Neural Networks to Compare High Dimensional Data having Skewed and Unknown Distributions

A. Ultsch

Department of Mathematics and Computer Science,
Philipps-University Marburg, D-35032 Marburg, Germany
ultsch@informatik.uni-marburg.de

Abstract: Data Mining and Knowledge Discovery aim at the detection of new knowledge in data sets. There are (at least) two important problems associated with real world data sets: missing values and unknown distributions. This is in particular a problem for clustering algorithms, be they statistical or neuronal. In this work a novel neural network, called ud-net is defined. Ud-nets are able to adapt to unknown distributions of data. The output of the networks may be interpreted as a distance. This distance is also defined for data with missing values in some components and the resulting value is comparable with complete data sets. Experiments with known distributions that were distorted by nonlinear transformations show that ud-nets produce values on the transformed data that are comparable to distance measures on the untransformed distributions. Ud-nets are also tested with a data set from a stock picking application. For this the results of the networks are very similar to results obtained by the application of hand tuned nonlinear transformations to the data set.

1 Introduction

In many applications, in particular in data mining, we are confronted with high dimensional data sets of unknown distributions in their variables (Fayyad et al (1994)). Their distributions are often highly non-normal. For many applications distances of the multidimensional data vectors need to be compared. In particular this needs to be solved in the context of Knowledge Discovery in Databases (KDD) (Fayyad et al (1994)). Euclidean Distances are not suitable in this case since the non-linear nature of the distributions distorts heavily the distance metrics. The application of non-linear transformations to a variable is a standard method in order to transform the problematic variables to a feasible distribution. The selection of a suitable transformation is an expert task (Hartung, Elpelt (1995)). This task consumes typically most of the time necessary to process the data set. It is estimated that 80 to 90 % of the whole time of a Data Mining task with respect to working-time is spent in this so called pre-processing of the data (Mannila (1997)). In this paper we follow the approach to define an artificial neural network, called

uniform distance net (ud-net), that adapts itself to the distributions of the variables. The network learns a distance measurement we call RelativeDistance. This RelativeDistance seems to be reasonably robust against distortions in the data like the application of nonlinear functions. This paper gives the definition of this network and shows the results of the application of ud-nets to an artificially constructed example and a real world data set.

2 Uniform Distance Networks

A ud-neuron takes two real numbers x and w as input and calculates from these a number in the unit interval. The output of the neuron is interpretable as follows: an output value close to 0 means x is far away from w; an output value close to 1 means x is relatively close to w. Input to the neuron are x and $w \in \Re$ and a system clock pulse with values $q \in \{0, 1\}$. A transition of q from 0 to 1 indicates the advent of a new and valid input in w and x. While $q = 1$ the relative nearness is calculated. The main functionality of the neuron is given by the difference function

$$d = d(x, w) = |x - w|$$

and the sigmoid function th:

$$rn = th(d, m, s) = e^{-(\frac{d}{m+s})^2}.$$

The values of m and s are adjusted during the learning of the neuron (see below). The purpose of the sigmoid function is to map all outputs onto the unit interval. The function t counts the changes of q from zero to one and may be interpreted as a time index. The value dm is the difference between the current and the last value of m: $dm = m - m_l$.

For ud-neurons the learning rules update in particular the values of m and s. For m learning is as follows:

$$m_0 = w(1);$$
$$m = m_l + \Delta_m \text{ with } \Delta_m = \tfrac{1}{t}(x - m_l),$$

where m_l is the last m value. For s the learning rule is as follows

$$s_0 = 0;$$
$$s = \sqrt{s_l^2 + t \cdot \Delta_m^2 - \tfrac{1}{t-1} \cdot s_l^2},$$

where s_l is the last value of s. For the calculation of a RelativeDistance between x and w, the output value of a ud-neuron may be "inverted". In this case after the sigmoid function an inverter I ($rd = 1 - rn$) is

