New Developments of Nonlinear Projections for the Visualization of Structures in Nonvectorial Data Sets

Teuvo Kohonen

Aalto University
SCIENCE + TECHNOLOGY
RESEARCH REPORT

Aalto University publication series
SCIENCE + **TECHNOLOGY** 8/2011

New Developments of Nonlinear Projections for the Visualization of Structures in Nonvectorial Data Sets

Teuvo Kohonen

Aalto University
School of Science
Department of Information and Computer Science
Adaptive Informatics Research Centre

Aalto University publication series
SCIENCE + **TECHNOLOGY** 8/2011

ISBN 978-952-60-4145-2 (pdf)
ISBN 978-952-60-4144-5 (printed)
ISSN-L 1799-4896
ISSN 1799-490X (pdf)
ISSN 1799-4896 (printed)

Aalto Print
Helsinki 2011

Finland

The publication can be read at http://lib.tkk.fi/SCIENCE_TECHNOLOGY/2011/isbn9789526041452.pdf

Publication orders (printed book):
series@ics.tkk.fi

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Teuvo Kohonen

Name of the publication
New Developments of Nonlinear Projections for the Visualization of Structures in Nonvectorial Data Sets

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series SCIENCE + TECHNOLOGY 8/2011

Field of research Computer science

Abstract

New nonlinear projections for the visualization of structures in nonvectorial data sets are suggested. Since there exist problems with the convergence of the traditional multidimensional scaling (MDS) when the data are nonvectorial, a new version of the MDS, called the nearest-neighbors multidimensional scaling (NN-MDS), is introduced. While it represents the local data structures more accurately and converges fast, two amendments had to be added, in order to describe the global structures as well. A new initialization method called the GENINIT is also introduced. It is very fast and may be used as a nonlinear projection, too, but it is more suitable for the initialization of the more accurate learning algorithms.

Keywords GENINIT, multidimensional scaling, nonlinear projection, similarity graph for string variables

ISBN (printed) 978-952-60-4144-5 **ISBN (pdf)** 978-952-60-4145-2

ISSN-L 1799-4896 **ISSN (printed)** 1799-4896 **ISSN (pdf)** 1799-490X

Location of publisher Espoo **Location of printing** Helsinki **Year** 2011

Pages 18 **The publication can be read at**
http://lib.tkk.fi/SCIENCE_TECHNOLOGY/2011/isbn9789526041452.pdf

Preface

The *nonlinear projections* are used for the visualization of topological relations between high-dimensional items, using a two-dimensional display on which the projections are shown. They can be used to map nonvectorial items, too, such as strings of symbols, into a two-dimensional order that illustrates their mutual distances, i.e., their dissimilarities. Such nonlinear projections have been used to classify, e.g., protein sequences and genetic codes of viruses.

In this work, new nonlinear projections for the visualization of structures in nonvectorial data sets are suggested. Since there exist problems with the convergence of the traditional multidimensional scaling (MDS) when the data are nonvectorial, a new version of the MDS, called the nearest-neighbors multidimensional scaling (NN-MDS), is introduced.

While NN-MDS represents the local data structures more accurately and converges fast, two amendments had to be added, in order to describe the global structures as well. A new initialization method called the GENINIT is also introduced. It is very fast and may be used as a nonlinear projection, too, but it is more suitable for the initialization of the more accurate learning algorithms.

Otaniemi, May 26, 2011,

Teuvo Kohonen

Contents

1 Introduction

Unlike the clustering methods, which are used to classify data into discrete sets, the *nonlinear projective mappings* have been meant to *visualize structures in data distributions by two-dimensional geometric displays*. In most applications, the data to be visualized consist of high-dimensional metric vectors, and the traditional nonlinear mappings are different versions of *multidimensional scaling (MDS)* [1]. In addition, the *self-organizing map (SOM)* [2] is used to display data structures.

It can be shown that both the MDS and the SOM can be constructed for *nonvectorial items*, too, such as strings of symbols, if a *distance matrix* for these items has been defined [2]. Tentatively it seems that the distance matrix $D(*,*)$ must define a *metric*, i.e., for any items A, B and C, $D(A,A) = 0$, $D(A,B) = D(B,A) \geq 0$, and $D(A,B) + D(B,C) \geq D(A,C)$.

