Séminaire Mathématique de Béjaïa (LaMOS)
Volume 6, 2008, pp. 33 - 37

7

On the numerical methods in Markov chain modeling

Karim ABBAS

LAboratory of Modelisation and Optimization of Systems LAMOS
University of Béjaïa, Algeria.
E-mail : karabbas2003@yahoo.fr

Résumé In this talk we review the main numerical methods used for numerical solving of Markov chains. Specifically, we are interested in obtaining stationary distributions of homogeneous, irreducible Markov chain. These stationary solutions can be obtained, both for continuous-time and discrete-time Markov chains, from a linear system. We classify traditional methods for solving such systems, arising during modeling with Markov chains.

Keywords : Markov chain ; Stationary distributions ; Linear system ; Numerical methods.

7.1 Introduction

In the context of performance evaluation, numerical analysis methods refer to those methods which work with a Markov chain representation of the system under evaluation and use techniques from the domain of numerical analysis to compute stationary state probabilities or other measures of interest. The use of mathematical models to analyze complex systems has a long history. With the advent of high powered workstations and cheap memory, these applications have greatly expanded. More and more frequently, however, the characteristics of the system to be modeled are such that analytical solutions do not exist or are unknown so that systems engineers turn to computing numerical solutions rather than analytical solutions.

In this talk we would like to make a review of choosen numerical algorithms used for numerical solving of Markov chains.

We are interested in stationary solutions of homogeneous, irreducible Markov chain. Such a chain can be described with an infinitesimal generator matrix Q defined for continuous-time Markov chains (CTMCs) as following :

$$Q = (q_{ij})_{1 \leq i \leq n,\ 1 \leq j \leq n},$$

$$q_{ij} = \lim_{\Delta t \to 0} \frac{p_{ij}(\Delta t)}{\Delta t} \ for\ i \neq j,$$

$$q_{ii} = -\sum_{i \neq j} q_{ij},$$

where $p_{ij}(\Delta t)$ is the probability that if the chain is in the state i it will be in the state j after the time Δt.

For a discrete-time Markov chain (DTMC) we can define the matrix Q as $Q = P - I$, where P is a stochastic matrix of the DTMC and it is defined :

$$P = (p_{ij})_{1 \leq i \leq n,\ 1 \leq j \leq n},$$

where p_{ij} is the probability that if the chain is in the state i it will be in the state j in the next time moment.

Stationary solutions can be obtained (both for CTMCs and for DTMCs) from a linear system :

$$\pi Q = 0,$$

where $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a vector of probabilities of particular states of the Markov chain (so $\pi \geq 0$ and $\sum_{i=1}^{n} \pi_i = 1$) which are to be found.

The same probability vector can be obtained as an eigenvector of the stochastic matrix P from :

$$\pi P = \pi.$$

For CTMCs we can define P as :

$$P = I + \alpha Q, \tag{7.1}$$

where $0 < \alpha < 1/(\max_{i=1,\ldots,n} |q_{ii}|)$.

For a convinient notation we assume $\pi = x^t$ and our problem is to solve :

$$Q^t x = 0, \tag{7.2}$$

(or as an eigenvector problem : $P^t x = x$) with the constraints :

$$x \geq 0,\ e^t x = 1. \tag{7.3}$$

Despite its familiar form the equation is rather spacial. The matrix Q is singular so the equation (7.2) has solutions and it can be proven [Ste94] that - if rank $Q = n - 1$ (it is true in interesting for us cases) – there exists exactly one solution satisfying (7.3). Moreover, the matrix Q is huge (sometimes millions of states or even more), very sparse and illconditioned. We have to chose a suitable algorithm for solving our problem (depending on our aims to achieve – accuracy, time or size).

There are following approaches to solve the equation (7.2) [Ste94] :

- direct methods ;
- iterative methods ;
- projection methods ;
- decompositional methods.

