Pattern Recognition 53 (2016) 184–194

Contents lists available at ScienceDirect

Pattern Recognition

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

Mixture of grouped regressors and its application to visual mapping

Lili Pan [a,*], Jason M. Saragih [b], Wen-Sheng Chu [c]

[a] University of Electronic Science and Technology of China, Chengdu, China
[b] TrueVisionSolutions Pty. Ltd., Brisbane, Australia
[c] Carnegie Mellon University, Pittsburgh, USA

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form
25 September 2015
Accepted 19 October 2015
Available online 11 November 2015

Keywords:
Mixture of regressors
Group partition
EM algorithm

ABSTRACT

Mixture of regressors (MoR) is a widely used regression approach for approximating nonlinear mappings between input and target outputs. However, existing learning procedures for MoR are prone to overfitting when only limited amounts of training data are available. To address this problem, we propose a new mixture regression model, named mixture of grouped regressors (MoGR). It partitions the individual regressors in the model into a set of groups, where the parameters of the regressors within each group are encouraged to take on similar values. As the parameters for each local regressor are learned using all data within a group, they tend to be better conditioned and more robust to noise in the training data. Extensive experiments on real-world head pose and gaze data demonstrate the benefits of our proposed MoGR model.

1. Introduction

Regression is the process of learning a mapping function from input to output variables, such that the target output can be predicted through the learned mapping function when given a new input. Many problems in pattern recognition and computer vision can be posed as regression problems, which include, but are not limited to, classification [1], discriminant analysis [2], object detection/tracking [3] and pose estimation [4–6]. Based on the form taken by the input–output relationship, regression approaches are often classified into two categories: linear and nonlinear regression.

In linear regression, it is assumed that the inputs are linearly related to their corresponding outputs. One advantage of this model is its low computational cost and ease of implementation. Its effectiveness has been shown in many computer vision problems, including visual tracking and image alignment, where the linear relationship assumption between input image features and target output is approximately satisfied [7,8]. However, many real-word problems do not exhibit a linear input–output relationship, limiting the utility of linear regression for these problems.

Compared with linear regression, nonlinear regression is more sophisticated and capable of accounting for the potentially complex relationship between inputs and their corresponding outputs. Approaches of this kind include Gaussian process regression (GPR) [9], mixture of regressors (MoR) [6,10], kernel partial least square regression (KPLSR) [11], support vector regression (SVR) [12], random regression forest (RRF) [4,13,14] and so forth. These methods have seen wide applicability, such as head pose estimation, human pose estimation and gaze estimation. In [15], Li et al. proposed adopting support vector regression (SVR) [12] for head pose estimation and in [11,16], Haj et al. applied the kernel version of partial least squares to both head pose and human pose estimation problems. Moreover, Williams et al. [9] proposed a sparse, semi-supervised Gaussian process regression model for gaze estimation. The disadvantage of the aforementioned three methods is the difficulty of finding the appropriate kernel function for mapping, which is highly application- and data-dependent. Random regression forests/ferns is an especially effective nonlinear regression approach and has been successfully used for head pose estimation, face alignment and age estimation in the work of [4,14,17]. Although highly robust results can be achieved through these approaches, they rely on large amounts of data and their generalization quickly deteriorates with limited data and labeling noise.

Mixture of regressors (MoR), as a nonlinear method, is best suited to problems where the data is easily partitioned into discrete set of categories. One of the well-known MoR methods is mixture of linear regressors (MoLR) [18]. It allows a soft and probabilistic split of input and output data into different clusters and employs a linear regressor to model the local mapping relationship between input and output data associated with each

* Corresponding author.
E-mail addresses: lilipan@uestc.edu.cn, panlili8255@gmail.com (L. Pan), jason.saragih@truevisionsolutions.net (J.M. Saragih), wschu@cmu.edu (W.-S. Chu).

