

Fuzzy Labeled Soft Nearest Neighbor Classification with Relevance Learning

Thomas Villmann

University Leipzig, Clinic for Psychotherapy
04107 Leipzig, Germany
villmann@informatik.uni-leipzig.de

Frank-Michael Schleif

University Leipzig, Dept. of Math. and C.S.
04109 Leipzig, Germany
schleif@informatik.uni-leipzig.de

Barbara Hammer

Clausthal University of Technology, Dept. of Math. and C.S.
38678 Clausthal-Zellerfeld, Germany
hammer@in.tu-clausthal.de

Abstract

We extend soft nearest neighbor classification to fuzzy classification with adaptive class labels. The adaptation follows a gradient descent on a cost function. Further, it is applicable for general distance measures, in particular task specific choices and relevance learning for metric adaptation can be done. The performance of the algorithm is shown on synthetic as well as on real life data taken from proteomic research.

keywords: fuzzy classification, LVQ, relevance learning

1 Introduction

KOHONEN'S Learning Vector Quantization (LVQ) belongs to the class of *supervised* learning algorithms for nearest prototype classification (NPC) [8]. NPC relies on a set of prototype vectors (also called codebook vectors) labeled according to the given data classes. The prototypes locations are adapted by the algorithm such that they represent their respective classes. Such, NPC is a local classification method in the sense that the classification boundaries are approximated locally by the prototypes. The classification provided by the trained LVQ is crisp, i.e., an unknown data point is uniquely assigned to a prototype based on their similarity, which itself is uniquely related to a class. Several extensions exist to improve the basic scheme.

Recently a new method, Soft Nearest Prototype Classification (SNPC), has been proposed by SEO ET AL. [11] in which soft assignments of the data vectors for the prototypes are introduced. The determination of soft assignments are based on a Gaussian mixture approach. However, the

class labels of the prototype vectors remain crisp and they are fixed a priori as usual in LVQ.

Generally, the crisp (unique) labeling in LVQ-methods has the disadvantage that the initial prototype labeling may be not sufficient for the real class label distribution of the data points in the data space. Data with different class labels may be assigned to the same prototype (misclassifications) because the classes are overlapping. A solution could be a post-labeling of the prototype labels according to the data statistics given by all data vectors represented by the considered prototype leading to a fuzzy labeling [13]. However, this method is not appropriate for online learning, since crisp prototype label information is essential for all classical LVQ-learning schemes to determine correct and incorrect classification during prototype adaptation.

In this article we introduce a dynamic fuzzy labeling of prototypes. This has the consequence that the required information of correct or incorrect classification during learning is lost and, hence, a new learning scheme has to be established. Based on SNPC we derive an adaptation scheme for labels and prototypes such that adaptive fuzzy labeling can be achieved.

We apply the new algorithm to profiling of mass spectrometric data in cancer research. During the last years proteomic¹ profiling based on mass spectrometry (MS) became an important tool for studying cancer at the protein and peptide level in a high throughput manner. Additionally, MS based serum profiling is a potential diagnostic tool to distinguish patients with cancer from normal subjects. The underlying algorithms for classification of the mass spectrometric data are one crucial point to obtain valid and competitive results. Usually one is interested in finding decision boundaries near to the optimal Bayesian decision. Especially, for

¹Proteome - is an ensemble of protein forms expressed in a biological sample at a given point in time [1].

high-dimensional data this task becomes very complicate. Further, for cancer research it is important to get a judgement about the safety of the classification. The proposed method offers a solution to these issues introducing fuzzy classification with relevance learning.

2 Crisp learning vector quantization

Usual crisp learning vector quantization is mainly influenced by the standard algorithms LVQ1..LVQ3 introduced by KOHONEN [8] as intuitive prototype-based clustering algorithms. Several derivatives were developed to improve the standard algorithms to ensure, for instance, faster convergence, a better adaptation of the receptive fields to optimum Bayesian decision, or an adaptation for complex data structures, to name just a few [5, 9, 12].

