2

A Vector-based Approach to Dialectometry

Erhard Hinrichs and Thomas Zastrow
University of Tübingen

Abstract

A novel unsupervised learning approach to computational dialectometry is presented which uses hard clustering. The approach relies on vector analysis over two-dimensional arrays of word lists collected for different geographical sites. The paper presents the underlying theory and applies the approach to a Bulgarian data set. The results of these experiments demonstrate the viability of the approach.

2.1 Computational Dialectometry

The study of language variation and of language change has a long and venerable tradition in linguistics. Traditional dialectology deals with the identification of dialect boundaries on the basis of historical evidence and on the basis of bundles of characteristic isoglosses. Relevant historical evidence includes information about lanugage contact, migration and settlement patterns, as well as processes of urbanisation. Isoglosses refer to dialect boundaries determined by individual linguistic features (such as word pronunciations, lexical choice or syntactic constructions). In contrast, computational dialectometry clusters phonetic or lexical data in the form of word lists into geographical dialect-regions by means of quantitatively defined distance measures (Göbl 1982, Kessler 1995, Nerbonne 2006, Nerbonne and

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

Hinrichs 2006, Prokić 2006).

Currently the method of choice for measuring the distance between two words, either in terms of their graphemic representation (in the case of lexical data) or in terms of their phonetic representation (in the case of pronunciation data), relies on the notions of alignment (Kondrak 2000) and of edit-distance (Heeringa 2004), particularly in the form of Levenshtein-Distance. The distance between lists of words is measured by an aggregate method that provides the summation of the distances in the word list.

Two disadvantages are implicit in these approaches:

- Aggregate methods consider in every pass just two data-records, not the entire data set. A comparison of the whole data set in a single step is not possible.
- It is only possible to compare pairs of individual words. For example, it is possible to compare two different pronunciations of the word *apple*, but it is not possible to track the occurrences of individual segments, e.g. the vowel *a* in different words, e.g. in *apple* and *banana*.

In this paper a new approach to computational dialectometry is proposed that is based on vector analysis and that avoids the above disadvantages of the aggregate-method. The approach is inspired by the Neogrammarian notion of regular sound correspondences. This notion has played a major role in the study of language change. Here it is applied to the study of language variation.

2.2 The Data

2.2.1 General format

The data takes the form of word lists, one such list per site. A site is a geographicaly defined point like a village or a town. Other properties such as size, geographical properties, more rural or more urban, are ignored at present.

For every site, the same words are collected and transcribed into X-Sampa (Wells n.d.), which is an electronic readable form of the IPA, the International Phonetic Alphabet (IPA 2003). The X-Sampa codes are the smallest units in the data-sets.

This data format allows investigations in two directions:

- Horizontal: In this direction all occurrences of a given element are traced in a single word from the word list across all sites. We will henceforth refer to such a horizontal trace as *single-word-all-sites* (SWAS-trace).
- Vertical: In this direction all occurrences of a given element are traced across the entire word list for a single site. We will henceforth refer to such a vertical trace as *single-site-all-words* (SSAW).

In the horizontal dimension, comparisons of a given element across different pronunciations of the same lexical item can detect regularities and irregularities

	Aldomirovci	Asparuhovo	...	Zheravna
агне (lamb)	"jAgne	"Agni	...	"Agni
аз (I)	"jA	"As	...	"As
бели (white-plural)	"beli	"beli	...	"beli
берат (pick up - 3rd plural)	"beru	bi"r7t	...	bi"r7t
...	...	...	...	...
ям (eat, 1st singular)	e"dem	"jAm	...	"jAm

Figure 2.1: general data-structure

of sound correspondences in the set of pronunciations. In the vertical dimension, comparisons of a given element across the word lists of different sites can reveal phonological and/or morphological proccesses such as insertion, deletion, and metathesis which are commonly found in language variation.

2.3 The Bulgarian Data-Set

In cooperation with the Bulgarian Academy of Science and the University of Sofia, a phonetic data-set of the Bulgarian language with 200 sites and 143 words[1] common to all sites is been collected (Osenova and Simov 2005). These 200 sites are spread across the whole territory of Bulgaria. At the moment, 121 sites are available in electronic form. XML is used as a container for the data. This data set forms the basis for all vector-based experiments reported in this paper.

