

Similarity Classifier based on Schweizer & Sklars t-norms

P. Luukka and J. Sampo

Laboratory of Applied Mathematics
 Lappeenranta University of Technology
 P.O. Box 20, FIN-53851 Lappeenranta, Finland
 Email:(pasi.luukka, jouni.sampo)@lut.fi

Abstract: In this article we have applied Schweizer & Sklars t-norm based similarity measures to classification task. We will compare results to fuzzy similarity measure based classification and show that sometimes better results can be found by using these measures than fuzzy similarity measure. We will also show that classification results are not so sensitive to p values with Schweizer & Sklars measures than when fuzzy similarity is used. This is quite important when one does not have luxury of tuning these kind of parameters but needs good classification results fast.

Keywords: Classification, Similarity, Differential evolution algorithm

1. Introduction

Because similarity is an equivalence relation that can be used to classify multi-valued objects, it is suitable for classifying problems that are possible to classify based on the clustering by finding similarities in objects. In instance-based classification methods the selection of a similarity measure is a critical consideration. Some times poor results, which has been got from the instance-based methods may actually originate in the underlying similarity measure, not the method itself.

In this article we have tested implications created by Schweizer and Sklar [1],[2] to our similarity classifier. Usually we have used fuzzy similarity measures in our classifier [3], [4],[5], [6] but also other measures can be used.

The data sets were chosen as diverse as possible so that classifiers properties would be apparent. Data sets were taken from a UCI Repository of Machine Learning Database [7] archive so that they were differently distributed and their dimensions varied. Classifier was implemented with MATLABTM-software.

2. Mathematical Background

Fuzzy similarity in Łukasiewicz structure can be defined as follows:

$$S_1(x_1, x_2) = \frac{\sum_{i=1}^n W_i \sqrt[1 - |x_1^p(f_i) - x_2^p(f_i)|]}{\sum_{i=1}^n W_i} \quad (1)$$

For more information about using fuzzy similarity in classification and more thorough mathematical background can be found in [4]. In this article we concentrate more on using Schweizer & Sklar's [1],[2] measures which are not applied to classification before to our knowlegde.

We are examining a choice situation where features of different objects can be expressed in values between [0,1]. Let X be the set of m objects. If we know the similarity value of the features f_1, \dots, f_n between objects, we can choose the object that has the highest total similarity value. The problem is to find for object x_i similar object x_j , where $1 \leq i, j \leq m$ and

$i \neq j$. By choosing structure defined by Schweizer & Sklars for features of the objects we get n similarities for comparing two objects (x_1, x_2) :

$$S_2(x_1, x_2) = \frac{1}{n} \sum_{i=1}^{dim} w_i \sqrt[p]{A(x_1) + B(x_2) - C(x_1, x_2)} \quad (2)$$

where $A(x_1) = (1 - x_1(f_i))^p$, $B(x_2) = (1 - x_2(f_i))^p$ and $C(x_1, x_2) = ((1 - x_1(f_i))(1 - x_2(f_i)))^p$

$$S_3(x_1, x_2) = \frac{1}{n} \sum_{i=1}^{dim} w_i e^{(-\sqrt[p]{|\ln x_1(f_i)|^p + |\ln x_2(f_i)|^p})} \quad (3)$$

$$S_4(x_1, x_2) = \frac{1}{n} \sum_{i=1}^{dim} w_i \frac{x_1(f_i)x_2(f_i)}{\sqrt[p]{x_1(f_i)^p + x_2(f_i)^p - (x_1(f_i)x_2(f_i))^p}} \quad (4)$$

From now on these measures are called as S_1, S_2, S_3 and S_4 .

