Signal Processing 118 (2016) 89–96

Contents lists available at ScienceDirect

Signal Processing

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

Fast communication

Generalized adaptive weighted recursive least squares dictionary learning

Yashar Naderahmadian [a,b,*], Mohammad Ali Tinati [a], Soosan Beheshti [b]

[a] Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz 51666-15813, Iran
[b] Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada

ARTICLE INFO

Article history:
Received 3 March 2015
Received in revised form
20 May 2015
Accepted 16 June 2015
Available online 25 June 2015

Keywords:
Dictionary learning
Sparse signal representation
Adaptive weighting
Recursive least squares

ABSTRACT

Recursive least squares (RLS) dictionary learning algorithm is one of the well-known dictionary update approaches which continuously update the dictionary per arrival of new training data. In RLS algorithm a forgetting factor is added to control the memory and the effect of the previous data in the dictionary update stage. In this paper, we generalize the RLS algorithm by introducing an additional correction weight for the arrival data. This additional correction weight adaptively controls the relative consistency between the arrival data and the existing dictionary estimate. Consequently, we show that the conventional RLS is a special case of our method. Synthetic data, with and without containing outliers, are used to train both methods. Experimental results verify that adding the correction weight in our proposed method improves the recovery of original dictionary and MSE of sparse representation for both types of training data. The improvement increases as the percentage of outliers increase.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Dictionary learning for sparse representation of signals is one of the active topics in various areas such as compression, denoising, inpainting, super-resolution, classification, and source separation [1–5]. Dictionary is a collection of atoms which can represent each training data by a linear combination of the atoms. The dictionary used in sparse representation is an over-complete one. Two types of over-complete dictionaries are available. In *Designed* dictionaries the atoms are selected from the available basis such as Fourier, Wavelet, Discrete Cosine Transform (DCT), etc. [6–11]. In *Trained* dictionaries the atoms of the dictionary are learned from the training data [12–15]. The latter deals with two parameters to optimize: the dictionary atoms and the sparse coefficients that linearly relate training data to the atoms. Dictionary learning algorithms generally use two-step alternating optimization method to find these two parameters. The first step, denoted as sparse coding, assumes that the dictionary is available and uses sparse approximation methods [9,16–19] to find the sparse representation of the data. The second step, denoted as dictionary update, uses the sparse coefficients from the previous step to update the dictionary atoms. Dictionary learning algorithms generally focus on the second step [12–15,20–23].

The focus of this paper is on recursive least squares (RLS) dictionary update algorithm [13]. The RLS dictionary update was proposed after method of optimal directions (MOD) algorithm [14] and both methods use the least squares to update the dictionary. In MOD a matrix inversion at each iteration increases the computational complexity in dealing with large data set. Consequently, RLS dictionary learning was proposed to solve the same least squares recursively to reduce the complexity of the

* Corresponding author at: Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz 51666-15813, Iran.
E-mail addresses: naderahmadian@gmail.com (Y. Naderahmadian), tinati@tabrizu.ac.ir (M.A. Tinati), soosan@ee.ryerson.ca (S. Beheshti).

http://dx.doi.org/10.1016/j.sigpro.2015.06.013
0165-1684/© 2015 Elsevier B.V. All rights reserved.

algorithm. In addition, the RLS algorithm uses a forgetting factor λ to improve the convergence speed. We denote this RLS by λ-RLS.

In this paper, we generalize the RLS algorithm by adding an extra correction weight for the *arrival* data. We denote our method by generalized adaptive weighted recursive least squares (GAW-RLS) and show the RLS and λ-RLS methods presented in [13] are special cases of GAW-RLS method. The simulation results show that our proposed method improves MSE (mean squared error) and original dictionary recovery of the RLS and λ-RLS methods. In addition, the generalized algorithm is more robust in dealing with training data containing outliers. The paper structure is as follows: Section 2 provides a brief review of dictionary learning; Section 3 introduces GAW-RLS method; and Section 4 contains the simulation results.

2. Dictionary learning and motivation

In dictionary learning, a set of vectors called dictionary as $\mathbf{D}=[\mathbf{d}_1,\mathbf{d}_2,\ldots,\mathbf{d}_n]\in\mathbb{R}^{m\times n}$ are trained by using a set of training data as $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_L]\in\mathbb{R}^{m\times L}$. Each data in the training data set can be represented by a linear combination of a small number of the dictionary columns as follows:

$$\mathbf{y}_i\simeq\mathbf{D}\mathbf{x}_i=\sum_{j=1}^{n}\mathbf{d}_j\mathbf{x}_i(j)=\{\mathbf{d}_1\mathbf{x}_i(1)+\mathbf{d}_2\mathbf{x}_i(2)+\cdots+\mathbf{d}_n\mathbf{x}_i(n)\}. \tag{1}$$

In this representation, dictionary $\mathbf{D}$ is over-complete (has more columns than rows $m<n$) and each $\mathbf{d}_j$ is called an atom of the dictionary. The $\mathbf{x}_i$ is the sparse representation of $\mathbf{y}_i$ over $\mathbf{D}$, where $\mathbf{x}_i(j)$ is the jth entry of $\mathbf{x}_i$ and only S (sparsity level and $S\ll m$) number of them are non-zero.

