Quantitative methods in dialectometry

Thomas Zastrow

October/November 2006

Contents

Information is the difference that makes a difference.
Gregory Bateson

1 The data

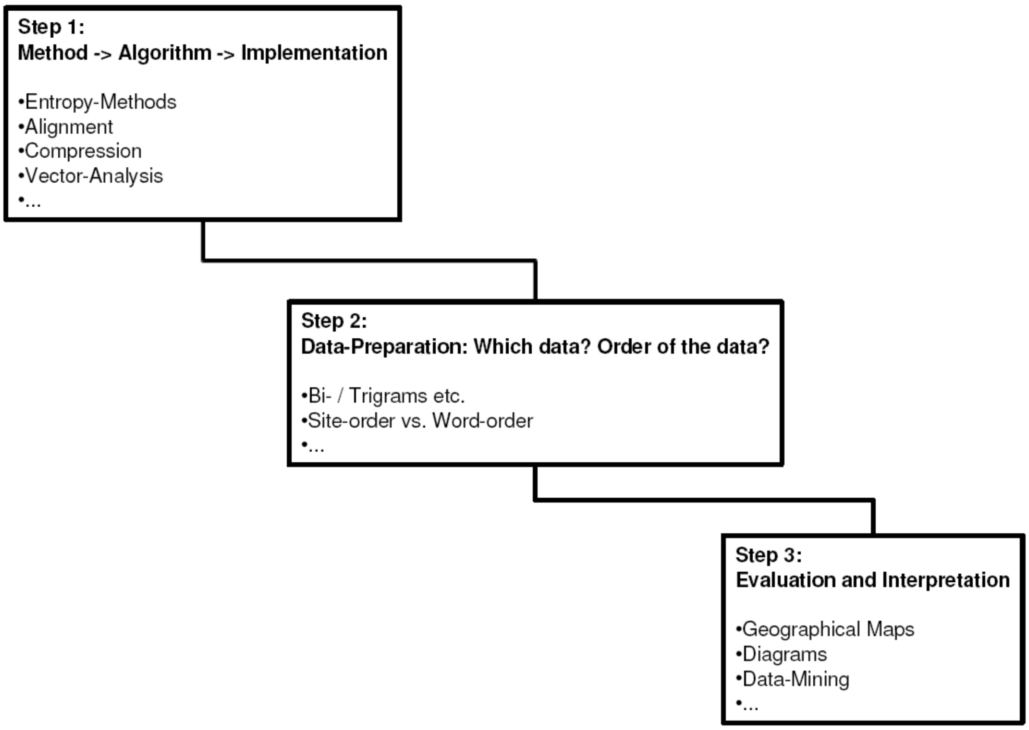
The data is organized as a 2-dimensional matrix:

	$Site_1$	...	$Site_n$
Key_1			
...			
Key_n			

This allows examinations in two directions:

1. Bundling the keys of site x (Direction D_s): In this direction, just pure quantitative examinations are possible
2. Bundling all the sites of key x (Directio D_k): Here are also alignment-examinations possible

2 The Procedure



3 Overview: The algorithms

	Direction	Alignment
Entropy	D_s	no
Alignment	D_k	yes
Compression	D_s	depends on the algorithm
Vector-analysis	D_s	(yes)
Levenshtein	D_k	yes

(The vector-analysis didn't needs an explicit alignment. It considers the relative positions of the elements to each other).

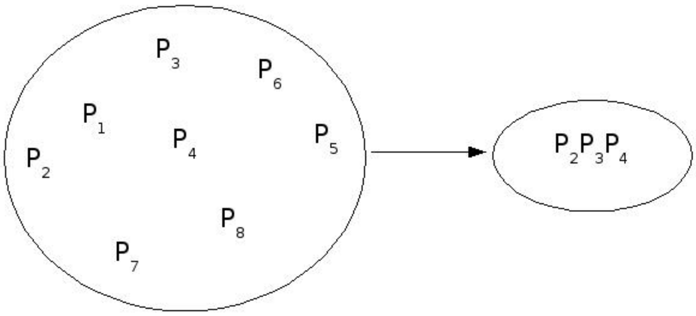
4 The class of entropy-related methods

4.1 Entropy as a quantitative approach

The original entropy was developed by Claude Elwood Shannon in 1948. Starting with this formula, a lot of derivations are possible.
All these derivations has one problematic property: When putting the data in the entropy-formula, it is handled like a set. The order of the elements is irrelevant. To consider the order of elements again, there seems to be two solutions:

