Applied Numerical Mathematics 125 (2018) 68–85

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Fast computation of stationary joint probability distribution of sparse Markov chains

Weiyang Ding [a,1], Michael Ng [a,*,2], Yimin Wei [b,3]

[a] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong
[b] School of Mathematical Sciences and Shanghai Key Laboratory of Contemporary Applied Mathematics, Fudan University, Shanghai, China

ARTICLE INFO

Article history:
Received 29 December 2016
Received in revised form 26 October 2017
Accepted 27 October 2017
Available online xxxx

Keywords:
Markov chains
Sparsity
Joint distribution
Optimization
Iterative methods

ABSTRACT

In this paper, we study a fast algorithm for finding stationary joint probability distributions of sparse Markov chains or multilinear PageRank vectors which arise from data mining applications. In these applications, the main computational problem is to calculate and store solutions of many unknowns in joint probability distributions of sparse Markov chains. Our idea is to approximate large-scale solutions of such sparse Markov chains by two components: the sparsity component and the rank-one component. Here the non-zero locations in the sparsity component refer to important associations in the joint probability distribution and the rank-one component refers to a background value of the solution. We propose to determine solutions by formulating and solving sparse and rank-one optimization problems via closed form solutions. The convergence of the truncated power method is established. Numerical examples of multilinear PageRank vector calculation and second-order web-linkage analysis are presented to show the efficiency of the proposed method. It is shown that both computation and storage are significantly reduced by comparing with the traditional power method.

1. Introduction

One of the core and important research problems in information retrieval is to evaluate the importance or popularity of objects that can assist to handle many data mining tasks [2,8,10,11,16]. For instance, we employ ranking for the input query in a search engine, finding genes relevant to disease or extracting relevant communities in a social network, see [2,10,15,18,23,25]. For a detailed discussion, see the recent survey paper [8]. The PageRank algorithm [20] developed by Google is well-known and useful to determine the importance of nodes in a directed graph which is used to represent the World Wide Web and its hyperlinks among webpages. For other applications of Markov chains, we refer to [3,7,17,22,28]. Mathematically, it is interesting to note that the PageRank algorithm is used to compute a stationary probability distribution $\mathbf{x} = [x_1, x_2, \cdots, x_n]^\top$ ($^\top$ denotes the transpose of a vector) of a modified first-order Markov chain with n states arising from a graph of n nodes [11,20]:

* Corresponding author.
E-mail address: mng@math.hkbu.edu.hk (M. Ng).

[1] This author's work was partially supported by the Hong Kong Research Grants Council (Grant No. C1007-15G).
[2] Research is supported in part by HKRGC GRF 12302715 and 12306616, and HKBU RC-ICRS/16-17/03.
[3] This author is supported by the National Natural Science Foundation of China under grant 11771099 and International Cooperation Project of Shanghai Municipal Science and Technology Commission under grant 16510711200.

https://doi.org/10.1016/j.apnum.2017.10.008
0168-9274/

$$\mathbf{x} = \beta \mathbf{P}\mathbf{x} + (1-\beta)\mathbf{g}, \quad \text{or} \quad x_i = \beta \sum_{j=1}^{n} p_{i,j} x_j + (1-\beta) g_i, \tag{1}$$

where $\mathbf{P} = [p_{i,j}]$ is the given transition probability matrix (stochastic matrix) arising from the first-order Markov chain (all the entries $p_{i,j}$ are non-negative and each column sum of $\mathbf{P}$ is equal to one), $\mathbf{g} = [g_1, g_2, \cdots, g_n]^\top$ is the given probability vector for the associated nodes [20], and β is a positive number in $(0, 1]$. When $\beta = 1$, Equation (1) is equivalent to solving a stationary probability distribution for a first-order Markov chain. When $\mathbf{P}$ is irreducible, the solution $\mathbf{x}$ is unique and its entries x_i are always positive [1]. For $\beta < 1$, it can be interpreted that a random walker iteratively visits from the current i-th node to the other nodes according to this probability vector: $\beta[p_{i,1}, p_{i,2}, \cdots, p_{i,n}]^\top + (1-\beta)[g_1, g_2, \cdots, g_n]^\top$. The probabilities are related to the transition probability matrix and the given probability vector for the associated nodes. The random walker has the steady state probabilities that will finally stay at different nodes. The corresponding solution $\mathbf{x}$ is also unique, see [10,23].

