Available online at www.sciencedirect.com

ScienceDirect

Neurocomputing 71 (2008) 1210–1219

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Learning dynamics and robustness of vector quantization and neural gas

Aree Witoelar[a,*], Michael Biehl[a], Anarta Ghosh[b,1], Barbara Hammer[c]

[a]*University of Groningen, Mathematics and Computing Science, P.O. Box 800, NL-9700 AV Groningen, The Netherlands*
[b]*WaNPRC, University of Washington, Seattle, WA 98195, USA*
[c]*Clausthal University of Technology, Institute of Computer Science, D-98678 Clausthal-Zellerfeld, Germany*

Available online 25 January 2008

Abstract

Various alternatives have been developed to improve the winner-takes-all (WTA) mechanism in vector quantization, including the neural gas (NG). However, the behavior of these algorithms including their learning dynamics, robustness with respect to initialization, asymptotic results, etc. has only partially been studied in a rigorous mathematical analysis. The theory of on-line learning allows for an exact mathematical description of the training dynamics in model situations. We demonstrate using a system of three competing prototypes trained from a mixture of Gaussian clusters that the NG can improve convergence speed and achieves robustness to initial conditions. However, depending on the structure of the data, the NG does not always obtain the best asymptotic quantization error.
© 2008 Elsevier B.V. All rights reserved.

Keywords: Vector quantization; Clustering; Online learning; Winner-takes-all algorithms; Neural gas

1. Introduction

Vector quantization (VQ) is an important unsupervised learning algorithm, widely used in different areas such as data mining, medical analysis, image compression, and speech or handwriting recognition [2]. The main objective of VQ is to represent the data points by a small number of prototypes or codebook vectors. This can directly be used for compression, clustering, data mining, or (with post-labeling of the prototypes) classification [9,14].

The basic "winner-takes-all" (WTA) or batch algorithms like the popular k-means clustering directly optimize the quantization error underlying VQ. However, these methods can be subject to confinement in local minima of the quantization error and can produce suboptimal results. A variety of alternatives to overcome this problem have been proposed, some of which are heuristically motivated while others are based on the minimization of a cost function related to the quantization error: the self-organizing map (SOM) [12], fuzzy-k-means [1], stochastic optimization [7], to name just a few. These algorithms have in common that each pattern influences more than one prototype at a time through a "winner-takes-most" paradigm. Neural gas (NG) as proposed in [13] is a particularly robust variation of VQ with the introduction of neighborhood relations. Unlike the SOM [12], the NG system takes into account the relative distances between prototypes in the input space and not on a predefined lattice.

In practice, NG algorithms yield better solutions than WTA; however, the effect of this strategy on convergence speed or asymptotic behavior has hardly been rigorously investigated so far.

Methods from statistical physics and the theory of on-line learning [8] allow for an exact mathematical description of learning systems for high dimensional data. In the limit of infinite dimensionality, such systems can be fully described in terms of a few characteristic quantities, the so-called *order parameters*. The evolution of these order parameters along the training procedure is characterized by a set of coupled ordinary differential equations (ODE).

[*]Corresponding author. Tel.: +31 50 3637049; fax: +31 50 3633800.
E-mail address: a.w.witoelar@rug.nl (A. Witoelar).
URL: http://www.cs.rug.nl/~aree (A. Witoelar).
[1]Present address: Nordic Bioscience, Imaging Division, Herlev Hovedgade 207, 2730 Herlev, Denmark.

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.11.022

By integrating these ODEs, it is possible to analyze the performance of VQ algorithms in terms of stability, sensitivity to initial conditions, and achievable quantization error. This successful approach has also been reviewed in [8,16], among others.

The extension of the theoretical analysis of simple (WTA-based) VQ with two prototypes and two clusters introduced in an earlier works [4,5] is not straightforward. Additional prototypes and clusters introduce more complex interactions in the system that can result in radically different behaviors, see [17] for an example. Also, the mathematical treatment becomes more involved and requires, for instance, several numerical integrations. Here we introduce an additional prototype and a mixture of clusters. We investigate not only WTA but also the popular NG approach [13] for VQ. This is an important step towards the investigation of general VQ approaches based on neighborhood interaction such as self-organizing maps.

2. WTA and NG

Assume input data $\boldsymbol{\xi} \in \mathbb{R}^N$, generated according to a given probability density function $P(\boldsymbol{\xi})$. VQ represents the input data in the same N-dimensional space by a set of prototypes $W = \{\mathbf{w}_i \in \mathbb{R}^N\}_{i=1}^S$. The primary goal of VQ is to find a faithful representation by minimizing the so-called quantization or distortion error

$$E(W) = \frac{1}{2}\int \mathrm{d}\boldsymbol{\xi} P(\boldsymbol{\xi}) \sum_{i=1}^{S} d(\boldsymbol{\xi}, \mathbf{w}_i) \prod_{j\neq i} \Theta_{ij} - \frac{1}{2}\int \mathrm{d}\boldsymbol{\xi} P(\boldsymbol{\xi}) \boldsymbol{\xi}^2, \qquad (1)$$