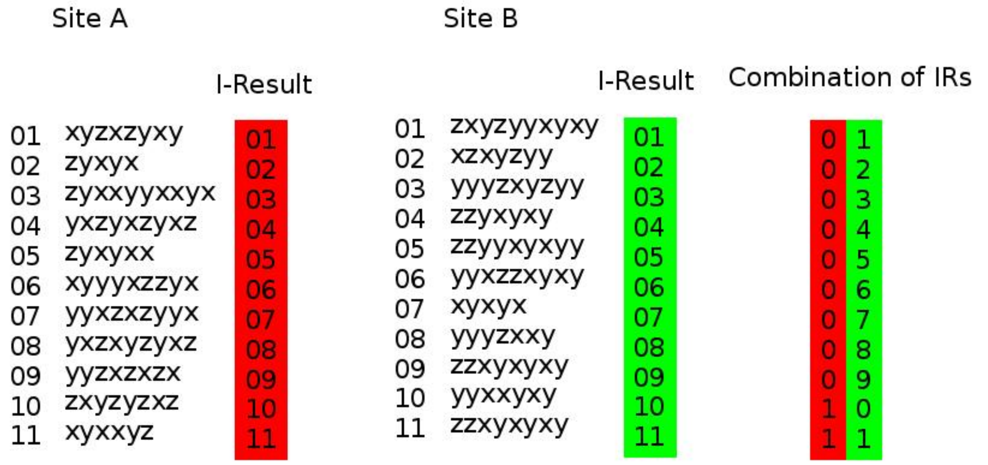
4.1.1 Possible solution: Data-preselection

Before putting the data into the entropy-formula. For example when creating a subset of a bigger data-set. This method seems not to be much flexible.



4.1.2 Possible solution 2: Take care of interim-results

When iterating over the elements, every element produces an interim result. When doing this with two or more data-sets which have the same order, the interim-results should be comparable:



Site A		I-Result	Site B		I-Result	Combination of IRs	
01	xyzxzyxy	01	01	zxyzyyxyxy	01	0	1
02	zyxyx	02	02	xzxyzyy	02	0	2
03	zyxxyyxxyx	03	03	yyyzxyzyy	03	0	3
04	yxzyxzyxz	04	04	zzyxyxy	04	0	4
05	zyxyxx	05	05	zzyyxyxyy	05	0	5
06	xyyyxzzyx	06	06	yyxzzxyxy	06	0	6
07	yyxzxzyyx	07	07	xyxyx	07	0	7
08	yxzxyzyxz	08	08	yyyzxxy	08	0	8
09	yyzxzxzx	09	09	zzxyxyxy	09	0	9
10	zxyzyzxz	10	10	yyxxyxy	10	1	0
11	xyxxyz	11	11	zzxyxyxy	11	1	1

In addition, a division of the interim-results by the actual value of the iteration-pass would take care of proportion:

$$\boxed{-\sum_{1}^{n} \frac{p_n \log_2(p_n)}{n}} \tag{1}$$

It has to be cherished, that the end-result would be the same as if the algorithm doesn't take care of the order of elements. Example:

1	+	2	+	3	+	4	=	10
4	+	3	+	1	+	2	=	10

5 Concrete methods

5.1 The relative entropy

The relative entropy is defined as the relation between the proper entropy and the maximum entropy of a dataset:

$$\boxed{H_{rel} = \frac{H_{prop}}{H_{max}}} \tag{2}$$

5.1.1 Algorithm

For every site

1. Calculate the proper entropy for the whole list of phones of the actual site
2. Calculate the maximum entropy for the whole list of phones of the actual site
3. The relative entropy is proper entropy / maximum entropy

5.1.2 Results

Bigrams: http://www.sfs.uni-tuebingen.de/dialectometry/maps2/index.html (Compared to the partial information)

5.2 The partial information

The partial information is one level of abstraction lesser than the entropy. It works directly on the information which a dataset contains[1]. The main difference is that it sums up the information of *every* element in the dataset and not just the alphabet.
The information of a single element is calculated by

$$\boxed{I(z_n) = \log_2 p(z_n)} \tag{3}$$

This information can be summed up for the whole dataset. The partial information is now the relation between two of these information-values:

$$\boxed{I_p(A, B) = \frac{I(A)}{I(B)} where A \subset B} \tag{4}$$

5.2.1 Algorithm

Calculate the information of the whole dataset I_{ds}.

