Measuring Dialect Pronunciation Differences using Levenshtein Distance

Heeringa, Wilbert Jan

Chapter 6

Analysing dialect distances

As mentioned in Chapter 5 we may use Levenshtein distance to find the distance between two pronunciations of the same word. The distance between two varieties is equal to the average of a sample of Levenshtein distances of corresponding word pairs. When we have n varieties, then the average Levenshtein distance is calculated for each possible pair of varieties. For n dialects $n \times n$ pairs can be formed. The corresponding distances are arranged in a $n \times n$ matrix which is comparable to distance tables published by auto clubs and often found in pocket calendars that show the distances between the main towns.

The distance at each variety with respect to itself is found in the distance matrix on the diagonal from upperleft to lowerright. These values are always zero and therefore give no real information, so that only $n \times (n-1)$ pairs are interesting. Furthermore, the Levenshtein distance is symmetric. This means that the distance between word 1 and word 2 is equal to the distance between word 2 and word 1. The result is that distance between variety 1 and variety 2 is equal to the distance between variety 2 and variety 1 as well. Therefore, the distance matrix is symmetric. We need to use only one half which contains the distances of $(n \times (n-1))/2$ word pairs.

To interpret the $(n \times (n-1))/2$ varieties, they can be visualized on a map. On the map each pair of points is connected by a line. Darker lines correspond to similar language varieties, lighter lines to more distant varieties. Very distant relations result in lines too faint to be seen (in the interest of overall contrast). An example of such a map can be found in Figure 9.4. The map shows distances between 360 Dutch varieties and is discussed in Section 9.2. On the map, dialect groups can already be distinguished to some extent. This way of visualizing is related to the beam maps of Séguy and Goebl (see Section 2.3.1). However, on the beam maps only neighboring points are connected while on our map all points are in principle connected showing all $(n \times (n-1))/2$ distances.

Another way of interpreting the distances is to examine the results of classification methods which are applied to the distances. Classification results show relations between elements in a way which is easy to understand. We used cluster

analysis and multidimensional scaling, two common techniques that complement each other. The result of cluster analysis is a dendrogram, a tree where the varieties are the leaves. The technique is described in Section 6.1. The result of multidimensional scaling is a map, where the distance between kindred varieties is small, and between different dialects great. This technique is explained in Section 6.2.

6.1 Cluster analysis

Jain and Dubes (1988, p. 55) define cluster analysis as 'the process of classifying objects into subsets that have meaning in the context of a particular problem.' The goal of clustering is to identify the main groups in complex data. In this section, we discuss a set of cluster methods that are referred to as SAHN (Sequential, Agglomerative, Hierarchical, Nonoverlapping) clustering methods by Sneath and Sokal (1973). Sequential means that the objects are processed one by one instead of simultaneously. Agglomerative procedures starts with placing each object in its own cluster and gradually merges smaller clusters in larger clusters until all objects are in one single cluster. A hierarchical classification is a nested sequence of partitions. Nonoverlapping means that for every split in the hierarchy each object belongs to exactly one cluster. SAHN clustering methods are suitable for classification of language varieties because they show both groupings and distances. The distances are reflected to some extent by the hierarchical structure.

6.1.1 Johnson's algorithm

The general scheme used for SAHN clustering is called Johnson's algorithm. Jain and Dubes (1988) mention that the scheme was suggested by King (1967) and formalized by Johnson (1967). We will demonstrate the algorithm by an example. Assume we get the following matrix, which shows the linguistic distances between some Dutch dialects[1]:

	Grouw	Haarlem	Delft	Hattem	Lochem
Grouw		42	44	46	47
Haarlem			16	36	38
Delft				38	40
Hattem					21
Lochem					

[1]In Chapter 9 linguistic distances between 360 Dutch dialects are calculated. Our small 5×5 table is a subtable of the large 360×360 table.

The value of each cell (i, i) is of course equal to 0 (the distance of a variety with respect to itself). Because the matrix is symmetric we do not need the distances in the left lower half.

Clustering with Johnson's algorithm is an iterative procedure. At each step of the procedure we select the shortest distance in the matrix, and then fuse the two data points which gave rise to it. Since we wish to iterate the procedure, we have to assign a distance from the newly formed cluster to all remaining points. To keep the example simple we calculate the distance from k to a newly formed cluster $[ij]$ as the mean of the distance between i and k and the distance between j and k. So for each k we calculate:

$$d_{k[ij]} = \frac{d_{ki} + d_{kj}}{2}$$

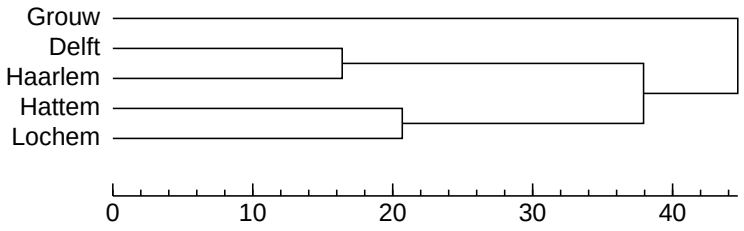
In the distance matrix the shortest distance is found between Haarlem and Delft. Both Haarlem and Delft are removed from the matrix, and a new cluster Haarlem & Delft is inserted. To iterate, we have to assign a distance from the newly formed cluster to all other points. For example, the distance between Grouw and Haarlem & Delft is calculated as follows:

$$\begin{aligned} d_{Grouw,\ [Haarlem\ \&\ Delft]} &= \frac{d_{Grouw,\ Haarlem} + d_{Grouw,\ Delft}}{2} \\ &= \frac{42 + 44}{2} \\ &= 43 \end{aligned}$$

