Probabilistic Prototype Classification Using t-norms

Tina Geweniger, Frank-Michael Schleif, and Thomas Villmann

Computational Intelligence Group, University of Applied Sciences Mittweida,
Technikumplatz 17, 09648 Mittweida, Germany

Abstract. We introduce a generalization of Multivariate Robust Soft Learning Vector Quantization. The approach is a probabilistic classifier and can deal with vectorial class labelings for the training data and the prototypes. It employs t-norms, known from fuzzy learning and fuzzy set theory, in the class label assignments, leading to a more flexible model with respect to domain requirements. We present experiments to demonstrate the extended algorithm in practice.

1 Motivation

Uncertainty is a general effect of most datasets and should not be neglected during learning. In this article we focus on classification problems where the uncertainty occurs in the data as well as in the label information. Both aspects have been addressed before in the field of probabilistic learning or using fuzzy sets [1,2]. However, often the obtained models are quite complex or lack sufficient flexibility to integrate additional expert knowledge. Recently, a multivariate formulation of Robust Soft Learning Vector Quantization (MRSLVQ) was proposed in [1] and independently recovered in [3], providing an interpretable prototype based model. Another alternative can be found in [4] for a so called *fuzzification* of Soft Nearest Prototype Classification for fuzzy labeled data and prototypes.

Prototypes are compact representations of a larger set of points, like the mean of a set of points, and partition the data space e. g. into disjunct regions. They can easily be inspected by experts and summarize large complex data sets.

All these models share many positive aspects of prototype based learning [5], such as metric adaptation [6], kernelization [5] or the processing of dissimilarity data [7]. Here we will focus on the MRSLVQ, although the presented concepts can be transferred to other approaches straight forward. The original formulation of MRSLVQ uses a multiplicative assignment rule for the class label assignments or fuzzy label membership which we will be replaced by the more generic concept of $t-$norms in this work. $t-$norms occur in the field of fuzzy logic to model boolean set operations for decision rules. It is assumed that the conjunction $\wedge$ is interpreted by a triangular norm ($t-$norm for short) and the disjunction $\vee$ is interpreted by a triangular co-norm ($t-$co-norm). In the considered classification task we like to express that the label of its closest prototype is consistent with the label of a data point which can be modeled by a logical conjunction and which is approximated by a corresponding $t-$norm.

In the following, first we will give a brief overview of MRSLVQ and review different t-norms. Subsequently we extend MRSLVQ by t-norms and show the effectiveness of the approach for two datasets with unsafe label information.

T. Villmann et al. (eds.), *Advances in Self-Organizing Maps and Learning Vector Quantization,* Advances in Intelligent Systems and Computing 295,
DOI: 10.1007/978-3-319-07695-9_9, © Springer International Publishing Switzerland 2014

2 Multivariate Robust Soft LVQ

The Robust Soft LVQ algorithm (RSLVQ) was introduced in [8] as a probabilistic prototype classifier. It is assumed that the probability density $p(\boldsymbol{v})$ of the data points $\boldsymbol{v} \in \mathbb{R}^d$, with d being the data dimensionality, can be described by a Gaussian mixture model. Every component of the mixture is assumed to generate data which belongs to only one of the N_C classes. The classification itself is based on a winner takes all scheme. The probability density of all the data points is given by

$$p(\boldsymbol{v}|W) = \sum_{k=1}^{N_C} \sum_{j:y_j=k}^{N_P} p(\boldsymbol{v}|j)P(j) \tag{1}$$

where $W = \{(\boldsymbol{w}_j, y_j)\}_{j=1}^{N_P}$ is the set of N_P labeled prototype vectors $\boldsymbol{w}_j \in \mathbb{R}^d$ and their assigned crisp class labels y_j. $P(j)$ stands for the probability that data points are generated by component j of the mixture and is commonly set to an identical value for all the prototypes. The conditional density $p(\boldsymbol{v}|j)$, which describes the probability that component j is generating a particular data point $\boldsymbol{v}$, is a function of the prototype $\boldsymbol{w}_j$ itself. The density $p(\boldsymbol{v}|j)$ can be chosen to have the normalized exponential form $p(\boldsymbol{v}|j) = K(j) \cdot e^{f(\boldsymbol{v}, \boldsymbol{w}_j, \sigma_j^2)}$ where $K(j)$ is the normalization constant and the hyper parameter σ_j^2 the width of component j.