For every site

1. Calculate the information of every site I_s
2. Calculate the partial information, dividing I_s by I_{ds}

[1] See also http://www.sfs.uni-tuebingen.de/dialectometry/meetings/2006groningen/entropy.pdf

5.2.2 Results

Bigrams: http://www.sfs.uni-tuebingen.de/dialectometry/maps2/index.html
(Compared to the relative entropy)

5.3 The conditional entropy

The conditional entropy compares two datasets: What is the relation between the entropies of a given, wellknown (X) and an unknown dataset (Y)?[2]

$$\boxed{H(Y|X) = H(X,Y) - H(X)} \tag{5}$$

where $H(X,Y)$ is the joint entropy.
In our case, the standard-pronunciation of the bulgarian words can be used as basis-set (X).

5.3.1 The joint entropy

When concatenating two datasets X and Y, the joint entropy gives the entropy of this new amalgamation:

$$\boxed{H(X,Y) = -\sum_{1}^{x,y} p_{x,y} \log_2(p_{x,y})} \tag{6}$$

5.3.2 Algorithm

1. Calculate the proper entropy of the standard-data-set (PE)
2. Calculate the probabilities of all phones in standard (PSt)
3. For every site: Calculate the probabilities of the phones (Ps) and then the joint-entropy of PSt and Ps (JE), the conditional entropy is JE - PE

5.3.3 Results

The Conditional Entropy, compared for every site against the standard: http://www.sfs.uni-tuebingen.de/dialectometry/maps2/ConditionalEntropyOfSites.html

[2] See also http://en.wikipedia.org/wiki/Conditional_entropy

6 Sound-correspondence

6.1 Primer

Sound Correspondence (SC) in dialectometry works on the basis of phones. It measures how a phone changes from dialect to dialect. This can be done in various ways, taking a look on two main features of a phone:

1. The position in the word.
   From region to region, the position of a phone within a word can change:

1	x	x	x	A	x	x	x
2	x	x	x	x	A	x	x
3	x	x	A	x	x	x	x

   In this example, the position of the phone A has changed from word 1 to 2 from position 4 to position 5 ....

2. The kind of the phone.
   spionf

The field of sound-correspondence is widely covered by the class of alignemnt-algorithms.
Alignment-algorithms are normaly used to identify related elements in two or more data-sets. In the Buldialect-project, exactly the opposite is wanted: How many and what kind of *differences* are between two or more sites?
With the attention to the unit of (single or combined) phones, two approaches are possible:

1. Putting the emphasis on one or more special phone(s). The question is: *Appears this phone in this key? And if so, where exactly?*

2. Instead of observing a phone, also a position - or more than one - within the keys could be the point of interest: *What phones are possible on this position(s)?*

6.2 Preparatory work

In gerneral, all alignment-methods needs a consistent data-set. For example, the key_1's in all records (here: a record is equal to one site) are compared to a "standard"-key_1.

Before starting with alingment, it has to be sure that:

1. Just the keys which appears in all records are used
2. In every record, the keys has to be in the same order
3. Another problems lies in the fact that in some cases we have more than one variant for a key. In this case, it should be necessary to use only one variant per key

(Talk to Petya about that.)

6.3 ALINE-algorithm (Kondrak)

For the algorithm see: 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, Seattle, April, 2000
(Online: http://www.cs.ualberta.ca/˜kondrak/publications.html#CL)
To be done. Some preparatory work:

1. ALINE-algorithm[3]
2. Feature-list of the used phones

6.4 N–Gram Similarity and Distance (Kondrak)

For the algorithm see: Kondrak, Grzegorz. N–gram similarity and distance. Proceedings of the Twelfth International Conference on String Processing and Information Retrieval, Buenos Aires, Argentina, November 2005
(Online: http://www.cs.ualberta.ca/˜kondrak/publications.html#CL)

6.5 The Number of Distsinct Alignments of Two Strings (Covington)

For the algorithm see: The Number of Distsinct Alignments of Two Strings. Michael A. Covington, Journal of Quantitative Lingustics 2004, Vol. 11, No. 3, pp. 173-182

7 Compression-algorithms

To be done. Take a look at Kiril's paper.

[3] http://www.cs.ualberta.ca/˜kondrak/

8 Vector analysis

8.1 Thoughts

A vector is an arrow which has two properties: Direction and length [4].