After calculating the distances between Hattem and Haarlem & Delft and between Lochem and Haarlem & Delft as well, we get the following matrix (new values are in bold type):

	Grouw	Haarlem & Delft	Hattem	Lochem
Grouw		**43**	46	47
Haarlem & Delft			**37**	**39**
Hattem				21
Lochem				

In each iteration the matrix is reduced in size. The iterations are repeated until no elements are left which can be fused to a new cluster. The final result is a complete hierarchical grouping of varieties. This grouping is visualized as a dendrogram, a tree in which the leaves are the varieties and the lengths of the branches correspond with the distances. In our example we get the following dendrogram:



In Figure 6.1 Johnson's algorithm is given in pseudo-code. On the basis of n elements $n-1$ clusters are obtained. The elements are numbered from 1 to n, and the clusters from $n+1$ to $n+n-1$. Therefore the variable k that gives the cluster index is set to the number of elements initially. The input of the procedure is $DistanceMatrix$ which contains the distances. The output is $Cluster$, containing for each of the $n-1$ clusters its subclusters and the distance between both subclusters. On the basis of $Cluster$ the dendrogram is constructed.

```
procedure cluster(DistanceMatrix,Cluster);
begin
  k:=number of elements;

  while elements or clusters are left that can be fused do begin
    k:=k+1;

    find pair (i,j) in DistanceMatrix that has smallest distance;
    store subclusters i and j in Cluster[k];
    distance between subclusters of Cluster[k]:=distance between i and j;

    delete rows and columns of i and j in DistanceMatrix;
    insert a row and a column of cluster k in the DistanceMatrix;
    calculate distances from cluster k to all remaining points;
  end;
end
```

Figure 6.1: Johnson's algorithm in pseudo-code.

6.1.2 Matrix updating algorithms

Each time two clusters are fused to a new cluster, the distances from the newly formed cluster to all other points (or clusters) need to be calculated. In our example, the distance from a new cluster ij to point k was calculated as the mean of the distance between i and k and the distance between j and k. The way in which the distances between a newly formed cluster and the remaining points is calculated is called a *matrix updating algorithm*. Sneath and Sokal

(1973, pp. 218–219) mention six matrix updating algorithms. Jain and Dubes (1988, p. 80) mention the same updating algorithms and added a seventh, Ward's method.

Assume points (or clusters) i and j are fused to one cluster ij. Then for calculating the distance from cluster ij to a point (or cluster) k the following data are (partly) needed: n_i (number of varieties in cluster i), n_j (number of varieties in cluster j), n_k (number of varieties in cluster k), d_{ij} (distance between i and j), d_{ki} (distance between k and i) and d_{kj} (distance between k and j). Now the seven matrix updating algorithms are defined as follows:

1. Single-link (nearest neighbor):

$$d_{k[ij]} = minimum(d_{ki}, d_{kj})$$

2. Complete-link (furthest neighbor):

$$d_{k[ij]} = maximum(d_{ki}, d_{kj})$$

3. Unweighted Pair Group Method using Arithmetic averages (UPGMA):

$$\begin{array}{rcccccc} d_{k[ij]} = & (n_i & / & (n_i + n_j)) & \times & d_{ki} & + \\ & (n_j & / & (n_i + n_j)) & \times & d_{kj} & \end{array}$$

4. Weighted Pair Group Method using Arithmetic averages (WPGMA):

$$d_{k[ij]} = (\tfrac{1}{2} \times d_{ki}) + (\tfrac{1}{2} \times d_{kj})$$

5. Unweighted Pair Group Method using Centroids (UPGMC):

$$\begin{array}{rcccccc} d_{k[ij]} = & (n_i & / & (n_i + n_j)) & \times & d_{ki} & + \\ & (n_j & / & (n_i + n_j)) & \times & d_{kj} & - \\ & ((n_i \times n_j) & / & (n_i + n_j)^2) & \times & d_{ij} & \end{array}$$

6. Weighted Pair Group Method using Centroids (WPGMC):

$$d_{k[ij]} = (\tfrac{1}{2} \times d_{ki}) + (\tfrac{1}{2} \times d_{kj}) - (\tfrac{1}{4} \times d_{ij})$$

7. Ward's method (minimum variance):

$$\begin{array}{rcccccc} d_{k[ij]} = & ((n_k + n_i) & / & (n_k + n_i + n_j)) & \times & d_{ki} & + \\ & ((n_k + n_j) & / & (n_k + n_i + n_j)) & \times & d_{kj} & - \\ & (n_k & / & (n_k + n_i + n_j)) & \times & d_{ij} & \end{array}$$

When finding the distance between a new cluster and an existing cluster, single-link finds the closest pair of elements in the two clusters and complete-link the most distant pair. UPGMA and WPGMA assess the dissimilarity between the new cluster and the existing cluster by the distance between the means. Instead of using means UPGMC and WPGMC use centroids, i.e. the hypothetical points at the centers of clusters i, j and k. Ward's method assigns a new distance in the way that results in the smallest increase in the within-cluster sum of squares, i.e., the sum of the squared distances between each point and the resultant cluster centroid (Wilks, 1995).