The aim of RSLVQ is to place the prototypes such that a given data set is classified as accurately as possible. Therefore the likelihood ratio

$$L = \prod_{i=1}^{N_V} L(\boldsymbol{v}_i, c_i) \ , \qquad \text{with } \ L(\boldsymbol{v}_i, c_i) = \frac{p(\boldsymbol{v}_i, c_i|W)}{p(\boldsymbol{v}_i|W)} \tag{2}$$

where N_V is the number of data points, has to be maximized. The ratio is built up of the particular probability density $p(\boldsymbol{v}_i, c_i|W)$, that data point $\boldsymbol{v}_i$ is generated by a mixture component of the correct class c_i

$$p(\boldsymbol{v}_i, c_i|W) = \sum_{j:y_j=c_i} p(\boldsymbol{v}_i|j)P(j) \tag{3}$$

with the total probability density $p(\boldsymbol{v}_i|W)$

$$p(\boldsymbol{v}_i|W) = \sum_j p(\boldsymbol{v}_i|j)P(j). \tag{4}$$

The cost function is given as

$$E_{RSLVQ} = \sum_{i=1}^{N_V} \log\left(\frac{p(\boldsymbol{v}_i, c_i|W)}{p(\boldsymbol{v}_i|W)}\right). \tag{5}$$

with learning rules as presented in [8].

While Robust Soft Learning Vector Quantization is very effective, it is only applicable for crisp labeled training data. An extension of this approach based on a vectorial adaption scheme for handling fuzzy labeled training data was presented in [9] leading to the following modifications:

2.1 Cost Function

The assumption of fuzzy labeled data points requires an adaption of the original RSLVQ algorithm. The originally crisp class label c_i for training data point $\boldsymbol{v}_i$ becomes a N_C-dimensional vector $\boldsymbol{c}_i$ of assignment probabilities with $\sum_{k=1}^{N_C} c_i^k = 1$ and $c_i^k \geq 0$. For RSLVQ, each prototype $\boldsymbol{w}_j$ describes exactly one class. Now we relax this condition and allow the prototypes to be (partial) representatives for different classes. Analogously to the notation for the data points, the class memberships of the prototypes are now expressed in vector notation yielding $\boldsymbol{y}_j$ with $\sum_{k=1}^{N_C} y_j^k = 1$ and $y_j^k \geq 0$. The classification of untrained data is still based on a winner takes all scheme. Taking the fuzzy class assignments of the data points into account, the particular probability density $p(\boldsymbol{v}_i, c_i|W)$ with crisp data labels c_i specified in equation (3) changes to

$$p(\boldsymbol{v}_i, \boldsymbol{c}_i|W) = \sum_{k=1}^{N_C} c_i^k \sum_{j=1}^{N_P} y_j^k \cdot p(\boldsymbol{v}_i|j)P(j) \tag{6}$$

where $p(\boldsymbol{v}_i, \boldsymbol{c}_i|W)$ now is the particular probability density that data point $\boldsymbol{v}_i$ is generated by the mixture components referred to by $\boldsymbol{c}_i$. Thereby, due to the factor c_i^k only a fraction of the sum of the respective probability densities is taken into account. The factor y_j^k ensures that only those prototypes are accounted for, which actually are representatives for the respective class.

The total probability density $p(\boldsymbol{v}_i|W)$ (4)

$$p(\boldsymbol{v}_i|W) = \sum_j p(\boldsymbol{v}_i|j)P(j)$$

is the probability that data point $\boldsymbol{v}_i$ is generated by *any* prototype. It is the sum over all prototypes independent of matching class assignments and, therefore, does not change.

The cost function of the Multivariate RSLVQ (MRSLVQ) can now be defined as

$$E_{MRSLVQ} = \sum_{i=1}^{N_V} \log\left(\frac{p(\boldsymbol{v}_i, \boldsymbol{c}_i|W)}{p(\boldsymbol{v}_i|W)}\right). \tag{7}$$

2.2 Derivation of Learning Rules

In order to optimize the classification, the cost function (7) has to be minimized, which can be done by a stochastic gradient descent.

