Gaussian Elimination

Geometric Algorithms
Lecture 3

CAS CS 132

Objectives

1. Motivation
2. Define the Gaussian Elimination (GE) algorithm
3. Analyze the GE algorithm

Keywords

echelon form

reduced echelon form

basic variables

free variables

Gaussian elimination

FLOPS

Motivation

Recall: Solving Systems of Linear Eqs.

Observation 1. Solutions look like simple systems of linear equations

said another way: it's easy to read off the solutions of some systems

Solving a system of linear equations is the same as row reducing its augmented matrix to a matrix which represents a solution.

What matrices represent solutions?

Recall: Number of Solutions

zero the system is inconsistent

one the system has a unique solution

many the system has infinity solutions

How does the number of solutions affect matrices representing solutions?

Recall: Elementary Row Operations

scaling	multiply a row by a number
interchange	switch two rows
replacement	add two rows (and replace one with the sum)
rep. + scl.	add a scaled equation to another

How do we use these operations to get to matrices representing solutions?

Motivating Questions

Let's consider these first

What matrices represent solutions? (which have solutions that are easy to read off?)

How does the number of solutions affect the shape of these matrix?

How do we use row operations to get to those matrices?

Unique Solution Case

Unique Solution Case

$$\begin{bmatrix} 2 & -3 & 5 & 11 \\ 2 & -1 & 13 & 39 \\ 1 & -1 & 5 & 14 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

$$x = 1$$
$$y = 2$$
$$z = 3$$

Nearly all the examples we've seen so far

The Identity Matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1s along the diagonal

0s elsewhere

Unique Solution Case

coefficient matrix

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right]$$

a system of linear equations whose **coefficient matrix** is the identity matrix represent a unique solution

No Solution Case

No Solution Case

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 2 & 3 & 4 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

two parallel planes

row representing $0 = 1$

No Solution Case

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

row representing $0 = 1$

a system with no solutions can be reduced to a matrix with the row

$$0\ 0\ \ldots\ 0\ 1$$

Infinite Solution Case

Infinite Solution Case

$$\begin{bmatrix} 2 & 4 & 2 & 14 \\ 1 & 7 & 1 & 12 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} x_1 + x_3 &= 2 \\ x_2 &= 1 \end{aligned}$$

a system with infinity solutions can be reduced to a system which leaves a variable <u>unrestricted</u>

Infinite Solution Case

$$x_1 + x_3 = 2$$
$$x_2 = 1$$

it doesn't matter what x_3 is if we want to satisfy this system of equations

$$x_1 = 2$$
$$x_2 = 1$$
$$x_3 = 0$$

Infinite Solution Case

$$x_1 + x_3 = 2$$
$$x_2 = 1$$

it doesn't matter what x_3 is if we want to satisfy this system of equations

$$x_1 = 1.5$$
$$x_2 = 1$$
$$x_3 = 0.5$$

Infinite Solution Case

$$x_1 + x_3 = 2$$

$$x_2 = 1$$

it doesn't matter what x_3 is if we want to satisfy this system of equations

$$x_1 = 20$$

$$x_2 = 1$$

$$x_3 = -18$$

Infinite Solution Case

$$x_1 + x_3 = 2$$
$$x_2 = 1$$

it doesn't matter what x_3 is if we want to satisfy this system of equations

$x_1 = 2 - x_3$
$x_2 = 1$
x_3 is free

general form

In Sum

none	reduces to a system with the equation $0 = 1$
one	reduces to a system whose coefficient matrix is the identity matrix
infinity	reduces to a system which leaves a variable unrestricted

Ideally, we want one *form* that handles all three cases

Motivating Questions

What matrices represent solutions? (which have solutions that are easy to read off?)

How does the number of solutions affect the shape of these matrix?

How do we use row operations to get to those matrices?

this is Gaussian elimination

Defining the Gaussian Elimination (GE) Algorithm

At a High Level

eliminations + back-substitution

we've already done this

but we'll take one step further and write down the algorithm as pseudocode

Keep in mind. How do we turn our intuitions into a formal procedure?

Defining the GE Algorithm (Outline)

1. `echelon forms`
2. `elimination phase`
3. `substitution phase`

Echelon Form

Leading Entries

Definition. the ***leading entry*** of a row is the first nonzero value

$$\begin{bmatrix} \boxed{1} & 2 & 3 \\ 0 & \boxed{-3} & 3 \\ 0 & 0 & 0 \\ \boxed{1} & -1 & 10 \end{bmatrix} \longleftarrow \text{no leading entry}$$

Echelon Form

Definition. A matrix is in ***echelon form*** if

1. The leading entry of each row appears to the right of the leading entry above it
2. Every all-zeros row appears below any non-zero rows

Echelon Form (Pictorially)

next leading entry
to the right

$$
\begin{bmatrix}
0 & \blacksquare & * & * & * & * & * & * & * & * \\
0 & 0 & 0 & \blacksquare & * & * & * & * & * & * \\
0 & 0 & 0 & 0 & \blacksquare & * & * & * & * & * \\
0 & 0 & 0 & 0 & 0 & \blacksquare & * & * & * & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \blacksquare & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

all-zero rows at
the bottom

$\blacksquare$ = nonzero, $*$ = anything

Why we care about Echelon Forms?

echelon forms aren't quite solutions, but their close

the goal of elimination is to reduce an augmented matrix to an echelon form

(more reasons we care in a moment)

Question

Is the identity matrix in echelon form?

