Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 69 (2006) 2425–2428
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Letters

Prototype-based fuzzy classification with local relevance for proteomics

T. Villmann[a,*], F.-M. Schleif[b], B. Hammer[c]

[a]Clinic for Psychotherapy, University Leipzig, K.-Tauchnitz-Str. 25, 04107 Leipzig, Germany
[b]Bruker Daltonik, Leipzig, Deutscher Platz 5d, 04109 Leipzig, Germany
[c]Institute for Computer Science, Clausthal University of Technology, Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany

Received 19 January 2005; received in revised form 6 February 2006; accepted 6 February 2006
Communicated by R.W. Newcomb
Available online 3 July 2006

Abstract

In this paper, we extend soft nearest prototype classification by local metric learning and fuzzy classification. Thereby, the metric is determined according to the given classification task. This may be done separately for each prototype or class specific. We apply the method to cancer detection based on proteomic data.

Keywords: Fuzzy classification; Learning vector quantization; Metric adaptation; Relevance learning

1. Introduction

Kohonen's learning vector quantization (LVQ) belongs to the class of *supervised* learning algorithms for crisp nearest prototype classification [3]. Originally LVQ is based on heuristics. Yet, several extensions of the basic scheme exist to obtain a gradient descent approach [2]. Recently a new method, soft nearest prototype classification (SNPC), has been proposed by Seo et al. [6] in which soft assignments of the data vectors to the prototypes, based on a Gaussian mixture approach, are introduced.

However, crisp classification by LVQ-methods has the disadvantage that overlapping classes cannot be represented adequately. A solution could be a post-labeling of the prototypes according to the data statistics leading to a fuzzy labeling. But in this case the prototype distribution would not be optimized, because fuzzy labels cannot be handled in LVQ methods during learning. Hence, a new learning scheme has to be established. Based on SNPC we derive such an adaptation scheme. Further, we improve the model by incorporation of metric adaptation into the learning dynamic.

We apply the new algorithm to profiling of mass spectrometic data in cancer research. The underlying algorithms for classification of the mass spectrometric data are one crucial point to obtain valid and competitive results. 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.

2. Crisp learning vector quantization

The standard algorithms for LVQ are LVQ1...LVQ3 introduced by Kohonen [3]. Several extensions have been proposed, one of them is SNPC introducing a gradient descent approach based on the expectation maximization of a cost function given as

$$E(\mathcal{S},\mathcal{W}) = \frac{1}{N_{\mathcal{S}}}\sum_{k=1}^{N_{\mathcal{S}}} lc(\mathbf{v}_k, c_{\mathbf{v}_k}), \tag{1}$$

$\mathcal{S} = \{(\mathbf{v}, c_{\mathbf{v}})\}$ the set of all inputs $\mathbf{v}$ and their class label $c_{\mathbf{v}}$, $N_{\mathcal{S}} = \#\mathcal{S}$, $\mathbf{W} = \{\mathbf{w}_{\mathbf{r}}\}$ 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 local costs $lc(\mathbf{v}_k, c_{\mathbf{v}_k})$ are defined as

$$lc(\mathbf{v}_k, c_{\mathbf{v}_k}) = \sum_{\mathbf{r}} u_\tau(\mathbf{r}|\mathbf{v}_k)(1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}}) \tag{2}$$

*Corresponding author.
E-mail addresses: thomas.villmann@medizin.uni-leipzig.de, villmann@informatik.uni-leipzig.de (T. Villmann).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.02.003

with value $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ equal the unity if $c_{\mathbf{v}_k} = c_{\mathbf{r}}$. Otherwise it is zero. $u_\tau(\mathbf{r}|\mathbf{v}_k)$ are the soft-assignments representing 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 winner for $\mathbf{v}_k$. The local error is the sum of the assignment probabilities $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ to all prototypes of an incorrect class, and, hence, $lc(\mathbf{v}_k, c_{\mathbf{v}_k}) \leqslant 1$.

In order to minimize (1) the variables $u_\tau(\mathbf{r}|\mathbf{v}_k)$ are taken as fuzzy assignments in SNPC. This allows a gradient descent on the cost function (1). As proposed in [6], the assignment probabilities are chosen to be of normalized exponential form

$$u_\tau(\mathbf{r}|\mathbf{v}_k) = \frac{\exp(-d(\mathbf{v}_k, \mathbf{w}_\mathbf{r})/2\tau^2)}{\sum_{\mathbf{r}'} \exp(-d(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}'})/2\tau^2)}, \tag{3}$$

whereby d is the standard Euclidean distance. As for standard LVQ2.1, SNPC requires a window rule for numerical stabilization [3,6]: the weight update is restricted to all weights for which the local value

$$\eta_\mathbf{r} = lc(\mathbf{v}_k, c_{\mathbf{v}_k}) \cdot (1 - lc(\mathbf{v}_k, c_{\mathbf{v}_k})) \tag{4}$$