In a geometrical way, an origin can be set for every site (Fig. 1). Starting from here, a chain of vectors can follow an observed phone through the site (Fig. 2.):

$$\boxed{\vec{v_1} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \vec{v_2} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \vec{v_3} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \vec{v_4} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}} \tag{7}$$

From this vector-chain, a new vector can be compiled (Fig. 3, blue vector):

$$\boxed{\vec{v_{ges}} = \vec{v_1} + \vec{v_2} + \vec{v_3} + \vec{v_4} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}} \tag{8}$$

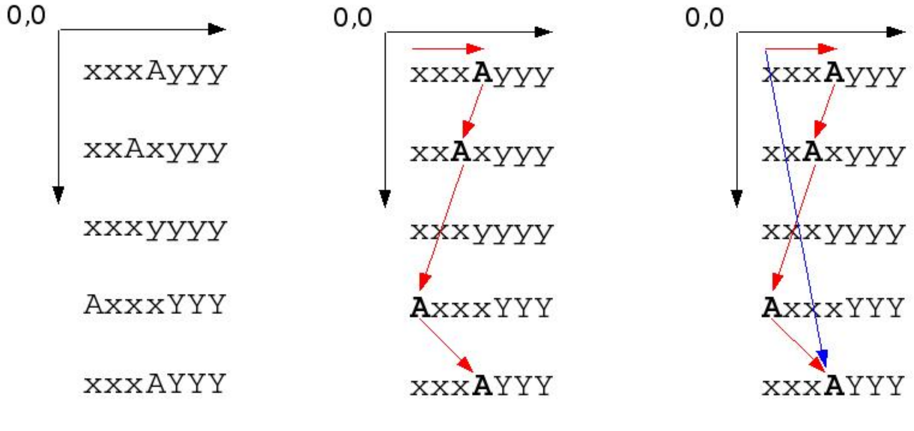


Fig. 1 Fig. 2 Fig. 3

When doing this for all sites, the angle between the compiled vectors should be site-specific (Fig. 4):

$$\boxed{\cos(\alpha) = \frac{\vec{a} * \vec{b}}{|\vec{a}| * |\vec{b}|}} \tag{9}$$

Here are two ”standard”-vectors possible: First, the vector of the ”standard”-bulgarian and second a vector which follows the X- or the Y-axxis. To get

[4] See in german http://de.wikipedia.org/wiki/Vektor and in english http://en.wikipedia.org/wiki/Vector_analysis

the angle from the cosinus-value, the arc-cosinus could be used.

In Java: All trigonometrical functions in Java are using the radial unit-systems. To get back degrees from the cosinus, the following transformations are necessary:

$$\boxed{\alpha^\circ = \frac{\arccos(\alpha) * 180}{Pi}} \tag{10}$$

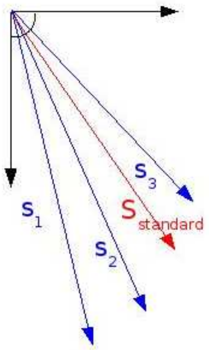


Fig. 4

8.2 Results

The maps are ready but still not online ...

9 Clustering

Up to now, the clustering-algorithm is a very simple one. In the future it should become necessary to use some more complex ones.

9.1 Parameters

The clustering-procedure is for all algorithms the same. It's properties:

1. Hard-clustering
2. Bottom-up
3. The number of clusters is variable

It expects:

1. A list of double-values
2. A treshold-value which represents a %-value
3. A boolean-value; If true, the colors of the clusters will be adapted so that the largest, the second largest cluster and so on will always get the same color

9.2 Algorithm

1. Sort the list of doubles from small to large values
2. Assign the first color to the first double-value
3. For each other double value:

   If the difference between the actual and the preceding value is smaller than the treshold, the actual double-value will get the same color as the preceding value. Else, it will get a new color.
4. If the colors should be adapted, correct the colors so that the largest, the second largest and so on will always have the same color (That sounds simple, but was realy difficult to implement ...)

10 The Maps

The maps are build with the help of QGIS[5] in the version 0.7. QGIS, the "Quantum Geographical Information System", is OpenSource-Software and via a Plug-In able to use the GRASS[6].

The coordinates for new sites have to be published in a comma-separated file. After reading this file with the "Add delimited text layer"-Plug-In, the sites can be given colored points, cricles or whatever in the rather XML-file (normaly, this is done by Java).

11 Miscellaneous ideas

1. Longest common subsequence (LCS): Probably not useful in the context of bulgarian phones, but perhaps useful in the context of alignments.

[5] http://www.qgis.org

[6] http://grass.itc.it/

2. Transition probabilities / Frequencies

3. "Putting the cart before the horse": Starting with *geographical* aligned data, assigning linguistic occurrences to geographical distributions

4. Splitting the phones into more smaller units: Kondrak has some thoughts on phonetical feature-lists for phones: Where is place of articulation, how is it build?