Note the use of the terms *unweighted* and *weighted*. *Weighted clustering* was introduced by Sokal and Michener (1958). In this approach clusters that merge get equal weights regardless of the number of elements in each cluster. In that case elements in small clusters are weighted relatively more heavily than elements in larger clusters. In *Unweighted clustering* each element in a cluster gets equal weight, regardless of the number of elements in that cluster (Sneath and Sokal, 1973, p. 228). Although this terminology was adopted by Jain and Dubes (1988) and others, it may be confusing since in unweighted clustering we weight clusters the merge by their size, and in weighted clustering we do not.

6.1.3 Experimentation

In our research, we have to decide which matrix updating algorithm should be used. Because both single-link and complete-link take only one cluster into account when merging two clusters, we did not use them. Weighting clusters by their size when fusing them seems more reasonable to us than weighting them equally, so we prefer unweighted clustering. Using the centroid-based methods, we sometimes get results in which the distance between two clusters is smaller than between the subclusters in (one of) the two clusters. When comparing dialects such results are not natural. So only two matrix updating algorithm are left: UPGMA and Ward's method.

Wilks (1995) indicates that Ward's method tends to create clusters of equal size, which is not always reasonable. In our research we found that varieties which appear as outliers in a UPGMA dendrogram are neatly ordered under a group of moderate size in a dendrogram obtained by Ward's method. We will compare on the basis of the following matrix:

(6.1)

	a	b	c	d
a		1	2	2
b			2	2
c				?
d				

We applied both UPGMA and Ward's method to the matrix, and experimented with different values for the '?', the distance between objects c and d. We found that for the values 0 through 1.9 (with a step size of 0.1) similar dendrograms are obtained. Setting the distance between objects c and d to 2.0, dendrograms are obtained which show that objects c and d are equally distant to the cluster containing objects a and b. However, in the dendrogram generated by the Ward's method objects c and d form a cluster. For values varying from 2.1 through 2.3 (with a step size of 0.1 again), in the UPGMA dendrogram object d is further apart from objects a and b than object c. In the dendrogram generated by Ward's method object c and d still form one cluster, which is unexpected and counterintuitive. For values 2.4 and higher the two methods will give similar results. Results for the values 1.0, 2.0, 2.2 and 2.4 are found in Figure 6.2.

A useful quantitative method for validating cluster results is developed by Sokal and Rohlf (1962). They proposed to calculate the *cophenetic correlation coefficient*, which is a measure of the agreement between the distances as implied by the dendrogram and those of the original distance matrix. This approach is also described in Sneath and Sokal (1973, pp. 277–284) and Jain and Dubes (1988, pp. 66–68 and 166–170). The *cophenetic correlation coefficient*, abbreviated as CPCC by Farris (1969), measures the correlation between the original distances and the cophenetic distances. Because dialect distances are numeric data, we used the Pearson correlation coefficient (see Section 3.6.2.5). Cophenetic values are the distances as suggested by the dendrogram. For finding the cophenetic distance between objects i and j we have to find the least significant (smallest) cluster in which both objects are first present. The cophenetic distance between i and j is equal to the distance between the subclusters of this cluster. Once we have a correlation coefficient, we can calculate to what extent the cophenetic distances explain the variance in the original distances. The variance is found by taking the square of the cophenetic correlation coefficient. The variance is expressed as a percentage when multiplied by 100. In this thesis this variance is given as a percentage for most dendrograms.

Using the CPCC we examined the difference between UPGMA and Ward's method further. For the '?' in the matrix (the distance between c and d) the values 0.0 through 3.0 are filled in with a step size of 0.1, and for each value the CPCC is calculated for both UPGMA and Ward's method. In Figure 6.3 the CPCC is plotted against *distance*(c,d). From 0.0 through 2.0 we see that the CPCC of the UPGMA is equal to 1.00. For values higher than 2.0 the CPCC decreases. For Ward's method the CPCC increases from 0.0 through 1.0, decreases from 1.0 through 2.1, increases from 2.1 through 2.3, and increases from 2.3. UPGMA and Ward's method are equal for *distance*(c,d)=1 where they have both a CPCC of 1.00. The greatest difference between UPGMA and Ward' method is found for *distance*(c,d)=2.0, where UPGMA has CPCC=1.00 and Ward's method CPCC=0.9439. From 2.1 through 2.3 Ward's method gives counterintuitive results, suggesting that c and d form a group which is not justified. For values 2.4

