NUMERICAL ANALYSIS TECHNIQUES TO SOLVE THE SUDOKU

Nehar Poddar

Numerical Analysis

Mini Project 1

THE SUDOKU PUZZLE

	10	11	12	13	14	15	16	17	18
1				4		9	6	7	
2					7	6	9		
3									3
4						1	7	4	
5	6	4						1	8
6		2	1	6					
7	1								
8			4	3	2				
9		6	2	9		4			

- The Sudoku puzzle is a discrete constraint satisfaction problem, where the constraints address uniqueness of the subsets of the puzzle, based on some observed "clues".
- There are several ways to solve the problem one of them being Back Propagation which includes logical elimination according to the distinct rules.
- The other way used is a more probabilistic approach that allows searching for potential solutions.
- Sinkhorn balancing means obtaining a unique doubly stochastic matrix from a arbitrary matrix.
- A doubly stochastic matrix is a matrix with non-negative elements whose rows each sum upto 1 and columns each sum upto 1.

SINKHORN – BALANCING ALGORITHM

If A is an n x n matrix with strictly positive elements, then there exist diagonal matrices D_1 and D_2 with strictly positive diagonal elements such that D_1AD_2 is doubly stochastic.

The matrices D_1 and D_2 are unique modulo multiplying the first matrix by a positive number and dividing the second one by the same number.

A simple iterative method to approach the double stochastic matrix is to alternately rescale all rows and all columns of A to sum to 1.

Sinkhorn and Knopp presented this algorithm and analyzed its convergence.

Max number of iterations M
Tolerance ϵ

Repeat :

For $j = 1 : N$ (Vertical : sum and scale columns)

$X_j = \sum_l q_{lj}^{[k]}$

$q_{ij} = q_{ij}^{[k]} / X_j \quad (i = 1,2,\ldots,N)$

End For j

For $i = 1 : N$ (Horizontal : sum and scale columns)

$X_i = \sum_l q_{il}^{[k]}$

$q_{ij}^{[k+1]} = q_{ij} / X_i \quad (j = 1,2,\ldots,N)$

End For i

If $||Q^{[k+1]} - Q^{[k]}|| < \epsilon$
Set $Q = Q^{[k+1]}$
Break

$k \leftarrow k + 1$
If $k > M$
$Q = Q^{[k]}$
Break

End Repeat

Return Q

PUZZLE DESCRIPTION AND REPRESENTATION

	10	11	12	13	14	15	16	17	18
1				4		9	6	7	
2					7	6	9		
3									3
4						1	7	4	
5	6	4						1	8
6		2	1	6					
7	1								
8			4	3	2				
9		6	2	9		4			

A Sudoku puzzle is an $N \; X \; N$ grid of cells partitioned into N smaller blocks of N elements each.

The puzzle problem is to fill in the cells so that the digits $1, \ldots\ldots, N$ appear uniquely in each row and column of the grid and in each block, starting from some initial set of filled-in cells called “clues.”

The uniqueness requirement imposes $3N$ constraints on the puzzle.

We denote the contents of cell n by $S_n \in \{1,2, \ldots\ldots, N\}$ for $n = 1,2, \ldots\ldots, N^2$, with cells numbered in row-scan order.

The row constraints are indexed by the numbers $1, \ldots\ldots, N$ down the side of the puzzle.

The column constraints are indexed by the numbers $N + 1, \ldots\ldots, 2N$ across the top of the puzzle.

Constraint m of the puzzle is satisfied if all N cells associated with it are distinct.

We model the contents of the cells probabilistically.

Let $(p_n = [P(S_n = 1) \; P(S_n = 2) \ldots\ldots\ldots P(S_n = 9) \;])$ be the probability row vector associated with cell S_n, with individual elements $p_{n,j}$.

Cells that are specified initially—the clue cells—place all their probability mass on the specified value.