Considering an universal parameter Θ with $\Theta \neq \boldsymbol{v}_i$ a general update rule can be derived:

$$\frac{\partial \log \frac{p(\boldsymbol{v}_i, \boldsymbol{c}_i|W)}{p(\boldsymbol{v}_i|W)}}{\partial \Theta_j} = (P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i) - P(j|\boldsymbol{v}_i))\left(\frac{1}{K(j)}\frac{\partial K(j)}{\partial \Theta_j} + \frac{\partial f(\boldsymbol{v}_i, \boldsymbol{w}_j, \sigma_j^2))}{\partial \Theta_j}\right). \tag{8}$$

The terms $P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i)$ and $P(j|\boldsymbol{v}_i)$ in (8), which are assignment probabilities, yield:

$$P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i) = \frac{\sum_k c_i^k y_j^k P(j)K(j)e^{f(\boldsymbol{v}_i,\boldsymbol{w}_j,\sigma_j^2,\lambda_j)}}{p(\boldsymbol{v}_i,\boldsymbol{c}_i|W)} \tag{9}$$

$$P(j|\boldsymbol{v}_i) = \frac{P(j)K(j)e^{f(\boldsymbol{v}_i,\boldsymbol{w}_j,\sigma_j^2,\lambda_j)}}{p(\boldsymbol{v}_i|W)} \tag{10}$$

$P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i)$ is the assignment probability of $\boldsymbol{v}_i$ to component j taking the partial class assignments of the data points and the prototypes into account. $P(j|\boldsymbol{v}_i)$ is the assignment probability of $\boldsymbol{v}_i$ to component j independent of the class membership.

Assuming the special case of a Gaussian mixture model with $P(j) = 1/N_P \forall j$, the similarity function is set to $f(\boldsymbol{v}_i,\boldsymbol{w}_j,\sigma_j^2) = \frac{d(\boldsymbol{v}_i,\boldsymbol{w}_j)}{2\sigma_j^2}$. Thereby, $d(\boldsymbol{v}_i,\boldsymbol{w}_j)$ is the distance between data point $\boldsymbol{v}_i$ and prototype $\boldsymbol{w}_j$, and $K(j)$ a normalization constant which can be set to $K(j) = (2\pi\sigma_j^2)^{(-N/2)}$.

The original RSLVQ algorithm uses the squared Euclidean distance as dissimilarity measure. In the following the update rules for the prototypes $\boldsymbol{w}_j$ and a hyper parameter σ_j^2 employing a general distance are derived. Afterwards the update rules based on specific distance measures are given.

To obtain the update rules for specific, cost function relevant parameters, Θ_j has to be substituted.

Updating the prototypes $\boldsymbol{w}$

Replacing Θ_j in (8) by the prototype $\boldsymbol{w}_j$ yields

$$\frac{\partial \log \frac{p(\boldsymbol{v}_i,\boldsymbol{c}_i|W)}{p(\boldsymbol{v}_i|W)}}{\partial \boldsymbol{w}_j} = \left(P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i) - P(j|\boldsymbol{v}_i)\right)\left(\frac{1}{2\sigma_j^2}\frac{\partial d(\boldsymbol{v}_i,\boldsymbol{w}_j)}{\partial \boldsymbol{w}_j}\right). \tag{11}$$

Updating the prototype labels $\boldsymbol{y}$

Analogously, the update rule for the fuzzy prototype labels $\boldsymbol{y}_j$ is obtained as

$$\frac{\partial \log \frac{p(\boldsymbol{v}_i,\boldsymbol{c}_i|W)}{p(\boldsymbol{v}_i|W)}}{\partial \boldsymbol{y}_j} = \left(\frac{\boldsymbol{c}_j}{P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i)} - \frac{1}{p(\boldsymbol{v}_i|W)}\right)\left(P(j)p(\boldsymbol{v}|j)\right). \tag{12}$$

3 T-Norms

T-norms are a generalization of the triangular inequality of metrics and were introduced by Menger [10]. They can also be used as generalizations of the Boolean logic conjunctive 'AND' operator to multi-valued logic. Applied in fuzzy logic t-norms represent the union of fuzzy sets. Its dual operation t-co-norm analogously refers to the 'OR' operator and can be used to represent the intersection of fuzzy sets. T-norms are widely used in fuzzy set theory with multiple applications [11,12,13]. Recently, $t-$norms have also been analyzed in alternative frameworks [14,15], motivating their use in general classification methods as shown here.