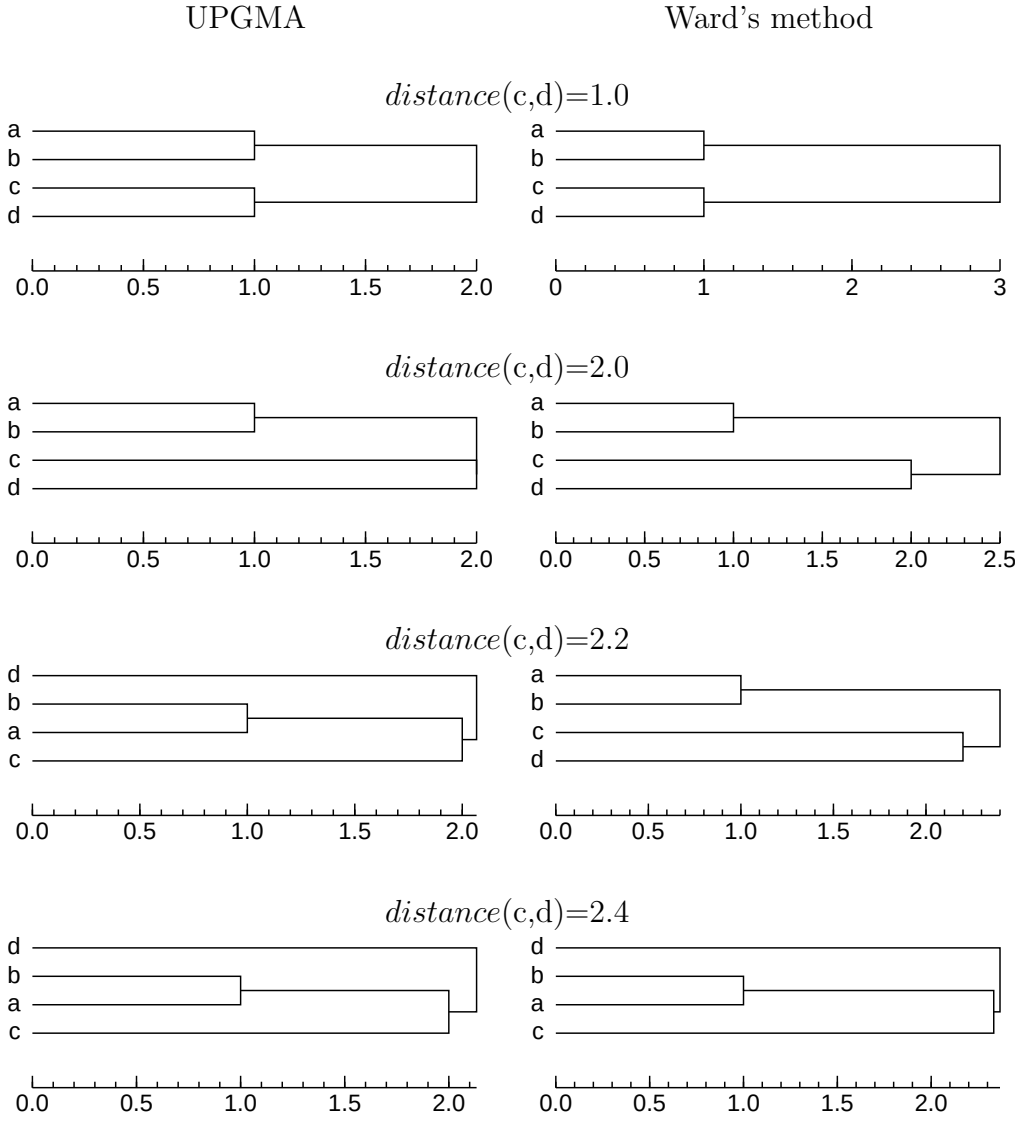


Figure 6.2: Results obtained from the matrix given in (6.1) using UPGMA and the Ward's method. In the matrix different values have been used as distance between objects c and d. When the distance between c and d is equal to 2.2, in the UPGMA dendrogram object d is further from objects a and b than object c. In the dendrogram generated by Ward's method object c and d form one cluster, which is perhaps counterintuitive with regard to seeking dialectological groups.

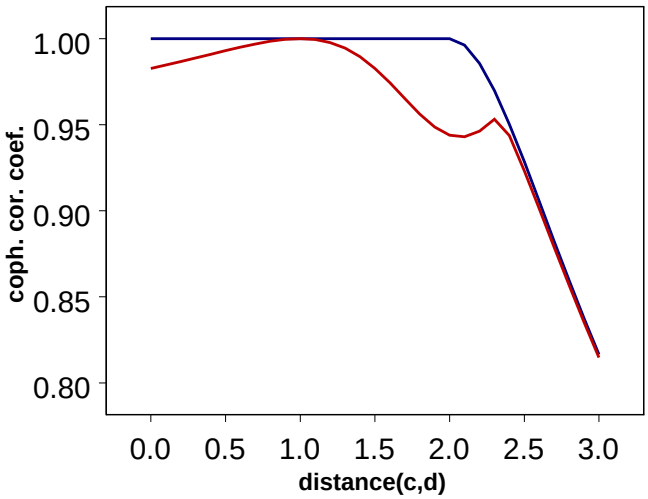


Figure 6.3: Comparison of UPGMA (upper line) and Ward's method (lower line) for the matrix given in (6.1) for $0 \leq$ *distance*(c,d) ≤ 3.0 with a step size of 0.1. To obtain a clear graph, points are connected by lines. For *distance*(c,d)=1.0 both methods have CPCC=1.00. Greatest difference was found for *distance*(c,d)=2.0, where UPGMA has CPCC=1.00 and Ward's method CPCC=0.9439. From 2.1 through 2.3 Ward's method gives counterintuitive results which is reflected in this graph by lower CPCC's with respect to the CPCC's of UPGMA.

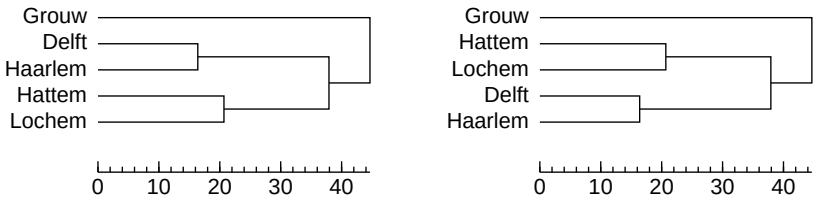
and higher both UPGMA and Ward's method run parallel, where for both the CPCC decreases. The UPGMA still performs better than the Ward's method. Only when *distance*(c,d)=16, both methods have the same CPCC when using four decimals.