Thus, for the puzzle:

$$p_4 = e_4 \qquad p_6 = e_9 \qquad p_7 = e_6 \;\; etc.$$

Where e_k is a vector of length N with a single 1 at position k and 0 in other positions.

For the non-clue cells (initially empty), the probabilities are uniformly distributed over the possible outcomes. For example,

$$p_1 = \frac{1}{4}[0\;1\;1\;0\;1\;0\;0\;1\;0].$$

SINKHORN SUDOKU ALGORITHM

Initialization : Set initial probability vectors $p_1, p_2, \ldots, p_{N^2}$ according to initial clues and uniformly in cells with no clues.

Set $k = 0$

Repeat:

For each constraint $m \in \{1,2,3,....,3N\}$:

Form the probability constraint matrix $Q_m^{[k]}$

Sinkhorn balance : $Q_m^{[k+1]} = SB(Q_m^{[k]})$

Extract the probabilities p_n from $Q_m^{[k+1]}$

End for m

Determine most probable contents S_n from p_n :

$$S_n = \arg\max p_{n,j}$$

If all constraints are satisfied, Break with success

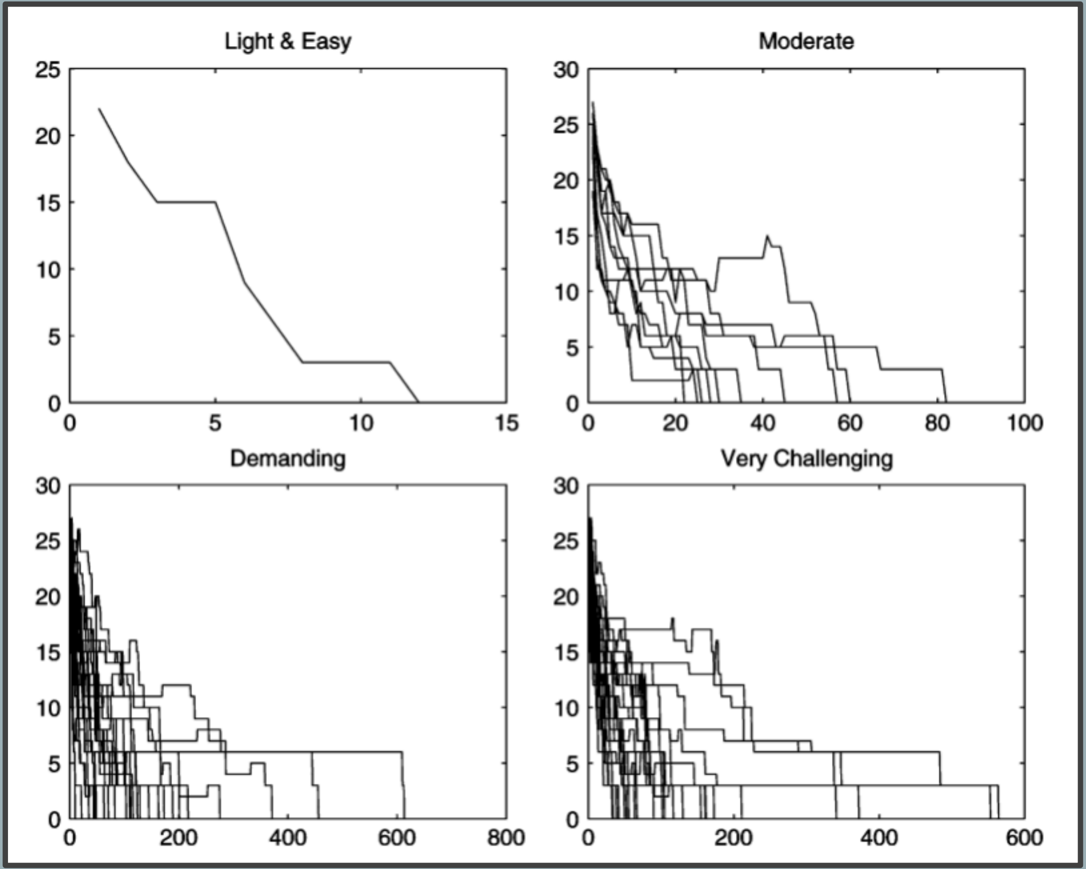
Increment iteration count : $k \to k + 1$

