Progress Report
21-30 June 2011

Raj K. Kumar, Amir Hesam Salavati
E-mail: raj.kumar@epfl.ch, hesam.salavati@epfl.ch

Supervisor: Prof. Amin Shokrollahi
E-mail: amin.shokrollahi@epfl.ch

Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)

September 20, 2011

1 Summary

In the neural error correcting algorithm, so far we had assumed that the constraint matrix was given and all that is required is a message paasing algorithm to correct the errors. Since we have found such an algorithm, the next step would be to find a way to determine the constraint matrix from a given data set that satisfy some constraints. This problem is closely related to fiding the null basis of a linear code (the parity check matrix). However, not any such matrix is within our field of interest since we look for a sparse and expander parity check matrix. To find a method to help us obtain such matrices, we have started reading some papers on this topic the datils of which are given below. Furthermore, we gave a talk on our work (entitled "Exponential Pattern Retrieval Capacity with Non-Binary Associative Memory") as an ALGO Seminar on June 29.

2 Null Space Problem

In [1], the authors propose an algorithm to find the a sparse null space to a given adjacency matrix. The algorithm uses complete matching to first identify non-zero positions in the null vectors of the matrix and then to solve a linear system of equations to find actual values of those non-zero elements. The authors have proposed two algorithms, one that computes a null space with an identity matrix as its part and the other one with a triangular matrix in it.

More formally, the problem of interest is to find a basis N for the null space of a $t \times n$ matrix A with rank t that *has the fewest non-zero elements*. In other words, given A, the problem is to find a sparase matrix N such that $AN = 0$.

A null vector for A is a vector that $Ax = 0$. It corresponds to a *dependent set*, i.e. a set of columns in A that are linearly dependent. A minimal dependent set in A is called a *circuit*. The authors propose the following circuit finding algorithm which will be used later in order to solve the null space problem:

Circuit finding algorithm: If A is an adjacency matrix with Weak Hall Property (WHP) and a complete matching, then if u is an unmatched column-node (the nodes whose connectivity are represented by columns of A) in A, by following all alternating paths, starting from u and ending in another column-node, after storing the end-point vertices one will have a dependent set which is minimal, i.e. we will have a circuit. This set is identified by the columns in A that the column-nodes in the dependent set represent. The corresponding dependent set is denoted by $n(u)$.

The reverse of the above algorithm is correct as well: *Every* circuit of a matrix A with WHP can be constructed by the circuit algorithm from some maximum matching of A. Using the above algorithm, **the fundamental null basis algorithm** is given below:

1. Let N be the empty set.
2. Find a complete matching ofA;
3. Partition the columns: A = [M—U], where M is a non-singular matrix. rows and columns in M are matched together and represent the complete matching above.
4. *For each* $u \in U$:
   - Construct $n(u)$ by the above circuit algorithm.

- Find the coefficients of the corresponding null vector for $n(u)$, i.e. the vector x for which $n(u).x = 0$.
- Add the obtained vector found above to N.

The authors have also proposed a similar algorithm that computes a basis with a triangular sub-matrix in it instead of the identity matrix as in the *fundamental null space*.

3 Conclusions and Future Works

One idea that was used by the authors to compute a matching in the fundamental null-space algorithm is to assign costs to each column, where the cost is the number of non-zero elements in that column. Then, for each row, one can match the corresponding row-column to the lowest cost *unmatched* column-node. In this way, we ensure to have a sparse null-sparse when we implement the fundamental null space algorithm on this matching. Such an idea might be useful in our algorithm, i.e. finding an expander parity check matrix of a linear code given its generator matrix.

References

[1] T. F. Coleman, A. Pothen, "The null space problem II. Algorithms", SIAM. J. on Algebraic and Discrete Methods, Vol. 8, No. 4, 1987, pp. 544-563.

[2] T. F. Coleman, A. Pothen, "The null space problem I: complexity", SIAM. J. on Algebraic and Discrete Methods, Vol. 7, No. 4, 1986, pp. 527-537.