The graph shows that the cophenetic correlation coefficient gives results which accord with our findings at the beginning of this section. On the basis of this we tend to judge UPGMA preferable to Ward's method. Therefore, we will use UPGMA troughout the rest of this thesis. We are aware that more situations are possible which should be tested. However, it is beyond the scope of this thesis to find and discuss them all.

6.1.4 Ordering of clusters

In Section 6.1.1 we show a dendrogram on the basis of five varieties. Below the dendrogram on the left is the same dendrogram as shown in Section 6.1.1, and the dendrogram on the right is an alternative. In the right one the clusters Delft & Haarlem and Hattem & Lochem are swapped. In the dendrogram Delft

and Haarlem can be swapped in the same way as Hattem and Lochem. Thus the same clustering can be visualized by different dendrograms. The question arises which of them is better. Examining the two dendrograms below we prefer the left one since the distance between Grouw and the cluster Delft & Haarlem ($(42+44)/2 = 43$, see the distance matrices in Section 6.1.1) is smaller than the distance between Grouw and the cluster Hattem & Lochem ($(46+47)/2 = 46.5$). In our implementation of the graphic display of the cluster algorithm the branches are ordered so that the more related varieties or clusters are located near each other in the dendrogram.



Assume we have a clustering that contains two clusters. The first cluster contains subclusters a and b (but a and b may also be leaves), and the second cluster contains subclusters c and d (c and d may again be leaves). Figure 6.4 shows that the clustering can be visualized in four different ways. So we have to decide which one is better. For this purpose we examine the distance between a, b, c and d and mirror one or both subclusters if necessary. Assume the subcluster which contains a and b is called i, the subcluster containing c and d is called j and the (sub)cluster containing i and j is called C. Now the procedure is as follows:

```
if minimum(b,c,C)
  then {nothing}
  else if minimum(a,c,C)
         then mirror subgroup i
         else if minimum(b,d,C)
                then mirror subgroup j
                else if minimum(a,d,C)
                       then mirror subgroup i and mirror subgroup j
```

The function *minimum* checks whether the distance for the pair of given elements is smaller than for each other possible pair of elements in (sub)cluster C. The result of this procedure is a dendrogram in which closest clusters (or terminals) are located near each other. This procedure was applied to the dendrograms in this thesis.

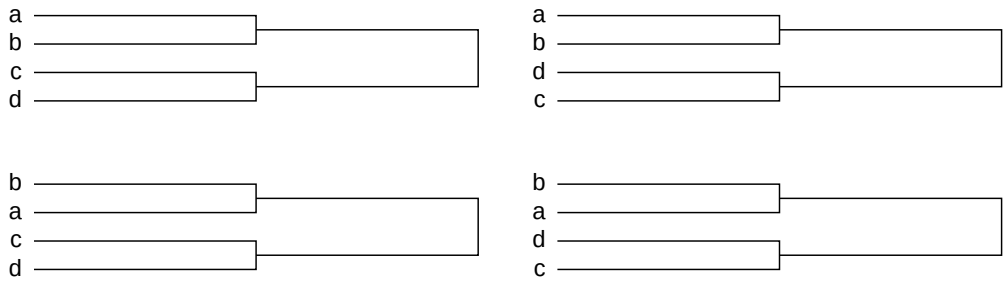


Figure 6.4: Four dendrograms, each of which is generated on the basis of the same distances using the same cluster method. We prefer the top left visualization in cases where $d(b,c) < d(a,c)$, $d(b,c) < d(b,d)$ and $d(b,c) < d(a,d)$.

6.1.5 Application

In Section 6.1.3 we mentioned that we use the UPGMA clustering method throughout this thesis. Examples of dendrograms can be found in Figures 8.3 and 9.5. The dendrogram in Figure 8.3 is based on distances between 55 Norwegian varieties, and the dendrogram in Figure 9.5 is based on 360 Dutch varieties. The two dendrograms are discussed in respectively Section 8.2.1 and Section 9.3.1.

For a dendrogram with n varieties the k most significant groups can be found, where $1 \leq k \leq n-1$. The choice of a suitable value for k depends on the number of varieties. When each group of the partition gets a unique color, the groups can be identified on a map. On such a map the most important groups in the dendrogram can easily be found. When neighboring points belong to different groups, the exact border between the points is found on the basis of triangulation. With this technique the two points are blown up to small areas until they touch each other (see Section 6.2.4). Figures 8.4 and 9.6 show maps based on the dendrograms in respectively Figure 8.3 and Figure 9.5. The two maps are discussed in respectively Section 8.2.2 and 9.3.2.

The groups as found in a dendrogram can also be represented geographically by a *composite cluster map*.[2] On this type of map groups are separated by borders which are represented by lines. Darker lines separate distant groups, lighter lines more similar groups. When creating this map, in the first step the border between the two most significant groups is drawn. In the second step, two borders are drawn which separate the three most significant groups. The first border, which was already drawn in the previous step, is drawn again, resulting in a darker color. The second border is drawn for the first time, so it will be lighter than the other one. In the i-th step, the $i-1$ borders of the i most

[2]Composite cluster maps were introduced by Peter Kleiweg, see also `http://www.let.rug.nl/~kleiweg/ccmap/`.

significant groups are drawn again (they get darker), and a new border is added. If the cluster contains n varieties, we start with drawing borders which separate the 2 most signficant groups, and end with drawing borders which separate the n most significant groups. Figure 9.7 shows a composite cluster map based on the dendrogram in Figure 9.5. The map is discussed in Section 9.3.3.