This work discusses new developments of nonlinear projections, especially those intended to cope with nonvectorial items. It has turned out that the traditional methods have problems with their convergence. It is also known that the initialization of nonlinear projection algorithms is a problem. This treatise introduces a new and extremely fast initialization method called the GENINIT ("general initialization") for nonvectorial items. It does not involve any learning, but it could be used as a nonlinear projective mapping, although its accuracy is not particularly good as such. However, it seems to be effective in speeding up the convergence of the other nonlinear projection methods.

2 The ordering initialization method GENINIT

In this subsection, the new ordering method called the GENINIT ("general initialization") is defined. It is based on the distances of the items from two pairs of properly selected *fixed points*. In order to understand the motivation underlying it, one may first have a look into the metric-vector case discussed in Sect.9 (Appendix).

1. The first pair of fixed points. Assume a finite set of sample items. Find the two items that have the *largest mutual distance*. These points are signified as the *first pair of fixed points*.

2. The first ordering sequence. All of the sample items can now be ordered one dimensionally according to their *differences of distances from*

the first pair of fixed points.

3. The second pair of fixed points. In order to find two other fixed points, one may compare all successive pairs of samples in the first ordering sequence. The pair, the mutual distance of which is largest, shall be regarded as the *second pair of fixed points*. The ordering of the items with respect to this pair is to a large extent *independent of the first ordering sequence*.

4. The second ordering sequence. Next, all of the original items are ordered one dimensionally according to their differences of distances from the second pair of fixed points.

5. Two-dimensional ordering. Let i be some running index that enumerates the input samples in any order. Let $ind_1(i)$ be the index in the first ordering sequence. Let $ind_2(i)$ be the index relating to the second ordering sequence, respectively. Then the simplest two-dimensionally ordered plot of the samples is defined by the coordinates $(ind_1(i), ind_2(i))$. This ordering is particularly effective as such for nonvectorial data, because no two input items coincide in the map. Notwithstanding it is also possible to use the sorted distance values in the two ordering sequences for the coordinates of the items in the map.

A comment at this point may be due. The nonlinear projections discussed in this work are not independent of the metric used to represent the input items. Therefore, in striving for good visualization, one is free to experiment, e.g., with different *powers* of the standard distance values in selecting a proper metric.

3 The Levenshtein distance for symbol strings

For *string variables*, various distance measures have been defined. The most popular one may be the *Levenshtein distance*, also called the *edit distance* [3]. A variant of it is the *maximum posterior probability distance* [3]. In bioinformatics, the distances in large databases of long symbol strings, such as the protein sequences [4] or genetic codes of viruses, are usually described by partly heuristically defined distances such as the FASTA measure [5]. In this work, for the simplicity of demonstration, we shall use the basic unweighted Levenshtein distance, which defines a metric.

The three basic types of error that occur in strings of symbols are the

replacement error (change of a symbol into a wrong one), the *deletion error*, and the *insertion error*, respectively. (The interchange of two successive symbols is reduced, e.g., to two replacement errors.)

The basic unweighted Levenshtein distance $LD(A, B)$ between the strings A and B is defined as

$$LD(A, B) = \min(a + b + c) \ , \tag{1}$$

where string B is obtained from string A by a replacements, b insertions, and c deletions of a symbol. There exists an indefinite number of combinations of a, b and c to implement the transformation, and the minimum is usually sought by a variational method called the *dynamic programming* [2].

4 Example: Two-dimensional ordering of names of different lengths by the GENINIT method

For a demonstration, 25 first names from the Author Index of the Proceedings of the WSOM 2009 [6] were chosen. Their lengths varied between 7 and 9 letters.

The most regular GENINIT initialization for nonvectorial items has been obtained using the *indices of the two ordering sequences as the coordinates* $x(i)$ and $y(i)$ in the two-dimensional ordered graph. Such a graph is shown in Fig. 1.

One has to emphasize that the ordering of the names in the horizontal and vertical directions is determined by the *global differences from the two pairs of fixed points*, whereas there is no reason for which the *differential distances between adjacent names in the sequences* should be correct. Nonetheless one can notice that the most distant items, i.e., the items, the sum of distances of which from all of the other items is largest, are mainly located at the edges of the 2D display. A better differential similarity of the mapping will be achieved in the fine tuning processes that follow.

Conclusion: The GENINIT method is very fast (the computing time in the present example was less than a millisecond) and robust, and it describes the overall structures approximately. It is not satisfactory for a nonlinear projection as such, but it is effective for the initialization of the learning projective algorithms.

