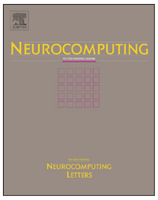
Neurocomputing 169 (2015) 272–280

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Analysis of high-dimensional data using local input space histograms

Jochen Kerdels*, Gabriele Peters

University of Hagen - Faculty of Mathematics and Computer Science, Human-Computer Interaction Universitätsstrasse 1, D-58084 Hagen, Germany

ARTICLE INFO

Article history:
Received 30 June 2014
Received in revised form
19 November 2014
Accepted 12 December 2014
Available online 4 April 2015

Keywords:
Local input space histograms
Prototype-based vector quantization
Growing neural gas
Curse of dimensionality
Minkowski distance

ABSTRACT

The idea of *local input space histograms* was recently introduced as a means to augment prototype-based vector quantization methods in order to gather more information about the structure of the respective input space. Here we investigate the utility of this new idea for analysing and clustering high-dimensional data. Our results demonstrate that the additional information gained about the input space structure can be used to enable and improve visualization and hierarchical clustering. Furthermore, we show that contrary to common view the Minkowski distance with $p > 1$ can be a meaningful distance measure for high-dimensional data.

1. Introduction

The analysis of data on a large scale is a challenging task. Commonly there is only few apriori knowledge available about structures contained within the data, e.g., information about possible classes the data could be partitioned into. In such a case methods that utilize forms of *unsupervised competitive learning* like the self-organizing map (SOM, [1]) or neural gas (NG, [2]) can be used to discover potential structures in the data. Both the SOM and NG are prototype-based vector quantization methods that use a set of prototypes to cover the particular input space as well as possible, i.e, to minimize the quantization error based on a given dissimilarity measure.

If there is little information about the structure of the data the Euclidean distance is often chosen as a "default" dissimilarity measure. In that case the individual prototypes can only represent local regions of the input space as convex polyhedrons and more complex structures must be approximated piecewise by multiple prototypes. In order to gather more information about the input space structure between prototypes the idea of *local input space histograms* [3] was introduced recently. As a proof of concept it has been shown that augmenting a growing neural gas (GNG, [4]) with local input space histograms can improve the discovery of non-convex clusters in two-dimensional datasets.

In this paper we investigate the utility of local input space histograms for analysing and clustering high-dimensional data. Section 2 introduces the methods and materials used in the subsequently described experiments. In particular, the section describes how a prototype-based vector quantization method – here a GNG – can be augmented by local input space histograms. In Section 3 the behavior of local input space histograms is analysed for high-dimensional random data as well as high-dimensional color histogram data. Section 4 discusses a number of interesting aspects of our results. Finally, a short conclusion and suggestions for further research are provided in Section 5.

2. Materials and methods

Growing neural gas revisited: To investigate the utility of local input space histograms for the analysis of high-dimensional data we extended a GNG as an exemplary prototype-based method. The GNG is a *topology representing network* [5], i.e., it uses a data-driven growth process to approximate the topology of the input space instead of using a fixed network topology like, e.g., a SOM does. Here we summarize the operation of the growing neural gas algorithm as described by Fritzke [4]. The growing neural gas is a network that consists of a set A of units and a set C of edges. Each unit $a \in A$ can be described by a tuple[1] (w, e) with the *prototype* $w \in \mathbb{R}^n$, with n being the dimension of the input space, and the accumulated error variable $e \in \mathbb{R}$. Each edge $c \in C$ can be described by a tuple (a, b, t) with the units $a, b \in A \wedge a \neq b$ that are connected by the edge and the variable $t \in \mathbb{N}$ which stores the current age of the edge. The direct neighborhood D_a of a unit a is defined as $D_a := \{b | \exists$

* Corresponding author.
E-mail address: Jochen.Kerdels@FernUni-Hagen.de (J. Kerdels).

[1] We use the notation $a^{(i)}$ to reference the ith element of a tuple beginning with index 1.

http://dx.doi.org/10.1016/j.neucom.2014.12.094
0925-2312/© 2015 Elsevier B.V. All rights reserved.

$(a,b,t)\in C, b\in A, t\in \mathbb{N}\}$. The network is initialized with two units that have random prototypes and accumulated error variables set to zero.

A given input $\xi\in\mathbb{R}^n$ is processed by the network in the following way:

- Find the two units s_1 and s_2 whose prototypes are closest to the input ξ:

  $$s_1 := \operatorname{argmin}\{a^{(1)}-\xi | a\in A\}, \quad s_2 := \operatorname{argmin}\{a^{(1)}-\xi | a\in A\backslash\{s_1\}\}.$$

- Increment the age of all edges connected to s_1:

  $$c^{(3)} := 0, \quad c\in C \wedge c^{(1)} = s_1 \wedge c^{(2)} = b, \quad \forall b\in D_{s_1}.$$

- If no edge exists between s_1 and s_2, create one:

  $$C := C \cup \{(s_1, s_2, 0)\}.$$

- Reset the age of the edge between s_1 and s_2 to zero:

  $$c^{(3)} := 0, \quad c\in C\wedge c^{(1)} = s_1 \wedge c^{(2)} = s_2.$$

- Add the squared distance between the input ξ and the prototype of unit s_1 to the accumulated error of s_1:

  $$s_1^{(2)} := s_1^{(2)} + \| s_1^{(1)} - \xi \|^2$$

- Adapt the prototype of s_1 and all prototypes of its direct neighbors $b\in D_{s_1}$:

  $$\Delta s_1^{(1)} := \epsilon_b\left(\xi - s_1^{(1)}\right), \quad \Delta b^{(1)} := \epsilon_n\left(\xi - b^{(1)}\right), \quad \forall b\in D_{s_1}.$$

