CSI 4103

Great Algorithms

Study Guide

Fall 2024

Gaussian Elimination

- Used to solve systems of linear equations
- Can be applied to any system with the same number of equations and unknowns
- Here's how it works

Algorithm GE

Input: A system S of n linear equations with n unknowns

1 if $n=1$ and the equation is $ax=b$, output b/a

2 Otherwise, pick a variable S and a equation e in S in which x has nonzero coefficient

3 Create a new system S′ of n-1 equations and n-1 unknowns

4 For every equation $e' \in S$ other than e

5 subtract enough copies of e from e′ to eliminate x and

6 include the resulting equation in S′

7 Apply GE to S′ to obtain an assignment to all variables except for x

8 Plug the partial assignment into e to recover x

9 Output the completed assignment

How to Analyze Algorithms

We ask questions like

- Why does it work? Does it ever fail?
- How efficient? What makes it attractive?
- How robust?
- Why is it useful?
- Can we apply the algorithm? Look for nails to hammer in

Problems vs Algorithms

- A computational problem is a relation R on instance-solution pairs
- An algorithm is a procedure that takes an instance x and produces solution(s) y such that $(x,y) \in R$
- Types of algorithms:
  - Search: output any solution y as long as $(x,y) \in R$
  - Decision: only need to answer if such solutions exist (yes/no)
  - Enumeration: output all possible solutions

In the case for Gaussian Elimination, an instance-solution pair is (S, sol), S is a system of linear equations and sol is an assignment

Correctness of Gaussian Elimination

- Algorithm GE has its issues, but we do expect it to work on instances with a unique solution
- The procedure of adding or subtracting multiples of one row from another is called an elementary row operation. They are invertible

Claim: Let S be a system of linear equations and e be an equation in S. Let S′ be obtained from S by adding a multiple of e to some other equation e′. Then S′ has the same solutions as S

Solution for S $\rightleftarrows$ Solution for S′

The reason for this is because being a solution is closed under taking linear combinations of equations

Proposition: Assuming S has a unique solution, then GE finds it

We can also tackle the problem where an equation reduces to $0=0$. We add the line: if S has the form $0=0$ but there are still unused variables, set all of them free. This should be added before line 2

There is also the problem of having a contradiction in the form $0=b$ for some nonzero b. Then insert the line: if S has the form $0=b$ for nonzero b, output no solution

Gauss-Jordan Elimination

Algorithm GJE

Input: A system S of m linear equations with n unknowns

1 If S has the form $0=0$ or is empty, output the assignment in which all variables are free
  If S has the form $0=b$ for $b \neq 0$, output no solution

2 Otherwise, pick a variable x and an equation e in S in which x has nonzero coefficient

3 Create a new system S′ of m-1 equations and n-1 unknowns

4 For every equation $e' \in S$ other than e

5 Subtract enough copies of e from e′ to eliminate x and

6 include the resulting equation in S′

7 Apply GE to S′ to obtain an assignment to all variables except for x

8 Plug this partial assignment into e to recover x in terms of the free variables

9 Output the completed assignment

Number of equations and number of unknowns no longer have to be the same

Theorem: GJE outputs a description of all the solutions of S if there are any, and no solutions if there aren't

Efficiency and Limitations of GE

Suppose a system of equations has m equations and n unknowns

Worst case: m-1 elementary row operations, then runs recursively on an instance with m-1 equations and n-1 unknowns, then completes assignment. There are $\Theta(mn)$ operations for step 4-6 and $\Theta(n^2)$ operations for step 8. The number of operations is then

$$C(m,n) = C(m-1,n-1) + \Theta(mn+n^2)$$

If $m=n$, we get

$$C(n) = C(n-1) + \Theta(n^2) = \Theta(n^3)$$

In general, it takes $mn+m$ numbers to describe a system with m equations so the worst case complexity is at most $\Theta((\text{input size})^{1.5})$

A linear system is sparse if the number of variables that participates in a typical equation is much smaller than n

For such sparse systems, the worst-case running time of GE is cubic in the size of the instance

Some Features of Gaussian Elimination

- Gaussian Elimination allows for in-place implementation. The space used to describe the instance can be reused to describe all intermediate states and the solution
- Can be implemented with perfect precision. If the input coefficients are provided as rational numbers and the additions and divisions in the elementary row operations are implemented without loss of accuracy, the algorithm will output an exact solution

Unsatisfiability of Gaussian Elimination

If we run Gaussian Elimination on a system and it outputs "no solution", how do we know there's indeed no solution?

A system cannot have a solution by coming up with some linear combination of the equations that makes the left-hand side vanish but not the right hand side. Such a linear combination is called a contradictory linear combination

Theorem: A system of equations has no solutions if and only if there exists a contradictory linear combination of the equations

If we use matrix notation and our system is $A\vec{x}=\vec{b}$, then if $\vec{y}^T$ is a row vector, we get that $A\vec{x}=\vec{b}$ has no solution if and only if $\vec{y}^T A = \vec{0}, \vec{y}^T\vec{b}=1$ has a solution

Learning Linear Functions

We can use data to train linear functions.

For example, for CSI4103 we want to know the grade weighting

h = homework

m = midterm

p = project

f = final grade

We have $f = hx_1 + mx_2 + px_3$ and we can solve for h, m, and p using past grade data and GE

However, the data is often noisy and rounding can significantly impact the accuracy of GE

Modular Gaussian Elimination

Problem: We have inputs $x_1, \dots, x_n$ that can be +1 or −1, then we have $f(x)$ is the product of a subset of the input. We want to find the subset.

We can use data to learn the subset.

This can be done by replacing −1 with 1 and 1 with 0 mod 2. Then GE can be used to solve the system and the subset can be found