3.1 Definition of General t-norms

A t-norm is a dual function $\top : [0,1] \times [0,1] \to [0,1]$ to generalize the triangle inequality of ordinary metric spaces and has the following properties:

1. Commutativity $\top(a,b) = \top(b,a)$
2. Monotonicity $\top(a,b) \leq \top(c,d)$, if $a \leq c$ and $b \leq d$
3. Associativity $\top(a, \top(b,c)) = \top(\top(a,b),c)$
4. Identity $\top(a,1) = a$

According to this definition, the values of t-norms are only specified on the corner points of a unit square and along the edges. In the middle area the values are restricted to the range $[0,1]$. Therefore, there exist a variety of different t-norms. In the following a short listing of common t-norms (some of them parametrized) is given. Selected plots based on the unit square are provided in Fig. 1:

Minimum/Zadeh t-norm	$\top_{min}(a,b) = \mathtt{min}(a,b)$
Product/Probabilistic t-norm	$\top_{prod}(a,b) = a \cdot b$
Łukasiewicz t-norm	$\top_{luka}(a,b) = \mathtt{max}(a+b-1,0)$
Drastic t-norm	$\top_{drastic}(a,b) = \begin{cases} a \text{ if } b=1 \\ b \text{ if } a=1 \\ 0 \text{ otherwise} \end{cases}$
Hamacher t-norm	$\top_{ham}(a,b) = \frac{ab}{\gamma+(1-\gamma)(a+b-ab)}$ with $\gamma > 0$
Weber t-norm	$\top_{weber}(a,b) = \mathtt{max}(\frac{a+b-1+\gamma ab}{1+\gamma},0)$ with $\gamma > -1$
Yager t-norm	$\top_{yager}(a,b) = \mathtt{max}(1-((1-a)^\gamma+(1-b)^\gamma)^{\frac{1}{\gamma}},0)$ with $\gamma > 0$
Aczel-Alsina t-norm	$\top_{acz}(a,b) = \exp(-((-\log(a))^\gamma+(-\log(b))^\gamma)^{\frac{1}{\gamma}})$ with $0 < \gamma < \infty$

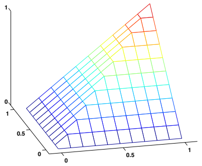
In accordance to the analysis provided in [15] we focus on the Product t-norm, the Hamacher t-norm, and the Aczel-Alsina t-norm. These three t-norms permit easy differentiation, avoiding further approximation steps as necessary in case of t-norms involving `max` operators [16]. Further, the Product t-norm was used implicitly in the original version of MRSLVQ as will be clarified in the next section.

Note that these three t-norms are related to each other:

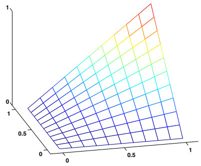
- for $\gamma = 1$ the Hamacher and the Aczel-Alsina t-norms are equivalent to the non-parametrized Product or Probabilistic t-norm

$$\top_{prod}(a,b) \equiv \top_{ham}(a,b) \equiv \top_{acz}(a,b)$$

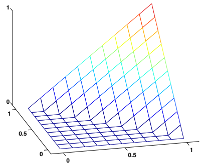
- for $\gamma \to +\infty$ (Hamacher t-norm) respectively $\gamma \to 0$ (Aczel-Alsina t-norm) the Drastic t-norm is approximated



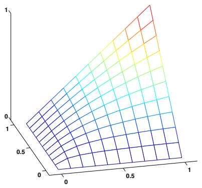
(a) Minimum t-norm



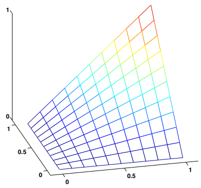
(b) Product t-norm



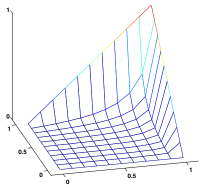
(c) Łukasiewicz t-norm



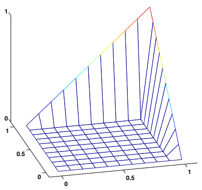
(d) Hamacher (γ=0.01)



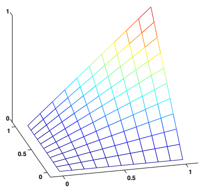
(e) Hamacher (γ=1)



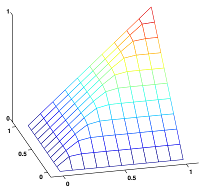
(f) Hamacher (γ=100)



(g) Aczel-Alsina (γ=0.1)



(h) Aczel-Alsina (γ=1)



(i) Aczel-Alsina (γ=10)

Fig. 1. Plots of various t-norms based on the unit square. For the parametrized Hamacher t-norm and the Aczel-Alsina t-norm three different values for the parameter γ are given.