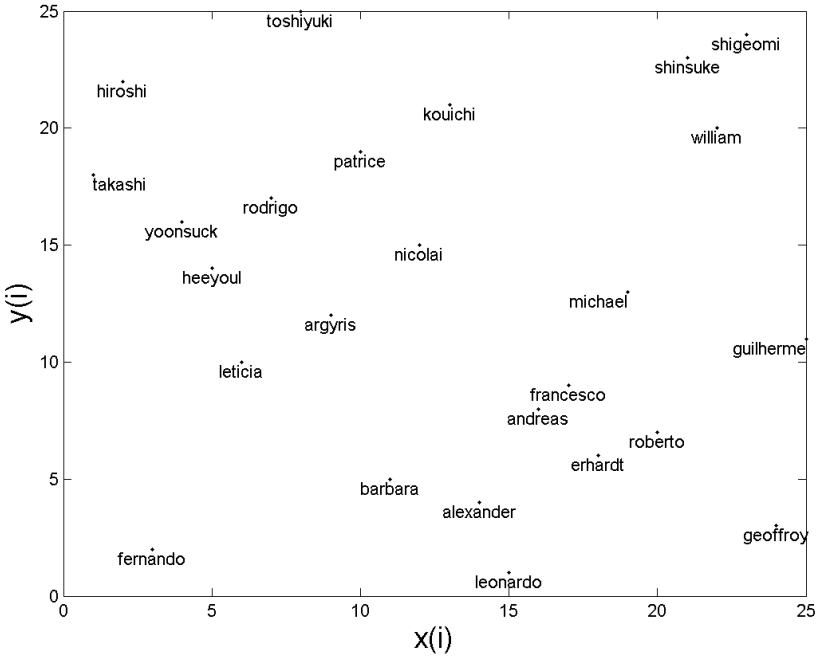

Figure 1. Two-dimensional ordering of 25 first names by the GENINIT method. The names are locatable according to the little points.

5 Nearest-neighbors multidimensional scaling (NN-MDS)

The previous approximative two-dimensional ordering does not yet guarantee that the global mutual distances in the two-dimensional map were even closely the same as in the original distance matrix D. However, one may use the GENINIT initialization as a starting point for *learning algorithms*. One may try to continue, e.g., with the *multidimensional scaling (MDS)* (cf., e.g., [1] or [2]). The basic objective in the latter is to *minimize the error function*

$$E = \sum_i \sum_{j<i} (D_2(i,j) - D(i,j))^2 \ , \tag{2}$$

where $D_2(i,j)$ is the distance between the images of the items i and j on the two-dimensional Euclidean plane, and $D(i,j)$ is the distance selected to describe the differences between these items (e.g., string variables). The minimum can be sought by gradient computation.

This first experiment has been carried out using the unweighted Levenshtein distances.

However, although the basic MDS works well with metric vectors, especially when their dimensionality is not large, in our numerous attempts to

use the MDS for nonvectorial items has not been successful. The convergence of the algorithm has been slow and the learning process has usually ended up in violent limit cycles. This is explainable by a few facts. The distances in this example are integer-valued and their domain is narrow. There are many equal distance values between the items. For these reasons the corrections to the locations of the items in the 2D display are often contradictory.

A variation of the MDS called the *Sammon mapping* [7] emphasizes local structures. The squared error is divided by the original distance $D(i,j)$. Also the convergence is thereby improved.

In this work we introduce a variation of the MDS, called the *nearest-neighbors multidimensional scaling (NN-MDS)*, in which the correction, relating to a particular item i, is computed only for i and its *nearest neighbor* (or neighbors, if the minimum is multiple). This variation of the MDS has brought about a very effective stabilization of the learning process, and the ordering criterion is more transparent than in the Sammon mapping. Let $j(i)$ be the index of the item closest to item i. The new variation of the algorithm is defined by:

if $j(i) = \text{argmin}_j(D(i,j))$ and $j(i) < i$,

$$E = \sum_i \sum_{j(i)} (D_2(i,j) - D(i,j))^2 \ . \tag{3}$$

One iteration cycle, in which the corrections to the coordinates are made conditionally, is the following. Let i run over all of the 25 indices of the words;

$$\begin{aligned}
&\text{for } \ i = 1 \ \text{ to } \ 25 \ , \\
&\text{if } \ j(i) = \text{argmin}_j(D(i,j)) \ \text{ and } \ j(i) < i \ , \\
&\Delta x(i) = -\lambda(D_2(i,j) - D(i,j)) * (x(i) - x(j))/D_2(i,j) \ , \\
&\Delta y(i) = -\lambda(D_2(i,j) - D(i,j)) * (y(i) - y(j))/D_2(i,j) \ , \\
&\Delta x(j) = -\Delta x(i) \ , \\
&\Delta y(j) = -\Delta y(i) \ .
\end{aligned} \tag{4}$$