- Remove all edges with an age above a given threshold $t_{\max}$ and remove all units that no longer have any edges connected to them.
- If an integer-multiple of λ inputs was presented to the network insert a new unit r. The new unit is inserted between the unit $q\in A$ with the maximum accumulated error and the unit $f\in D_q$ which has the largest accumulated error among the neighbors of q, i.e., the prototype of unit r is set to

  $$r^{(1)} := (q^{(1)} + f^{(1)})/2.$$

  Create edges between q and r as well as f and r, and remove the edge between the units q and f. Decrease the accumulated errors of q and f by a factor α and set the accumulated error of the new unit r to the decreased accumulated error of unit q.
- Finally, decrease the accumulated error of all units in A by a factor β.

Typically, the inputs ξ are randomly chosen from a set of training data and fed into the network until a given halting criterion (e.g., a maximum network size) is met. In all experiments the following parameter values were used:

$$\begin{array}{lll} \epsilon_b = 0.01, & \epsilon_n = 0.0001, & t_{\max} = 500, \\ \lambda = 2000, & \alpha = 0.5, & \beta = 0.0005. \end{array}$$

The parameters deviate from the values proposed by Fritzke [4]. They result in a slower development of the GNG which turns out to be more robust with respect to high-dimensional inputs. A slower development compensates for possible inhomogeneities in the training data, which are in general more likely to occur in high-dimensional data as the ratio between the number of available training data points and the size of the input space typically diverges with increasing dimension.

Local input space histograms: As described above, edges in a GNG network are created between the first and second best matching units (BMUs) s_1 and s_2 of each input ξ and are maintained as long as they are used regularly. Thus, the neighborhood relations among units represented by the GNG network indicate that the input space between connected units is not empty. However, the mere existence of an edge does not provide any further information about the underlying input space structure. The core idea of local input space histograms is to increase the available information in this regard by adding a small histogram $H=\{h_0,\dots,h_{k-1}\}$, e.g., with $k=16$ bins, to each edge $c\in C, c=(a,b,t,H)$ of the GNG network and to update this histogram for those inputs ξ that are mapped to the corresponding edge using a distance ratio r:

$$r := \frac{\| s_1^{(1)} - \xi\| - \| s_2^{(1)} - \xi\|}{\| s_1^{(1)} - s_2^{(1)}\|} + 1,$$

with $s_1^{(1)}$ and $s_2^{(1)}$ being the prototypes of the first and second BMUs for the given input ξ, respectively.

The ratio r lies in the interval [0,1] and describes how close the prototype of the best matching unit s_1 is to the input ξ in relation to the prototype of the second best matching unit s_2. A geometric interpretation of the distance ratio is depicted in Fig. 1a. As a local input space histogram $c^{(4)}$ is part of an edge $c\in C$ it is shared by the two units $c^{(1)}$ and $c^{(2)}$. Thus, the ratio r is used to either update the upper or the lower half of the histogram depending which of the units is the BMU s_1:

$$\Delta h_u = 1, \quad u = \begin{cases} \lfloor k(r/2)\rfloor & \text{if } c^{(1)} = s_1, \\ \lfloor k(1-r/2)\rfloor & \text{if } c^{(2)} = s_1, \end{cases} \quad h_u\in c^{(4)} = \{h_0,\dots,h_{k-1}\}.$$

The resulting histogram represents the distribution of the approximate, relative positions of those inputs that are located somewhere around the two connected units. Fig. 1b provides an example of local input space histograms occurring in a two-dimensional GNG that received uniform, random input.

The additional information provided by the local input space histograms allows us to characterize the input space in more detail. For example, it can be estimated if the input space between two connected units is sparse or dense. One measure to quantify this property is the average bin error[2] $\overline{e}_H$ of a histogram H:

$$\overline{e}_H := \frac{1}{k}\sum_{i=0}^{k-1} e_i, \quad e_i := \begin{cases} \sqrt{h_i}/h_i & \text{if } h_i > 0, \\ 1 & \text{if } h_i = 0, \end{cases} \quad h_i\in H = \{h_0,\dots,h_{k-1}\}.$$

In case of a local input space histogram $c^{(4)}$ the value of $\overline{e}_{c^{(4)}}$ will be near 1 if the corresponding region of input space is sparse and it will be close to 0 in case the input space is dense.

Distance measures: The analysis of high-dimensional data spaces is accompanied by a number of problems that are commonly referred to as the “curse of dimensionality” [6]. In this context a major problem is that the ability to discriminate data points by their relative distances diminishes with increasing dimensionality [7]. To observe the impact of different distance measures on the GNG and the local input space histograms we use the Minkowski distance d_p in our analysis with varying values for p:

$$d_p(x,y) := \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}, \quad x=(x_1,\dots,x_n),\ y=(y_1,\dots,y_n).$$

By choosing the Minkowski distance a range of popular distance measures can be covered: for $p=1$ it is equivalent to the Manhattan distance, for $p=2$ it is equivalent to the Euclidean distance , and for $p\rightarrow\infty$ it approaches the Chebyshev distance.

[2] Note: the definition of the average bin error given here differs from [3].