Given a set of training data $\mathbf{Y}$, number of dictionary atoms n, and sparsity level S, dictionary learning finds $\hat{\mathbf{D}}$ (dictionary) and $\hat{\mathbf{X}}$ (sparse representation of $\mathbf{Y}$ over $\mathbf{D}$) by minimizing a desired cost function as follows:

$$\left\{\hat{\mathbf{D}},\hat{\mathbf{X}}\right\}=\arg\min_{\mathbf{D},\mathbf{X}} C(\mathbf{Y},\mathbf{D},\mathbf{X})\quad \text{s.t.}$$
$$\forall j\in\{1:n\}\,\|\mathbf{d}_j\|_2=1,\quad \forall i\in\{1:L\}\,\|\mathbf{x}_i\|_0\le S \tag{2}$$

where $\|\cdot\|_0$ is the l_0 norm and shows the number of non-zero elements in $\mathbf{x}$ and $\|\cdot\|_2$ is the l_2 norm. One of the most used cost functions considered for optimization is the total sparse representation error for the training data set as follows:

$$\left\{\hat{\mathbf{D}},\hat{\mathbf{X}}\right\}=\arg\min_{\mathbf{D},\mathbf{X}}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2\quad \text{s.t.}$$
$$\forall j\in\{1:n\}\,\|\mathbf{d}_j\|_2=1,\quad \forall i\in\{1:L\}\,\|\mathbf{x}_i\|_0\le S \tag{3}$$

in which $\|\cdot\|_F$ is the Frobenius norm and for a matrix like $\mathbf{A}\in\mathbb{R}^{m\times n}$ can be calculated as $\|\mathbf{A}\|_F^2=\sum_{all\,i,j}a_{ij}^2$ where a_{ij} is the entry in the ith row and jth column of $\mathbf{A}$. The cost function in (3) is non-convex, since both $\mathbf{D},\mathbf{X}$ are unknown. Consequently, this optimization problem is solved by alternating minimization approach which uses the two steps, briefly described in the following section.

2.1. Two steps of dictionary learning

The two steps of dictionary learning are as follows:

1) *Sparse coding step*: Considers a fixed dictionary $\mathbf{D}$ and estimates $\mathbf{X}$ in the following cost function by using sparse approximation methods:

$$\forall i\in\{1:L\},\quad \hat{\mathbf{x}}_i=\arg\min_{\mathbf{x}_i}\|\mathbf{y}_i-\mathbf{D}\mathbf{x}_i\|_2\quad \text{s.t.}\ \|\mathbf{x}_i\|_p\le S \tag{4}$$

in which $\|\cdot\|_p$ is the l_p-norm. To solve the cost function in (4), the following three approaches are proposed: *greedy algorithms* for $p=0$, *relaxation methods* for $p=1$, and methods such as focal under-determined system solver (FOCUSS) for $0<p<1$. Details of these methods can be found in [9,12,13,16–19].

2) *Dictionary update step*: Uses the sparse coefficients matrix $\mathbf{X}$ from step 1 and updates the dictionary $\mathbf{D}$ by solving the following cost function:

$$\hat{\mathbf{D}}=\arg\min_{\mathbf{D}}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2\quad \text{s.t.}\,\forall j\in\{1:n\}\,\|\mathbf{d}_j\|_2=1. \tag{5}$$

Several approaches have been proposed to find the solution of (5) [12–15,19,24,25]. The focus of this paper is on the least squares based methods. For example, the least squares solution for (5) is provided in MOD algorithm [14] as follows:

$$\hat{\mathbf{D}}=\mathbf{Y}\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1} \tag{6}$$

This solution is straightforward, however, includes matrix inversion at each step which increases the complexity of the problem in dealing with large number of training data. To avoid this matrix inversion, the RLS algorithm [13] uses online and recursive implementation of the MOD algorithm. The RLS algorithm has simple parameters update and the data have the same value of importance in training of this algorithm. The RLS dictionary learning to improve the algorithm has adapted the idea of using forgetting factor from the adaptive filters [13]. Motivated by the role of the forgetting factor as a proportional weight on the *passed* data, in this work we plan to study and analyze introducing an additional weight for the *arrival* data.

3. Generalized adaptive weighted recursive least squares dictionary learning (GAW-RLS)

The λ-RLS algorithm presented in [13] considers the following cost function:

$$f_R(\mathbf{D})=\sum_{j=1}^{i}\lambda^{i-j}\|\mathbf{r}_j\|_2^2=\sum_{j=1}^{i-1}\lambda^{i-j}\|\mathbf{r}_j\|_2^2+\|\mathbf{r}_i\|_2^2 \tag{7}$$

where subscript R in $f_R(\mathbf{D})$ is used for RLS algorithm and i is the number of training data, $\mathbf{r}_j$ represents the sparse representation error for the jth data, and λ is the forgetting factor. The update of the dictionary is found as follows [13]:

$$\mathbf{D}_i=\arg\min_{\mathbf{D}} f_R(\mathbf{D})=\arg\min_{\mathbf{D}}\sum_{j=1}^{i-1}\lambda^{i-j}\|\mathbf{r}_j\|_2^2+\|\mathbf{r}_i\|_2^2 \tag{8}$$

Download English Version:

https://daneshyari.com/en/article/566301

Download Persian Version:

https://daneshyari.com/article/566301

Daneshyari.com