



Growing topology representing network

Kazuhiro Tokunaga*

National Fisheries University, 2-7-1 Nagata-Honmachi, Shimonoseki, Yamaguchi 759-6595, Japan



ARTICLE INFO

Article history:

Received 30 August 2012
Received in revised form 22 February 2014
Accepted 21 April 2014
Available online 4 June 2014

Keywords:

Growing Neural Gas
Gaussian mixture model
Online learning

ABSTRACT

This paper describes a method for finding the topology of a data distribution online using a new growing graph network architecture. Many growing neural networks for finding the topology of data online, such as the Growing Neural Gas, depend on the order and number of input data. For this reason, conventional methods have certain drawbacks: weakness to noise, generating redundant nodes, requiring a great deal of input data, and so on. The proposed method is robust with respect to these issues since it has been developed from the viewpoint of a generative model. This paper presents both the theory and an algorithm in this paper. Moreover, the effectiveness of the proposed method is shown by experiments comparing the proposed method with various growing graph networks.

© 2014 Elsevier B.V. All rights reserved.

Introduction

Aim of the work

Finding the topology of input data vectors is important in various applications such as object recognition, character recognition, structure recognition, and so on. Finding the topology of data online is necessary particularly for intelligent agents, because the structure of the data distribution is dynamically transformed by changes in the environment around the intelligent agents.

The Growing Neural Gas (GNG) proposed by Fritzke [1] is a typical method for finding the topology of data dynamically using a growing graph network. The network of a GNG has a graph structure where each node and each graph-path represent a reference vector and the similarity between nodes, respectively. The nodes and graph-paths are generated and updated in a self-organizing manner according to the input data vectors. Moreover, the required graph-paths survive automatically. This behavior of a GNG is performed in online learning. In other words, the GNG can dynamically adjust to new input data or transformation of the structure of the data distribution. As such, the GNG can be applied to hand recognition [2], map building by a robot [3], and so on.

In past works, various methods other than the GNG have been proposed for finding the structure of a data distribution by growing a graph network. (In this paper, these methods are generically called “growing neural networks”.) The many conventional growing neural networks include building the graph network directly from the observed data vectors. Hence, growing the graph network using a conventional growing neural network is commonly affected by noise. Besides, the learning result of the graph has no consistency, since the growth of the graph depends on the input order of the data. Moreover, conventional growing neural networks tend to generate redundant nodes. In the GNG for example, the number of nodes increases permanently in proportion to the number of learning steps. The reason for this is because conventional growing neural networks represent the distribution structure of the data using a large number of nodes. These characteristics in conventional growing neural networks have a negative influence on their practical application to intelligent agents. For example, redundant nodes waste the limited memory resources

of intelligent agents. Moreover, stability of the result is closely associated with the reliability of operating the system for practical purposes.

The above problems can mostly be attributed to the fact that the input data vector is used directly in the learning process. To solve this problem, the growth of the graph should be controlled by estimating the generative model representing the distribution structure of the data.

The aim of this work is to develop an algorithm for a growing neural network for application to intelligent agents from the perspective of a generative model. In particular, the generation of nodes is controlled by the information criterion. Likewise, the generation and update of graph-paths are controlled by generative models.

In this paper, the algorithm for the proposed method is developed based on an online Gaussian mixture model (GMM); henceforth, this proposed method is referred to as the “GTR: growing a topology representing network”. The GMM represents a probability density function (pdf) combining multiple Gaussian kernels. Thus, the GMM builds the generative model represented by the pdf. The GTR is an extension of the GMM that includes the following mechanisms: online learning, generation of Gaussian kernels according to the information criterion, and finding topologies using connected graph-paths between kernels.

This paper discusses the theory and an algorithm for the GTR. In addition, results of experiments comparing the GTR with three typical GNNs, namely, the GNG, Evolving Self-Organizing Map (ESOM), and Self-Organizing Incremental Neural Network (SOINN), are presented.

