Linear Algebra and Probabilistic Implications

Guillermo Angeris   Alex Evans

Session 1, SPLA Study Club

Outline

High level ideas

Linear algebra

Probabilistic implications

Where to from here?

At a high level

- Take a bunch of concepts from succinct proofs (ZK)
- Reduce them to linear algebra
- (and a bit of error correcting codes)

At a high level (cont.)

- Introduce succinct (!) notation
- Relax traditional logic to ‘probabilistic’ versions
- Get ‘proof-carrying’ protocols at the end!
- Show a (weak) bound on the soundness of FRI

At a high level (cont.)

- Introduce succinct (!) notation
- Relax traditional logic to 'probabilistic' versions
- Get 'proof-carrying' protocols at the end!
- Show a (weak) bound on the soundness of FRI
- To do this, we have to eat some veggies first...

Why do this work?

Why do this work?

- We start with something we know exists
- Try to (a) find minimal requirements for it to work
- And (b) try to use this to clean up exposition

Why do this work?

- We start with something we know exists
- Try to (a) find minimal requirements for it to work
- And (b) try to use this to clean up exposition
- Why?

Why do this work? (Cont.)

- Often, removing requirements helps understanding
- Allows generalizations and new discoveries
- Lets us divide a protocol into its constituent parts

Why do this work? (Cont.)

- Often, removing requirements helps understanding
- Allows generalizations and new discoveries
- Lets us divide a protocol into its constituent parts
- For more on this check out ep. 294 with Kobi and Anna :)

But in reality

But in reality

- We do it because it's fun :)

Outline

Vectors

- An *n-vector* is an ordered collection of n elements
- Example: a 3-vector x

$$x = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$$

- We will sometimes write this as $x = (3, 5, 1)$

Vectors

- An *n-vector* is an ordered collection of n elements
- Example: a 3-vector x

$$x = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$$

- We will sometimes write this as $x = (3, 5, 1)$
- Can *index* this collection: $x_1 = 3$, $x_2 = 5$, $x_3 = 1$

Operations on vectors

- We can *scale* vectors by a *scalar*
- Example: $x = (3, 5, 1)$ scaled by 2

$$2x = 2 \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 10 \\ 2 \end{bmatrix}$$

Operations on vectors

- We can *scale* vectors by a *scalar*
- Example: $x = (3, 5, 1)$ scaled by 2

$$2x = 2 \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 10 \\ 2 \end{bmatrix}$$

- We can also add vectors too

$$x + \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 4 \end{bmatrix}$$

Linear combinations

- Of course, we can do both at the same time
- Given vectors x, y, and z, we can take

$$x + 2y + 3z$$

- Called a *linear combination* of vectors

Scalar and vector notation

- Vectors will almost always be *lowercase Roman* letters
- Such as x, y, z, ...
- Scalars will almost always be *lowercase Greek* letters
- For example, α, β, γ, ...

Linear combinations

- We are often interested in linear combinations of vectors
- For this we need, (a) a collection of vectors and (b) a collection of scalars

Linear combinations

- We are often interested in linear combinations of vectors
- For this we need, (a) a collection of vectors and (b) a collection of scalars
- From before: (b) is just a vector

Linear combinations

- We are often interested in linear combinations of vectors
- For this we need, (a) a collection of vectors and (b) a collection of scalars
- From before: (b) is just a vector
- But (a) is what we call a *matrix*

Matrices

- An $m \times n$ matrix is an ordered list (of length n) of m-vectors

Matrices

- An $m \times n$ matrix is an ordered list (of length n) of m-vectors
- We can take a linear combination of the n available m-vectors

Matrices

- An $m \times n$ matrix is an ordered list (of length n) of m-vectors
- We can take a linear combination of the n available m-vectors
- If a vector $z = (4, 8, 2)$ contains the scalars
- And A is a $m \times 3$ matrix, then

$$Az = 4a_1 + 8a_2 + 2a_3$$

where a_i is the ith column of A

Matrices

- An $m \times n$ matrix is an ordered list (of length n) of m-vectors
- We can take a linear combination of the n available m-vectors
- If a vector $z = (4, 8, 2)$ contains the scalars
- And A is a $m \times 3$ matrix, then

  $$Az = 4a_1 + 8a_2 + 2a_3$$

  where a_i is the ith column of A
- Very compact notation!