(Zhobov 2006) provides detailed information about the selection of words that have been chosen for the data set and about the sources that have been consulted for their pronunciation.

2.4 The Vector-based Approach

2.4.1 Background: Vector Analysis

Vector analysis is a subarea of geometry. It deals with arrays (vectors) in a two- or higher dimensional space. In these spaces, vectors are defined by two points, each identified by one coordinate for each dimension. The arrays in our particular dialectometry application are always two-dimensional (one dimension for the canonical order of words in the word list and one for the order of elements within the individual word). Figure 2.2 gives an example of a vector $\vec{v_1}$ in two-dimensional space with the starting point (2,2) and the end point (4,4):

The length of a vector can be calculated on the basis of the Pythagorean theorem:

$$|\vec{v_1}| = \sqrt{\triangle x^2 + \triangle y^2} \tag{2.1}$$

[1]Some of the sites contain more words, but these 143 words are included in every site.

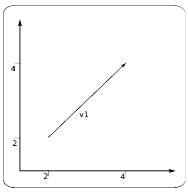

Figure 2.2: a two-dimensional vector

where $\triangle x$ and $\triangle y$ are the relative position-changes of the vector on the X and the Y axis.

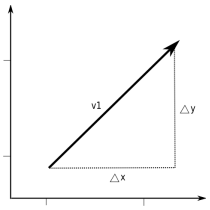

Figure 2.3: calculating the length of a vector

To compute the angle between two 2-dimensional vectors:

$$cos(\alpha) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \tag{2.2}$$

2.4.2 The Algorithm

In this method one element of interest is selected. This can be a single segment, a bigram or even longer sequences of segments. By the use of vectors, the element in focus is traced either horizontally (SWAS) or vertically (SSAW) through the entire data set. Each occurrence of the focus item is represented by a single vector. Combining these vectors into a chain of vectors, the relative position changes of the relevant element are recorded. The pseudocode for constructing such a vector chain for an *SWAS* trace or an *SSAW* trace is shown in figure 2.4.

On the X-axis of the coordinate system the units of measurement are the positions of X-Sampa codes in individual words. On the Y-axis, the words are the unit of measurement. By assumption, a shift on the X-axis of one X-Sampa to the left or to the right has the value 1. On the Y-axis going down one line to the next word, without shift on the X-axis, has the value of 1.

A vector chain constitutes a unique fingerprint of the occurrence of the element in focus either in a single word across all sites in the horizontal dimension or in

```
delta[X] = 0;
delta[Y] = 0;

for i=1 to number of occurrences of element A

    delta[X] = X(A[i]) - delta[X];
    delta[Y] = Y(A[i]) - delta[Y];
    addToVectorChain(<delta[X], delta[Y]>);
```

Figure 2.4: pseudocode for constructing a vector chain for a single focused element

all words of the word list for a single site in the vertical dimension. Moreover, in each dimension such fingerprints can be compared across sites or across words.

In the following example (Figure 2.5), a hypothetical element A is followed through a data record. The origin and starting point of the first vector is set to the first element of the data record.

Starting in the upper lefthand corner (0,0), the first appearance of "A" can be achieved by the vector $\vec{v} = (3, 0)$. The first coordinate represents the movement on the X- and the second one the movement on the Y-axis: this means, that they are showing the *relative* movement of a vector from the actual element to the next one, not the absolute position in the coordinate-system. From here, a second vector is drawn down to the second appearance of "A" (-1, +1), and so on:

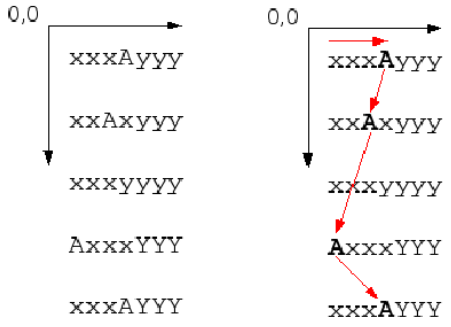

Figure 2.5: artificial example for tracing an element

Figure 2.6 shows an excerpt from the Bulgarian data set. From left to right: The first 13 words of the site Rakovica, located in western Bulgaria. In the middle, the vector chain for the vowel "e" is drawn. On the righthand side, the complete "e"-vector for the 143 words of site Rakovica is shown:

Figure 2.6: from left to right: partial word list, partial vector chain, complete vector chain

2.4.3 The Length of a Vector Chain

In the previous section, we have shown how vector chains can be created. Graphically such vector chains can be rendered as shown in Figure 2.6. However, in order to be able to compare vector chains with one another, a quantitative measure is needed. Such a measure can be obtained on the basis of the length of a vector chain[2]. This length can be calculated by adding together the lengths of the individual vectors contained in the vector chain.

$$|\vec{v_c}| = \sum_{i=1}^{n} \sqrt{\triangle x_{v_i}^2 + \triangle y_{v_i}^2} \tag{2.3}$$

where *n* is the number of single vectors in the vector chain.

For illustration, Figure 2.7 shows some typical vector chains and their lengths:

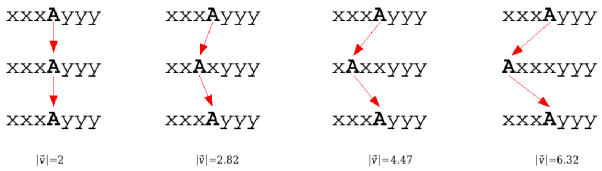

Figure 2.7: some typical vector lengths

[2]Another possibility would be to sum up the absolute movement of the element to the right and to the left. This *fluctuation* has one disadvantage: it cannot handle words which has more than one element correctly.

When calculating the length of vector chains, two questions arise: first, how to treat words with zero occurrences of the element and second, what to do when a word contains more than one occurrence of the same element. If a word has no occurrence of the element in focus, the vector chain will pass through this word and will extend to the next word in the word list that contains the element in focus. In consideration of the Pythagoraean Theorem, the resulting length of such a vector chain differs from a vector chain where each word contains exactly one occurrence of the focused element. This is illustrated in figure 2.8.

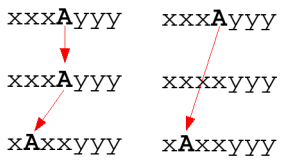

Figure 2.8: vectors when in a word the element doesn't occur

If a word has more than one element "A", additional vectors are drawn (Figure 2.9).

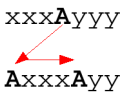

Figure 2.9: vectors when an element occurs more than one times in a word

Depending on the direction of analysis (horizontal or vertical), the length of a vector chain can be interpreted in two ways:

- In the horizontal direction: a higher value means that the specific element has more fluctuation than with a lower value. Elements with a high value are of particular interest since they carry a high degree of information about the linguistic distance across sites.
- In the vertical direction: the vector chain provides a site-specific, individual "fingerprint" of that element. The values of the individual vector chains for each site can then be clustered as described in more detail in section 2.5.

Some prominent values for the vector length are:

- If the element doesn't appear in the complete data-set, the length of the vector chain is zero.
- If the element always appears at the same position, the vector chain's length is identical to the number of words in the chain: there is just movement on the Y-, but none on the X-axis.

- The maximum length of a vector chain depends on the length of the words and their order

2.5 Vector-based Analysis: Selection and Clustering of Elements

As described in section 2.1, computational dialectometry deals with geographically defined dialect regions. Considering this goal, the examination in the vertical direction, which yields site-specific fingerprints is of central concern. This constitutes one more contrast to aggregate methods where pairwise comparison of individual words (in the horizontal dimension) provide the most important dataset.

In the vector-based analysis, the horizontal dimension can be used to determine which elements carry the highest degree of information about the linguistic distance between individual words. The elements with the highest information content can be selected to create particularly content-rich fingerprints of individual sites.

Such fingerprints can then be used to cluster the sites. The clustering is done by a bottom-up, hard clustering algorithm. Hard clustering is used so that each site can appear as a member in exact one cluster. Clustering proceeds in an iterative fashion. At the beginning, every site is its own cluster. In subsequent iterations clusters are merged until a fixed number of clusters has been reached. The target number of clusters is set in advance and depends on the desired granularity of geographic distribution.