Comparing the color area map with the composite cluster map, the benefit of the composite cluster map is that the weigth of borders between groups is visualized. On the other hand, composite cluster maps have the disadvantage that they cannot show that varieties which are geographically separated by varieties of other groups, belong to the same group. In the color area map varieties of the same group get simply the same color.

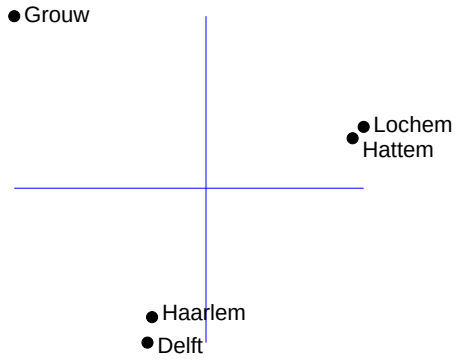
6.2 Multidimensional scaling

On the basis of geographic coordinates, the distances between locations can be determined. The reverse is also possible: on the basis of the known distances, an optimal coordinate system can be determined with the coordinates of the locations in it. The latter is realized by a technique known as 'multidimensional scaling' (MDS). In a multidimensional scaling plot, strongly related dialects are close to each other, while strongly different dialects are located far away from each other. MDS has its origins in psychometrics. Different persons are judged as similar if they tend to give similar responses to the same stimuli. MDS helps to understand the results of similar experiments (Oh and Raftery, 2001). Togerson (1952) proposed the first MDS method and coined the term.

6.2.1 Basic idea

The purpose of multidimensional scaling (MDS) is to provide a visual representation of the pattern of distances among a set of elements. On the basis of distances between a set of elements a set of points is returned so that the distances between the points are approximately equal to the original distances. The result is that on the plot like concepts are plotted nearby and unlike concepts are distant.

In Section 6.1 we use a small distance matrix which contains the distances between five varieties. On the basis of these distances with multidimensional scaling the varieties are plotted on a map, where the distances between the elements reflect the original distances as close as possible. This gives the following result:



In the original distance matrix small distances were found between Haarlem and Delft and between Hattem and Lochem. In the MDS plot, Haarlem and Delft, and Hattem and Lochem appear as two close clusters. The original distance matrix shows large distances between Grouw and the four other dialects, between Haarlem and Hattem, Haarlem and Lochem, Delft and Hattem and Delft and Lochem. These large distances are clearly reflected in the MDS plot. The x-axis represents the first dimension, and the y-axis the second dimension. If required the axes may be swapped. MDS values may also be used inversely. Both swapping axes and using values inversely are allowed since they do no change the distances between the elements on the plot.

6.2.2 Algorithms

Having three elements a, b and c, it is not difficult to place them in two-dimensional space so that the distances between them are correctly rendered. First a and b are placed with the right distance between them, and next c is placed so that is has the right distance with respect to both a and b. However, when adding a fourth element d, it is more difficult and it may be impossible to locate it so that the distances with respect to a, b and c are reflected perfectly. MDS assigns coordinates so that the Euclidean distances between the assigned points reflect the original distances as closely as possible. Normally, MDS is used to scale to two dimensions since three or more dimensions are difficult to display on paper. However, MDS can also be used to scale to three or more dimensions. In our research we used MDS routines as implemented in the statistical R package.[3] The program provides three MDS procedures: Classical Multidimensional Scaling, Kruskal's Non-metric Multidimensional Scaling and Sammon's Non-Linear Mapping.

As mentioned above, Togerson (1952) proposed the first MDS method which is known as Classical Multidimensional Scaling. The method is also described in

[3]The program R is a free public domain program and available via `http://www.r-project.org/`.

Togerson (1958) and is a *metric* procedure. The MDS plot in the example above is obtained on the basis of this procedure.

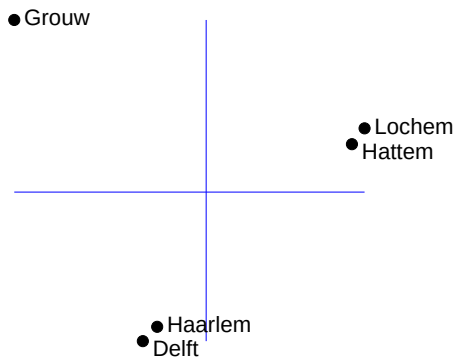
Both Kruskal's Non-metric Multidimensional Scaling and Sammon's Non-Linear Mapping are *non-metric* procedures, i.e. the ranks of the distances are used. Kruskal's Non-metric Multidimensional Scaling was the first non-metric multidimensional scaling procedure. In Shepard (1962), Kruskal (1964) and Kruskal and Wish (1978) the procedure is explained. Sammon's Non-Linear Mapping is described by Sammon (1969). In both procedures the MDS coordinates are found by an iterative algorithm. The algorithm starts with an initial configuration. Usually random values are assigned to coordinates of each of the elements. In R, however, the initial coordinates are found with Classical Multidimensional Scaling. Next, a range of steps is repeated until the optimal coordinates are found. First, the Euclidean distances between the elements on the basis of their coordinates are calculated. These distances are compared to the original distances using a *stress* function. The smaller the stress value, the closer the correspondence. The function is discussed below. Next, the coordinates are adjusted to reduce the stress. The most optimal coordinates are found when the stress can drop no further.