If too many iterations, Break with failure

End Repeat

RESULTS

Convergence of Sinkhorn Balancing - Sudoku



The horizontal axes show the number of iterations.
The vertical axes show the number of unsatisfied constraints

- The Sinkhorn balancing has lower computational complexity (per iteration).
- It is more robust and can solve more number of problems
- It does not suffer from cycles in the Tanner graph (all cells are in cycles of length four) as the back propagation method.
- The Sinkhorn balanced matrix is said to be closer in Kullback Leibler (KL) distance to the constraint probability matrix for a solved puzzle.

Comparison Table

	A	B	C	D
Total number of puzzles	10	15	20	25
No. Sinkhorn Solved	10	15	19	23
No. of iterations	37	52	153	180
No. Back Propogation Solved	4	7	11	13
No. of iterations	6	8	7	9

Unsolved Example

CORNER CASES

- The sequence of matrices converges to a doubly stochastic limit if and only if the matrix 'A' contain at least one positive diagonal.
- A necessary and sufficient condition that there exist diagonal matrices D_1 and D_2 with positive main diagonals such that D_1AD_2 is both doubly stochastic and the limit of the iteration is that $A \neq 0$ and each positive entry of A is contained in a positive diagonal.
- The diagonal matrices and the solution are all unique.
- The form D_1AD_2 is unique, and D_1 and D_2 are unique up to positive scalar multiple if and only if A is fully indecomposable.
- For a lot of cases of very easy (more filled elements) sudoku puzzles, I often got the 'nan' error in the middle of the computation. Generally it works better for sparce matrices.

APPLICATIONS

- Sinkhorn balancing also known as Sinkhorn scaling has been widely studied and makes an appearance in a variety of applications.
- Not surprisingly, the simplicity of the method has led to its repeated discovery.
- It is claimed to have first been used in the 1930's for calculating traffic flow.
- It appeared in 1937 as a method for predicting telephone traffic distribution
- .In the numerical analysis community it is most usually named after Sinkhorn andKnopp, who proved convergence results for the method in the 1960's, but it is also known by many other names, such as the RAS method and Bregman's balancing method.
- It is compared with some well known alternatives, including PageRank.
- It is shown that with an appropriate modifications, the Sinkhorn-Knopp algorithm is a natural candidate for computing the measure on enormous data sets.

REFERENCES

- T. Davis, "The mathematics of Sudoku," [Online]. Available: http:// www.geometer.org/mathcircles/sudoku.pdf T. K. Moon and J. H. Gunther, "Multiple constraint satisfaction by belief propagation: An example using Sudoku," in *Proc. SMCals/IEEE Mountain Workshop Adaptive Learn. Syst.*, Jul. 2006, pp. 122–126.
- S. Kirkpatrick, "Message-passing and Sudoku," [Online]. Available: http://www.www.cs.huji.ac.il/~kirk/Sudoku.ppt R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Ann. Math. Statist.*, vol. 35, pp. 876–879, Jun. 1964.
- R. Sinkhorn, "Diagonal equivalence to matrices with prescribed row and column sums," *Amer. Math. Monthly*, vol. 35, pp. 876–879, 1967. H. Balakrishnan, I. Hwang, and C. J. Tomlin, "Polynomial approxi- mation algorithms for belief matrix maintenance in identity manage- ment," in *Proc. 43rd IEEE Conf. Decision Control*, Dec. 14–17, 2004, pp. 4874–4979.
- R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.
  W. S. ed. , *Pocket Sudoku*. New York: St. Martins Paperbacks, 2005. M. Huckvale, *The Big Book of Sudoku # 2*. New York: New Market Press, 2005.