Here λ is the *learning rate* such that $0 < \lambda < 1$, and with the number n of learning cycles, a useful expression for λ could be $0.2/(1 + An)$, where A is a value that depends on the number of learning cycles. With 100,000 cycles, the value of A could be equal to .001.

The fine tuning episode consisted of 100,000 learning cycles. Fig. 2 shows the map after the NN-MDS fine tuning.

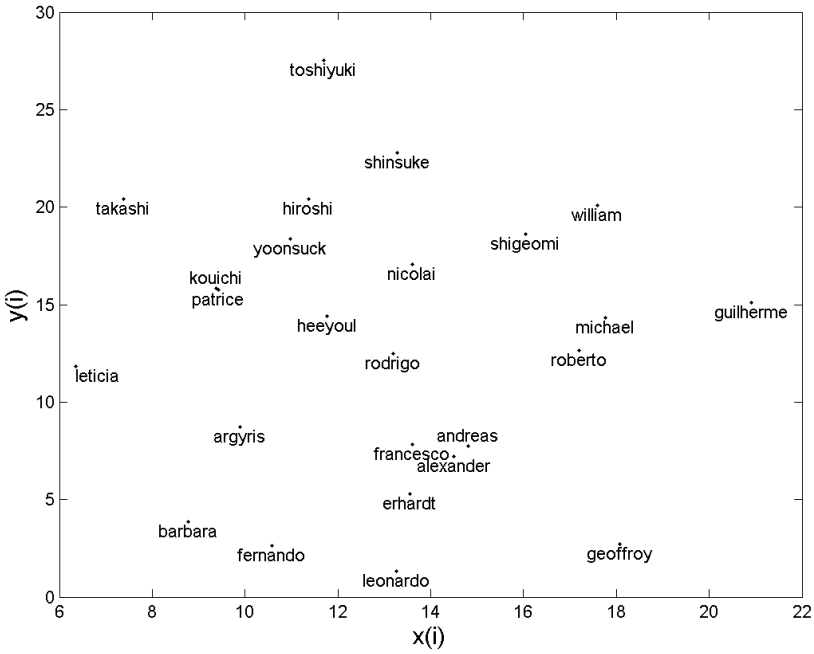

Figure 2. The coordinates $(x(i), y(i))$ of the items after 100,000 iteration cycles of the basic NN-MDS algorithm.

Notice that the new values $x(i)$ and $y(i)$ are no longer integers but real-valued coordinates.

Conclusion: The convergence of the basic MDS algorithms is a severe problem in the case of nonvectorial data. Since the representation of local structures is usually most important for visualization, the NN-MDS algorithm is preferred, also because its convergence is very good. However, without any amendments, the mapping of large distances sometimes becomes erroneous.

6 The first amendment to the NN-MDS

Fig. 2 has still an unsatisfactory property. Although the items that have a small mutual distance are usually located near to each other in the display, the items may be close to each other in the map, although their original distance is large. This is a usual flaw in all of the earlier MDS methods and the SOM, too. Eq. (4), however, can be amended by alternating two learning phases: first, the NN-MDS algorithm, and second, an algorithm in which the corrections are