Standard LVQ does not possess a cost function in the continuous case; it is based on the heuristic to minimize misclassifications using Hebbian learning. The first version of learning vector quantization based on a cost function, which formally assesses the misclassifications, is the Generalized LVQ (GLVQ) [10]. To be insensitive on the initialization and avoiding local minima, GLVQ can be combined with neighborhood learning yielding the Supervised Neural Gas (SNG). Further, both approaches can be extended by metric adaptation according to the given classification task. The respective relevance oriented algorithms are GRLVQ and SRNG [4],[6]. It has been shown that GLVQ as well as the extension SNG, GRLVQ, SRNG optimize the hypothesis margin [3].

Soft Nearest Prototype Classification (SNPC) has been proposed as alternative based on another cost function. It introduces soft assignments for data vectors to the prototypes in order to obtain a cost function for classification such that adaptation forms a gradient descent. In the standard variant of SNPC provided in [11] one considers as the cost function

$$E(\mathcal{S}, \mathcal{W}) = \frac{1}{N_S} \sum_{k=1}^{N_S} \sum_{\mathbf{r}} u_{\tau}(\mathbf{r}|\mathbf{v}_k) \left(1 - \alpha_{\mathbf{r}, c_{\mathbf{v}_k}}\right) \quad (1)$$

with $\mathcal{S} = \{(\mathbf{v}, c_{\mathbf{v}})\}$ the set of all inputs \mathbf{v} and their class label $c_{\mathbf{v}}$, $N_S = \#\mathcal{S}$, $\mathcal{W} = \{\mathbf{w}_{\mathbf{r}}\}$ the set of all codebook vectors and $\mathcal{W} = \{(\mathbf{w}_{\mathbf{r}}, c_{\mathbf{r}})\}$ whereby $c_{\mathbf{r}}$ is the class label of $\mathbf{w}_{\mathbf{r}}$. The value $\alpha_{\mathbf{r}, c_{\mathbf{v}_k}}$ equals the unit if $c_{\mathbf{v}_k} = c_{\mathbf{r}}$. Otherwise it is zero. $u_{\tau}(\mathbf{r}|\mathbf{v}_k)$ is the probability that the input vector \mathbf{v}_k is assigned to the prototype \mathbf{r} . In case of a crisp *winner-takes-all* mapping one has $u_{\tau}(\mathbf{r}|\mathbf{v}_k) = 1$ iff \mathbf{r} is wiiner for \mathbf{v}_k .

In order to minimize (1) in [11] the variables $u_{\tau}(\mathbf{r}|\mathbf{v}_k)$ are taken as fuzzy assignments. This allows a gradient descent on the cost function (1). As proposed in [11], the assignment probabilities are chosen to be of normalized exponential form

ponential form

$$u_{\tau}(\mathbf{r}|\mathbf{v}_k) = \frac{\exp\left(-\frac{d(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}})}{2\tau^2}\right)}{\sum_{\mathbf{r}'} \exp\left(-\frac{d(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}'})}{2\tau^2}\right)} \quad (2)$$

whereby d is the standard Euclidean distance. The cost function (1) can be rewritten into

$$E_{soft}(\mathcal{S}, \mathcal{W}) = \frac{1}{N_S} \sum_{k=1}^{N_S} lc(\mathbf{v}_k, c_{\mathbf{v}_k}) \quad (3)$$

and local costs

$$lc(\mathbf{v}_k, c_{\mathbf{v}_k}) = \sum_{\mathbf{r}} u_{\tau}(\mathbf{r}|\mathbf{v}_k) \left(1 - \alpha_{\mathbf{r}, c_{\mathbf{v}_k}}\right) \quad (4)$$

i.e., the local error is the sum of the assignment probabilities $\alpha_{\mathbf{r}, c_{\mathbf{v}_k}}$ ² to all prototypes of an incorrect class, and, hence,

$$lc(\mathbf{v}_k, c_{\mathbf{v}_k}) \leq 1 \quad (5)$$

Because the local costs $lc(\mathbf{v}_k, c_{\mathbf{v}_k})$ (for short lc in the following) are continuous and bounded, the cost function (3) can be minimized by stochastic gradient descent using the derivative of the local costs:

$$\Delta \mathbf{w}_{\mathbf{r}} = \begin{cases} \frac{u_{\tau}(\mathbf{r}|\mathbf{v}_k) \cdot lc}{2\tau^2} \cdot \frac{\partial d_{\mathbf{r}}}{\partial \mathbf{w}_{\mathbf{r}}} & c_{\mathbf{v}_k} = c_{\mathbf{r}} \\ -\frac{u_{\tau}(\mathbf{r}|\mathbf{v}_k) \cdot (1-lc)}{2\tau^2} \cdot \frac{\partial d_{\mathbf{r}}}{\partial \mathbf{w}_{\mathbf{r}}} & c_{\mathbf{v}_k} \neq c_{\mathbf{r}} \end{cases} \quad (6)$$

using

$$\frac{\partial lc}{\partial \mathbf{w}_{\mathbf{r}}} = -\frac{u_{\tau}(\mathbf{r}|\mathbf{v}_k)}{2\tau^2} (\zeta - lc) \cdot \frac{\partial d_{\mathbf{r}}}{\partial \mathbf{w}_{\mathbf{r}}} \quad (7)$$

This leads to the learning rule

$$\mathbf{w}_{\mathbf{r}} = \mathbf{w}_{\mathbf{r}} - \epsilon(t) \cdot \Delta \mathbf{w}_{\mathbf{r}} \quad (8)$$

with learning rate $\epsilon(t)$ restricted to $\sum_{t=0}^{\infty} \epsilon(t) = \infty$ and $\sum_{t=0}^{\infty} (\epsilon(t))^2 < \infty$ as usual. Note that this adaptation is quite similar to classical LVQ2.1.

A window rule like for standard LVQ2.1 can be derived for SNPC. The window rule is necessary for numerical stabilization [8],[11]. The update is restricted to all weights for which the local value

$$\eta_{\mathbf{r}} = lc \cdot (1 - lc) \quad (9)$$

is less than a threshold value η with $0 \ll \eta < 0.25$.

3 Dynamic fuzzy labeling for soft nearest prototype classification

Up to now, the prototype labels are crisped and fixed in advance. In *Dynamic Fuzzy Labeling* for SNPC (FSNPC)

²We introduce $\zeta = (1 - \alpha_{\mathbf{r}, c_{\mathbf{v}_k}})$ as an abbreviation

we allow dynamic fuzzy labels $\alpha_{\mathbf{r},c}$ to indicate the responsibility of weight vector $\mathbf{w}_{\mathbf{r}}$ to class c such that $0 \leq \alpha_{\mathbf{r},c} \leq 1$ and $\sum_{c=1}^{N_{\mathcal{L}}} \alpha_{\mathbf{r},c} = 1$. These labels are *automatically* adapted during learning. We remark that the class information used to distinguish the adaptation rules for correct and incorrect prototypes needed in (6) is no longer available now. Hence, in addition to an update rule for the fuzzy labels, we have to introduce a new methodology for appropriate prototype adaptation.

For this purpose we can use the cost function introduced for SNPC. Obviously, the loss boundary property (5) remains valid. The stochastic derivative of the cost function (3) according to the weights yields the prototype adaptation. It is determined by the derivative of the local costs (4):

$$\frac{\partial lc}{\partial \mathbf{w}_{\mathbf{r}}} = -\frac{u_{\tau}(\mathbf{r}|\mathbf{v}_k)}{2\tau^2} \cdot (\zeta - lc) \cdot \frac{\partial d_{\mathbf{r}}}{\partial \mathbf{w}_{\mathbf{r}}} \quad (10)$$

Parallely, the fuzzy labels $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ can be optimized by gradient descent on the cost function, too. Taking the local cost one has

$$\Delta \alpha_{\mathbf{r},c_{\mathbf{v}_k}} = -\frac{\partial lc}{\partial \alpha_{\mathbf{r},c_{\mathbf{v}_k}}} \quad (11)$$

$$= -u_{\tau}(\mathbf{r}|\mathbf{v}_k) \quad (12)$$

and subsequent normalization.