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

3. Dynamic fuzzy labeling and metric adaptation

In *Fuzzy Labeled* SNPC (FSNPC) we allow fuzzy values for $\alpha_{\mathbf{r},c}$ to indicate the responsibility of weight vector $\mathbf{w}_\mathbf{r}$ to class c such that $0 \leqslant \alpha_{\mathbf{r},c} \leqslant 1$ and $\sum_{c=1}^{N_\mathscr{Y}} \alpha_{\mathbf{r},c} = 1$. These labels should be adjusted *automatically* during training. However, doing so, the crisp class information for prototypes, assumed in the learning dynamic of SNPC (or generally required in LVQ) [6], is no longer available. Yet, a corresponding learning dynamic can be derived: in complete analogy to the original SNPC one has

$$\frac{\partial lc(\mathbf{v}_k, c_{\mathbf{v}_k})}{\partial \mathbf{w}_\mathbf{r}} = -\frac{T}{2\tau^2} \cdot \frac{\partial d_\mathbf{r}}{\partial \mathbf{w}_\mathbf{r}} \tag{5}$$

with $T = u_\tau(\mathbf{r}|\mathbf{v}_k) \cdot (1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}} - lc(\mathbf{v}_k, c_{\mathbf{v}_k}))$. Thereby, the loss boundary property (2) remains valid. Parallely, the fuzzy labels $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ can be optimized using $\partial lc(\mathbf{v}_k, c_{\mathbf{v}_k})/\partial \alpha_{\mathbf{r},c_{\mathbf{v}_k}}$:

$$\Delta \alpha_{\mathbf{r},c_{\mathbf{v}_k}} = -u_\tau(\mathbf{r}|\mathbf{v}_k) \tag{6}$$

followed by subsequent normalization.

To adjust the window rule to now fuzzified values $\alpha_{\mathbf{r},c_{\mathbf{v}_k}}$ we consider T. Using the Gaussian form (3) for $u_\tau(\mathbf{r}|\mathbf{v}_k)$, the term T can be rewritten as $T = (T_{lc} - T_\alpha) \cdot \Pi(\alpha_{\mathbf{r},c_{\mathbf{v}_k}})$ with

$$\Pi(\alpha_{\mathbf{r},c_{\mathbf{v}_k}}) = \frac{\exp(-d(\mathbf{v}_k, \mathbf{w}_\mathbf{r})/2\tau^2)}{\sum_{\mathbf{r}'} (1 - \alpha_{\mathbf{r},c_{\mathbf{v}_k}} - \alpha_{\mathbf{r}',c_{\mathbf{v}_k}})/\exp(d(\mathbf{v}_k, \mathbf{w}_{\mathbf{r}'})/2\tau^2)} \tag{7}$$

and $T_{lc} = lc(\mathbf{v}_k, c_{\mathbf{v}_k})(1 - lc(\mathbf{v}_k, c_{\mathbf{v}_k}))$ and $T_\alpha = \alpha_{r,c_{\mathbf{v}_k}}(1 + \alpha_{r,c_{\mathbf{v}_k}})$.

As in the original SNPC, $0 \leqslant lc(\mathbf{v}_k, c_{\mathbf{v}_k})(1 - lc(\mathbf{v}_k, c_{\mathbf{v}_k})) \leqslant 0.25$ because $lc(\mathbf{v}_k, c_{\mathbf{v}_k})$ fulfills the loss boundary property (2) [6]. Hence, we have $-2 \leqslant T_0 \leqslant 0.25$ using the fact that $\alpha_{r,c_{\mathbf{v}_k}} \leqslant 1$. Further, the absolute value of the factor $T_0 = T_{lc} - T_\alpha$ has to be significantly different from zero to have a valuable contribution in the update rule [6]. This yields the *window condition* $0 \ll |T_0|$, which can be obtained by balancing the local loss $lc(\mathbf{v}_k, c_{\mathbf{v}_k})$ and the value of the assignment variable $\alpha_{r,c_{\mathbf{v}_k}}$.

Now we apply the idea of metric adaptation to FSNPC [2,7]: we replace the similarity measure $d(\mathbf{v}_k, \mathbf{w}_\mathbf{r})$ by a *local* and prototype dependent parametrized general similarity measure $d_\mathbf{r}^{\boldsymbol{\lambda}_\mathbf{r}}(\mathbf{v}_k, \mathbf{w}_\mathbf{r})$ with so-called relevance parameters $\boldsymbol{\lambda}_\mathbf{r} = (\lambda_1(\mathbf{r}), \ldots, \lambda_m(\mathbf{r}))$, $\lambda_j \geqslant 0$, $\sum \lambda_j = 1$. An example is the scaled Euclidean metric $\sum_j \lambda_j(\mathbf{r}) \cdot (v^j - w^j)^2$. Metric adaptation takes place as gradient descent on the cost function with respect to the relevance paremeters $\boldsymbol{\lambda}_\mathbf{r}$ (relevance learning): $\Delta \boldsymbol{\lambda}_\mathbf{r} = -\partial lc(\mathbf{v}_k, c_{\mathbf{v}_k})/\partial \boldsymbol{\lambda}_\mathbf{r}$ with

$$\frac{\partial lc(\mathbf{v}_k, c_{\mathbf{v}_k})}{\partial \lambda_j(\mathbf{r})} = -\frac{T}{2\tau^2} \cdot \frac{\partial d_\mathbf{r}^{\boldsymbol{\lambda}_\mathbf{r}}(\mathbf{v}_k, \mathbf{w}_\mathbf{r})}{\partial \lambda_j(\mathbf{r})} \tag{8}$$

using the local cost (2) and subsequent normalization of the $\lambda_j(\mathbf{r})$. In case of $\boldsymbol{\lambda} = \boldsymbol{\lambda}_\mathbf{r}$ for all $\mathbf{r}$ (global parametrized metric) one gets

$$\frac{\partial lc(\mathbf{v}_k, c_{\mathbf{v}_k})}{\partial \lambda_j} = -\sum_\mathbf{r} \frac{T}{2\tau^2} \cdot \frac{\partial d^{\boldsymbol{\lambda}}(\mathbf{v}_k, \mathbf{w}_\mathbf{r})}{\partial \lambda_j}. \tag{9}$$

We refer to this variant as FSNPC-R.

4. Experiments and results

We successfully applied the FSNPC to a synthetical set of two overlapping two-dimensional Gaussian classes results and in a more challenging application to the well-known prostate cancer set from the National Research Cancer Institute (NCI) [4].

4.1. 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 an overlapping region, 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 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