1.1. Sparse second-order Markov chains

The main objective of this paper is to study efficient solution methods for the computation of stationary joint probabilities of sparse second-order Markov chains/multilinear PageRank:

$$x_{i,j} = \beta \sum_{k=1}^{n} p_{i,j,k} x_{j,k} + (1-\beta) g_{i,j}, \quad i, j = 1, 2, \ldots, n. \tag{2}$$

The quantity $p_{i,j,k}$ represents the probability that the process will make a transition to the i-th node given that currently the process is in the j-th node and the process was in the k-th node previously. Also $g_{i,j}$ is the given probability for the associated i-th and j-th nodes [9,20]. It is interesting to note in many practical applications that there may be many $p_{i,j,k}$ equal to zero. For instance, when we study the citation ranking among researchers in a publication community, second-order transition probabilities can be constructed by using their successive citations [14,19]. As a researcher may not cite the papers written by other researchers in the whole community especially when the number of researchers (n) is very large, we expect many $p_{i,j,k}$ would be zero. Here the i-th researcher cites the k-th researcher via the citation by the j-th researcher. For example, when we study successive links of webpages in World Wide Web, second-order transition probabilities can be constructed that the i-th webpage is linked to the k-th webpage via the j-th webpage. We expect most of these probabilities are zero especially when the number of webpages is very large. The technique for handling dangling states has been considered in [9,14,19], see Section 2.2. Moreover, there are many zeros in $g_{i,j}$ in these applications. For example, we target for some specific query keywords, publications and researchers in communities [21], and we only set a few relevant keywords, publications and researchers in multilinear PageRank applications in which they refer the corresponding non-zeros in $g_{i,j}$.

We note that any second-order Markov chain with n nodes, i.e., $\beta = 1$ in (2), can be reformulated into an n^2-by-n^2 linear system. The block form of the equations can be described as follows:

$$\begin{bmatrix} x_{1,j} \\ x_{2,j} \\ \vdots \\ x_{n,j} \end{bmatrix} = \begin{bmatrix} p_{1,j,1} & p_{1,j,2} & \cdots & p_{1,j,n} \\ p_{2,j,1} & p_{2,j,2} & \cdots & p_{2,j,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,j,1} & p_{n,j,2} & \cdots & p_{n,j,n} \end{bmatrix} \begin{bmatrix} x_{j,1} \\ x_{j,2} \\ \vdots \\ x_{j,n} \end{bmatrix}, \quad j = 1, 2, \cdots, n, \tag{3}$$

or

$$\mathbf{X}(:, j) = \mathbf{P}_j \mathbf{X}(j, :)^\top, \quad j = 1, 2, \cdots, n.$$

We remark that the left hand side vector $\mathbf{X}(:, j)$ is different from $\mathbf{X}(j, :)^\top$ and therefore the linear system is not separable in each block. By adopting Matlab notations, $\mathbf{X}(:, j)$ is the j-th column of $\mathbf{X}$ and $\mathbf{X}(j, :)$ is the j-th row of $\mathbf{X}$. Also $\mathbf{X}(j, :)^\top$ is the transpose of $\mathbf{X}(j, :)$ and $\mathbf{X}(j, :)^\top$ is a column vector. It is clear that the column summation of $\mathbf{P}_j$ is equal to 1, i.e., $\mathbf{P}_j$ is a stochastic matrix.

If the corresponding transition probability matrix in the above equation arising from a second-order Markov chain is irreducible, then the stationary joint probability distribution $\mathbf{X}$ is unique. Similar to (1), we can deal with the case of modified higher-order Markov chains for multilinear PageRank applications. A detailed discussion can be found in [9]. The main computational problem is to calculate and store an n-by-n matrix $\mathbf{X}$. According to (2), $\mathbf{X}$ can be obtained by solving a matrix system where its size is n^2-by-n^2. For example, when $n = 10000$, the matrix size would be of hundred million by hundred million, and the solution would have hundred million unknowns. Indeed, many large-scale applications can be found in web analysis, publication networks, social networks and bioinformatics, see for instance [8,14,15,19]. In these applications, n refers to the number of webpages, the number of authors, the number of users, and the number of genes respectively that can be very large.