2.6 Experiments with the Bulgarian Data-Set

This section reports on the application of the vector-based analysis, whose underlying theoretical assumptions have been presented in the previous sections, to the Bulgarian data set introduced in section 2.3. These experiments follow the strategy outlined in section 2.5. In a first step, an SWAS trace is performed for all single element X-Sampa codes contained in the entire data set. In a second step, the most content-rich elements are identified. In a third step, a SSAW trace is performed, which generates site-specific fingerprints for each of these most content-rich elements. In a fourth step, the lengths of the vector chains for each of these fingerprints is computed as described in section 2.4.3. Finally, these characteristic lengths are used in the hard clustering algorithm that was described in the previous section.

2.6.1 Finding content-rich Elements

Figure 2.10 shows the results of the first analysis steps. It displays the 10 most content-rich segments rendered in their respective X-Sampa codes.

Notice that, most of these elements are vowels or semi vowels (palatalized j). This quantitative finding corroborates the often-cited observation by traditional dialectologists that vowels tend to exhibit the highest degree of dialect variation.

X-Sampa-Code	Length of Vector chain
e	40015.1759910523
stress	35731.207131129
7 *(close-mid back, unrounded)*	35653.6778159966
A	35432.7572223606
i	34438.756791175
u	34120.3965759371
n	33581.1330654058
s	33038.0473845845
o	32878.0780176776
_j *(palatalized)*	32317.4612226377

Figure 2.10: the 10 most content-rich segments in the Bulgarian data set

The fact that the vector-based analysis is able to induce this observation by purely automatic means attests the viability of this method.

There is a second finding contained in Figure 2.10 that directly conforms to observations found in the traditional literature on Bulgarian dialect variation. It is the observation that different stress placements play a prominent role in the identification of dialect regions. Once again, the vector-based analysis induced this finding by purely quantitative means since the X-Sampa code for stress is identified as the second most content-rich element.

2.6.2 Creating Vector Chains

With the use of the elements in Figure 2.10, vector chains can be build for every site. Figure 2.11 shows 6 of these fingerprints, using the X-Sampa code “e”. Three of these sites are located in the eastern part and three in the western part of Bulgaria. Figure 2.12 shows their exact locations. The graphical rendered fingerprints of the 6 sites shows that individual fingerprints are an indicator for the sites’ geographical position.

2.6.3 Clustering the Sites

For every of the above described fingerprints the length of the vector chain can be computed. This results in a single value for every site, representing the variation of the focused element.

Using the above described clustering algorithm on these values, a distinction between the eastern part and the western part of Bulgaria can be seen for the entire data set in Figure 2.13. This east/west split once again conforms to the claim found in the traditional literature that the major division among Bulgarian dialects follows this orientation.

This distinction between the east and the west of Bulgaria can be seen in nearly every element. In general, the vowels are producing better results than the conso-

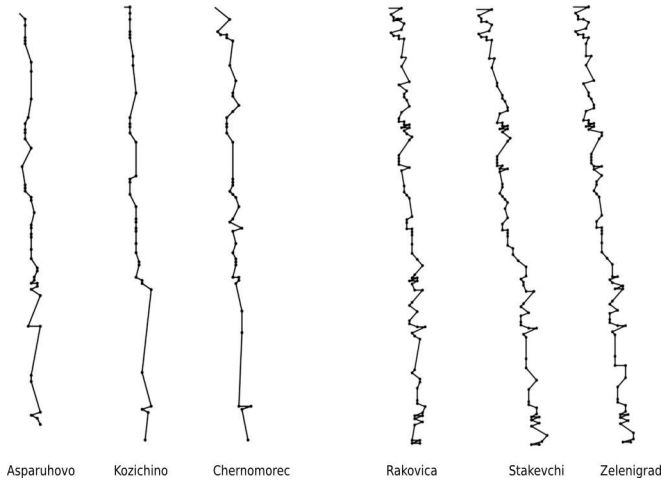

Figure 2.11: fingerprints of six sites, three in the east and three in the west of Bulgaria

Figure 2.12: the locations of the six sites

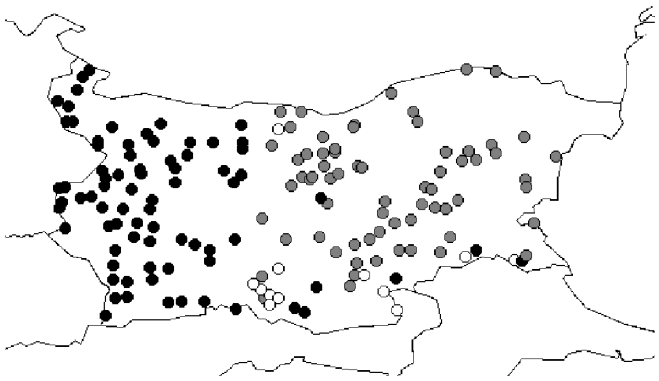

Figure 2.13: east-west distinction of Bulgaria, using the X-Sampa code “e”

nants since the variability of vowels is by comparison much larger than that of consonants.

