Neurocomputing 169 (2015) 262–271

Contents lists available at ScienceDirect

Neurocomputing

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

Asynchronous gossip principal components analysis

Jerome Fellus*, David Picard, Philippe-Henri Gosselin

ETIS – UMR CNRS 8051 – ENSEA – Universite de Cergy-Pontoise, 6 Avenue du Ponceau, 95014 Cergy, France

ARTICLE INFO

Article history:
Received 30 June 2014
Received in revised form
13 October 2014
Accepted 21 November 2014
Available online 4 April 2015

Keywords:
Distributed machine learning
Dimensionality reduction
Gossip protocols

ABSTRACT

This paper deals with Principal Components Analysis (PCA) of data spread over a network where central coordination and synchronous communication between networking nodes are forbidden. We propose an asynchronous and decentralized PCA algorithm dedicated to large scale problems, where "large" simultaneously applies to dimensionality, number of observations and network size. It is based on the integration of a dimension reduction step into a gossip consensus protocol. Unlike other approaches, a straightforward dual formulation makes it suitable when observed dimensions are distributed. We theoretically show its equivalence with a centralized PCA under a low-rank assumption on training data. An experimental analysis reveals that it achieves a good accuracy with a reasonable communication cost even when the low-rank assumption is relaxed.

1. Introduction

Dimensionality reduction plays an important role in solving large scale machine learning problems where input data usually consists of a huge number of observations in a high-dimensional space. Classification, regression, or similarity ranking of such raw data often raise computation and storage issues. In practice, the intrinsic dimensionality of observed phenomena is much lower than the extrinsic dimension of the input space. Dimensionality reduction then aims at projecting input data into a lower-dimensional space such that subsequent learning stages keep a maximal accuracy.

Principal Components Analysis (PCA) is a linear approach to dimensionality reduction [1,2]. Given a sample matrix $\mathbf{X} \in \mathbb{R}^{D\times n}$ made of n observations in $\mathbb{R}^D$, PCA finds an orthonormal basis $\mathbf{U}^\star = [\mathbf{u}_1 \dots \mathbf{u}_q]$, $\mathbf{u}_k \in \mathbb{R}^D$ that projects the input sample $\mathbf{X}$ into the q-dimensional subspace, $q < D$, that retains the maximal variance in $\mathbf{X}$. Equivalently, the PCA solution is the linear projection that best conserves the Gram matrix (*i.e.*, the matrix of pairwise inner-products):

$$\mathbf{U}^\star = \underset{\mathbf{U}\in\mathbb{R}^{D\times q}}{\arg\min} \|\mathbf{X}^\top\mathbf{X} - \mathbf{X}^\top\mathbf{U}\mathbf{U}^\top\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \mathbf{U}^{\star\top} = \mathbf{U}^{\star-1} \tag{1}$$

The optimal conservation of the inner product makes PCA particularly suited to feed algorithms that solely rely on the inner product instead of the input data [3] (*e.g.*, Support Vector Machines). PCA enjoys a closed-form solution, as $\mathbf{U}^\star$ is made of the q leading eigenvectors of the sample covariance matrix [2]:

$$\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top - \mu\mu^\top \quad \text{where } \mu = \frac{1}{n}\mathbf{X}\mathbf{1} \text{ is the sample mean} \tag{2}$$

Like most statistical learning tools, PCA was formulated for centralized setups where all data are available at a single location. This assumes that the solution can be computed by a single machine and that all intermediary results fit in the main memory. However, this assumption is unrealistic in most applicative fields that deal with very large sample matrices. For instance, in biomedical, multimedia, or remote sensing applications, D and n often grow up to millions. The sample and covariance matrices then scale in Terabytes. Moreover, the $O(D^3)$ complexity of covariance eigendecomposition translates into an exaflop computation cost. Besides, along with the democratization of connected devices, data tends to originate from an increasing number of distributed sources with reasonable computing capacity, immersed in large unreliable networks without any central coordination. This has led to a number of so-called Distributed PCA algorithms, designed to deal with the spread of input data over multiple networking nodes.

Because computing μ and $\mathbf{C}$ would involve the full data $\mathbf{X}$ which is unknown to individual nodes, distributed PCA requires dedicated algorithms combining node-local optimization and inter-node communications. Distributed PCA encompasses two main scenarios, depending on the way the entries of $\mathbf{X}$ are spread over the networking nodes. Consider a network made of N (strongly connected) nodes. In a Distributed Samples (DS) scenario, each node i holds a distinct sample $\mathbf{X}_i \in \mathbb{R}^{D\times n_i}$ of n_i observations (*i.e.*, the *columns* of $\mathbf{X}$ are distributed). Conversely, in a Distributed Coordinates (DC) scenario, each node holds all n observations, but only gets a subset $\mathbf{X}_i \in \mathbb{R}^{D_i\times n}$ of their components (*i.e.*, the *rows* of $\mathbf{X}$ are distributed). On average, each $\mathbf{X}_i$ is then N times smaller than $\mathbf{X}$, thus $n_i \gg D$ in DS case while $D_i \gg n$ in DC

* Corresponding author.
E-mail addresses: jerome.fellus@ensea.fr (J. Fellus), picard@ensea.fr (D. Picard), gosselin@ensea.fr (P.-H. Gosselin).

http://dx.doi.org/10.1016/j.neucom.2014.11.076
0925-2312/

case. No assumption is made on their exact sizes, as they may be very different at each node. In both DS and DC scenarios, a typical objective is to provide all nodes with compressed representations of their locally hosted observations that account for the contribution of all components. Nodes then have to find a consensus basis $\mathbf{U}^{\star}$ that minimizes the PCA objective defined in Eq. (1) over the complete data. Usually, one also wants an operator that allows projection of future observations into the same output space, but not all approaches are able to provide such operator at a low cost. In spite of similar goals, the DS and DC scenarios have been tackled separately with very different approaches in the distributed PCA literature [4].

