Estimating Relevant Input Dimensions for Self-organizing Algorithms

Barbara Hammer[1] and Thomas Villmann[2]

[1] University of Osnabrück, Department of Mathematics/Computer Science, Albrechtstraße 28, 49069 Osnabrück, Germany
[2] University of Leipzig, Clinic for Psychotherapy and Psychosomatic Medicine, Karl-Tauchnitz-Straße 25, 04107 Leipzig, Germany

Summary. We propose a new scheme for enlarging generalized learning vector quantization with weighting factors for the several input dimensions which are adapted according to the specific task. This leads to a more powerful classifier with little extra cost as well as the possibility of automatically pruning irrelevant input dimensions. The method is tested on real world satellite image data and compared to several well known algorithms which determine the intrinsic data dimension.

1 Introduction

Self-organizing methods such as the self-organizing map (SOM) or vector quantization (VQ) as introduced by Kohonen provide a successful and intuitive method of processing data for easy access [12]. Assumed data are labeled, an automatic clustering can be learned via attaching maps to the SOM or enlarging VQ with a supervised component to so-called learning vector quantization (LVQ) [13,17]. Various modifications of LVQ exist which ensure 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 [13,20,21]. Concerning SOM, one major problem consists in finding an appropriate topology of the initial lattice of codebooks such that the prior topology of the neural architecture mirrors the intrinsic topology of the data. Hence various approaches exist in order to measure the degree of topology preservation, to adapt the topology to the data, to define the lattice a posteriori, or to evolve structures which are appropriate for real world data [2,6,14,19,25]. In all tasks the intrinsic dimensionality of the data plays a crucial role since it determines large parts of the optimum neural network, i.e., the lattice for SOM. Moreover, superfluous data dimensions slow down the training for LVQ as well. They may even cause a decrease in accuracy since they add possibly noisy or misleading terms to the Euclidian metric where LVQ is based on. Hence a data dimension as small as possible is desirable for the above mentioned methods in general, for the sake of efficiency, accuracy, and simplicity of neural network processing. We will consider possibilities of pruning irrelevant data dimensions and computing the intrinsic data dimension in this paper. Approaches like [11] clearly indicate that often a considerable reduction of the data dimension is possible without loss of information.

We will focus on LVQ since it combines the elegancy of simple and intuitive updates in unsupervised algorithms with the accuracy of supervised methods. The

N. Allinson et al., *Advances in Self-Organising Maps*

main idea of our approach is to introduce weighting factors to the data dimensions which are adapted automatically such that the classification error becomes minimal. Small factors indicate that the respective data dimension is irrelevant. This idea can be applied to any generalized LVQ (GLVQ) scheme as introduced in [20] such that a robust and efficient method results which can push the classification borders near to the optimum Bayesian decision. This method, generalized relevance LVQ (GR-LVQ), generalizes relevance LVQ (RLVQ) [3] which is based on simple Hebbian learning and leads to worse and instable results in case of noisy real life data. However, like RLVQ, GRLVQ has the advantage of an intuitive update rule and allows efficient input pruning compared to other approaches which adapt the metric to the data involving additional transformations as proposed in [7,9,22] or depend on less intuitive differentiable approximations of the original dynamics [15]. We will apply GRLVQ to classify a real life satellite image with approx. 3 mio. data points. Apart from the above mentioned methods, dimensionality reduction is possible via standard methods like principal component analysis or independent component analysis [10,18]. Furthermore, a growing SOM (GSOM) automatically adapts the lattice of neurons to the data and hence gives hints about the intrinsic dimensionality as well. We compare our GRLVQ experiments to the results provided by GSOM and a Grassberger-Procaccia analysis respectively, and obtain comparable results concerning the intrinsic dimensionality of our data.

2 The GRLVQ Algorithm

Assume a finite training set $X = \{(x^i, y^i) \subset \mathbb{R}^n \times \{1, \ldots, C\} \mid i = 1, \ldots, m\}$ of training data is given and the clustering of the data into C classes is to be learned. We denote the components of a vector $x \in \mathbb{R}^n$ by $(x_1, \ldots, x_n)$ in the following. GLVQ chooses a fixed number of vectors in $\mathbb{R}^n$ for each class, so called codebooks. Denote the set of codebooks by $\{w^1, \ldots, w^K\}$ and assign the label $c^i = c$ to w^i iff w^i belongs to the cth class, $c \in \{1, \ldots, C\}$. The receptive field of w^i is defined by $R^i = \{x \in X \mid \forall w^j \, (j \neq i \rightarrow |x - w^i| < |x - w^j|)\}$. The training algorithm adapts the codebooks w^i such that for each class $c \in \{1, \ldots, C\}$, the corresponding codebooks represent the class as accurately as possible. That means, the difference of the points belonging to the cth class, $\{x^i \in X \mid y^i = c\}$, and the receptive fields of the corresponding codebooks, $\bigcup_{c^i=c} R^i$, should be as small as possible for each class. For a given data point $(x, y) \in X$ denote by $\mu(x)$ some function which is negative if x is classified correct, i.e., it belongs to a receptive field R^i with $c^i = y$, and which is positive if x is classified wrong, i.e., it belongs to a receptive field R^i with $c^i \neq y$. Denote by $f : \mathbb{R} \rightarrow \mathbb{R}$ some monotonically increasing function. The general scheme of GLVQ consists in minimizing the error term

$$S = \sum_{i=1}^{m} f(\mu(x_i)) \tag{1}$$

via a stochastic gradient descent. The concrete choice of f as the identity and $\mu(x) = \eta \cdot d$, d being the squared Euclidian distance of x to the nearest code-