3. A Classifier

A classifier based on previously presented similarity measures is implemented. It simply compares the data vector to the ideal vector that represents the class as well as possible. In the algorithmic form, a classifier is:

```

Require: test,learn[1...n],weights,dim
scale test between [0, 1]
scale learn between [0, 1]
for  $i = 1$  to  $n$  do
    idealvec[i] = IDEAL[learn[i]]
    maxsim[i] =  $\frac{\sum_{j=1}^{dim} weights[j] \sqrt[p]{1 - |idealvec[i][j]^p - test[j]^p|}}{\sum_{j=1}^{dim} weights[j]}$ 
end for
class = arg max $_i$  maxsim[i]
    
```

In the algorithm, the similarity relation based on the Łukasiewicz structure is used. However, it can be replaced with any other similarity relation in very straightforward way. $IDEAL[i]$ is the vector that best characterizes the class i and in this paper we have used the mean vector of the class as an $IDEAL$ -operator. For this algorithm, weights

This paper is partially supported by East Finland Universities Graduate School in Computer Science and Engineering (ECSE) and Tekniikan Edistämissäätiö.

can be optimized for example by using Differential evolution algorithm in optimization. In following there is a short description of Differential evolution algorithm.

3.1. Differential evolution

The basic idea of evolutionary algorithms is that we create a population V_0 of trial solutions (vectors) for optimization problem. Next we combine the members of V_0 in a certain way and check if combined solutions are better than original trial solutions in V_0 . Best solutions continue to population V_1 , the next step of evolution and then the procedure starts again.

Differential evolution (DE) can be considered as quite new algorithm, since it was developed around 1995 by Storn and Price [8]. Unlike genetic algorithms (GA), DE works directly with continuous variables without encoding and decoding of vectors.

We denote the population at evolution step n by set V_n . Fitness function $f : \mathbb{R}^D \rightarrow [0, 1]$ is set so that if $f(\mathbf{v}) < f(\mathbf{u})$ then \mathbf{v} is better solution for our problem than \mathbf{u} .

As GA the basic DE contains also crossover operation but it has not a real mutation operation. Instead of mutation, DE has *differential variation* operation. So in one evolution step we make the following two basic operations for all $\mathbf{v}_n = \{v_1, \dots, v_D\} \in V_n$:

1. *Differential variation.* The basic idea of differential variation is to add "noise vector" \mathbf{n} to vector $\mathbf{w} \in V_n$. Both \mathbf{n} and \mathbf{w} can be chosen in many ways. In our DE \mathbf{n} is difference vector between two randomly chosen vector from V_n . Vector \mathbf{w} is chosen to be best vector in V_n . From \mathbf{n} and \mathbf{w} we form a new vector

$$\mathbf{u} = \mathbf{w} + F\mathbf{n}, \quad (5)$$

where usually parameter F is constant and between 0 and 1.

2. *Crossover.* From $\mathbf{u} = \{u_1, \dots, u_D\}$ and \mathbf{v}_n we form a trial vector $\mathbf{t} = \{t_1, \dots, t_D\}$ by setting

$$t_i = \begin{cases} u_i & \text{if } x_i < CR \\ v_i & \text{otherwise} \end{cases} \quad (6)$$

where $x_i \in [0, 1]$ is uniformly distributed random variable and CR is crossover probability. Usually values $F = 0.9$ and $CR = 0.8$ works pretty well [10]. Finally the vector \mathbf{v}_{n+1} which is chosen to next generation V_{n+1} is

$$\mathbf{v}_{n+1} = \begin{cases} \mathbf{t} & \text{if } f(\mathbf{t}) < f(\mathbf{v}) \\ \mathbf{v}_n & \text{otherwise} \end{cases} \quad (7)$$

3.2. Data sets

We used three different data sets which are all freely available in [7]. The fundamental properties of the data sets are shown in Table 1.

All three data sets were splitted in half; one half was used for training and the other for testing the classifier.

Table 1. Test data sets and their properties.

Name	No classes	Dimen	No cases
Thyroid data set	3	5	215
wave form data set	3	21	5000
Image data set	7	19	2100