Answer: Yes

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

the leading entries of each row appears to the right of the leading entry above it

it has no all–zero rows

Question

Is this matrix in echelon form?

$$\begin{bmatrix} 2 & 3 & -8 \\ 0 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

Answer: No

$$\begin{bmatrix} 2 & 3 & -8 \\ 0 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

The leading entry of the least row is not to the right of the leading entry of the second row

The Problem with Echelon Forms

1. they're not unique (uniqueness makes it easier to define an algorithm)
2. we can't read off the complete solution from an echelon form

Reduced Echelon Form

Reduced Echelon Form

Definition. A matrix is in ***reduced* *echelon form*** if

1. The leading entry of each row appears to the right of the leading entry above it
2. Every all-zeros row appears below any non-zero rows
3. The leading entries of non-zero rows are 1
4. the leading entries are the only non-zero entries of their columns

Reduced Echelon Form (Pictorially)

leading entries are 1

$$\begin{bmatrix} 0 & 1 & * & 0 & 0 & 0 & * & * & 0 & * \\ 0 & 0 & 0 & 1 & 0 & 0 & * & * & 0 & * \\ 0 & 0 & 0 & 0 & 1 & 0 & * & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 1 & * & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

other column
entries are 0

Reduced Echelon Form (A Simple Example)

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Reduced Echelon Form (A Simple Example)

$$x_1 + x_3 = 2$$
$$x_2 = 1$$

$x_1 = 2 - x_3$

$x_2 = 1$

x_3 `is free`

The Fundamental Point

Theorem. every matrix is row equivalent to a <u>unique</u> matrix in reduced echelon form

Definition. a ***pivot position*** (i, j) in a matrix is the position of a leading entry in it's reduced echelon form

we can read off the solutions of a system of linear equations by looking at its pivot positions

Basic and Free Variables

Definition. A variable is ***basic*** if its column has a pivot position (this is called a ***pivot row***). It is ***free*** otherwise.

x_1 is basic

x_2 is basic

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

x_3 is free

Solutions of Reduced Echelon Forms

the row of a pivot position in row i describes the value of x_i in a solution to the system, in terms of the free variables

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$x_1 = 2 - x_3$

$x_2 = 1$

x_3 is free

General Form Solution

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$x_1 = 2 - x_3$

$x_2 = 1$

x_3 is free

for each pivot position (i, j), isolate x_i in the equation in row j

if x_i does not have a pivot position, write

x_i is free

Inconsistent Echelon Forms

Corollary. A matrix represents an inconsistent system if its echelon form has a row of the form

just echelon

$$0\ 0\ 0\ \ldots\ 0\ 0\ 1$$

if it didn't, we could read off a solution

Why we care about Reduced Echelon Forms?

the goal of <u>back-substitution</u> is to reduce an echelon form matrix to a ***reduced*** *echelon form*

the goal of <u>Gaussian elimination</u> is to reduce an ***augmented*** *matrix to a* ***reduced*** *echelon form*

echelon forms describe solutions to linear equations

Question

write down a solution in general form for this reduced echelon form matrix

$$\begin{bmatrix} 1 & 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Answer

$$\begin{bmatrix} 1 & 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$x_1 = 1 - 3x_4$

x_2 is free

$x_3 = 4 - 2x_4$

x_4 is free

The Algorithm

Gaussian Elimination (Specification)

Input: (augmented) matrix A of size $m \times (n+1)$

Output: reduced echelon form of A

Notation:

$A[i]$ = ith row of A

$A[i,j]$ = entry in the the ith row and jth column

Gaussian Elimination (High Level)

```
Given A:
  convert A to an echelon form A′
  if A′ is consistent:
    convert A′ to reduced echelon form
```

Gaussian Elimination (Pseudocode)

```
FUNCTION GE(A):
  GE_elim_stage(A)
  IF is_consistent_echelon(A):
    GE_back_sub_stage(A)
```

Elimination Stage

Elimination Stage (High Level)

Input: (augmented) matrix A of size $m \times (n+1)$

Output: echelon form of A

starting at the top left and move down, find a leading entry and eliminate it from latter equations