We now turn to the derivation of a window rule for FSNPC in analogy to LVQ2.1 and SNPC, which is necessary for numerical stabilization of the adaptation process [8],[11]. For this purpose we consider in (7) the term

$$T = u_{\tau}(\mathbf{r}|\mathbf{v}_k) (\zeta - lc) \quad (13)$$

paying attention to the fact that now the $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ are fuzzy. Using the Gaussian form (2) for $u_{\tau}(\mathbf{r}|\mathbf{v}_k)$, the term T can be rewritten as $T = T_0 \cdot \Pi(\alpha_{\mathbf{r},c_{\mathbf{v}_k}})$ with

$$\Pi(\alpha_{\mathbf{r},c_{\mathbf{v}_k}}) = \frac{\exp\left(-\frac{d(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}})}{2\tau^2}\right)}{\sum_{\mathbf{r}'} (\zeta - \alpha_{\mathbf{r}',c_{\mathbf{v}_k}}) \exp\left(-\frac{d(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}'})}{2\tau^2}\right)} \quad (14)$$

and $T_0 = \left(lc \cdot (1 - lc) - \alpha_{\mathbf{r},c_{\mathbf{v}_k}} (1 + \alpha_{\mathbf{r},c_{\mathbf{v}_k}}) \right)$

Obviously, $0 \leq lc \cdot (1 - lc) \leq 0.25$ because lc fulfills the loss boundary property (5). Hence, we have $-2 \leq T_0 \leq 0.25$ using the fact that $\alpha_{\mathbf{r},c_{\mathbf{v}_k}} \leq 1$. Now, a similar argumentation as in [11] can be applied: The absolute value of the factor T_0 has to be significantly different from zero to have a valuable contribution in the update rule. This yields the *window condition* $0 \ll |T_0|$, which can be obtained by balancing the local loss lc and the value of the assignment variable $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$.

4 Relevance Learning in SNPC and FSNPC (FSNPC-R)

It is possible to apply the idea of relevance learning known from GRLVQ/SRNG to both, SNPC and FSNPC. The idea behind is to introduce a parametrized distance measure. Then, relevance adaptation extracts the most important parameters for the given classification task by weighting all parameters. To do so, we replace the similarity measure $d(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}})$ by a *local* and prototype dependent parametrized similarity measure $d_{\mathbf{r}}^{\lambda_{\mathbf{r}}}(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}})$ with relevance parameters $\lambda_{\mathbf{r}} = (\lambda_1(\mathbf{r}), \dots, \lambda_m(\mathbf{r}))$, $\lambda_j \geq 0, \sum_j \lambda_j = 1$. An example is the scaled Euclidean metric $\sum_j \lambda_j(\mathbf{r}) \cdot (v^j - w^j)^2$. The update of the relevance parameters $\lambda_{\mathbf{r}}$ is obtained by the derivative of the cost function, determined by the gradient $\frac{\partial lc}{\partial \lambda_j(\mathbf{r})}$ using the local cost (4) and $\gamma = (\zeta - lc)$:

$$\frac{\partial lc}{\partial \lambda_j(\mathbf{r})} = \frac{\partial}{\partial \lambda_j(\mathbf{r})} \left[\frac{\sum_{\mathbf{r}} \exp\left(-\frac{d_{\mathbf{r}}^{\lambda_{\mathbf{r}}}(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}})}{2\tau^2}\right) \cdot \zeta}{\sum_{\mathbf{r}'} \exp\left(-\frac{d_{\mathbf{r}'}^{\lambda_{\mathbf{r}'}}(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}'})}{2\tau^2}\right)} \right] \quad (15)$$

$$= -\sum_{\mathbf{r}} \frac{u_{\tau}(\mathbf{r}|\mathbf{v}_k)}{2\tau^2} \cdot \frac{\partial d_{\mathbf{r}}^{\lambda_{\mathbf{r}}}(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}})}{\partial \lambda_j(\mathbf{r})} \cdot \gamma \quad (16)$$

with subsequent normalization of the $\lambda_j(\mathbf{r})$.