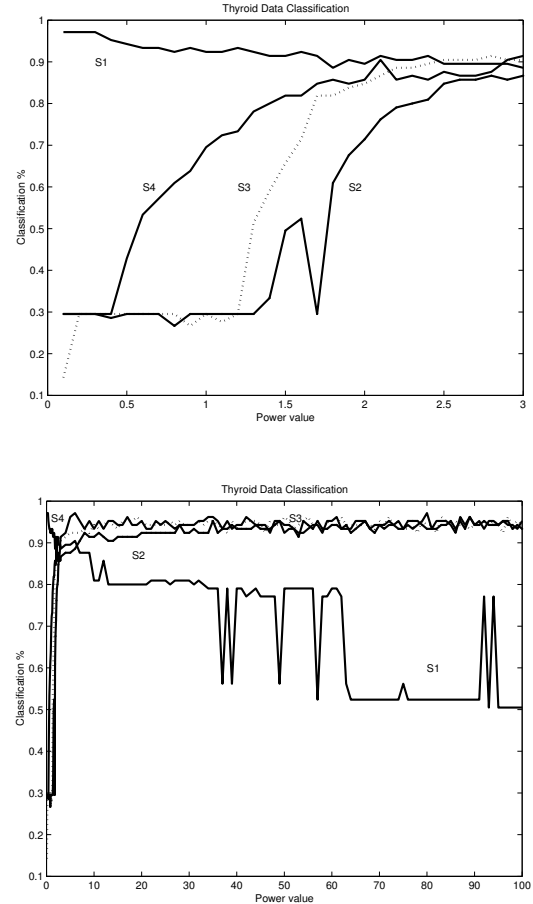


Fig. 1. Classification results for Thyroid data set

4. Empirical Classification Results

Classification results can be seen in figures 1 – 3. Figures are represented in two parts. In first part classification results with small p values are presented and second part much wide range of classification results with respect to p values are represented.

Thyroid data set: Classification results can be seen in figure 1. Similarity based on Łukasiewicz structure (S_1) seems to work best with small p values when other three similarities did not work so well with small similarities but their classification results improved when p value got bigger. Results between $p=[35 \ 120]$ had little difference with Schweizer & Sklars equations. Maximum classification results were following: $S_1 = 97.14\%$, $S_2 = 96.19\%$, $S_3 = 96.19\%$ and $S_4 = 97.14\%$. On average best results was achieved with S_4 .

Waveform data set: Classification result for wave form

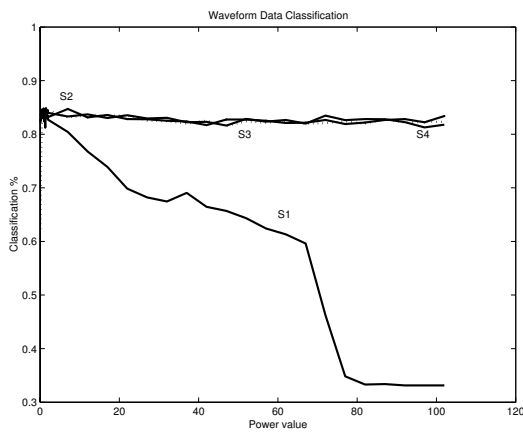
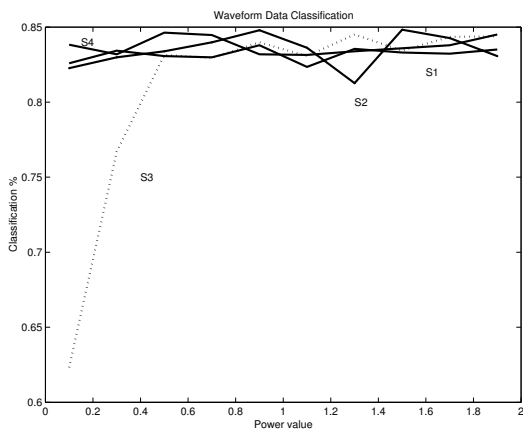


Fig. 2. Classification results for waveform data set

data set can be seen in figure 2. As seen from the figure classification results with S_1 worked again well when p values were small but got worse quite rapidly when power value increased. With other measures S_2 and S_4 got quite stable results and power value had very little effect on classification results. With S_3 at small p values results were not so good but quite rapidly they got better when p value was increased. Maximum classification results were following: $S_1 = 83.79\%$, $S_2 = 84.83\%$, $S_3 = 84.51\%$ and $S_4 = 84.63\%$.