In this work, we consider the asynchronous decentralized PCA problem, which specializes distributed PCA by adding the following constraints:

(C1) No sample exchange Samples cannot be exchanged between nodes, for size, privacy or property reasons.
(C2) Asynchrony Nodes must never wait for each other.
(C3) Decentralization All nodes and links must play the same role. Formally, nodes and links must be selected for communication with the same probability and all nodes must run the same procedures.
(C4) Consensus All nodes must obtain the same orthogonal basis. Node-local solutions must allow projection of future observations.

Satisfying these four constraints, a distributed PCA algorithm gains applicability to a wider range of networking situations such as sensor networks, Internet-enabled multimedia devices, and cloud computing systems, where central coordination or synchronous functioning can be inapplicable. This work focuses on distributed setups where the main limiting factor is not the node local computing capacity but rather the network infrastructure. In such setups, communication cost and latency, synchronization constraints and unpredictable (dis)connections can be much more harmful than slightly more demanding computations. This is typically the case in sensor or mobile networks with extremely large number of nodes.

In this paper, we propose a decentralized and asynchronous algorithm called Asynchronous Gossip Principal Component Analysis (AGPCA) that satisfy all the above constraints. Our algorithm is a revision and extension of previous work presented in [5]. The original contributions of this paper are the following:

- We give a formal and in-depth description of AGPCA in the DS case, as well as the intuitions leading to the algorithm.
- We propose an extension to the DC case through a dual transcription.
- Provided a low rank property is met on the data, we give a theoretical guarantee that AGPCA yields the exact solution of the centralized PCA.
- We present experiments for both DS and DC scenarios, as well as results for various network topologies.

The remaining of this paper is organized as follows: In the next section, we detail related works on distributed PCA algorithms. Then, we present our AGPCA algorithm for the DS case in Section 3. We present the extension to the DC case in Section 4. In Section 5, we theoretically show that the output of AGPCA is identical to a centralized PCA under a low-rank assumption on the data. The last section gathers experimental results both in DS and DC cases, before we conclude.

2. Related work

In this section, we present existing algorithms for distributed PCA. We first present methods that integrate prior information on the input data. Then, we compare existing algorithms in terms of decentralization and asynchrony. Finally, we discuss the benefits of approaches based on model aggregation over those based on iterative optimization passes over the data.

2.1. Prior knowledge about input data

Existing distributed PCA approaches can be first distinguished by their level of prior knowledge about the input data matrix $\mathbf{X}$. Indeed, $\mathbf{X}$ can either carry node-local observations independently of their network relationships or integrate properties of the network graph itself. In the latter case, a typical object of interest is the adjacency matrix of the weighted network graph, whose entries correspond to some scalar relationship *between* nodes. For instance, when these entries represent estimated geographic distances between neighboring sensors, their absolute geographic position can be recovered by computing the three principal components through distributed PCA [6].

Other methods have considered the case where the data distribution inherits specific characteristics from the network structure, such as statistical dependencies. This happens when, *e.g.*, data is generated by flowing through directed paths along the network structure, thus making data at downstream nodes dependent on data at upstream nodes, but independent from each other. Properly modeling these statistical dependencies through Graphical Models, either undirected (*e.g.*, Decomposable Gaussian Graphical Models [7]) or directed (*e.g.*, Directed Gaussian Acyclic Graphical Models [8]), one can benefit from the natural sparsity of the concentration matrix (*i.e.*, the inverse covariance) or its Cholesky factor to estimate the principal subspace with reduced communication costs.

On the contrary, in this work the network topology has no relevance in the desired result, as information is solely carried by the nodes and not by the links. Still, link-related data can be seen as observations relative to one or both of their ends, making methods aimed at node-related data suitable for link-related data.

2.2. Decentralization and asynchrony

Another classification criterion separates decentralized approaches from those which assign node-specific roles and asynchronous approaches from those based on synchronous communications.

In [9], a parallel PCA algorithm is proposed. Sufficient statistics $\mathbf{X}_i\mathbf{1}$ and $\mathbf{X}_i\mathbf{X}_i^{\top}$ are locally computed at all nodes and transmitted to a master node that performs a global Singular Value Decomposition (SVD) to obtain the PCA result. This approach is only suitable when the master node can handle the $O(D^3)$ complexity of the SVD and assumes that $D \times D$ covariance matrices can be exchanged on the network, which is unrealistic in many large scale contexts.

In [4], a distributed PCA algorithm for the DC scenario is proposed. Exchanging only $q \times q$ matrices, nodes iteratively maximize the variance retained by the reduced basis. Even though the process is decentralized, nodes have to update their estimates synchronously before performing any further computation, thus violating (**C2**). The whole system performance is then limited by the slowest networking node. Moreover, synchronous updating is hard to sustain in large networks and can result in overwhelming idle phases even when nodes have identical computing resources.

A fully asynchronous and decentralized Power Iteration method is proposed in [10] using random matrix sparsifications and a nested Sum–Weight Gossip averaging protocol to reduce communication costs. However, its original formulation only provides the first principal component. Synchronous passes would be required to sequentially obtain the next ones.