2.7 Conclusion

A novel unsupervised learning approach to computational dialectometry is presented which uses hard clustering. The approach relies on vector analysis over two-dimensional arrays of word lists collected for different geographical sites. The paper presents the underlying theory and applies the approach to a Bulgarian data set. The results of these experiments demonstrate the viability of the approach since it is able to reproduce by purely quantitative means the major findings that have been obtained by traditional methods of dialectology for Bulgarian language variation.

In future research we plan to conduct further experiments with the full Bulgarian data set once it has become available. A second future field of experimentation concerns the length of the elements traced in the horizontal or vertical dimension. In the experiment described in section 2.6 we only investigated unigrams. Further experiments with bigrams and trigrams need to be conducted.

A third direction for further experimentation concerns the order of words in the word list. Currently, the words are ordered alphabetically. An anonymous reviewer has raised the issue whether the results depend on the order of the words and has suggested to compare the vector chains for different random permutations in the word list.

Finally, in the current experiments, we only used a single hard clustering approach. The investigation of different variants of hard clustering could well be another area where the current results may be improved.

2.8 Acknowledgement

This research was conducted by a collaborative project entitled *Measuring linguistic unity and diversity in Europe* with the following project partners: Bulgarian Academy of Science, Sofia, Bulgaria; Rijksuniversiteit Groningen, Alfa-informatica; Eberhard Karls University Tübingen, Seminar für Sprachwissenschaft. We should like to acknowledge the financial support given by the Volkswagen-Stiftung for this project[3].

We are much indebted to our colleagues in the project, in particular to Georgi Kolev, John Nerbonne, Petya Osenova, Petar Shishkov, Kiril Simov, and Vladimir Zhobov, for extended discussions on scientific matters related to this paper.

References

Göbl, H.(1982), *Dialektometrie. Prinzipien und Methoden des Einsatzes der Numerischen Taxonomie im Bereich der Dialektgeographie*, Denkschriften der Österreichischen Akademie der Wissenschaften, phil.-hist. Klasse, Österreichischen Akademie der Wissenschaften, Vienna.

Heeringa, W.(2004), *Measuring Dialect Pronunciation Differences using Levenshtein Distance*, PhD thesis, University of Groningen, Groningen.

IPA(2003), *Handbook of the International Phonetic Association*, Cambridge University Press.

Kessler, B.(1995), Computational Dialectology in Irish Gaelic, *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL-1995)*, Association for Computational Linguistics.

Kondrak, G.(2000), A new algorithm for the alignment of phonetic sequences, *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, Seattle, pp. 288–295.

Nerbonne, J.(2006), Identifying linguistic structure in aggregate comparison, *Literary and Linguistic Computing* **21**(4), 463–476.

Nerbonne, J. and Hinrichs, E. (eds)(2006), *Linguistic Distances*, Proceedings of the ACL-COLING-2006 Workshop, Association for Computational Linguistics, Sydney.

Osenova, P. and Simov, K.(2005), An infrastructure for storing and processing dialect data, Unpublished manuscript, Bulgarian Academy of Sciences. Available at: www.sfs.uni-tuebingen.de/dialectometry/documents.shtml.

Prokić, J.(2006), *Identifying Linguistic Structure in a Quantitative Analysis of Bulgarian Dialect Pronunciation*, Master's thesis, University of Tübingen.

Wells, J.(n.d.), Computer-coding the IPA: A Proposed Extension of SAMPA, http://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm.

[3]See `http://www.sfs.uni-tuebingen.de/dialectometry/` for further information

Zhobov, V.(2006), Description of the Sources for the Pronunciation Data, Unpublished manuscript. Department of Slavic Philologies, University of Sofia.