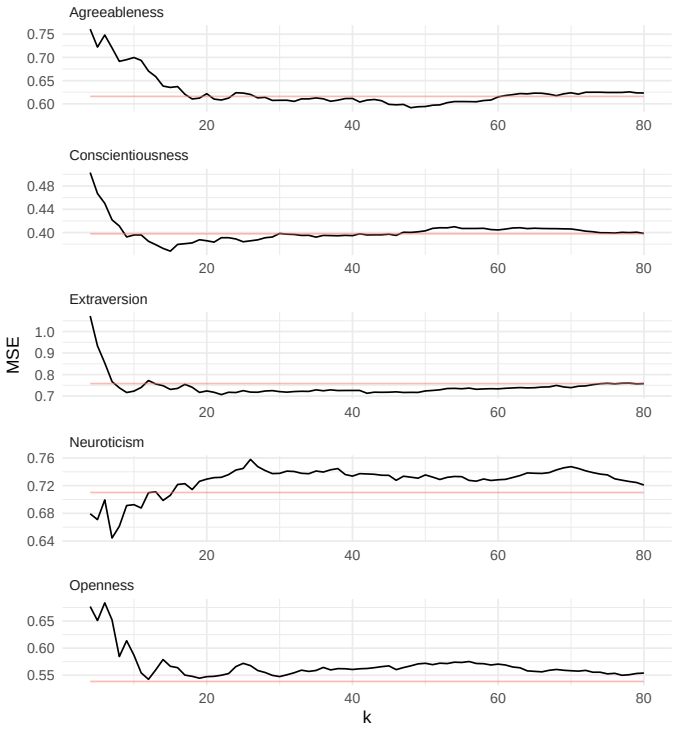

Figure 3.1: MSE by cluster for base setup.

Overall, it seems that using the sequential data for prediction results in little improvement compared to the trivial prediction except for neuroticism where the MSE takes a minimum value around $k = 5$.

Furthermore, for conscientiousness and openness, the MSE does not seem to increase again as k increases, which is expected when using kNN, due to overfitting. Moreover, for openness, the prediction with kNN is always worse than the trivial prediction. For conscientiousness, the MSE takes a minimum value around $k = 15$ and after $k = 30$ the MSE curve stays flat.

For agreeableness, the MSE is minimum around $k = 50$ and increases again. However, this minimum is not considerably lower than the trivial prediction. Similarly, for extraversion, the MSE takes a minimum value after $k = 20$, but is not a significant improvement compared to the trivial prediction.

Given that the performance of the predictions is just slightly better than aver-

age in most cases, we contemplate other scenarios with different variations of the hyperparameters considered in this section.