Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 71 (2008) 1191–1202
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Following non-stationary distributions by controlling the vector quantization accuracy of a growing neural gas network

Hervé Frezza-Buet

Supélec, 2 rue Édouard Belin, F-57070 Metz, France

Available online 1 February 2008

Abstract

In this paper, an original method extended from growing neural gas (GNG-T) [B. Fritzke, A growing neural gas network learns topologies, in: G. Tesauro, D.S. Touretzky, T.K. Leen (Eds.), Advances in Neural Information Processing Systems 7, MIT Press, Cambridge, MA, 1995, pp. 625–632] is presented. The method performs continuously vector quantization over a distribution that changes over time. It deals with both sudden changes and continuous ones, and is thus suited for the video tracking framework, where continuous tracking is required as well as fast adaptation to incoming and outgoing people. The central mechanism relies on the management of the quantization resolution, that copes with stopping condition problems of usual GNG inspired methods. Application to video tracking is presented.
© 2008 Elsevier B.V. All rights reserved.

Keywords: Vector quantization; Non-stationary distribution; Video tracking; Growing neural gas

1. Introduction

Unsupervised learning consists in mapping the distribution of some phenomenon with some reduced information, like principal components in principal components analysis (PCA) for example. In this paper, analysis by vector quantization (VQ) is rather addressed. This consists in finding a finite set of vectors that are located in order to cover places where the phenomenon happens. Representing an unknown continuous density probability function by few vectors reduces the information and allows us to analyze, compress or represent the complexity of the problem.

In this paper, that is an extended version of [4], unsupervised learning by VQ is applied to non-stationary distributions, which requires to adapt the usual techniques to this issue. The problem with non-stationary distributions is that the process should keep a high level of plasticity, in order to adapt to sudden changes, as well as being able to stabilize on constant or smoothly changing parts, which rather requires stability. The algorithm proposed here deals with both these features, and can thus be applied to video scene analysis, since objects in a scene can appear, disappear, move continuously or just be motionless.

2. VQ and non-stationary distributions

Since a decade, many methods have been proposed for VQ. Some of them are related to information theory and signal processing, as reviewed in [27], but more recent ones deal with the mapping of a distribution by a finite set of vectors that try to cover at best a continuous probability density function. These latter are reported in [3,16]. Let us first sketch the basic elements of any VQ procedure, and then show how it has been extended to deal with non-stationary distributions.

2.1. VQ basics and notation

In order to introduce VQ, let us recall some of its features and define some notations as in [8]. Let $\xi \in X$ be a sample of an unknown distribution $\mathscr{P}_X$ over space X, according to a stationary density of probability $p(\xi)$. VQ consists classically in finding a discrete set $\{w_i\}_{1 \leqslant i \leqslant n} \subset X$ of *prototypes* such that this set "matches" $\mathscr{P}_X$. The so-called

E-mail address: Herve.Frezza-Buet@supelec.fr

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.12.024

neuron i is the computational element that gathers information related to the w_i prototype. Let $w(\xi)$ be $\text{argmin}_{w_i} d(\xi, w_i)$, where d is some proximity function (usually $d(\xi, w_i) = \|\xi - w_i\|^2$), less restrictive than pure distances, returning low values for similar arguments and higher values for less similar arguments. The quality of the fitting depends on how well the prototypes $\{w_i\}$ are scattered over the distribution. More formally, this scattering has to minimize distortion defined in Eq. (1), where V_i is the Voronoï cell around w_i (i.e. the region in space where points are closer to w_i than to any other w_j). The V_is form a partition of X.

$$E = \int_X d(w(\xi), \xi) p(\xi)\, \mathrm{d}\xi = \sum_{i=1}^{n} E_i, \tag{1}$$

where $E_i = \int_{V_i} d(w_i, \xi) p(\xi)\, \mathrm{d}\xi$ and $V_i = \{\xi \in X : w(\xi) = w_i\}$.