4 Integrating t-norms in MRSLVQ

The large number of different (parametric) t-norms is due to different domain specific interpretations of the (dis-)similarity of multivariate vectors compared by a t-norm. Here we consider the (dis-)similarity between multivariate label vectors. In the MRSLVQ the authors made implicit use of the Probabilistic respectively Product t-norm in (6) by taking the fuzzy labels of the prototypes into the inner sum. Replacing the probabilistic t-norm in (6) we get:

$$p(\boldsymbol{v}_i, \boldsymbol{c}_i|W) = \sum_{k=1}^{N_C} \sum_{j=1}^{N_P} \top(c_i^k, y_j^k, \tau) \cdot p(\boldsymbol{v}_i|j)P(j) \tag{13}$$

with $\top(c_i^k, y_j^k, \tau)$ being a t-norm as defined before with a potential parameter τ. Due to the generalization to any t-norm the update of the prototype positions and the prototype labels has to be changed. Accordingly we replace in the equation of the assignment probabilities (9) the product of the fuzzy label assignments by a t-norm:

$$P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i) = \frac{\sum_k \top(c_i^k, y_j^k, \tau) P(j) K(j) e^{f(\boldsymbol{v}_i, \boldsymbol{w}_j, \sigma_j^2, \lambda_j)}}{p(\boldsymbol{v}_i, \boldsymbol{c}_i|W)} \tag{14}$$

This substitution also has to be considered in the prototype update of Eq. (11).

For the update of the fuzzy prototype labels $\boldsymbol{y}_j$ the gradient of the t-norm with respect to the prototype label $\frac{\partial \top(c_i^k, y_j^k, \tau)}{\partial \boldsymbol{y}_j}$ has to be taken into account yielding the general form

$$\frac{\partial \log \frac{p(\boldsymbol{v}_i, \boldsymbol{c}_i|W)}{p(\boldsymbol{v}_i|W)}}{\partial \boldsymbol{y}_j} = \left(\frac{\frac{\partial \top(c_i^k, y_j^k, \tau)}{\partial \boldsymbol{y}_j}}{P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i)} - \frac{1}{p(\boldsymbol{v}_i|W)} \right) (P(j) p(\boldsymbol{v}|j)) \tag{15}$$

By replacing $\top(c_i^k, y_j^k, \tau)$ by a specific t-norm the particular update rule is obtained. For example the Product t-norm yields

$$\frac{\partial \log \frac{p(\boldsymbol{v}_i, \boldsymbol{c}_i|W)}{p(\boldsymbol{v}_i|W)}}{\partial \boldsymbol{y}_j} = \left(\frac{\boldsymbol{c}_j}{P_{\boldsymbol{c}_i}(j|\boldsymbol{v}_i)} - \frac{1}{p(\boldsymbol{v}_i|W)} \right) (P(j) p(\boldsymbol{v}|j)) \,. \tag{16}$$

which is equivalent with update rule (12) as expected.

It would also be possible to update parameters of the t-norm by providing the corresponding gradients similar as for the metric adaptation or the σ learning, see e. g. [17]. For simplicity we will specify t-norm parameters using a grid search on an independent test set. In the following we focus on the before chosen parametrized t-norms Hamacher t-norm and Aczel-Alsina t-norm and provide experiments for different datasets taken from the life science domain. We compare with the approach using the standard Probabilistic t-norm, which is identical with the original MRSLVQ.

5 Experiments

We now apply the priorly derived approach to two datasets with multivariate labels. We chose the Hamacher t-norm and the Aczel-Alsina t-norm due to their easy differentiability. We show the effectiveness for a classification task and compare the results to the standard MRSLVQ approach based on the implicitly implemented Probabilistic or Product t-norm. Potential parameters of the t-norms have been optimized using a grid search on an independent test set. Using the optimized parameters the model performance was evaluated on the remaining data in a 10-fold cross-validation.