5 Experiments and results

5.1 FSNPC on synthetical data

First, we apply the FSNPC to a synthetical set of two Gaussian classes, each consisting of 900 data points in two dimensions with different variances per data class and a small overlapping region, see Fig. 1. We use the FSNPC as a standalone algorithm with 50 prototypes. The initial fuzzy labeling is random at nearly around 50% for each class per prototype corresponding to an initial accuracy of around 46%. The FSNPC algorithm now optimizes in each step the codebook vector positions and label information. Because of the fuzzy property the codebook labels can change during optimization. Indeed the labeling becomes nearly perfect until the 50th complete step of FSNPC which leads to a prediction of 92%. To assess the classification rate we assign prototypes with responsibility of at least 60% to this class. By this threshold we obtain a sufficiently good labeling after 300 complete steps. Note, that codebook vectors which are clearly located within the region of a data class show very pronounced fuzzy labels of about 80% – 100% for the correct class. Only codebook vectors close to a class boundary or in the overlapping class region are still undecided with fuzzy labels of approximately 50% for each class. It can be seen during training that the codebook vectors in the overlap region switch frequently their labeling. This indicates

for the respective data points that the classification should be taken as an unknown class label. This behavior is shown in Fig. 1.

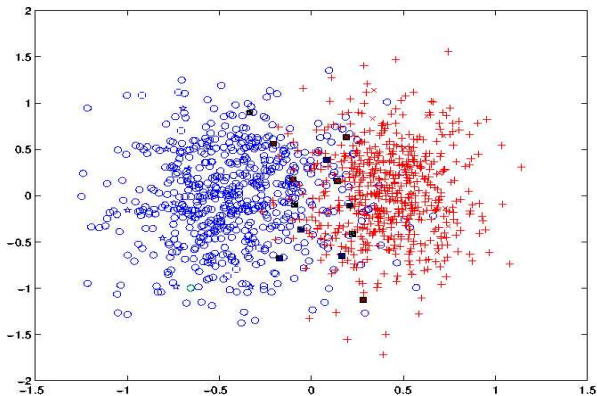


Figure 1. Plot of the overlapping synthetic data sets. Data of class 0 are given as circles together with respective codebook vectors as stars, data of class 1 are given by '+'-signs with their codebook vectors as \times . Codebook vectors without clear decision (labeling) are indicated by a surrounding black square.

5.2 Application to proteomic data

The proteomic data set used for this analysis is based on the well known prostate cancer set from the National Research Cancer Institute (NCI) [2]. This set contains mass spectra of prostate cancer specimens. One class is the control class, whereas the other are due to different cancer stages. Overall, the set consists of 222 training and 96 test data with input dimension $D_V = 130$ after preprocessing. The spectra were first processed using a standardized workflow [2]. Model building was performed by application of FSNPC-R onto the training data sets. Thereby, the number of cycles of the FSNPC-R has been set to $nc = 3000$ with $nn = 88$ as number of neurons.

The recognition rates obtained from the training data set and the prediction rates from the test data set are compared with respect to results obtained by the NCI-study [7] and Supervised Neural Gas as an alternative state of the art LVQ-approach (see above) in Tab. 1.

We see that FSNPC-R is capable to generate a suitable classification model which led to prediction rates of about 85%. The results are mostly better than those reported from NCI and SNG. Besides the good prediction rates obtained from FSNPC-R we get a fuzzy labeling and thus a judgement about the classification safety. Some of the prototypes show fuzzy labels close to 100% whereas others have values of about 65 – 70% or even lower. For a further anal-

	FSNPC-R	SNG	NCI study ([7])
Benign	86%	80.4%	78%
$PSA \leq 1$	81%	20%	<i>n.a.</i>
$PSA \in [4, 10]$	100%	29%	71%
$PSA > 10$	70%	48%	95%

Table 1. Prediction rates of correct classification for the FSNPC-R and SNG compared with NCI results on the NCI prostate cancer data set. The different PSA levels refer to different cancer states.

ysis of the prototypes with unclear prediction and the data points which were wrong classified one has to consider the receptive fields. In particular, the receptive fields of prototypes with unclear prediction may indicate regions which are overlapping in the data space. For the special kind of the NCI data, this effect reflects partially the different, overlapping cancer stages.