http://dx.doi.org/10.1016/j.patcog.2015.10.016
0031-3203/

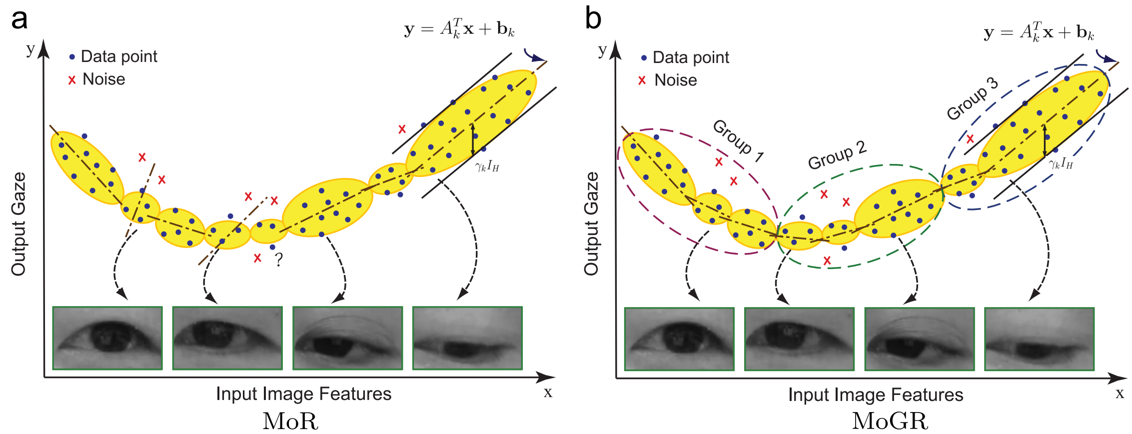

Fig. 1. Illustration of MoGR. Fig. 1(a) and (b) show MoR and MoGR, respectively, for the problem of gaze estimation. MoGR avoids the overfitting and underdetermined problems of MoR caused by limited training data. Here, $\mathbf{A}_k$ and $\mathbf{b}_k$ are the regression matrix and bias of each local regressor and $\gamma_k \mathbf{I}_H$ is the noise covariance of each regressor.

cluster [18]. Based on MoLR, mixture of experts (MoE), proposed by Jacobs in [10], define the mixing coefficients as functions of the input. Agarwal et al. successfully applied MoE to human pose estimation problem in [6]. Tian et al. [19] proposed to fit MoE in the latent space of input and output observations to improve the prediction accuracy for human pose estimation. MoR has been shown to accurately capture the complex global mapping relationship between input and output when local linear regressors are used as their mixing elements. This is particularly useful as it leads to a simple iterative learning procedure that alternates between cluster assignment and solving for the parameters of each local linear regressor. However, this often leads to poorly conditioned solutions when data is limited as each mixture element only *sees* a subset of all available data (see Fig. 1(a)). To address this drawback, mixture of lasso regressors and mixture of ridge regressors have been proposed in the related works [20,21]. Incorporating a Laplace or Gaussian prior on regression parameters leads to better determined solutions. However, this is achieved at the cost of biasing the solution towards smaller values.

In this work, we propose a novel mixture of grouped regressor (MoGR) that effectively circumvents the aforementioned learning problem inherent in most conventional MoR approaches. As illustrated in Fig. 1, the main idea of MoGR is to partitions the individual regressors in the model into a set of groups, where the parameters of the regressors within each group are encouraged to take on similar values. As the parameters for each local regressor are learned using all data within a group, they tend to be better conditioned and more robust to noise in the training data. We motivate our construction through a probabilistic interpretation and demonstrate its efficacy on the head pose and gaze estimation problems, where it is shown to outperform conventional approaches.

We begin in Section 2 with a review of mixture of linear regression. We develop our MoGR method in Section 3 and provide experimental results in Section 4. We conclude in Section 5 and propose directions for future work.