Note. this may require interchanging rows

Elimination (Pseudocode)

```
FUNCTION GE_elimination_stage(A):
  FOR i from 1 to m:     # for all rows from top to bottom
    IF rows i...m are all-zeros then STOP
    (j,k) ← position of leftmost nonzero entry in rows i...m of A
    swap rows A[i] and A[j]     # make sure row i has the pivot
    apply row operations to zero out all entries below (i,k) in A
    IF A has an inconsistent row then STOP
```

Elimination Stage (Example)

$$\begin{array}{c} \\ \\ \text{leftmost nonzero entry} \end{array} \begin{bmatrix} 0 & 3 & -6 & 6 & 4 & -5 \\ 3 & -7 & 8 & -5 & 8 & 9 \\ \boxed{3} & -9 & 12 & -9 & 6 & 15 \end{bmatrix}$$

Swap R_1 and R_3

Elimination Stage (Example)

$$\begin{array}{r} \\ \texttt{next entry to zero} \\ \\ \end{array} \begin{bmatrix} 3 & -9 & 12 & -9 & 6 & 15 \\ 3 & -7 & 8 & -5 & 8 & 9 \\ 0 & 3 & -6 & 6 & 4 & -5 \end{bmatrix}$$

$$R_3 \leftarrow R_3 - R_1$$

Elimination Stage (Example)

leftmost
nonzero
entry

$$\begin{bmatrix} 3 & -9 & 12 & -9 & 6 & 15 \\ 0 & 2 & -4 & 4 & 2 & -6 \\ 0 & 3 & -6 & 6 & 4 & -5 \end{bmatrix}$$

swap R_2 with R_2

Elimination Stage (Example)

$$
\begin{array}{c c}
 & \begin{bmatrix} 3 & -9 & 12 & -9 & 6 & 15 \\ 0 & 2 & -4 & 4 & 2 & -6 \\ 0 & 3 & -6 & 6 & 4 & -5 \end{bmatrix} \\
\texttt{next entry to zero} &
\end{array}
$$

$$R_3 \leftarrow R_3 - \frac{3R_2}{2}$$

Elimination Stage (Example)

$$\begin{array}{r} \\ \\ \texttt{next entry} \\ \texttt{to zero} \end{array} \begin{bmatrix} 3 & -9 & 12 & -9 & 6 & 15 \\ 0 & 2 & -4 & 4 & 2 & -6 \\ 0 & 0 & 0 & 0 & \boxed{1} & 4 \end{bmatrix}$$

swap R_3 with R_3

Elimination Stage (Example)

$$\begin{bmatrix} \boxed{3} & -9 & 12 & -9 & 6 & 15 \\ 0 & \boxed{2} & -4 & 4 & 2 & -6 \\ 0 & 0 & 0 & 0 & \boxed{1} & 4 \end{bmatrix}$$

done with elimination stage
going to back substitution stage

Back Substitution Stage

Back Substitution Stage (High Level)

Input: (augmented) matrix A of size $m \times (n+1)$ in echelon form

Output: reduced echelon form of A

scale pivot positions and eliminate the variables for that column from the other equations

Back Substitution Phase (Pseudocode)

```
FUNCTION GE_back_sub_stage(A):
  FOR i from 1 to m:
    IF row i has a pivot position (i, j):
      A[i] ← A[i] / A[i, j]
      apply row operations to zero-out entries above (i, j)
```

Gaussian Elimination (Example)

$$\begin{array}{l l}\texttt{pivot position} & \begin{bmatrix} \boxed{3} & -9 & 12 & -9 & 6 & 15 \\ 0 & 2 & -4 & 4 & 2 & -6 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}\end{array}$$

$$R_1 \leftarrow R_1 \;/\; 3$$

Gaussian Elimination (Example)

$$
\begin{array}{r l}
 & \\
\texttt{pivot position} & \left[\begin{array}{cccccc} 1 & -3 & 4 & -3 & 2 & 5 \\ 0 & 2 & -4 & 4 & 2 & -6 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{array}\right] \\
 &
\end{array}
$$

$$R_2 \leftarrow R_2 \,/\, 2$$

Gaussian Elimination (Example)

next entry
to zero

$$\begin{bmatrix} 1 & -3 & 4 & -3 & 2 & 5 \\ 0 & 1 & -2 & 2 & 1 & -3 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

$$R_1 \leftarrow R_2 + 3R_2$$

Gaussian Elimination (Example)

pivot position

$$\begin{bmatrix} 1 & 0 & -2 & 3 & 5 & -4 \\ 0 & 1 & -2 & 2 & 1 & -3 \\ 0 & 0 & 0 & 0 & \boxed{1} & 4 \end{bmatrix}$$

$$R_3 \leftarrow R_3 \ / \ 1$$

Gaussian Elimination (Example)

next entry
to zero

$$\begin{bmatrix} 1 & 0 & -2 & 3 & 5 & -4 \\ 0 & 1 & -2 & 2 & 1 & -3 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