The minimization of E is performed by successive stages, until some stopping condition is met. Some methods proceed minimization from a given finite set of examples, chosen from $\mathscr{P}_X$, and minimize the distortion measured on this set [15,11]. Some other methods work on-line, since examples are provided continuously. In this latter case, at each stage, an input ξ is first chosen, according to $\mathscr{P}_X$. Second, the so-called winner-take-all procedure (WTA) allows to determine the winning prototype $w_{i_1} = w(\xi)$. Third, w_{i_1} is modified so as to be closer to ξ.

The previous procedure describes the reduction of distortion by placing vectors in the input space. For many algorithms in the literature, the feature of topology preservation is also addressed. This consists in endowing the prototypes with a graph structure, that defines a neighborhood relation between them. Preserving topology, roughly speaking, means ensuring that two prototypes that are close in the graph are also close according to the metrics d in the input space. Early self-organizing maps by Kohonen [14] have this feature since they try to map an a priori graph (usually a grid) to the distribution in the input space. This distribution can have a dimension higher than two. In this case, the resulting grid of prototypes is distorted to fit the distribution. The work presented here is rather inspired by the growing neural gas (GNG) algorithm by Fritzke [6] since it adapts to the actual topology of the distribution by building a suited graph with the prototypes. This prevents from distorting an arbitrary low dimension graph (as a grid) if some higher dimension connectivity is needed. The idea is, each time an example is given, to link the two best matching prototypes. This method, called competitive Hebbian learning (CHL), has been introduced by Martinez and Schulten. Authors have demonstrated that it builds a graph that approximates the Delaunay triangulation of the prototypes [17]. One other interesting feature of GNG is its incremental nature. Each prototype w_i has an accumulation variable with it that measures the sum of the errors w_i makes when it actually wins. This accumulated value is used to add neurons where quantization is not accurate enough (i.e. local distortion is too high).

Accumulation of local error, an incremental algorithm, and CHL are of primary relevance in our approach, that directly inherits these features from the GNG algorithm. Some refinements have been proposed in the literature, in order to accelerate the growing of the network [24] or manage the unstability induced by outliers in the distribution [23]. Some other approaches modulate the spatial sensitivity of a prototype, using some variable radius receptive field, to allow different accuracies in the network [1]. In this latter case, adaptation and growth is driven by a value that is a goal to be reached by local errors. Such a criterion is very close to the *target* that is used to drive the growth of our algorithm.

Last, in [24], the preservation of topology is analyzed numerically. The authors stress that, for a given distribution, the quality of the topology preservation increases with the number of prototypes until some bound is reached. More prototypes, then, will not improve the preservation. This bound depends on the probability density function p itself, and it may differ drastically from one distribution to another. This is why setting in advance the number of prototypes is difficult, and difficulty increases if the distribution is subject to change with time. This is why, as explain in next section, controlling continuously the number of prototypes is the main difficulty when using VQ for non-stationary distributions.

2.2. Endowing VQ with the ability to track non-stationary distributions

As explained in [8], algorithms whose plasticity decreases with time, as for example the self-organizing maps, are not suitable for non-stationary distributions. What is less intuitive is that even incremental neural networks, where all parameters are constant, encounter difficulties when fed with changing distributions. They are actually able to create new prototypes if needed, but the problem is that some neurons may become useless (dead units) in case of a sudden change. So the challenge is to exploit the properties of algorithms like GNG, while controlling the additions and removals of neurons, in order to stick to the changes in the distribution. This requires computing some kind of statistics for each prototype, in order to detect when it needs to have one more neighbor or, rather, that it needs to be removed.

Let us first consider the problem of removing useless prototypes. Some utility measure has proposed by Fritzke [7], and adapted to GNG [8]. The resulting GNG-U algorithm uses two accumulators for each prototype. One accumulator stores the sum of errors made when the prototype actually wins, as in GNG. The second stores the ''utilities'' of a prototype when it wins as well. This utility is the difference between the error made by the second closest prototype and the error made by the winning prototype itself. If this is low, it means that if the winning prototype were missing, distortion would have been approximately the same. Both accumulators for all prototypes are subject

Download English Version:

https://daneshyari.com/en/article/409127

Download Persian Version:

https://daneshyari.com/article/409127

Daneshyari.com