5.1 Overlapping Gaussian Distributions

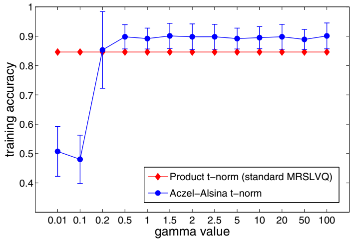
The first data set is a simulated one consisting of two overlapping Gaussian distributions. 1000 samples are drawn randomly mixed from the two distributions. The mixing coefficients are used as fuzzy labels. Applying the Aczel-Alsina t-norm, the grid search for the optimal parameter γ reveals improvements for $\gamma \geq 0.2$ compared to the standard MRSLVQ (see Fig. 2a). These improvements are measured in terms of training accuracy on randomly selected training data. The performance test was conducted on separate test data with $\gamma = 0.5$ reaching a test accuracy of 85.77% which is slightly better than standard MRSLVQ (see Tab. 1). The Hamacher t-norm turned out to be a less effective. First, the range for the grid search for the optimal parameter γ has to be enlarged to show any effect ($0.001 \leq \gamma \leq 10000$), and second, the classification accuracy of standard MRSLVQ cannot be reached (see Tab. 1).

5.2 Barley Grain Plant Data

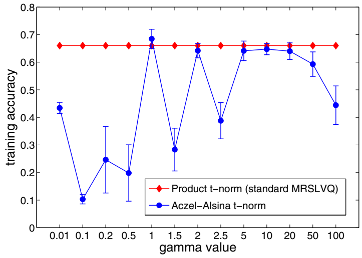
The second dataset are images of serial transverse sections of barley grains at different developmental stages. Developing barley grains consist of three genetically different tissue types: the diploid maternal tissues, the filial triploid endosperm, and the diploid embryo. Because of their functionality, cells of a fully differentiated tissue show differences in cell shape and water content and accumulate different compounds. Based on those characteristics, scientists experienced in histology are able to identify and to label differentiated tissues within a given section of a developing grain (segmentation). However, differentiating cells lack these characteristics. Because differentiation occurs along gradients, especially borders between different tissue types of developing grains often consist of differentiating cells, which cannot be identified as belonging to one or the other tissue type. Thus, fuzzy processing is highly desirable. However, since (training) examples, manually labeled by a biological expert, are costly and rarely available, one is interested in automatic classification based on a small training subset of the whole data set. In our example, the training set consists of 4418 data points (vectors) whereas the whole transverse section of the image contains 616×986 samples, which finally have to be classified and visualized as an image for immediate interpretation by biologists. The data vectors are 22-dimensional, the number of classes is $N_c = 11$. Using standard MRSLVQ based on the Product t-norm to classify the plant data yields a classification accuracy of 64.16% (see Tab. 1). Before testing our derived method the optimal parameter values were obtained again by a grid search using a training dataset and comparing the training accuracy to the standard MRSLVQ training accuracy. The plot of the accuracies obtained by the Aczel-Alsina t-norm is depicted in Fig. 2b. Interestingly, the parameter value yielding a slightly better classification accuracy than MRSLVQ is exactly that value, for which Azcel-Alsina t-norm and Product t-norm are equivalent. But nevertheless, as observed before for the Gaussian dataset, applying Aczel-Alsina t-norm with $\gamma = 1.0$ yields an improvement. For the current dataset this improvement amounts

Table 1. Average classification accuracy for the Gaussian dataset and the Barley grain plant data. Note that the non-parametric Product or Probabilistic t-norm is equivalent to the standard MRSLVQ model.

	Gaussian distributions		Barley grain plant data	
	class. acc.	γ	class. acc.	γ
Probabilistic/Product t-norm	0.8517 ± 0.0388	–	0.6416 ± 0.0317	–
Hamacher t-norm	0.8257 ± 0.0530	0.01	0.6664 ± 0.0464	0.01
Aczel-Alsina t-norm	0.8577 ± 0.0429	0.5	0.7857 ± 0.0317	1.0



(a) Gaussian dataset



(b) Barley grain plant data

Fig. 2. Grid search for the optimal parameter based on the training accuracy for MRSLVQ incorporating Aczel-Alsina t-norm. The red line indicates the training accuracy for the non-parametric Product t-norm respectively standard MRSLVQ.

to 22.5% in the test accuracy (see Tab. 1). Again, the Hamacher t-norm is less effective. Setting $\gamma = 0.01$ yields an improved classification accuracy of only 3.9% (see Tab. 1).