$R_2 \leftarrow R_2 - R_1$

Gaussian Elimination (Example)

next entry
to zero

$$\begin{bmatrix} 1 & 0 & -2 & 3 & \boxed{5} & -4 \\ 0 & 1 & -2 & 2 & 0 & -7 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

$$R_1 \leftarrow R_1 - 5R_3$$

Gaussian Elimination (Example)

$$\begin{bmatrix} \boxed{1} & 0 & -2 & 3 & 0 & -24 \\ 0 & \boxed{1} & -2 & 2 & 0 & -7 \\ 0 & 0 & 0 & 0 & \boxed{1} & 4 \end{bmatrix}$$

done with back substitution phase

Gaussian Elimination (Example)

$$x_1 = (-24) + 2x_3 - 3x_4$$

$$x_2 = (-7) + 2x_3 - 2x_4$$

$$x_3 \texttt{ is free}$$

$$x_4 \texttt{ is free}$$

$$x_5 = 4$$

Gaussian Elimination (Example)

$x_1 = (-24) + 2x_3 - 3x_4$

$x_2 = (-7) + 2x_3 - 2x_4$

x_3 is free

x_4 is free

$x_5 = 4$

$$\begin{bmatrix} 1 & 0 & -2 & 3 & 0 & -24 \\ 0 & 1 & -2 & 2 & 0 & -7 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

(columns 3 and 4 don't have pivot positions)

Question

Why do we check if the system is consistent before doing back substitution?

Answer

```
We only back substitute if we want to be able
to get a solution in general form
```

Analyzing the Algorithm

Analyzing the Algorithm

We will not use $O(\cdot)$ notation!

For numerics, we care about number of **FL**oating-oint **OP**erations (FLOPs):

>> addition
>> subtraction
>> multiplication
>> division
>> square root

$2n$ vs. n is very different when $n \sim 10^{20}$

Dominant Terms

that said, we don't care about *exact* bounds

A function $f(n)$ is ***asymptotically equivalent*** to $g(n)$ if

$$\lim_{i \to \infty} \frac{f(i)}{g(i)} = 1$$

for polynomials, they are equivalent to their dominant term

Dominant Terms

the dominant term of a polynomial is the monomial with the highest degree

$$\lim_{i \to \infty} \frac{3x^3 + 100000x^2}{3x^3} = 1$$

$3x^3$ dominates the function even though the coefficient for x^2 is so large

Parameters

n : `number of variables`

m : `number of equations (we will assume` $m = n$`)`

$n+1$: `number of rows in the augmented matrix`

The Cost of a Row Operation

$$R_i \leftarrow R_i + aR_j$$

$n+1$ multiplications for the scaling

$n+1$ additions for the row additions

Tally: $2(n+1)$ FLOPS

Cost of First Iteration of Elimination

$$R_2 \leftarrow R_2 + a_2 R_1$$
$$R_3 \leftarrow R_3 + a_3 R_1$$
$$\vdots$$
$$R_n \leftarrow R_n + a_n R_1$$

`repeated row operations for each row except the first`

`Tally:` $\approx 2n(n+1)$ `FLOPS`

Rough Cost of Elimination

repeating this last process at most n times gives us a dominant term $2n^3$

we can give a better estimation...

Tally: $\approx 2n^2(n+1)$ FLOPS

Cost of Elimination

$$\begin{bmatrix} 0 & \blacksquare & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & \blacksquare & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

At iteration i, we're only interested in rows after i

And to the right of column i

Cost of Elimination

```
Iteration 1:
```
$2n(n+1)$

```
Iteraiton 2:
```
$2(n-1)n$

```
Iteration 3:
```
$2(n-2)(n-1)$

$$\vdots \qquad +$$

$$\sum_{k=1}^{n} 2k(k+1) \approx \frac{2n(n+1)(2n+1)}{6} \sim (2/3)n^3$$

`Tally:` $\sim (2/3)n^3$ `FLOPS`

Cost of Back Substitution

(Let's assume no free variables)

for each pivot, we only need to:

>> zero out a position in 1 row (0 FLOPS)
>> add a value to the last row (1 FLOP)

at most 1 FLOP per row per pivot $\sim n^2$

Tally: $\sim (2/3)n^3$ FLOPS

Cost of Gaussian Elimination

Tally: $\sim (2/3)n^3$ FLOPS

(dominated by elimination)

Summary

row echelon forms describe solutions to systems of linear equations

Gaussian elimination is an algorithmic process for solving systems of linear equations

Gaussian elimination requires about $(2/3)n^3$ FLOPS