7.2 Direct Methods

Methods which would give us an exact solution in a finite number of steps – if the machine accuracy were infinite – are called direct methods (or sometimes : exact methods). The traits of the direct methods are :

- ✓ constant execution time (or rather a constant number of computation steps for a given matrix size) known in advance ;
- ✓ modification (or rather complete reconstruction) of the given matrix ;
- ✓ the fill-in ;
- ✓ rather good accuracy.

The fill-in is a very troublesome phenomenon. It consists in appearing nonzero elements in the output matrices in places of zero elements in the input matrix. It is very unconvinient if we want to store matrices in a compact manner (i.e. without zero entries) – what is very efficient and indeed necessary for such huge and sparse matrices. In compact storage schemes we must implement some routines to insert new nonzero elements – or provide some space for such entries. However, amount of this space must be estimated in advance what is not a trivial problem. Sometimes we simply have not the space needed for the fill-in.

7.3 Iterative Methods

All the iterative methods have the similar scheme. They start with a starting vector $x^{(0)}$ and then they generate a sequence $(x^{(0)}, x^{(1)}, \ldots)$ which – hopefully – converges to the solution vector x.

The advantages of the iterative methods :

- ✓ they need no modification of the given matrix (so no fill-in is generated and we do not need any additional space for new elements and we spend no additional time on inserting these elements into a complicated storage structure) ;
- ✓ they need very little additional memory ;
- ✓ they are usually faster than direct methods – especially when we do not need very good accuracy offered by the direct methods ;
- ✓ they are easy to implement efficiently and easy to vectorize and to parallelize.

However, the iterative methods have some disadvantages too. We do not know the time needed to achieve required accuracy. Moreover, sometimes we can have even troubles with convergency and we can achieve a solution not satisfying us – especially when the required accuracy is high (what can be an issue in our applications).

7.4 Projection Methods

The projection methods consists in approximating the solution vector with a vector from a small-dimension subspace. Such approximations are repeated until our approximation

is sufficiently close to the solution – in some sense the projection methods are iterative methods.

The projection methods need more space than iterative methods (because they have to store huge basis vectors of subspaces), but can converge faster than classical iterative methods – although the convergence rate is much better for the matrices 'more beautiful' in their structure than the ones arising in solving Markov chains.

7.5 Decompositional Methods

In Markov chain models it is frequently the case that the state space can be meaningfully partitioned into subsets. Perhaps the states of a subset interact only infrequently with the states of other subsets, or perhaps the states possess some property that merits special consideration. In these cases it is possible to partition the matrix Q accordingly and to develop iterative methods based on this partition. In general such decompositional iterative methods require more computation per iteration, but this is offset by a faster rate of convergence.

The IAD - Iterative Aggregation / Disaggregation methods are related to decompositional iterative methods. They are particularly powerful when the Markov chain is NCD - Nearly Completely Decomposable, as the partitions are chosen based on how strongly the states of the Markov chain interact with one another, [Cou77, Mey89]. The choice of good partitions for both decompositional iterative methods and IAD methods is an active area of current research.

7.6 Conclusion

In this talk we classified traditional methods for solving, $Q^t x = 0$, arising during modeling with Markov chains.

Selection of a suitable solution method is by no means easy. The choice depends on many questions :

- △ the matrix structure and its degree of decomposability (e.g. is it NCD ?) ;
- △ the matrix closeness to a suitable structure (and possibility to convert it) ;
- △ the matrix sparseness ;
- △ the matrix size (and our storage possibilities) ;
- △ time to find the solution ;
- △ desired accuracy ;
- △ the matrix conditioning.

Références

[Cou77] P.J. Courtois. *Decomposability.* Academic Press, New York, 1977.

[Mey89] C.D. Meyer. Stochastic complementation, uncoupling Markov chains and the theory of nearly reducible systems. *SIAM Rev.*, 31 :240–272, 1989.

[Ste94] W. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Chichester, West Sussex, 1994.