

Overview

Computational
lin alg.

Announcements

Example 9 ✓

Office hour changes ✓

HW4 ✓

IEF ✓

Outline

Python, Numpy, and Matplotlib
Making Models with Polynomials
Making Models with Monte Carlo

Error, Accuracy and Convergence
Floating Point

Modeling the World with Arrays

The World in a Vector

What can Matrices Do?

Graphs

Sparsity

Norms and Errors
The 'Undo' Button for Linear Operations: LU

LU: Applications

Linear Algebra Applications

Interpolation

Repeating Linear Operations:
Eigenvalues and Steady States

Eigenvalues: Applications

Approximate Undo: SVD and Least Squares

SVD: Applications

Solving Funny-Shaped Linear Systems

Data Fitting

Norms and Condition

Numbers

Low-Rank Approximation

Iteration and Convergence

Solving One Equation

Solving Many Equations

Finding the Best: Optimization in 1D

Optimization in n Dimensions

Some Perspective

- ▶ We have so far (mostly) looked at what we can do with single numbers (and functions that return single numbers).
- ▶ Things can get *much* more interesting once we allow not just one, but *many* numbers together.
- ▶ It is natural to view an *array of numbers* as one object with its own rules.
The simplest such set of rules is that of a **vector**.
- ▶ A 2D array of numbers can also be looked at as a **matrix**.
- ▶ So it's natural to use the tools of **computational linear algebra**.
- ▶ 'Vector' and 'matrix' are just *representations* that come to life in many (*many!*) applications. The purpose of this section is to explore some of those applications.

Vectors

What's a vector?

$$\begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}$$

a vector is just
a thing \in Vector
Space V

$$3x + 5x = 8x$$

+ "arithmetic
make sense"

$$\begin{cases} \vec{x} \in V & y \in V \\ \vec{x} + \vec{y} \in V \\ \alpha \vec{x} \in V & \alpha \in \mathbb{R} \end{cases}$$

Vectors from a CS Perspective

What would the concept of a vector look like in a programming language (e.g. Java)?

```
interface Vector {  
    Vector add ( Vector  
                other );  
    Vector scale ( float  
                 scalar );  
}
```

Vectors in the 'Real World'

Demo: Images as Vectors

Demo: Sound as Vectors

Demo: Shapes as Vectors



Outline

Python, Numpy, and Matplotlib
Making Models with Polynomials
Making Models with Monte Carlo

Error, Accuracy and Convergence
Floating Point

Modeling the World with Arrays

The World in a Vector
What can Matrices Do?
Graphs
Sparsity

Norms and Errors
The 'Undo' Button for Linear Operations: LU

LU: Applications
Linear Algebra Applications
Interpolation

Repeating Linear Operations:
Eigenvalues and Steady States
Eigenvalues: Applications
Approximate Undo: SVD and Least Squares

SVD: Applications

Solving Funny-Shaped Linear Systems
Data Fitting
Norms and Condition Numbers
Low-Rank Approximation

Iteration and Convergence

Solving One Equation
Solving Many Equations
Finding the Best: Optimization in 1D
Optimization in n Dimensions

Matrices

What does a matrix do?

It represents a *linear function* between two vector spaces $f : U \rightarrow V$ in terms of bases $\mathbf{u}_1, \dots, \mathbf{u}_n$ of U and $\mathbf{v}_1, \dots, \mathbf{v}_m$ of V . Let

$$\mathbf{u} = \underline{\alpha}_1 \mathbf{u}_1 + \dots + \underline{\alpha}_n \mathbf{u}_n \quad \leftarrow$$

and

$$\mathbf{v} = \underline{\beta}_1 \mathbf{v}_1 + \dots + \underline{\beta}_m \mathbf{v}_m.$$

Then f can *always* be represented as a matrix that obtains the β s from the α s:

$$\underbrace{\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}}_{\text{Matrix}} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}.$$

\downarrow

Example: The 'Frequency Shift' Matrix

Assume both u and v are linear combination of sounds of different frequencies:

$$u = \alpha_1 u_{110 \text{ Hz}} + \alpha_2 u_{220 \text{ Hz}} + \dots + \alpha_4 u_{880 \text{ Hz}}$$

(analogously for v , but with β s). What matrix realizes a 'frequency doubling' of a signal represented this way?

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} 0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}$$

$$\text{output} = 0 u_{110} + \alpha_1 u_{220} + \alpha_2 u_{440} + \alpha_3 u_{880}$$

Matrices in the 'Real World'

What are some examples of matrices in applications?

Demo: Matrices for Geometry Transformation

Demo: Matrices for Image Blurring

In-class activity: Computational Linear Algebra

Outline

Python, Numpy, and Matplotlib
Making Models with Polynomials
Making Models with Monte Carlo

Error, Accuracy and Convergence
Floating Point

Modeling the World with Arrays

The World in a Vector
What can Matrices Do?
Graphs
Sparsity

Norms and Errors
The 'Undo' Button for Linear Operations: LU

LU: Applications
Linear Algebra Applications
Interpolation

Repeating Linear Operations:
Eigenvalues and Steady States
Eigenvalues: Applications
Approximate Undo: SVD and Least Squares

SVD: Applications

Solving Funny-Shaped Linear Systems
Data Fitting
Norms and Condition Numbers
Low-Rank Approximation

Iteration and Convergence

Solving One Equation
Solving Many Equations
Finding the Best: Optimization in 1D
Optimization in n Dimensions

Graphs as Matrices

How could this (directed) graph be written as a matrix?