Image data set: Classification results for image data set can be seen in figure 3. With image data set fuzzy similarity S_1 gave best classification result with small p values. S_2 gave quite stable results again with all tested p values and S_3 worked again not so good with small p values but results enhanced after p value increased. S_4 gave quite bad results with this data set and did not work at all. Maximum classification results were following: $S_1 = 83.67\%$, $S_2 = 78.57\%$, $S_3 = 79.59\%$ and $S_4 = 29.59\%$

5. Conclusions

In this study we have shown that similarity measures constructed from Schweizer & Sklar's t-norm are very useful in classification. They clearly have something new to offer in

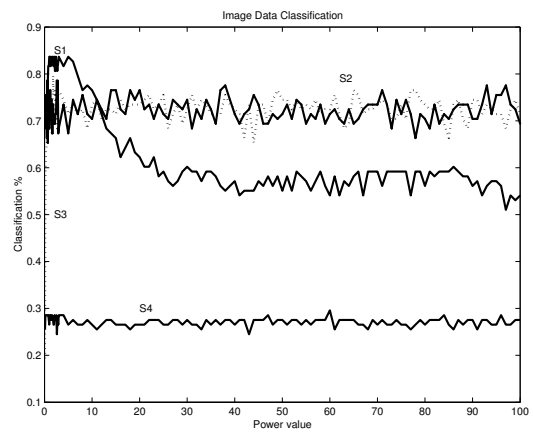
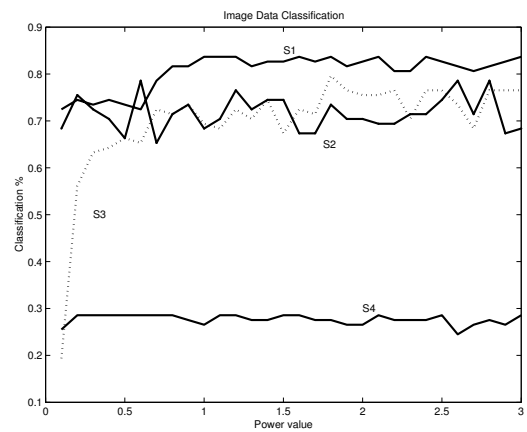


Fig. 3. Classification results for Image data set

classification compared to traditional similarity measures. Another interesting result was that these measures are not so sensitive to p value but much longer range of good classification results can be achieved by these measures compared to usually used fuzzy similarity measure.

The major advantages of the method is that it provides semantic information about the classification task by allowing partial membership of the class.

References

- [1] B. Schweizer, & A. Sklar, *Triangle Inequalities in a Class of Statistical Metric Spaces* Journal London Math. Soc. 38 1963, pp. 401-406.
- [2] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, 1995
- [3] V. K on onen, P. Luukka and K. Saastamoinen, *New Classifier Based on Maximal Fuzzy Similarity*. Unpublished.
- [4] P. Luukka, K. Saastamoinen, V. K on onen, *A Classifier Based on the Maximal Fuzzy Similarity in the Generalized Lukasiewicz-structure*. Paper published in the proceedings of the FUZZ-IEEE 2001 conference, Melbourne, Australia, 2001
- [5] K. Saastamoinen, V. K on onen and P. Luukka, *A Clas-*

sifier Based on the Fuzzy Similarity in the Lukasiewicz-Structure with Different Metrics. Paper published in the proceedings of the FUZZ-IEEE 2002 Conference, Hawaii, USA.

- [6] K. Saastamoinen and P. Luukka, *Classification in the Lukasiewicz Algebra with Different Means* paper published in the proceedings of FUZZ-IEEE 2003 conference, St Louis, USA 2003.
- [7] UCI Repository of Machine Learning Databases network document. Referenced 4.11.2003. Available: <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- [8] K.V. Price, *Differential Evolution: A Fast and Simple Numerical Optimizer*. Biennial Conference of the North American, 1996.
- [9] P. Luukka and J. Sampo, *Weight Optimization Using Differential Evolution and Genetic Algorithm in Classification of High Dimensional Data with Arithmetic, Geometric and Harmonic Similarity*, Published in the proceedings of InTech03 Fourth International Conference on Intelligent Technologies, Chiang Mai, Thailand 2003.
- [10] J. Lampinen, Laboratory of Information Technology, Lappeenranta University of Technology, personal communication 2003.