Solving SDD linear systems in nearly-linear time

Maguy Trefois Jean-Charles Delvenne Paul Van Dooren

Dept. of Applied Mathematics
Université catholique de Louvain
Avenue G. Lemaître, 4-6
1348 Louvain-La-Neuve
Belgium
Email: maguy.trefois, jean-charles.delvenne, paul.vandooren@uclouvain.be

1 Introduction

Symmetric and diagonally-dominant (SDD) linear systems appear in many applications of computer science. These systems are usually of important size and solving them is the main computational task. Direct methods for solving linear systems are much too slow in the case of huge systems. In 2013, Kelner *et al.* [1] proposed a new approach for solving SDD systems in time which is nearly-linear in the number of nonzero entries. The original paper [1] presents this method using a physical interpretation of the problem. From this approach, it is unclear how this method works in a matrix theoretic point of view. In this talk, we explain Kelner's algorithm using only matrix theoretic arguments.

2 Problem Formulation

A particular case of linear system is when the coefficient matrix is the Laplacian matrix of an undirected graph with positive weights along the edges. The Laplacian matrix of such a graph is defined as

$$L = D - A,$$

where D is the diagonal matrix of the vertex degrees and A is the weighted adjacency matrix. These particular systems are called Laplacian systems and are SDD linear systems. They are of particular interest since it has been shown that if one can solve any Laplacian system in nearly-linear time, then one can solve any SDD system in nearly-linear time as well.

Consequently, we consider a Laplacian system

$$Lv = b.$$

This system is supposed to have an exact solution v_{opt}.

The goal is finding an ε-approximate solution v_K, namely v_K satisfies

$$||v_K - v_{opt}||_L \leq \varepsilon \cdot ||v_{opt}||_L,$$

in nearly-linear time, i.e. in time

$$\mathcal{O}(m \log^c n),$$

where n is the number of vertices (or the size of the system) and m is the number of edges (or the number of nonzero entries in L).

In this talk, we explain the Kelner algorithm through matrix theoretic arguments.

3 The Kelner algorithm

The method is based on the usual factorization of the Laplacian matrix through an incidence matrix $B \in \mathbb{R}^{m \times n}$:

$$L = B^T R^{-1} B,$$

where $R^{-1} \in \mathbb{R}^{m \times m}$ is the diagonal matrix of the weights along the edges.

We use the variable change $f = R^{-1}Bv$. The method is made up of the two following steps:

1. Solve the Laplacian system in variable f, namely solve
$$B^T f = b$$
and compute an ε-approximation f_K of the minimal R-norm solution f_{opt} to $B^T f = b$.
In order to find f_K, a coordinate descent method is used.
2. Given f_K, use the variable change $f_K = R^{-1}Bv$ to find an ε-approximate solution to $Lv = b$.

In our talk, we explain the Kelner method in details and we show how to perform the crucial steps in order to get a running time in

$$\mathcal{O}(m \log^2 n).$$

References

[1] J.A. Kelner, L. Orecchia, A. Sidford, Z. Allen Zhu, A simple, combinatorial algorithm for solving SDD systems in nearly-linear time, in Proceedings of the 45th annual ACM Symposium on Theory Of Computing (STOC), pp. 911-920, New York, NY, USA, 2013.