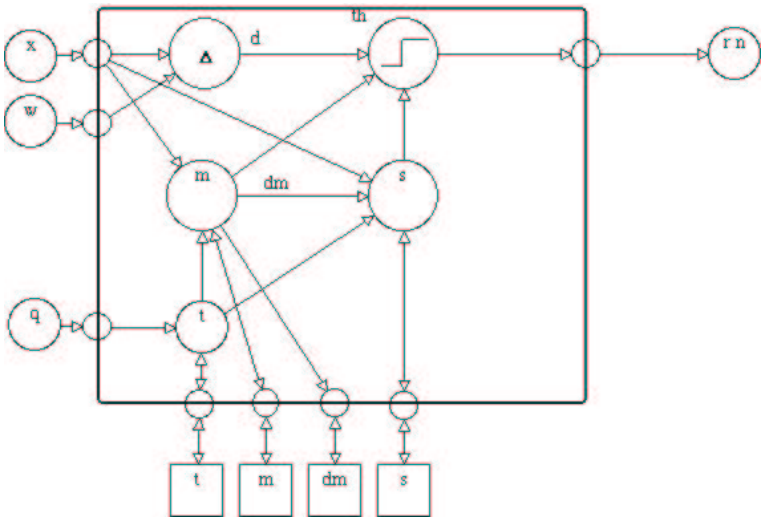

Figure 1: Circuit diagram of a ud-neuron

build into the ud-neuron. Since the RelativeDistances measured in each component of the vectors are all in the same unit interval, the square root of the sum (srd) of all rd of all components is a possible measure of RelativeDistance. In the above definition the maximum of RelativeDistances in one component is at most 1. A value of d as RelativeDistance means, roughly spoken, that vectors x and w are different in about d aspects (dimensions). If srd is divided by the dimensionality of each vector pair x, w the resulting value may be compared to srd of other vector pairs with different dimensions. This allows, for example, to measure valid and comparable distances even in the case of missing values (Ultsch, Rolf (2000)).

3 Ud-nets measuring Distances in Data with different Distributions

To analyze ud-nets in the case of skewed distributions, a data set E, consisting of 300 three-dimensional data vectors, was generated. In each variable the values were uniformly distributed in [0;1]. Euclidean Distances between all pairs of vectors in E are normally distributed and so are all RelativeDistances. As a first experiment a logarithmic function was applied to the variable y (lny = $ln(y)$) and an exponential transformation to z (expz = e^z). This transformed data set is called T1. The distribution of Euclidean Distances is shifted from normal in the untransformed E to a skewed distribution. Comparing the Euclidean Distances measured after transformation T1 to the original distances gives Figure 2.

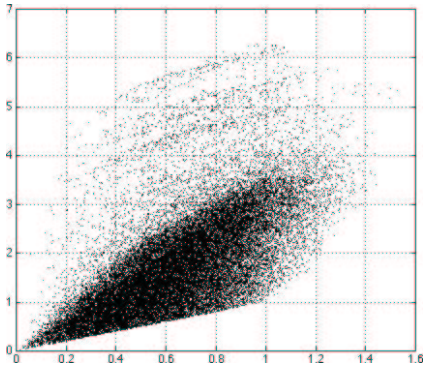

Figure 2: Euclidean Distances measured after transformation T1 vs. the original Euclidean Distances

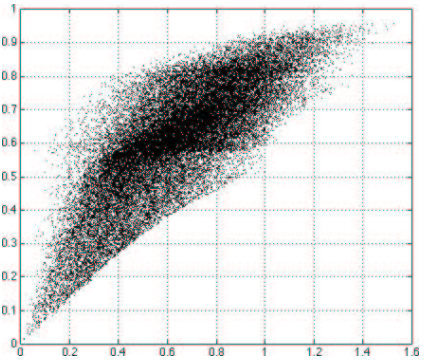

Figure 3: Euclidean Distances on T1 vs. RelativeDistances on the untransformed set

Pearson's correlation coefficient for the correlation between Euclidean Distances in E and in T1 is 0.66. The detailed relationship between Euclidean Distances in the original set E and RelativeDistances in the transformed set T1 shows Figure 3.

Pearson's correlation coefficient for the correlation between Euclidean Distances in E and RelativeDistances in T1 is in this case 0.81. If the data set T is transformed into T2 = $(x, ln(y), ln(z))$, the same phenomena can be observed: the correlation between Euclidean Distances in E and Euclidean Distances in T2 is relatively small (0.59), while the correlation between Euclidean Distances in E and RelativeDistances in T2 remains high (0.71). In this case the "gain" in correlation percentages is 22 %. Similar results could be obtained by other combinations of ln, sqrt, squaring and exponentiation as transformations.

4 Ud-nets in a Practical Example

In this section we apply ud-nets to a data set stemming from a financial problem domain: stock market analysis (Deboeck, Ultsch (1999)). The data set used consists of 331 real valued vectors describing companies emitting stocks at US-American stock exchanges. The data for each stock consists of 17 variables. Quantile/quantile (Q/Q) plots of the variables exhibit that, except for the first variable (Held), all distributions are rather skewed (see Figure 4).