As mentioned above the degree of correspondence between the Euclidean distances between the MDS coordinates and the original distances is measured by a *stress* function. Assume d_{ij} is the original distance between elements i and j, and D_{ij} is the Euclidean distance between elements i and j as found on the basis of the coordinates. When using Kruskal's Non-metric Multidimensional Scaling the stress is calculated with the following formula:

$$STRESS = \sqrt{\frac{\sum_{i<j}(f(d_{ij}) - D_{ij})^2}{\sum_{i<j} {D_{ij}}^2}} \tag{6.2}$$

In this formula, $f(d_{ij})$ is a weakly monotonic transformation of the original distances. The function maps the original distances to values that best preserve the rank order. The transformation is found via *monotonic regression* (Jain and Dubes, 1988, pp. 49–50). Monotone regression is a step-function which is constrained to always increase from left to right. First, a monotonic transformation of the original distances is performed. Next, linear regression is applied to these transformed original distances and the coordinate-based distances. Subsequently, on the basis of the regression formula, original distances are predicted on the basis of the coordinate-based distances. In the formula, such a predicted value is noted as $f(d_{ij})$. Using monotone regression, the correlation between $f(d_{ij})$ and D_{ij} will be maximized. Monotone regression is also known as *isotonic regression*. Therefore, in R, Kruskal's Non-metric Multidimensional Scaling is also known as isoMDS. The denominator of the fraction is a constant scaling factor that assures

that stress values are between 0 and 1. Applying Kruskal's Non-metric Multidimensional Scaling to the distances of the five dialects in our example we get the following plot:

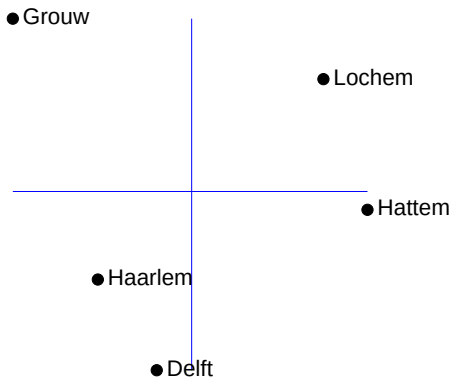


The x-axis represents the first dimension, and the y-axis the second dimension. Compared to the plot obtained on the basis of Classical Multidimensional Scaling the distance between Haarlem and Delft has become a bit smaller, and the distance between Hattem and Lochem has become a bit greater. Otherwise the two plots are very similar.

When using Sammon's Non-Linear Mapping another stress function is used. The formula is:

$$STRESS = \frac{\sum_{i<j} \frac{(d_{ij} - D_{ij})^2}{d_{ij}}}{\sum_{i<j} d_{ij}} \tag{6.3}$$

The main difference with stress in Kruskal's Non-metric Multidimensional Scaling is that the squared differences between the original distances and the coordinate-based distances are weighted by the original distances. Because of this normalization the preservation of small distances will be emphasized. The whole sum in the numerator is divided by the sum of the original distances in the denominator in order to scale the stress to a value between 0 and 1. The following plot show the result of Sammon's Non-Linear Mapping when applied to the distances of the five varieties in our example:



The x-axis represents the first dimension, and the y-axis the second dimension. When comparing the plot to the plots obtained by Classical Multidimensional Scaling and Kruskal's Non-metric Multidimensional Scaling we see that the distance between Haarlem and Delft and between Hattem and Lochem has become relatively much larger, although still two clusters can be recognized. The plot shows clearly that small distances are emphasized and greater distances are weakened. Using a larger set of points it appears that groups which are sharply distinguished on plots obtained by Classical Multidimensional Scaling and Kruskal's Non-metric Multidimensional Scaling form a continuum on a plot obtained by Sammon's Non-Linear Mapping.

6.2.3 Experimentation

R provides us with stress values only for Kruskal's Non-metric Multidimensional Scaling and Sammon's Non-Linear Mapping. However, stress values calculated with different formulas are not comparable. In order to compare the results of the different MDS procedures, we need a measure of fitness which can applied to each MDS result. This was found in ALSCAL, an MDS program using the alternating least square algorithm. A description of the algorithm is given by Takane et al. (1977) and Norušis (1997).[4] In ALSCAL the squared Pearson's correlation coefficient is calculated between the original distances and the Euclidean MDS coordinate-based distances. A higher correlation coefficient indicates that the multidimensional scaling values are a good representation of the original distances. The square of this correlation coefficient is equal to the variance of the original distances as explained by the chosen number of dimensions. We extended the R procedure so that Pearson's correlation coefficient r between the original distances and the final Euclidean distances on the plot is calculated by default. On the basis of this correlation coefficient the r^2 value was calculated and given for each plot in this thesis.

[4]The ALSCAL program is a free public domain program and available via: `http://forrest.psych.unc.edu/research/alscal.html`. ALSCAL is also included in the statistical package SPSS.