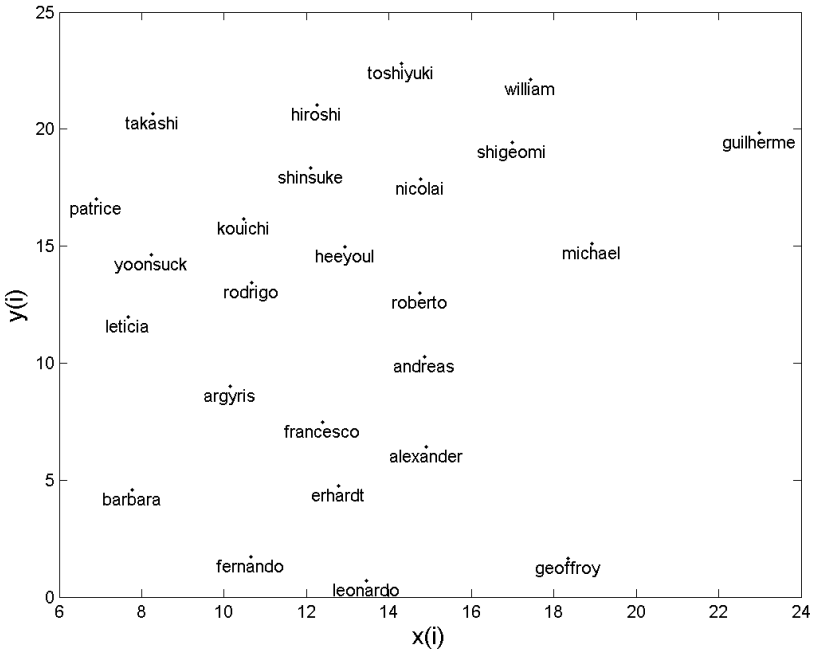

Figure 3. The coordinates $(x(i), y(i))$ of the items after 100,000 iteration cycles, when the first amendment to the NN-MDS algorithm was applied.

$$
\begin{aligned}
&\text{for all } i \text{ and } j \ , \ \text{if } (i,j) = \text{argmin}(D_2(i,j)) \ , \\
&\Delta x(i) = -\lambda (D_2(i,j) - D(i,j)) * (x(i) - x(j))/D_2(i,j) \ , \\
&\Delta y(i) = -\lambda (D_2(i,j) - D(i,j)) * (y(i) - y(j))/D_2(i,j) \ , \\
&\Delta x(j) = -\Delta x(i) \ , \\
&\Delta y(j) = -\Delta y(i) \ .
\end{aligned}
\tag{5}
$$

In other words, *we are looking for the items that are closest in the 2D display. We carry out a correction on them that is repulsive in the case that their 2D distance is too small*. This variation has a regularizing effect on the map, without causing any instabilities in the convergence, or major changes in the overall point density. Fig. 3 shows what happened to Fig. 2.

Conclusion: The errors in the largest distances, caused by the basic NN-MDS algorithm, will be corrected to a large extent by the first amendment.

7 The second amendment to the NN-MDS

Since the nonlinear projection methods have been intended for visualization purposes, it is the *rank order* of the items in the display that is more important than their exact distances. The *nonmetric MDS* [8], [9] tries to guarantee the best possible configuration of the 2D display by transforming the original distances $D(i,j)$ into a new scale $f(D(i,j))$, defined by a monotonically increasing function f that best preserves the rank order. The function f is usually determined experimentally.

The NN-MDS algorithm already abandoned the accurate metric distances by restricting the corrections to the items that are closest to each other in the original space. However, when striving for the best rank order in the 2D display, it will be possible to apply the nonmetric transformation to the NN-MDS, too.

If we define the distance $D(i,j)$, say, as the *third power* of $LD(i,j)$, we actually make a bigger distinction between small and large distances, which improves the 2D visualization, as seen from Fig. 4.

In this experiment we had $\lambda = .5/(1+.0001n)$ and the number of learning cycles was 1,000,000.

There is only one pair of items, namely, (*fernando*, *leonardo*) that has a mutual Levenshtein distance of three edit operations. The mutual distance of these names in the 2D diplay is smallest.The following five pairs of names have the distance of four edit operations:

(*erhardt*, *fernando*),
(*erhardt*, *leonardo*),
(*hiroshi*, *nicolai*),
(*hiroshi*, *takashi*),
(*roberto*, *rodrigo*).

These pairs of names have small mutual distances and are located around the middle of the display.

On the other hand, the following four names have the largest sums of Levenshtein distances from the others. All of them lie at different edges of the 2D display.

alexander *guilherme* *toshiyuki* *francesco*

It has now been demonstrated that the NN-MDS, with its two amendments, is a viable alternative of the familiar MDS methods, especially for nonvectorial items.