Transformations like square root and logarithmic transformations to regularize the distributions were hand selected. The effect of these transformations can also be seen in the distributions of the distance values.

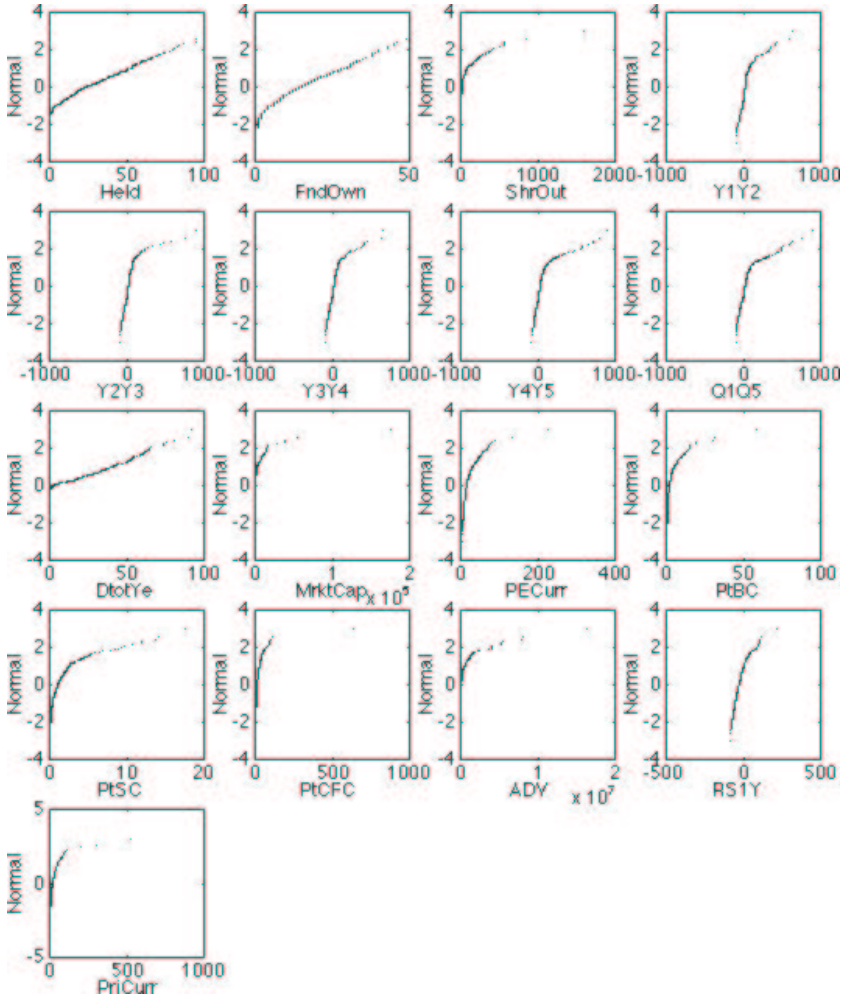

Figure 4: Q/Q plots of the untransformed distributions

While Euclidean Distances in the original data set are non uniformly distributed, the distribution of RelativeDistances is close to normal (Ultsch (1999b)). The sensitivity of Euclidean Distance to the non-linear transformations can be estimated by the Pearson's correlation factor of 0.67.

Figure 5 gives the relation of Euclidean Distance of the transformed data vs. Euclidean Distance of untransformed data. It can be seen that the correlation is very weak for larger distances. The most important observation is how ud-net distances of untransformed data sets correlate to Euclidean Distances of the transformed data. Figure 6 plots these distances against each other. It can be seen that there is a rather strong correlation which is also visible in a Pearson correlation factor of 0.79.

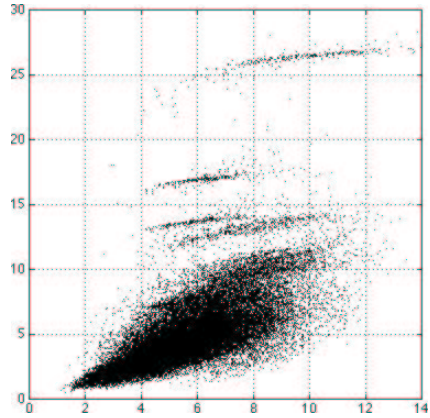

Figure 5: Euclidean Distance of transformed data vs. Euclidean Distance of untransformed data

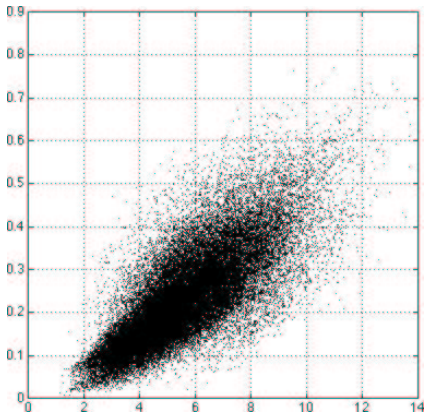

Figure 6: Euclidean Distance of transformed data vs. RelativeDistance of untransformed data

Note also that the "smoke stack cloud" of Figure 6 gets smaller near the origin. This indicates that for small distances correlation is even higher.