Example of a matrix

- Useful to write matrices as 'rectangles of numbers'
- Example matrix A of dimensions 3×2

$$A = \begin{bmatrix} 3 & 1 \\ 5 & 2 \\ 1 & 3 \end{bmatrix}$$

- Matrices will almost always be *uppercase Roman* letters
- Such as A, B, C, ...

Finite fields (a quick aside)

- The numbers in the vectors (and matrices) have to 'exist somewhere'
- We usually assume these numbers lie in a *finite field* $\mathbf{F}$
- An n-vector x from a finite field $\mathbf{F}$ is written

$$x \in \mathbf{F}^n$$

- An $m \times n$ matrix A with elements in $\mathbf{F}$ is written

$$A \in \mathbf{F}^{m \times n}$$

Vector spaces

- The natural sets where linear algebra works are *vector spaces*

Vector spaces

- The natural sets where linear algebra works are *vector spaces*
- A set $V \subseteq \mathbf{F}^n$ is a vector space if it is *closed* under linear combinations
- If we take $x, y \in V$ then

  $$\alpha x + \beta y \in V$$

  for any $\alpha, \beta \in \mathbf{F}$

Vector space examples

- The simplest vector space: $V = \{0\}$

Vector space examples

- The simplest vector space: $V = \{0\}$
- Second simplest: $V = \mathbf{F}^n$

Vector space examples

- The simplest vector space: $V = \{0\}$
- Second simplest: $V = \mathbf{F}^n$
- Third (?) simplest, for fixed $x \in \mathbf{F}^n$: $V = \{\alpha x \mid \alpha \in \mathbf{F}\}$

Range and nullspace

- The *range* of a matrix $A \in \mathbf{F}^{m \times n}$ is defined as

$$\mathcal{R}(A) = \{Ax \mid x \in \mathbf{F}^n\}$$

is a vector space (see homework)

Range and nullspace

- The *range* of a matrix $A \in \mathbf{F}^{m\times n}$ is defined as

$$\mathcal{R}(A) = \{Ax \mid x \in \mathbf{F}^n\}$$

  is a vector space (see homework)

- Similarly, the *nullspace* of the matrix A

$$\mathcal{N}(A) = \{y \in \mathbf{F}^n \mid Ay = 0\}$$

  is also a (very different!) vector space

Codes

- We will call a matrix $G \in \mathbf{F}^{m \times n}$ a (linear) *error correcting code*
- The matrix G takes in an n-vector and spits out an m-vector
- Given a *message* $x \in \mathbf{F}^n$ then

$$y = Gx$$

  is an *encoding* of the message x

Codes (examples)

- Some classic codes

Codes (examples)

- Some classic codes
- The identity code

$$G = I$$

such that $Gx = x$

Codes (examples)

- Some classic codes
- The identity code

$$G = I$$

  such that $Gx = x$

- The repeated code!

$$Gx = \begin{bmatrix} x \\ x \\ \vdots \\ x \end{bmatrix}$$

Codes (examples, cont.)

- More examples!

Codes (examples, cont.)

- More examples!
- The Hadamard code $G \in \mathbf{F}^{m \times n}$: every possible n-tuple is a row of G

Codes (examples, cont.)

- More examples!
- The Hadamard code $G \in \mathbf{F}^{m \times n}$: every possible n-tuple is a row of G
- The Reed–Solomon code $G \in \mathbf{F}^{m \times n}$ (see homework!)

Distance

- We will only mainly use one definition which is the *distance* of a code
- Defined as
$$d = \min_{x \neq 0} \|Gx\|_0$$
- Here, $\|y\|_0$ is the number of nonzero elements in y

Distances of some codes

- The identity code $Gx = x$ has $d = 1$

Distances of some codes

- The identity code $Gx = x$ has $d = 1$
- The repeated code has $d = k$, where k is the number of repetitions

Distances of some codes

- The identity code $Gx = x$ has $d = 1$
- The repeated code has $d = k$, where k is the number of repetitions
- The Hadamard code has distance $d = |\mathbf{F}|^n - |\mathbf{F}|^{n-1}$

Distances of some codes

- The identity code $Gx = x$ has $d = 1$
- The repeated code has $d = k$, where k is the number of repetitions
- The Hadamard code has distance $d = |\mathbf{F}|^n - |\mathbf{F}|^{n-1}$
- The Reed–Solomon code has distance $d = m - n + 1$ (see homework!)

