# Problem Solving in Computer Science: Lecture 10

Scribe notes by Quentin Bonnard

October 21, 2008

## Introduction

The lecture reported in this document was held by Verena Wolf. It started with a quick progress report of each group. Then, Verena exposed several ideas to solve the problem presented in the previous lectures. Finally, she gave an overview of numerical methods for solving linear equation systems, and described more precisely the power method.

## 1 Ideas project 2

- Solve the system for small $m$ with direct methods, and check if the model scales by increasing $m$ (e.g. if patterns can be discovered).
- Try to bring $Q$ in block structure by choosing a "good" enumeration.
- Construct an abstract model that is significantly smaller than the original one, using macrostates (e.g. $([0, 10], [20, 30], ...)$) and a good approximation for the transition rates between these macrostates.
- Restrict the state space to a manageable subset of states (that may change during the computation). Ignore states having very small probability.
- Use iterative methods to solve $\pi Q = 0$; implement a fast vector-matrix multiplication. Choose a "good" initial vector (close to steady state).

## 2 Numerical Methods for Solving Large Linear Equation Systems

### 2.1 Overview

#### Direct methods

- Obtains the solution in a *fixed* number of steps.
- Problem: suffers from matrix fill in.
- May build up rounding errors.

**Iterative methods**

- Start with an initial approximation $x^0$ of the solution and generate $x^1$, $x^2$, ... with hopefully increasing accuracy.
- Iteration can be stopped when accuracy is high enough
- Preserve sparseness of $A$. $A$ is not needed explicitely, only the product $Ax^i$.
- Examples of basic iterative methods: Jacobi, Gauss-Seidel, SOR (successive over-relaxation), power method... See [1] for more details.
- Projection methods: compute approximation in a space of smaller dimension and expand the space until the decided accuracy is reached. See [2] for more details.

## 2.2 Power Method

- Let $Q$ be a generator matrix of an ergodic Markov chain.
- Define $P = I + \frac{1}{q}Q$, where $q \geq \max_{i \in S}(-q_{ii})$. For example, the generator matrix $Q$ corresponding to the graph shown in Figure 1 is

$$Q = \begin{pmatrix} -3 & 3 & 0 \\ 2 & -3 & 1 \\ 4 & 5 & -9 \end{pmatrix}.$$

Then with $q = 9$,

$$P = \begin{pmatrix} 6/9 & 3/9 & 0 \\ 2/9 & 6/9 & 1/9 \\ 4/9 & 5/9 & 0 \end{pmatrix}.$$

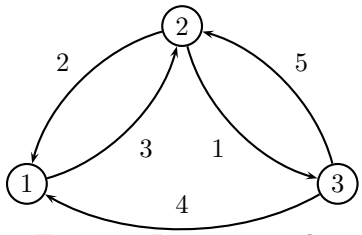

Figure 1: Intensitygraph.

- Then $P$ is the transition matrix of a discrete time Markov chain $\tilde{\mathrm{X}}$.
- $\tilde{\mathrm{X}}$ and $X$ have the same limiting distribution, and $X$ is ergodic iff $\tilde{\mathrm{X}}$ is.
- Let $\pi^*$ be the steady state of $X$ (and $\tilde{\mathrm{X}}$), then $\pi^* = \pi^* P$ and

$$\lim_{k \to \infty} \pi(0) P^k = \pi^*$$

for all initial distributions, because $X$ is ergodic.

| $\lvert\lambda_2\rvert$ | 0.1 | 0.5 | 0.6 | ... | 0.9 | 0.95 |
|---|---|---|---|---|---|---|
| $k$ | 6 | 20 | 27 | ... | 131 | 269 |

Table 1: Examples of number of steps $k$ needed to converge given the subdominant eigenvalue $\lambda_2$ with a fixed accuracy $\epsilon = 10^{-6}$

- Iteration
$$x^0 = \pi(0)$$
$$x^{k+1} = x^k P$$
- also used for the computation of the left-hand eigenvector (for eigenvalue 1)
- The rate of convergence depends on subdominant eigenvalue[1] $\lambda_2$ of P (the smaller the better) :
$$O(|\lambda_2|^k) = ||x^k - \pi^*||$$
and thus, for accuracy $\epsilon > 0$,
$$k = O\left(\frac{\log \epsilon}{\log |\lambda_2|}\right)$$
See Table 1 for examples of convergence speed.

# Conclusion

Because of the time spent in various discussion during the lecture, others iterative methods could not be presented. This is planned for the next lecture.

# References

[1] William J. Stewart. Numerical Methods for Computing Stationary Distributions of Finite Irreducible Markov Chains.

[2] Yousef Saad. Iterative methods for sparse linear systems (1st edition)

[1] Note that computing the eigenvalue can be a problem, except if the matrix is a block matrix.