where $\Theta_{ij} \equiv \Theta(d(\boldsymbol{\xi}, \mathbf{w}_j) - d(\boldsymbol{\xi}, \mathbf{w}_i))$. For each input vector $\boldsymbol{\xi}$ the closest prototype $\mathbf{w}_i$ is singled out by the product of Heaviside functions, $\Theta(x) = 0$ if $x<0$; 1 else. Here we restrict ourselves to the quadratic Euclidean distance measure $d(\boldsymbol{\xi}, \mathbf{w}_i) = (\boldsymbol{\xi} - \mathbf{w}_i)^2$. The constant $\frac{1}{2}\int \mathrm{d}\boldsymbol{\xi} P(\boldsymbol{\xi})\boldsymbol{\xi}^2$ term is independent of prototype positions and is subtracted for convenience.

The input data are presented sequentially during training and one or more prototypes are updated on-line. Algorithms studied here can be interpreted as stochastic gradient descent procedures with respect to a cost function $H(W)$ related to $E(W)$. The generalized form reads

$$H(W) = \frac{1}{2}\int \mathrm{d}\xi P(\boldsymbol{\xi}) \sum_{i=1}^{S} d(\boldsymbol{\xi}, \mathbf{w}_i) f(r_i) - \frac{1}{2}\int \mathrm{d}\boldsymbol{\xi} P(\boldsymbol{\xi}) \boldsymbol{\xi}^2, \qquad (2)$$

where r_i is the rank of prototype $\mathbf{w}_i$ with respect to the distance $d(\boldsymbol{\xi}, \mathbf{w}_i)$, i.e. $r_i = S - \sum_{j\neq i} \Theta_{ij}$. Rank $r_J = 1$ corresponds to the so-called *winner*, i.e. the prototype $\mathbf{w}_J$ closest to the example $\boldsymbol{\xi}$. The rank function $f(r_i)$ determines the update strength for the set of prototypes and satisfies the normalization $\sum_{i=1}^{S} f(r_i) = 1$; note that it does not depend explicitly on distances but only on the ordering of the prototypes with respect to the current example.

The corresponding stochastic gradient descent in $H(W)$ is of the form

$$\mathbf{w}_i^\mu = \mathbf{w}_i^{\mu-1} + \Delta\mathbf{w}_i^{\mu-1} \text{ with}$$
$$\Delta\mathbf{w}_i^{\mu-1} = \frac{\eta}{N} f(r_i)(\boldsymbol{\xi}^\mu - \mathbf{w}_i^{\mu-1}), \qquad (3)$$

where η is the learning rate and $\boldsymbol{\xi}^\mu$ is a single example drawn independently at time step μ of the sequential training process. We compare two different algorithms:

(i) *WTA*: Only one prototype, the winner, is updated for each input. The cost function directly minimizes the quantization error with $H(W) = E(W)$. The corresponding rank function is

$$f_{\mathrm{WTA}}(r_i) = \prod_{j\neq i} \Theta_{ij}. \qquad (4)$$

(ii) *NG*: The update strength decays exponentially with the rank controlled by a parameter λ. The rank function is $f(r_i) = (1/C(\lambda))h_\lambda(r_i)$ where $h_\lambda(r_i) = \exp(-r_i/\lambda)$ and

$$C(\lambda) = \sum_{r_i=1}^{S} \exp(-r_i/\lambda)$$

is a normalization constant. The parameter λ is adjusted during training; it is frequently set large initially and decreased in the course of training. Note that for $\lambda \to 0$ the NG algorithm becomes identical with WTA. We divide $f(r_i)$ according to its ranks as

$$f_{\mathrm{NG}}(r_i) = \frac{1}{C(\lambda)} \sum_{k=1}^{S} h_\lambda(k) g_i(k), \qquad (5)$$

where $g_i(k) = 1$ if $r_i = k$; 0 else and $\sum_k g_i(k) = 1$. In a model with three prototypes, this can be written in terms of Heaviside functions

$$g_i(1) = \prod_{j\neq i} \Theta_{ij},$$
$$g_i(2) = \sum_{k\neq i} \prod_{j\neq k,i} \Theta_{ij}(1 - \Theta_{ik}),$$
$$g_i(3) = \prod_{j\neq i}(1 - \Theta_{ij}). \qquad (6)$$

3. Model

We choose the model data as a mixture of M spherical Gaussian clusters:

$$P(\boldsymbol{\xi}) = \sum_{\sigma=1}^{M} p_\sigma P(\boldsymbol{\xi}|\sigma) \text{ with}$$
$$P(\boldsymbol{\xi}|\sigma) = \frac{1}{(2\pi\upsilon_\sigma)^{N/2}} \exp\left(-\frac{1}{2\upsilon_\sigma}(\boldsymbol{\xi} - \ell_\sigma \mathbf{B}_\sigma)^2\right), \qquad (7)$$