Ad-hoc Networking – Models and Methods

Holger Hermanns Sven Johr

Universität des Saarlandes

June 15, 2004

Part V

Numerical Solution of Markov Chains

Steady-State and Transient Behaviour

- Steady-State behavior,
  - DTMC: $\underline{\pi} P = \underline{\pi}$,
  - CTMC: .
- Transient behaviour,
  - DTMC: $\underline{\pi}(n) = \underline{\pi}(0) P^n$,
  - CTMC: , can be solved via
    $\underline{\pi}(t) = \underline{\pi}(0) e^{Qt} =$.
-
  - $\underline{r} = (r_1, r_2, \ldots, r_n)$ vector,
  - $\vec{c} = (c_1, c_2, \ldots, c_n)^T$ vector,
  - π probability vector (column or row),
  - $\pi(t)$ probability vector at (column or row).

Use of Direct Methods

- Used to solve systems of linear equations, ,
  - matrix A, dimension n,
  - vectors $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ and $\vec{b} = (b_1, b_2, \ldots, b_n)^T$.
- Computing steady-state $\underline{\pi} M = \underline{0}$, with
  - for CTMC,
  - for DTMC.
- Rephrasing necessary,
  - .

Gaussian Elimination – The Method

- Transfer $A\vec{x} = \vec{b}$ to with,
  - U upper triangular.
- Obtaining U from A called
  - $n-1$ steps,
  - i^{th} step: eliminate elements below i^{th} diagonal element.
- $\vec{x}$ obtained in ,
  - ,
  - $x_i =$, $i = n-1, \ldots, 2, 1$.

Gaussian Elimination – Reduction Phase

- Elements , $i = 1, 2, \ldots, n-1$,
  - $a_{kl}^{(i)} =$ for $k \leq i$, $l = 1, 2, \ldots, n$,
  - $a_{kl}^{(i)} =$ for $k > i$ and $l = 1, 2, \ldots, n$.

    , called .
- Elements element set to 0.
- $a_{ii}^{(i)} \neq 0$.
- May is necessary (stability).
- Remark: $\vec{b}$ has to be accordingly.

Gaussian Elimination – Markov Chains

- Right hand side equals .
- Rank of system is $n-1$ (of matrix is 0),
  - resulting steady-state vector,
  - normalisation equation directly.
- Example: $\mathbf{Q} = \begin{pmatrix} -4 & 2 & 2 \\ 1 & -2 & 1 \\ 6 & 0 & -6 \end{pmatrix}$, $\mathbf{Q}^T = \begin{pmatrix} -4 & 1 & 6 \\ 2 & -2 & 0 \\ 2 & 1 & -6 \end{pmatrix}$

LU Decomposition – The Method

- Write A as ,
  - $A\vec{x} = \vec{b} \Rightarrow$ $= \vec{b}$,
  - L is ,
  - U is .
- Solve $L\vec{z} = \vec{b}$,
  - simple forward substitution.
- Solve $U\vec{x} = \vec{z}$,
  - simple backward substitution.

LU Decomposition – Computation of L and U

- n^2 equations,

$$a_{ij} = \quad , \quad i,j = 1,2,\ldots,n \,. \qquad (1)$$

- Find unkowns,
  - $l_{ik}, i = 1,2,\ldots,n,$,
  - $u_{kj}, j = 1,2,\ldots,n,$.
- Choose n unknowns,
  - $l_{ii} = 1, 1 \leq i \leq n,$,
  - $u_{ii} = 1, 1 \leq i \leq n,$.
- Rewrite Equation (1),
  - $i \leq j$: $a_{ij} = u_{ij} + \sum_{k=1}^{i-1} l_{ik}u_{kj} \Leftrightarrow$,
  - $i > j$: $a_{ij} = l_{ij}u_{jj} + \sum_{k=1}^{j-1} l_{ik}u_{kj} \Leftrightarrow$,
  - and solve iteratively.

LU Decomposition – Comparison to Gauss

- LU decomposition and are intimately related.
- L matrix of used in Gaussian elimination.
- U upper triangular matrix obtained from .
- Benefit,
  - matrix L available, hence
  - solving for more than one quickly possible.

Iterative vs. to Direct Methods

- Result is computed in ,
  - no precise solution available.
- Number of steps depends on ,
  - not a-priori known.
- Efficient can be used,
  - sparce matrices,
  - decision diagramms.
- No (see Example).
- Solution of systems possible;
  - direct methods to systems with a states/equations.

Power Method – The Method

- For DTMC,
  - multiply (steady-state) vector with $\mathbf{P}$ until .
- For CTMC,
  - determine DTMC $\mathbf{P} =$, $\lambda \geq \max_i\{|q_{ii}|\}$.
- $\underline{\pi}^{(i)}\mathbf{P} = \underline{\pi}^{(i+1)}$.
- Used to solve for left eigenvector with eigenvalue 1.
- Not very effcient.
-

Jacobi Method – The Method

- Rewrite i^{th} equation: $p_i = \quad \left(\sum_{j=1}^{i-1} p_j a_{ij} + \sum_{j=i+1}^{n} p_j a_{ij}\right)$.
- Use first estimate $\vec{\pi}^{(0)}$,
  - uniform distribution good choice.
- Next estimate
  - $p_i^{(k+1)} = \frac{1}{|a_{ii}|}\left(\sum_{j=1}^{i-1} p_j^{(k)} a_{ij} + \sum_{j=i+1}^{n} p_j^{(k)} a_{ij}\right)$.
- Stop iterating: (difference criterion),
  - not necessarily solution vector found!
- Check (residual criterion);
  - more expensive, hence
  - use combination of both.
- Slow conversion: use on non-successive iterates.

