Lecture 416: Biases in Data
(through the lens of SVD)

We’ve seen a myriad of uses for SVD

1
2
3
4
5

Like last time, suppose we have a data matrix

\[ A = \begin{bmatrix} a_1 & a_2 & \ldots & a_p \end{bmatrix}_{n \times p} \]

with SVD given by \( A = U \Sigma V^T \)
To make things simple, let's assume it's centered:
\[
\sum_{i=1}^{p} a_i = 0
\]

Last time we said the **reconstruction error** of projecting onto
\[
\text{span} (u_1, u_2 \ldots u_k)
\]
(first k columns of U)
is given by:
\[
\sum_{i=1}^{p} \| a_i - u_{1:k} u_{1:k}^T a_i \|^2
\]

Moreover this is the best you can do — any other projection onto a k-dimensional subspace has the same or worse reconstruction error.

Q: But is it always “fair”?
This is a vague question, but let’s make it precise

Poll: If we look for the best 1-dimensional projection, which subgroup is worse off?

(a) the blue subgroup  (b) the red subgroup

If we looked at either subgroup in isolation, the reconstruction error would be small

But lumped together, the blue subgroup overwhelms the data
Takeaway:

Check out Joy Buolamwini's work at the Media Lab!

We can already understand a harmful misconception with just what we've seen with SVD.

The problem is with the word average, it doesn't mean it's good for everyone.
Let's do another example, where things go wrong

**Application**: Word Embeddings

Popular technique in **natural language processing**

Let's motivate it through a concrete application,

**solving analogies**:

It turns out you can solve this by mapping words to vectors via SVD

**Step #1**

**Step #2**

\[ V = \# \text{ of distinct words in all the documents} \]

How often does word \( i \) occur in a document in a window of size \( t \) along with word \( j \)?
For example, what if we have three documents and a window of size 2

\[
\begin{align*}
\text{document \#1} & \\
\text{document \#2} & \\
\text{document \#3} & \\
\end{align*}
\]

\[
\begin{bmatrix}
\text{I like} & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\
\text{enjoy} & 2 & 0 & 0 & 1 & 0 & 1 & 0 \\
\text{linear} & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\text{algebra} & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\text{optimization} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\text{tests} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Step #3

Each row of $U_{1,2}$ represents a word and is a length k vector.
Q3: So how do you use $U_{1:k}$ to solve analogies?

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>America</td>
<td></td>
</tr>
<tr>
<td>Canada</td>
<td></td>
</tr>
<tr>
<td>Burger</td>
<td></td>
</tr>
</tbody>
</table>
```

Now look for the word $c$ where

But again we're using linear algebra on data that was generated/collected by humans, so we might inherit its biases.

Q4: In what ways might word embeddings be biased?

Example is from Bolukbasi et al., many more way scarier examples
Let's end on some notes of optimism:

1) we have made progress, because identifying word biases isn't easy

  e.g. counting frequency of word pairs doesn't work:

But SVD, and word embeddings more generally, can!

2) how can we (at least somewhat) debias word embeddings?

Again, the answer is PCA

  step #1: Identify a gender subspace
  take ten gender pair differences

  \[ y_1 = w_{\text{grandmother}} - w_{\text{grandfather}} \]
  \[ y_2 = \ldots \]
  \[ \vdots \]
  \[ y_{10} = \ldots \]

  And take the top left singular vector
Step #2: various debiasing techniques for removing this direction.