Outline

High level ideas

Linear algebra

Probabilistic implications

Where to from here?

Probabilistic implications

- We'll first start with 'normal' logic
- Then expand to a probabilistic logic framework

‘Traditional’ logic

- In classical logic, we have statements, say P and Q

‘Traditional’ logic

- In classical logic, we have statements, say P and Q
- They are either true or false

‘Traditional’ logic

- In classical logic, we have statements, say P and Q
- They are either true or false
- In some cases, we can say basic things, such as

$$P \wedge Q$$

  (read: P and Q)
- Note: statements in math are **assertions**!

Implications

- Other possible statements include
- P implies Q (*i.e.*, some statement implies another)

Implications

- Other possible statements include
- P implies Q (*i.e.*, some statement implies another)
- This is the same as saying

$$\neg(P \wedge \neg Q)$$

(Consequence: if P implies Q, and Q implies T, then P implies T)

- See homework!

Probabilistic? Implications

- At a high level:
- Zero knowledge proofs deal with implications *with some error*

Probabilistic? Implications

- At a high level:
- Zero knowledge proofs deal with implications *with some error*
- How do we codify this idea in notation?

Probabilistic? Implications

- At a high level:
- Zero knowledge proofs deal with implications *with some error*
- How do we codify this idea in notation?
- By relaxing implications to *probabilistic implications*

Probabilistic implications

- In zero knowledge protocols, statements depend on *randomness*
- *i.e.*, we have some statement P_r
- Which depends on a randomly drawn r (from some distribution)

Probabilistic implications (cont.)

- Traditional implication: P implies Q if

$$\neg(P \wedge \neg Q)$$

- Equivalently: implication doesn't hold if $P \wedge \neg Q$

Probabilistic implications (cont.)

- Traditional implication: P implies Q if
$$\neg(P \wedge \neg Q)$$
- Equivalently: implication doesn't hold if $P \wedge \neg Q$
- A relaxation is: if P_r and $Q_{r'}$ depend on randomness r and r' then
$$\Pr(P_r \wedge \neg Q_{r'}) \leq p$$
where p is probability of error
- Recover original definition whenever $p = 0$

Consequences

- Define convenient notation

$$P_r \underset{p}{\Longrightarrow} Q_{r'}$$

for $\Pr(P_r \wedge \neg Q_{r'}) \leq p$

- Then we can chain implications!

$$P_r \underset{p}{\Longrightarrow} Q_{r'} \quad \text{and} \quad Q_{r'} \underset{p'}{\Longrightarrow} T_{r''}$$

implies that

$$P_r \underset{p+p'}{\Longrightarrow} T_{r''}$$

Consequences (cont.)

- We can also take contrapositives
- Given

$$P_r \underset{p}{\Longrightarrow} Q_{r'}$$

- This is the same as

$$\neg Q_{r'} \underset{p}{\Longrightarrow} \neg P_r$$

- And a few others (see homework!)

As a side note

- One can create a basic logical language
- Acts much like a syntax with rules for (probabilistic) proofs
- Except it can also spit out probability of failure
- An open project would be to formalize this!

Outline

High level ideas

Linear algebra

Probabilistic implications

Where to from here?

Continuing

- From here, we have all the basics!

Continuing

- From here, we have all the basics!
- Next up: rewriting some (many?) basic tools of succinct proofs

Continuing

- From here, we have all the basics!
- Next up: rewriting some (many?) basic tools of succinct proofs
- We'll finally start saying real things!

Continuing

- From here, we have all the basics!
- Next up: rewriting some (many?) basic tools of succinct proofs
- We'll finally start saying real things!
- Homework will be released after lecture and Q&A