Gauss-Seidel – The Method

- Structures the Jacobi method,
  - Jacobi method requires storage of two vectors $\vec{\pi}^{(k)}$ and $\vec{\pi}^{(k+1)}$,
  - now, results are used as soon as computed.
- $p_i^{(k+1)} = \frac{1}{|a_{ii}|}\left(\sum_{j=1}^{i-1} p_j^{(k+1)} a_{ij} + \sum_{j=i+1}^{n} p_j^{(k)} a_{ij}\right)$,
  - order of computation is assumed to be from p_1 to p_n.
- Storage of only one probability vector.
- Iteration scheme $D\vec{\pi}^{(k+1)} = L\vec{\pi}^{(k+1)} + U\vec{\pi}^{(k)}$,
  - $\vec{\pi}^{(k+1)} = (D-L)^{-1}U\vec{\pi}^{(k)}$,
  - iteration matrix $\Phi_{GS} = (D-L)^{-1}U$.

Example

- Reconsider Q from above examples.
- $\Phi_P = I + Q/\lambda = \frac{1}{6}\begin{pmatrix} 2 & 1 & 6 \\ 2 & 4 & 1 \\ 2 & 1 & 0 \end{pmatrix}$. 10 iterations.
- $\Phi_J = D^{-1}(L+U) = \begin{pmatrix} 0 & \frac{1}{4} & \frac{3}{2} \\ 1 & 0 & 0 \\ \frac{1}{3} & \frac{1}{6} & 0 \end{pmatrix}$. 8 iterations.
- $\Phi_{GS} = (D-L)^{-1}U = \begin{pmatrix} 0 & \frac{1}{4} & \frac{3}{2} \\ 0 & \frac{1}{4} & \frac{3}{2} \\ 0 & \frac{1}{8} & \frac{3}{4} \end{pmatrix}$. 2 iterations.

The Method

- Solving differential equation system $\underline{\pi}'(t) = \underline{\pi}(t)Q$, numerically.
- Approximate $\underline{\pi}(t)$ by a $\underline{p}_i$, $i \in \mathbb{N}$,
  - use h, hence,
  - $\underline{p}_i = \underline{\pi}(ih)$.
- $\underline{p}_0$ chosen as $\underline{\pi}(0)$.
- The smaller h,
  - result more accurate,
  - and more expensive.
- $\underline{p}_i$ is used to compute $\underline{p}_{i+1}$,
  - but not $\underline{p}_0$ to $\underline{p}_{i-1}$,
  - method.
- Fairly efficient.
- Methods of different order exist.

Main notion

- Solution via expansion
  $\underline{\pi}(t) = \underline{\pi}(0)e^{Qt} =$ for generator matrix Q
  not applicable,
  - infinite summation cannot be ,
  - severe (positive and negative entries of Q).
- Overcome this by .
- Define ,
  - $\lambda \geq \max_i\{|q_{ii}|\}$ called .
- P .
- $\underline{\pi}(t) = \underline{\pi}(0)e^{Qt} =$ $= \underline{\pi}(0)e^{-\lambda t}e^{\lambda P t}$.
- $\underline{\pi}(t) =$.

Benefit

- Still dealing with Taylor series.
- All values are between 0 and 1.
- Allows for iterative solution,
  - $\underline{\pi}(t) = \sum_{n=0}^{\infty} \psi(\lambda t; n)\,(\underline{\pi}(0)P^n) =$,
  - $\underline{\pi}_n$ computed recursively ($\underline{\pi}_0 = \underline{\pi}(0)$,).
- Sum is after steps,
  - k_ϵ is computed a-priori with respect to accuracy ϵ,
  - $||\underline{\pi}(t) - \tilde{\underline{\pi}}(t)|| \leq 1 - \sum_{n=0}^{k_\epsilon} \psi(\lambda t; n)$,
  - $\sum_{n=0}^{k_\epsilon} \frac{(\lambda t)^n}{n!} \geq \frac{1-\epsilon}{e^{\lambda t}} = (1-\epsilon)e^{\lambda t}$.
- For large λt,
  - steady-state reached? Stop computing .

Main Computational Challenges

- Steady-state analysis, ($A\vec{\pi} = \vec{0}$)
  - DTMC: product with until convergence,
  - CTMC: product with until convergence.
- Transient analysis,
  - DTMC: matrix-vector product, ($\underline{\pi}(n) = \underline{\pi}(n-1)P$)
  - CTMC: uniformisation, summation over with .

Uniformisation

- Consider $Q = \begin{pmatrix} -3 & 2 & 0 & 1 \\ 0 & -4 & 1 & 3 \\ 1 & 0 & -1 & 0 \\ 0 & 2 & 0 & -2 \end{pmatrix}$.
- Assume $\underline{\pi}(0) = (1,0,0,0)$.
- Compute the probability distribution for $t = 1$ with uniformisation and answer thereby the following questions.
  - How large is the uniformisation rate λ?
  - How large is k_ϵ for $\epsilon = 10^{-n}$, $n = 1,2,3,4,5$?

European Championship

Germany

vs.

Netherlands

Today, 20:45h
Result will be added.