2. Mixture of linear regressors (MoLR)

Mixture of linear regressors (MoLR) is a commonly used MoR model. In this section, we give a brief review of MoLR that will form the basis of our proposed construction in Section 3. Given a set of input and output pairs $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$, where $\mathbf{x}_n \in \Re^{D\times 1}$ and $\mathbf{y}_n \in \Re^{H\times 1}$, MoLR [18] models the mapping function from input to target output with a mixture of K local linear regressors, each of which is governed by its regression matrix $\mathbf{A}_k \in \Re^{D\times H}$, bias

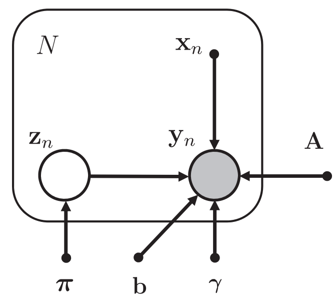

Fig. 2. Probabilistic directed graph representing MoLR. Here, $\mathbf{z}_n$ is a latent variable indicating which regressor the input and output pair $\{\mathbf{x}_n, \mathbf{y}_n\}$ belongs to.

$\mathbf{b}_k \in \Re^{H\times 1}$, and the covariance of regression noise $\gamma_k \mathbf{I}_H \in \Re^{H\times H}$, $k = 1, 2, \ldots, K$ (the mean of regression noise is assumed to be zero). This is illustrated in Fig. 1. To indicate regressor association of every data point $\{\mathbf{x}_n, \mathbf{y}_n\}$, a K-dimensional binary latent variable $\mathbf{z}_n$ is introduced, where $z_{kn} \in \{0, 1\}$ and $\sum_k z_{kn} = 1$. If data point n is assigned to component k then $z_{kn} = 1$, otherwise $z_{kn} = 0$. The marginal distribution over $\mathbf{z}_n$ is specified in terms of mixing coefficients π_k where $0 \le \pi_k \le 1$ and $\sum_k \pi_k = 1$. If we use $\mathbf{\Theta}$ to denote all the parameters in MoLR model, namely $\mathbf{A} = \{\mathbf{A}_k\}_{k=1}^K$, $\mathbf{b} = \{\mathbf{b}_k\}_{k=1}^K$, $\boldsymbol{\gamma} = \{\gamma_k\}_{k=1}^K$, and $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^K$, the log likelihood function for MoLR takes the form

$$\ln p(\mathbf{Y}|\mathbf{X}, \mathbf{\Theta}) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{y}_n | \mathbf{A}_k^T \mathbf{x}_n + \mathbf{b}_k, \gamma_k \mathbf{I}_H\right), \tag{1}$$

where $\mathbf{X} \in \Re^{D\times N}$ is a matrix in which the nth column is given by $\mathbf{x}_n$, $\mathbf{Y} \in \Re^{H\times N}$ and $\mathbf{Z} \in \Re^{K\times N}$ are defined similarly. The graphical model of MoLR is shown in Fig. 2.

The maximal likelihood solution of $\mathbf{\Theta}$ can be found by using EM algorithm [18]. In the E-step, the current parameter values $\mathbf{\Theta}^{old}$ is used to calculate the posterior probability, or responsibility of each point belonging to the kth regressor as

$$p\left(z_{kn} = 1 | \mathbf{x}_n, \mathbf{y}_n, \mathbf{\Theta}^{old}\right) = \frac{\pi_k \mathcal{N}\left(\mathbf{y}_n | \mathbf{A}_k^T \mathbf{x}_n + \mathbf{b}_k, \gamma_k \mathbf{I}_H\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{y}_n | \mathbf{A}_j^T \mathbf{x}_n + \mathbf{b}_j, \gamma_j \mathbf{I}_H\right)}. \tag{2}$$

Denoting $w_{kn} = p\left(z_{kn} = 1 | \mathbf{x}_n, \mathbf{y}_n, \mathbf{\Theta}^{old}\right)$ in order to keep the notation uncluttered, the so-called $\mathcal{Q}$ function in EM algorithm is constructed as follows:

$$\mathcal{Q}\left(\mathbf{\Theta}, \mathbf{\Theta}^{old}\right) = \sum_{\mathbf{Z}} p\left(\mathbf{Z} | \mathbf{X}, \mathbf{Y}, \mathbf{\Theta}^{old}\right) \ln p\left(\mathbf{Y}, \mathbf{Z} | \mathbf{X}, \mathbf{\Theta}^{old}\right) \tag{3}$$