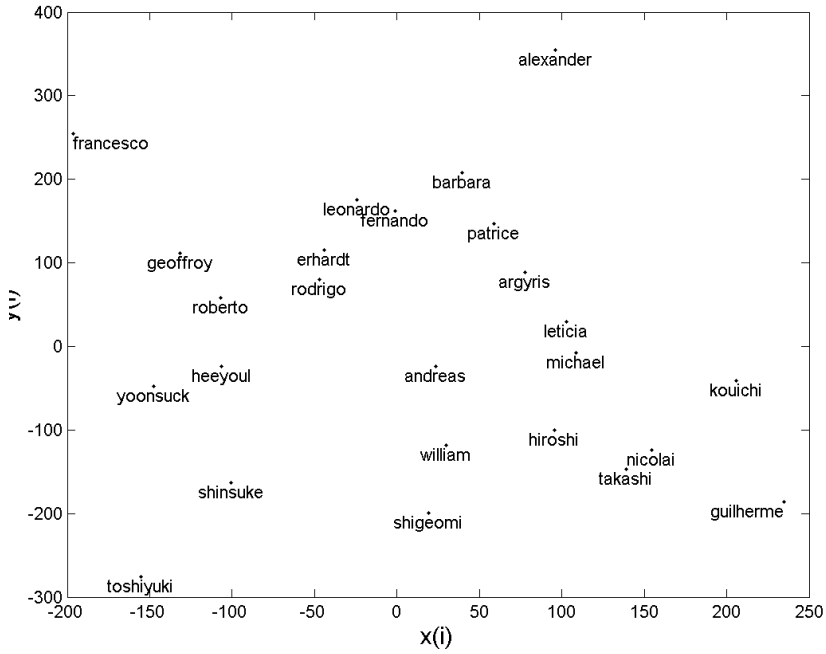

Figure 4. The coordinates $(x(i), y(i))$ of the items after 1,000,000 iteration cycles when both the first and the second amendment to the NN-MDS algorithm were applied.

Conclusion: A good agreement with both local and global structures will be obtained by the NN-MDS algorithm, if the first and the second amendment to it are introduced. The convergence is reasonably robust, although not as fast as without the amendments.

At any rate, it is recommendable to use the GENINIT initialization with all versions of the MDS algorithms.

8 How do we compute an SOM for general distance measures?

First of all it may be necessary to remark that the MDS and the SOM have been meant for quite different problems.

1. If one wants to have a unique image for every unique input item in the plot, as the case normally is in taxonomy, then one ought to resort to MDS, where the projection errors have been minimized.

2. If, on the other hand, there are plenty of randomly scattered samples, and one is mainly interested in the local averages of data and their cluster structures, the usual nonlinear projection methods such as the MDS cannot be used. The SOM display is the only realistic method for that purpose.

This distinction relates to some other neural-network mappings, too, that have been suggested for a replacement of the SOM.

3. The point density of the local averages on the SOM display is tending to approximate the density function of the input items, at least some low power p of it $(0 < p < 1)$.

4. A few further comments to clarify the differences between the SOM and the different MDS methods should be made. In the basic MDS, the optimization of the 2D map was made by minimizing the distance errors of the 2D projections vs. the original distances over all pairs of items, but there may remain even large errors in the individual distances between samples that are neighbors in the map. The Sammon mapping, the basic NN-MDS, as well as the SOM tend to approximate the local topological structures in subsets of samples that are neighbors in the map, and do not pay attention to the correctness of longer distances.

In order to demonstrate the real capability of the SOM and the benefit of the GENINIT initialization method for it, we need so much bigger databases and more computation that it must be left to a further study. Here we only delineate the method by which an SOM is constructed for nonvectorial data items such as, for example, string variables.

The *ordering criterion* of an SOM, for which only the distance matrix of its input items has been given, is that in the neighborhood set of nodes around every node, the item associated with the node is the *set median* of the items in the neighborhood nodes. The neighborhood of a node consists of the set of nodes within a fixed radius from the node. The set median of a set, on the other hand, is defined as the item that has the *smallest sum of distances from all members of the set* [2].

Starting from any initial state of the SOM (in which some input item is associated with every map node), a *batch training procedure* can be defined. First, every *training input* is mapped into that map node ("winner node"), from which it has the smallest distance. Thus one obtains a list of selected input items associated with every node.