6 Conclusion

In this paper, we introduced the fuzzy labeling for SNPC (FSNPC) approach and an accompanied relevance learning method. For this purpose we developed a new methodology for prototype update during learning because the unique class information of prototypes necessary in standard LVQ-methods is not longer valid and, therefore, cannot be used. We derived an adaptation dynamic for fuzzy labeling and prototype adjustment according to a gradient descent on a cost function. This cost function is obtained by appropriate modification of the SNPC costs.

In general the FSNPC-R algorithm will be used as a standalone algorithm or in combination with alternative LVQ-schemes. The new fuzzy labeling learning vector quantization can be efficiently applied to the classification of proteomics data and leads to results which are competitive and mostly better than results as reported by NCI. The extraction of fuzzy assignments through the algorithm allows the introduction of an additional class label - *unclassified* if the fuzzy assignment for the predicted class is under a predetermined threshold (e.g. 60%). Therewith, it is possible to get a deeper understanding of the underlying data space, since, by use of a sufficiently large number of prototypes, we get a model which is capable to reflect the class boundaries for overlapping classes and indicates this by unsafe fuzzy assignment values near to those boundaries.

ACKNOWLEDGEMENT: The authors are grateful to U. Clauss and J. Decker both Bruker Daltonik GmbH Leipzig for support by preprocessing of the NCI prostate cancer data set.

References

- [1] P. A. et al. Binz. Mass spectrometry-based proteomics: current status and potential use in clinical chemistry. *Clin. Chem. Lab. Med.*, 41:1540–1551, 2003.
- [2] Adam BL et al. Serum protein finger printing coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. *Cancer Research*, 62(13):3609–3614, July 2002.
- [3] B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of GRLVQ networks. *Neural Proc. Letters*, 21(2):109–120, 2005.
- [4] B. Hammer and Th. Villmann. Generalized relevance learning vector quantization. *Neural Netw.*, 15(8-9):1059–1068, 2002.
- [5] B. Hammer and Th. Villmann. Mathematical aspects of neural networks. In M. Verleysen, editor, *Proc. Of Europ. Symp. on Art. Neural Netw. (ESANN'2003)*, pages 59–72, Brussels, Belgium, 2003. d-side.
- [6] Barbara Hammer, Marc Strickert, and Thomas Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21(1):21–44, February 2005.
- [7] E.F. Petricoin III, D.K. Ornstein, C.P. Haweletz, A. Ardekani, P.S. Hackett, B.A. Hitt, A. Velasco, C. Trucco, L. Wiegand, K. Wood, C.B. Simone, W.M. Linehan P.J. Levine, M.R. Emmert-Buck, S.M. Steinberg, E.C. Kohn, and L.A. Liotta. Serum proteomic patterns for detection of prostate cancer. *Journal of the National Cancer Institute*, 94(20):1576–1578, 2002.
- [8] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (2nd Ext. Ed. 1997).
- [9] Teuvo Kohonen, Samuel Kaski, and Harri Lapalainen. Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. *Neural Computation*, 9:1321–1344, 1997.
- [10] A. S. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 423–429. MIT Press, 1995.
- [11] S. Seo, M. Bode, and K. Obermayer. Soft nearest prototype classification. *IEEE Transaction on Neural Networks*, 14:390–398, 2003.
- [12] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15:1589–1604, 2003.
- [13] G. Van de Wouwer, P. Scheunders, D. Van Dyck, M. De Bodt, F. Wuyts, and P.H. Van de Heyning. Wavelet-FILVQ classifier for speech analysis. pages 214–218. IEEE Press, 1996. In Proc. of the Int. Conf. Pattern Recognition.