Related works for growing neural networks

If a growing neural network is used by an intelligent robot, the following characteristics are important: ability to operate with limited resources, robustness to noise, and fast creation of a useful graph network. In an autonomous robot that operates in a variety of environments such as on other planets and the sea floor, resources such as memory, are limited. Thus, it is preferable not to generate redundant nodes. Moreover, robustness to noise is required, since noise is often included in sensor information. In addition, a useful graph network should be created quickly from as little information as possible in order to adapt quickly to changes in the environment. The SOINN and ESOM are examples of growing neural networks that can adapt to such situations.

* Tel.: +81832865111.

E-mail address: tokunaga@fish-u.ac.jp

Apart from the GNG, the SOINN, proposed by Furoo et al. [4] has greater robustness to noise than any of the other growing neural networks. Moreover, in the SOINN, the graph networks are dynamically incremented even if the data are observed from different environments. The SOINN has been adapted for use in various applications including associative memory for noisy input [5], path planning for robots, and so on [6]. The ESOM [7], proposed by Deng et al., supports the feature of a fast growing graph network. Moreover, it is able to find the topology with very limited input data.

The above two methods are extremely useful compared with other growing neural networks, and provide very effective techniques depending on the task. In applications for intelligent robots, however, various problems need to be solved. The SOINN requires a lot of learning data to find the topology. In addition, a great many nodes are generated depending on the task. Hence, it is difficult to use the SOINN in practice for the evolution of intelligent agents where resources such as memory, time, and input data are limited. Furoo et al. proposed an enhanced self-organizing incremental neural network (ESOINN) which is an improvement of the SOINN. The ESOINN, which is more useful than the SOINN, can perform classifications accurately even if the distribution between classes is close [8]. However, the ESOINN still does not provide a solution to the problems mentioned above. On the other hand, in the ESOM, there are cases where the distribution of the input data cannot be represented appropriately by the generated graph network, since the node increment is controlled by only one threshold value (the radius of the territory at the node).

In contrast, the algorithm for the proposed method has been developed from the viewpoint of a generative model to solve these problems. The graph network is generated based on a stochastic approach. Thus, the GNG, ESOM, and SOINN are considered in the comparative experiments. The ESOINN is not used for comparison because the aim of the comparative experiments is to verify the performance characteristic of the proposed method and to highlight the problems of conventional growing neural networks.

Related works for incremental learning based GMMs

Various methods for incremental learning based on the GMM have been proposed. Arandjelovic et al. proposed incremental learning for temporally-coherent GMMs [9]. In this method, the GMM is executed with online learning. Moreover, components of the Gaussian kernels are split or merged according to the input data. For this reason, an appropriate number of Gaussian kernels is generated. Bouchachia et al. proposed incremental learning based on growing GMMs [10]. This method also has mechanisms for merging and splitting the Gaussian kernel. Moreover, this method can label each Gaussian kernel using semi-supervised learning. In other words, online estimation of the parameter for the GMM and clustering are performed at the same time. However, these methods cannot perform unsupervised clustering. On the contrary, this is possible in the proposed method, using the generated graph network.

Framework and theory

The proposed method, the GTR, performs three processes concurrently: (1) online parameter estimation of the GMM, (2) controlling the generation of Gaussian kernels using an information criterion, and (3) generating a graph-path representing the topology between kernels and updating the graph-path's strength. First, a framework for the proposed method is presented, and then each of the above processes are explained in turn.

Framework

The framework of tasks to which the proposed method will be applied, is given below.

- Sequentially-observed data vectors cannot be stored in memory.
- The appropriate number of Gaussian kernels is unknown. (In the first stage of learning, the number of kernels is one.)
- The parameters (mixing parameter, and mean and covariance matrices) for each kernel are unknown.
- The class information for data vectors is unknown.
- Noise is added to the data vector.

Under the above conditions, processes for finding the generative model of the input data and generating the graph-paths are performed simultaneously in online learning.