The r^2 value may help us to decide which of the MDS procedures available in R should be used. Calculating the r and r^2 values for the three MDS plots of our example, we get the following outcomes:

Method	r	r^2
Classical Multidimensional Scaling	0.989	97.7%
Kruskal's Non-metric Multidimensional Scaling	0.990	98.0%
Sammon's Non-Linear Mapping	0.935	87.4%

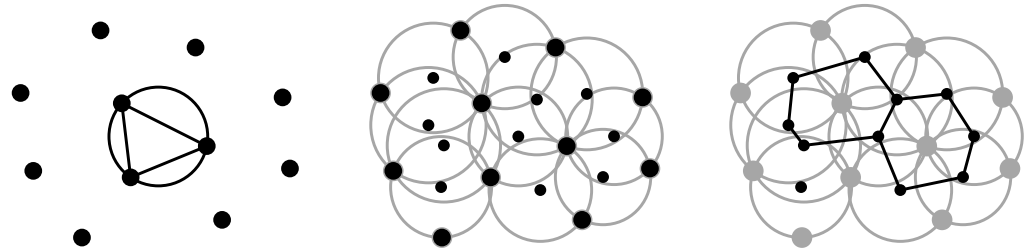
We used a three-digit precision to distinguish the values from each other. All correlation coefficients are significant, but none is significantly higher than the others.[5] Because the data set of our example contains only five varieties, no firm conclusions can be drawn. However, when applying the procedures to other data sets, in general Kruskal's Non-metric Multidimensional Scaling gets the highest r and r^2 value. Therefore, we will use this procedure throughout this thesis. Note that the non-metric Kruskal MDS performs better than the metric classical MDS as well although all distances we measured (between segments or between dialects) are metric data. We cannot explain this. Sammon's Non-Linear Mapping sometimes outperforms Classical Multidimensional Scaling, but the opposite was also observed many times.

6.2.4 Application

In Section 6.2.3 we mentioned that we use Kruskal's Non-metric Multidimensional throughout this thesis. An example of a three-dimensional multidimensional scaling plot can be found in Figure 8.5. The plot is based on distances between 55 Norwegian varieties. The plot is discussed in Section 8.3.1. On the basis of three dimensions of a three-dimensional solution, each variety can be represented by a color. If we let the first dimension be the intensity of red, the second the intensity of green and the third the intensity of blue, each variety gets an unique color. This approach can be used to create a dialect map. Each dialect point gets a color according to its MDS values. Colors can be assigned to the MDS dimensions so that a color scheme is obtained that is as similar as possible to existing dialect maps. Space between points can be colored in two ways. In the first approach dialect points are blown up to small areas. The areas are found by using the *Delaunay triangulation* (Krämer, 1995). Triangles connect points so that the circumcircle (circle that passes through all three points) does not contain any other point (see left picture below). For each circle that connects the three points of a triangle the center can be found (see below the picture in the middle). The centers of circles corresponding with adjacent triangles are connected. In this

[5]For finding significances we used the Mantel test which is explained in Section 3.8.2. As significance level we choose $\alpha = 0.05$.

way a pattern of polygons arises known as *Voronoi polygons*, *Thiessen polygons* or *Dirichlet tessellation* (see right picture below). The same technique for finding polygons is also used by Goebl (1982) (see also Goebl (1993) and Figure 2.1 and 2.2).



Sometimes MDS plots clearly show that varieties are language islands in the continuum. In that case we do not derive a Voronoi cell from these varieties. On the map they are marked with a diamond, where only the diamond is colored on the basis of the MDS values. Varieties which fit in the continuum are marked with a black dot. An example of such a map is found in Figure 9.32. The map is based on MDS values of a three-dimensional solution which was obtained on the basis of distances between 360 Dutch varieties. The map is discussed in Section 9.5.2.

In the second approach space between points is colored by interpolation. Assume point a is yellow and point b is blue. When no other points are located between points a and b, an unknown point exactly in the middle of both points will be green. An unknown point closer to point a will be more yellow and a point closer to point b will be more blue. In our research we used the most simple interpolation procedure, which is known as *inverse distance weighting*. Assume we have a map with n points. Each point has geographic coordinates (x_i, y_i) and MDS coordinates (R_i, G_i, B_i) where $1 \leq i \leq n$. Now for an intermediate point with geographic coordinates (x_p, y_p) we want to calculate MDS coordinates (R_p, G_p, B_p) where $1 \leq p \leq m$ and m is the number of intermediate points. These points form a regular grid over the area. Obviously, m determines the density. A higher m will result in a map on which colors more gradually change. R_p is found as follows:

$$R_p = \frac{\sum_{i=1}^{n} R_i \times \frac{1}{\delta(i,p)^s}}{\sum_{i=1}^{n} \frac{1}{\delta(i,p)^s}} \tag{6.4}$$

$$\text{where } \delta(i,p) = (x_i - x_p)^2 + (y_i - y_p)^2 \text{ and } s = 2.$$

Coordinates G_p and B_p are found in an analogous way. When $s = 0.5$, then $\delta(i,p)$ is just the Euclidean distance between i and p. Higher values for s give higher

color contrasts. In our research we used $s = 2$ which results in color contrasts which are strong enough to be seen on the one hand, and realistic on the other hand.

Just as when applying triangulation dialect islands are excluded from interpolation. As they would be in triangulation they are marked with a diamond on the map, where only the diamond is colored on the basis of the MDS values. Varieties which fit in the continuum are marked with a black dot. The color of the space immediately around this dot will nearly reflect the color of the variety itself. Examples of this type of map are given in Figures 8.6 and 9.33. Figure 8.6 is based on the MDS values which are represented by the multidimensional scaling plot in Figure 8.5. The map is discussed in Section 8.3.2. Figure 9.33 is based on the same MDS values as the map in Figure 9.32 and described in Section 9.5.2.

Triangulation takes less computation time than interpolation. When the network of data points has a high density, interpolation may hardly be needed. On the other hand, interpolation does justice to the idea that the dialect landscape may be regarded as a continuum.