5 Discussion

As testbed for the claimed robustness of the distances measured by ud-nets an artificially generated example of 300 points was used. Non-linear transformations like ln, exp, etc. distort Euclidean Distances heavily. A Pearson correlation between the original distances and the Euclidean Distances in the transformed data is in the range of 0.6. If distances are learned by a ud-net, the correlation between the Euclidean Distances in the untransformed set and RelativeDistances in the transformed data remains relatively high, i. e. in the 0.8 range. Comparing the plots of the original distances vs. RelativeDistances show that for small distances the correlation remains even higher.

Ud-nets have been applied to a dataset from an application in the domain of stock marketing. Except one, all of the variables showed non-normal distributions. For each of these variables non-linear transformations like sqrt and ln had been hand selected, such that the transformed variables were sufficiently close to a normal distribution. The Euclidean Distances of transformed vs. untransformed data points decorrelate more and more as the distances increase. RelativeDistances on the untransformed data set measured with ud-nets show a high correlation to the Euclidean Distances measured in the transformed data set (consider also Figure 6).

This is a strong indication that trained ud-nets are able to measure dis-

tances in a high dimensional vector space that are close to the "real" distances of the vectors, even if the distribution in the are rather skewed. This property allows to use ud-nets as distance metric on original, untransformed data sets.

6 Conclusion

In particular in the context of Knowledge Discovery in Databases a problem arises when it's necessary to compare high dimensional data points of unknown distributions in their variables. Euclidean Distances are misleading if these distributions are highly non-normal. This problem is typically overcome by a statistical expert who hand-selects and adapts a non-linear transformation like ln, exp, etc. to each of the variables (Hartung, Elpelt (1995)).

Due to the very different distributions observed in practice an analytical definition of a distance-metric that is suitable for all situations seems to be impractical. In this work an adaptive neural network, called ud-net, is proposed. It adapts itself to the distribution of each variable. After its learning phase a ud-network produces an output that can be interpreted as a RelativeDistance. Using an artificially generated example with known properties and a data set from a real application we could demonstrate that this learned distance is reasonably robust against standard types of non-linear transformations.

As practice shows, missing values are present in almost all data sets from real applications (Little, Rubin (1987)). Ud-nets may be used in these algorithms to produce valid results for such incomplete data sets. In other works we could show that ud-nets are superior to standard approaches for dealing with missing and/or incomplete values (Ultsch, Rolf (2000)).

Acknowledgment
The author whishes to thank Guido Deboeck, author of "Visual Explorations in Finance using self-organizing maps" and "Trading on the Edge" for the provision of the financial data set.

References

DEBOECK, G. and ULTSCH, A. (1999): Picking Stocks with Emergent Self-organizing Value Maps, to appear in Proc. PASE 2000.

FAYYAD, U.S. and UTHURUSAMY, R. (Eds.) (1994): Knowledge Discovery in Databases; Papers from the 1994 AAAI Workshop. Menlo Park, CA, AAAI Press.

HARUNG, J. and Elpelt, B. (1995): Multivariate Statistik, Oldenbourg, Wien.

LITTLE, R.J.A. and RUBIN, D.B. (1987): Statistical Analysis with Missing Data, Wiley, New York.

MANNILA, H. (1997): Methods and Problems in Data Mining, in: Afrati, F., Kolatis, P. (Eds.): Proc. ICDT, Delphi, Greece, Springer, pp. 41–55.

ULTSCH, A. (1999): Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series, in: Oja, E., Kaski, S. (Eds.): Kohonen Maps, Elsevier, Amsterdam, pp. 33–45.

ULTSCH, A. (1999): Neural Networks for Skewed Distributions, Technical Report 12/99, Department of Mathematics and Computer Science, University of Marburg.

ULTSCH, A. and ROLF, S. (2000): The completion of Missing Values by Neural Nets for Data Mining, to appear in Proc. GfKL, Passau, 2000.