Online GMM

The GMM is widely known as a parametric approach for estimating the probability density function and represents the probability density function using a mixture of multiple Gaussian kernels.

Now, at time t , let a d -dimensional data vector \mathbf{x}_t be observed. Then, in the GMM composed of K kernels, the probability density function $p(\mathbf{x}_t)$ is described as

$$p(\mathbf{x}_t) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad \sum_{k=1}^K \pi_k = 1. \quad (1)$$

Here, the Gaussian kernel is represented by a normal probability distribution $\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, π^k is the mixing parameter of the k th kernel, and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance matrices, respectively, in the k th Gaussian kernel.

In online parameter estimation for an online GMM, the extremum of the objective function

$$L(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \log \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} - \lambda \left(\sum_{k=1}^K \pi_k - 1 \right), \quad (2)$$

defined using Lagrange's method for undetermined multipliers, is found using a hill climbing method. Updating expressions for each parameter are defined as follows:

$$\boldsymbol{\mu}_k^{\text{new}} = \boldsymbol{\mu}_k^{\text{old}} + \epsilon_\mu (\boldsymbol{\Sigma}_k^{\text{old}})^{-1} \gamma_k(\mathbf{x}_t) \{ \mathbf{x}_t - \boldsymbol{\mu}_k^{\text{old}} \}, \quad (3)$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \boldsymbol{\Sigma}_k^{\text{old}} + \frac{1}{2} \epsilon_\Sigma \gamma_k(\mathbf{x}_t) (\boldsymbol{\Sigma}_k^{\text{old}})^{-1} \left\{ (\mathbf{x}_t - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_t - \boldsymbol{\mu}_k^{\text{new}})^T (\boldsymbol{\Sigma}_k^{\text{old}})^{-1} - \mathbf{I} \right\}, \quad (4)$$

$$\pi_k^{\text{new}} = \pi_k^{\text{old}} + \epsilon_\pi \left\{ \frac{\gamma_k(\mathbf{x}_t)}{\pi_k^{\text{old}}} - 1 \right\}. \quad (5)$$

Here, ϵ_μ , ϵ_Σ , and ϵ_π are the learning rates, such that $0 < \epsilon_\mu, \epsilon_\Sigma, \epsilon_\pi < 1.0$. The setting of parameters is done through trial-and-error. Usually ϵ_μ is set to a larger value than ϵ_Σ and ϵ_π . Moreover, $\gamma_k(\mathbf{x}_t)$ is the posterior probability (responsibility) defined as:

$$\gamma_k(\mathbf{x}_t) = \frac{\pi_k^{\text{old}} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_k^{\text{old}}, \boldsymbol{\Sigma}_k^{\text{old}})}{\sum_{k'=1}^K \pi_{k'}^{\text{old}} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{k'}^{\text{old}}, \boldsymbol{\Sigma}_{k'}^{\text{old}})}. \quad (6)$$

Generation of Gaussian kernel using information criteria

Generation of the Gaussian kernel is controlled by the Bayesian Information Criteria (BIC) [11]. The BIC are defined as:

$$\text{BIC}(K) = -2 \ln L(K) + 2C(K) \ln(n). \quad (7)$$

Here, K , $L(K)$, and $C(K)$ are the number of kernels, the maximum likelihood in K kernels, and the degrees of freedom of the model, respectively. Moreover, n is the number of data points. In the GMM, the number of kernels is chosen to maximize the information criteria, $\text{BIC}(K)$. In other words, the number of kernels is determined so that the maximum likelihood is given by as few kernels as possible. Generally, in the GMM, the number of kernels is controlled offline using all the input data. However, the GTR controls the generation of the kernel from a single observed data vector in online learning.

Suppose that data vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T-1}$ have been observed by time $T-1$. Now, at time T , let a new input data

Download English Version:

<https://daneshyari.com/en/article/495355>

Download Persian Version:

<https://daneshyari.com/article/495355>

[Daneshyari.com](https://daneshyari.com)