6 Conclusions

In this work we proposed an extension of Multivariate Robust Soft LVQ incorporating t-norms in the learning dynamic. This is the first proposal of this type for prototype based learning to the authors best knowledge. Unsafe label information is very common for many real life data but not yet sufficiently addressed by appropriate learning methods and our method is a proposal to improve the current situation. The data can reflect the fuzziness in the labeling e.g. by similar scores for different class indices. This is a very similar setting to classical fuzzy-theory and a motivation for the use of t-norms to judge the similarity of label vectors. We considered different t-norms for MRSLVQ and observed that the used t-norms might lead to (slight) improvements in the model accuracy. Especially we found that the implicitly and unwittingly used Product t-norm may not be the best choice. The Aczel-Alsina t-norm performed best in our experiments but a wider study is necessary to get a sufficient support for generic statements. In future work we will address in more detail the theoretical links of the used label norm with respect to a large margin classifier and its generalization capabilities.

Acknowledgment. Marie Curie Intra-European Fellowship (IEF): FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS) is greatly acknowledged. We would like to thank Petra Schneider for providing the code of the MRSLVQ implementation.

References

[1] Schneider, P., Geweniger, T., Schleif, F.M., Biehl, M., Villmann, T.: Multivariate class labeling in Robust Soft LVQ. In: Verleysen, M. (ed.) 19th European Symposium on Artificial Neural Networks (ESANN 2011), pp. 17–22. d-side publishing (2011)

[2] Bhattacharya, S., Bhatnagar, V.: Fuzzy data mining: A literature survey and classification framework. International Journal of Networking and Virtual Organisations 11(3-4), 382–408 (2012)
[3] Bonilla, E.V., Robles-Kelly, A.: Discriminative probabilistic prototype learning. In: ICML. icml.cc / Omnipress (2012)
[4] Geweniger, T., Villmann, T.: Extending FSNPC to handle data points with fuzzy class assignments. In: Verleysen, M. (ed.) Proc. of European Symposium on Artificial Neural Networks (ESANN 2010), Brussels, Belgium. d-side publications (2010)
[5] Schleif, F.-M., Villmann, T., Hammer, B., Schneider, P.: Efficient kernelized prototype based classification. Int. J. Neural Syst. 21(6), 443–457 (2011)
[6] Schneider, P., Biehl, M., Hammer, B.: Distance learning in discriminative vector quantization. Neural Computation 21(10), 2942–2969 (2009)
[7] Hammer, B., Schleif, F.-M., Zhu, X.: Relational extensions of learning vector quantization. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part II. LNCS, vol. 7063, pp. 481–489. Springer, Heidelberg (2011)
[8] Seo, S., Obermayer, K.: Soft learning vector quantization. Neural Computation 15, 1589–1604 (2003)
[9] Schneider, P.: Advanced methods for prototype-based classification. PhD thesis, Rijksuniveriteit Groningen (2010)
[10] Menger, K.: Statistical metrics. Proceedings of the National Academy of Sciences 28(12), 535–537 (1942)
[11] Gosztolya, G., Dombi, J., Kocsor, A.: Applying the generalized dombi operator family to the speech recognition task. CIT 17(3), 285–293 (2009)
[12] Senthil Kumar, A.V.: Diagnosis of heart disease using fuzzy resolution mechanism. Journal of Artificial Intelligence 5(1), 47–55 (2012)
[13] Ciaramella, A., Tagliaferri, R., Pedrycz, W., di Nola, A.: Fuzzy relational neural network. Int. J. Approx. Reasoning 41(2), 146–163 (2006)
[14] Quost, B., Masson, M.-H., Denaux, T.: Classifier fusion in the dempster-shafer framework using optimized t-norm based combination rules. International Journal of Approximate Reasoning 52(3), 353–374 (2011)
[15] Farahbod, F., Eftekhari, M.: Comparison of different t-norm operators in classification problems. International Journal of Fuzzy Logic Systems 2(3), 33–39 (2012)
[16] Lange, M., Villmann, T.: Derivatives of l_p-norms and their approximations. Machine Learning Reports 04/2013, pp. 43–59 (2013)
[17] Schneider, P., Biehl, M., Hammer, B.: Hyperparameter learning in probabilistic prototype-based models. Neurocomputing 73(7-9), 1117–1124 (2010)