The *updating* of the SOM takes place by replacing the old item at every node by the *set median of the union of the lists in the neighborhood of this node*. The updating shall occur as a *two-rank operation*, i.e., the replacement of all old items shall occur concurrently, after the lists have been constructed. The mapping and updating operations are reiterated until the map becomes steady, i.e., until no further changes in the above lists occur.

To the knowledge of this author there does not exist any mathematical proof for that the batch ordering with nonvectorial data converges. In practice, one is always able to check whether the mapping is steady. *This has so far occurred in a finite number of updating cycles, if the neighborhood sets of the SOM do not vary with time*.

9 Appendix

Let us now consider coordinate systems that are defined by *Euclidean distances from fixed points*. On a Euclidean plane, an *ellipse* is the set of points, the sum of distances of which from two fixed points is constant. With different sum values one obtains different ellipses.

A *hyperbola* is the set of points, the difference of distances of which from two fixed points is constant. The two symmetric branches of a hyperbola are defined by differences, the magnitudes of which are equal but which have opposite signs.

By a family of elliptic and hyberbolic curves, one usually defines a *bifocal coordinate system* on a 2D Euclidean plane.

It may now be intriguing to learn that on a Euclidean plane, *the set of points that has a constant difference of squares of distances from two fixed points is a straight line*. In a Euclidean space of *arbitrary dimensionality* such a set of points defines a *hyperplane* that is orthogonal to the line passing the fixed points. Relating to a set of such differences, the corresponding hyperplanes are parallel, and their distance is a linear function of the difference. This can be shown by simple analytic geometry.

Consider an n-dimensional Euclidean space and the point $x = [x_1, x_2, ..., x_n]$ in it. Consider further two fixed points $a = [0, 0, ..., 0]$ and $b = [c, 0, 0, ..., 0]$. The squared distance between x and a we write

$$D(x, a) = {x_1}^2 + \sum_{i=2}^{n} {x_i}^2 \quad , \tag{6}$$

and the squared distance between x and b is

$$D(x, b) = (x_1 - c)^2 + \sum_{i=2}^{n} {x_i}^2 \quad . \tag{7}$$

The difference of $D(x, a)$ and $D(x, b)$, denoted d, is equal to $2cx_1 - c^2$. In other words, if d is constant, the set of the corresponding points is described by $x_1 = (d + c^2)/2c$. This is the equation of a *hyperplane* that is orthogonal to the x_1 axis and intersects it at $(d + c^2)/2c$.

It is deducible that if one has *four* fixed points that do not lie on the same straight line, one can define two families of intersecting hyperplanes that can be used for independent coordinate planes in the hyperspace. It would be desirable that these two families of hyperplanes were *orthogonal to each other*.

Motivated by the above finding, we introduced the GENINIT ordering method. However, when the metric of the source data is not Euclidean, there exist no grounds for using differences of *squared* distances in the definition of the general GENINIT method. One is free to experiment with other scale transformations.

Bibliography

[1] J. Kruskal and M. Wish, *Multidimensional scaling*. Newbury Park, CA: Sage Publications, 1978.

[2] T. Kohonen, *Self-Organizing Maps,* 3rd edition. Berlin, Heidelberg: Springer-Verlag, 2001.

[3] T. Okuda, E.Tanaka and T. Kasai, A method for the correction of garbled words based on the Levenshtein metric. IEEE Trans. C-25, 172-178, 1976.

[4] A. Bairoch and R. Apweiler, The SWISS-PROT protein sequence data bank and its supplement. TrEMBL in 1999. Nucleic Acids Res., 27:49-54, 1999.

[5] W. Pearson and D. Lipman, Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. USA, 85:2444-2448, 1988.

[6] J. Principe and R. Miikkulainen, eds.: Advances in Self-Organizing Maps. LNCS 5629. Berlin, Heidelberg, New York: Springer-Verlag, 2009.

[7] J.W. Sammon, Jr., A nonlinear mapping for data structure analysis. IEEE Trans. Comp., 18:401-409, 1969.

[8] R.N. Shepard, The analysis of proximities: multidimensional scaling with an unknown distance function. Psychometrica, 27:125-246, 1962.

[9] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrica, 29:1-27, 1964.

Aalto-ST 8/2011

ISBN: 978-952-60-4145-2 (pdf)
ISBN: 978-952-60-4144-5
ISSN-L: 1799-4896
ISSN: 1799-490X (pdf)
ISSN: 1799-4896

Aalto University
School of Science
Department of Information and Computer Science
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS