Bag of words

- The **bag of words** model represents documents as **vectors of word counts**, i.e. **term frequencies**.
  The matrix below was created using the bag of words model.

- Each **row** in the bag of words matrix is a **vector representation** of a document.

	big	data	class	science
big big big big data class	4	1	1	0
data big data science	1	2	0	1
science big data	1	1	0	1

- For example, we can represent the document 2, **data big data science**, with the vector $\vec{d}_2$:

$$\vec{d}_2 = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

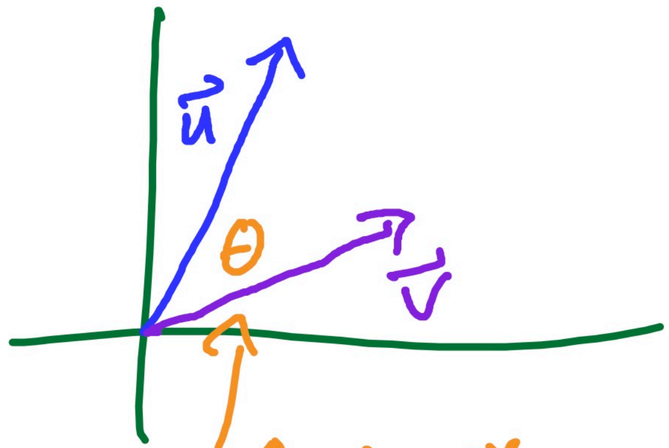
big science data data would have the same representation.

Math recap

$$\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

e.g. 2D

$$\vec{u} = \begin{bmatrix} 3 \\ 5 \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$



θ is the angle between $\vec{u}$ and $\vec{v}$

$\Rightarrow$ the smaller θ is, the more similar $\vec{u}$ and $\vec{v}$ are!

Dot product

Two definitions that are equivalent!

① $\vec{u} \cdot \vec{v} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$

② $\vec{u} \cdot \vec{v} = \|\vec{u}\| \, \|\vec{v}\| \cos\theta$

$\|\vec{u}\|$ = "magnitude" / "length" / "norm".

$\|\vec{u}\|_2 = \sqrt{u_1^2 + u_2^2 + \cdots + u_n^2}$ "general Pythagorean"

equivalence of dot products

$$u_1 v_1 + u_2 v_2 + \cdots + u_n v_n = \|\vec{u}\| \, \|\vec{v}\| \cos\theta$$

$$\cos\theta = \frac{u_1 v_1 + u_2 v_2 + \cdots + u_n v_n}{\|\vec{u}\| \, \|\vec{v}\|} = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \, \|\vec{v}\|}$$

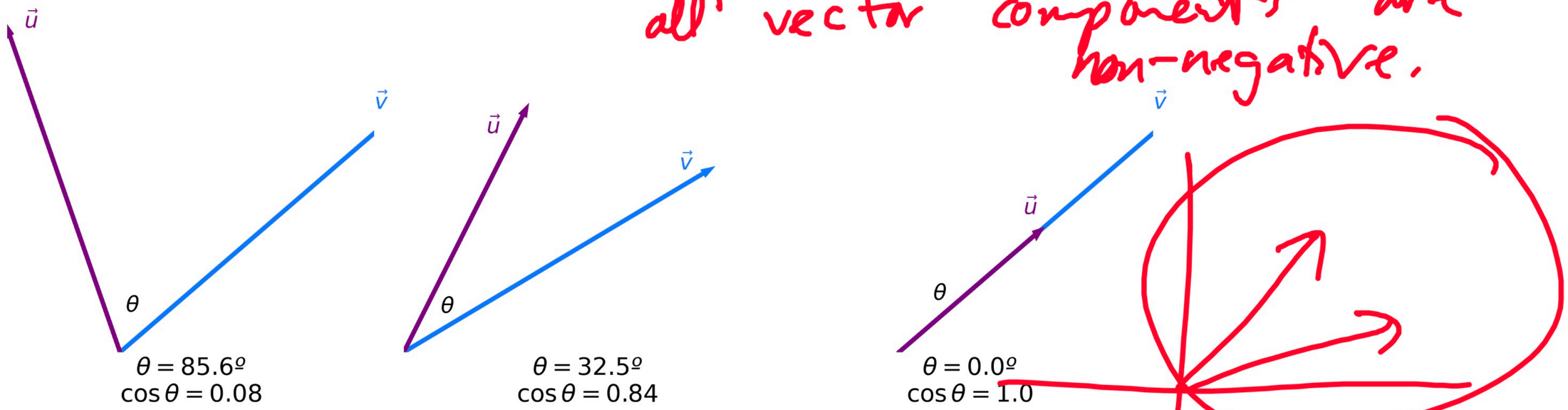
$$\theta = \cos^{-1}\left(\frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \, \|\vec{v}\|}\right)$$

in text processing, we mostly care about $\cos\theta$

Angles and similarity

- **Key idea**: The more similar two vectors are, the **smaller** the angle θ between them is.

in bag of words, all vector components are non-negative.



so angle θ is always $0 \leq \theta \leq 90$.

- The smaller the angle θ between two vectors is, the **larger** $\cos\theta$ is.
- The maximum value of $\cos\theta$ is 1, achieved when $\theta = 0$.
- **Key idea: The more similar two vectors are, the larger $\cos\theta$ is!**

Consider the matrix of word counts we found earlier, using the bag of words model:

	big	data	class	science
big big big big data class	4	1	1	0
data big data science	1	2	0	1
science big data	1	1	0	1

1. Which two documents have the highest **dot product**? $\vec{d}_1, \vec{d}_2$
2. Which two documents have the highest **cosine similarity**? $\vec{d}_2, \vec{d}_3$

Issues with the bag of words model

- Recall, the bag of words model encodes a document as a vector containing **word frequencies**.

- It doesn't consider the **order** of the terms.
  "big data science" and **"data science big"** have the same vector representation, but mean different things.

- It doesn't consider the **meaning** of terms.
  "I really really hate data" and **"I really really love data"** have nearly identical vector representations, but very different meanings.

- It treats all words as being equally important. **This is the issue we'll address today.**
  In **"I am a student"** and **"I am a teacher"**, it's clear to us humans that the most important terms are **"student"** and **"teacher"**, respectively. But in the bag of words model, **"student"** and **"I"** appear the same number of times in the first document.

Inverse document frequency

- The **inverse document frequency** of a term t in a **set** of documents $d_1, d_2, \ldots$ is:

$$\text{idf}(t) = \log\left(\frac{\text{total \# of documents}}{\text{\# of documents in which } t \text{ appears}}\right)$$

- **Example**: What is the inverse document frequency of **"billy"** in the following three documents?
  - "my brother has a friend named **billy** who has an uncle named **billy**"
  - "my favorite artist is named jilly boel"
  - "why does he talk about someone named **billy** so often"

$$\text{idf} = \log\left(\frac{3}{3}\right)$$

- **Answer**: $\log\left(\frac{3}{2}\right) \approx 0.4055.$

  Here, we used the natural logarithm. It doesn't matter which log base we use, as long as we keep it consistent throughout all of our calculations.

- Intuition: If a word appears in every document (like **"the"** or **"has"**), it is probably not a good summary of any one document.

- Think of $\text{idf}(t)$ as the "rarity factor" of t across documents – the larger $\text{idf}(t)$ is, the more rare t is.

$$\text{idf}(t) \text{ large} \implies t \text{ rare across all documents}$$
$$\text{idf}(t) \text{ small} \implies t \text{ common across all documents}$$

Computing TF-IDF

- **Question**: What is the TF-IDF of **"science"** in **"data big data science"**?

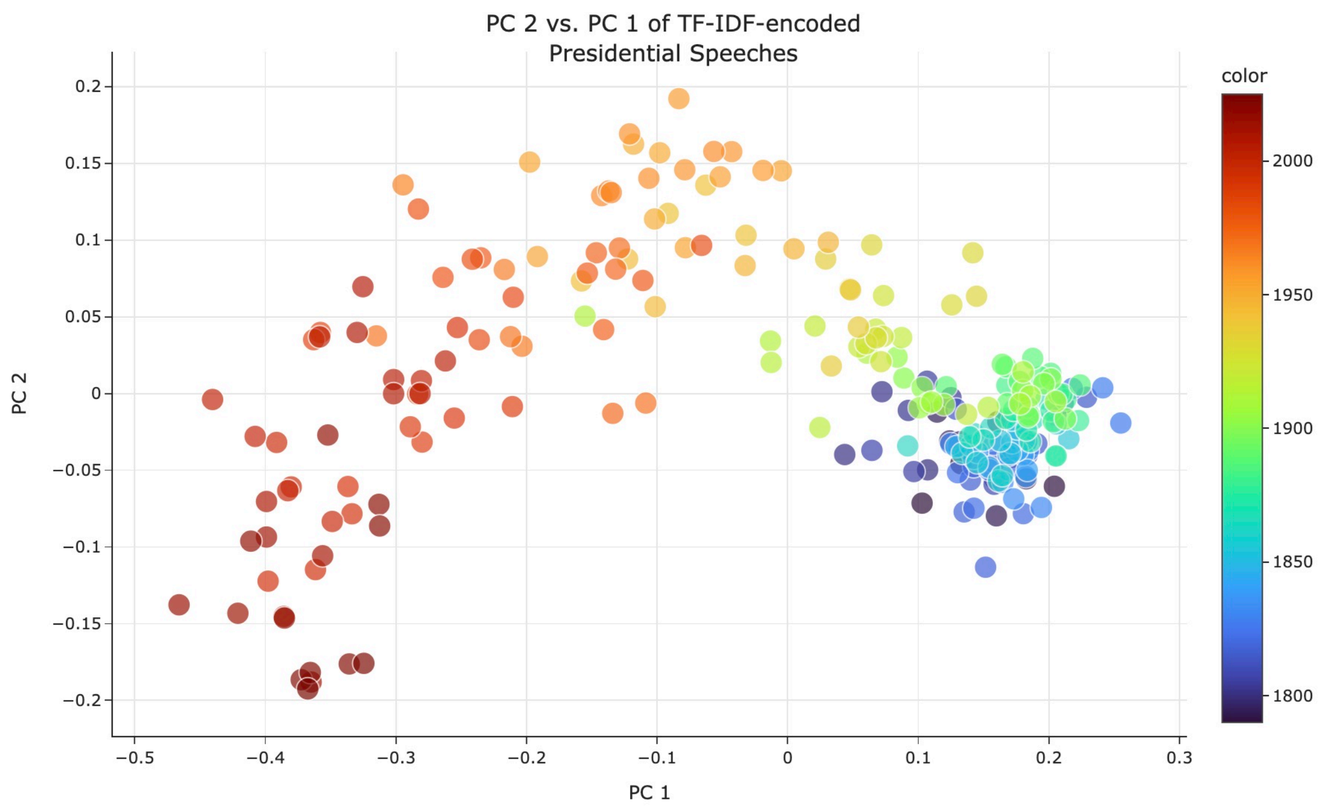
	big	data	class	science
big big big big data class	4	1	1	0
data big data science	1	2	0	1
science big data	1	1	0	1

$$\text{tf-idf}(\text{"science"}, \text{"data big data science"}) = \frac{1}{4} \cdot \log\left(\frac{3}{2}\right)$$

```
    PCA(n_components=2),
)
pipeline.fit(speeches['text'])
scores = pipeline.transform(speeches['text'])
fig = px.scatter(x=scores[:, 0],
           y=scores[:, 1],
           hover_name=speeches['text'].index,
           color=speeches['text'].index.str.split(', ').str[-1].astype(int),
           color_continuous_scale='Turbo',
           size_max=12,
           size=[1] * np.ones(len(scores)))
fig.update_layout(xaxis_title='PC 1',
                  yaxis_title='PC 2',
                  title='PC 2 vs. PC 1 of TF-IDF-encoded<br>Presidential Speeches',
                  width=1000, height=600)
```

$\mathbb{R}^{23998} \rightarrow \mathbb{R}^2$

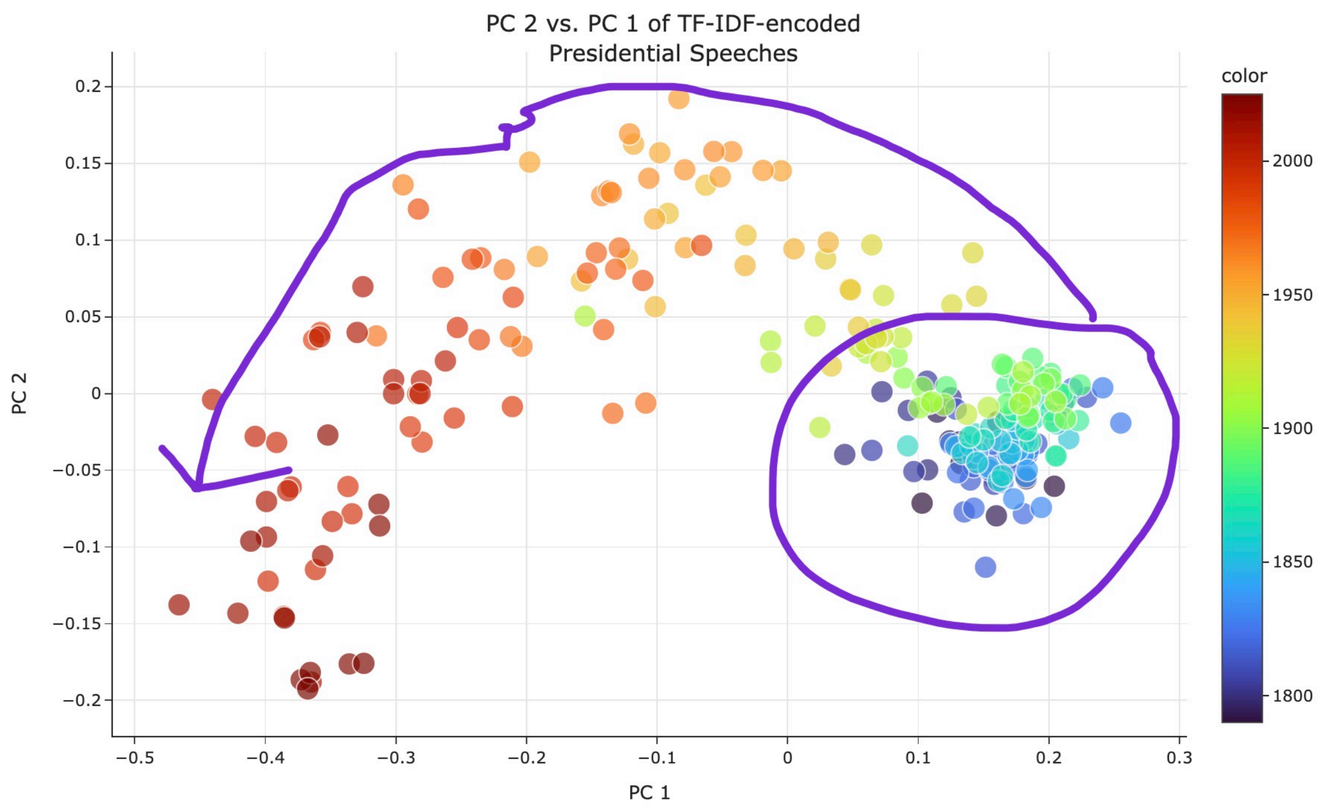
"Principal Components Analysis".



```
    PCA(n_components=2),
)
pipeline.fit(speeches['text'])
scores = pipeline.transform(speeches['text'])
fig = px.scatter(x=scores[:, 0],
           y=scores[:, 1],
           hover_name=speeches['text'].index,
           color=speeches['text'].index.str.split(', ').str[-1].astype(int),
           color_continuous_scale='Turbo',
           size_max=12,
           size=[1] * np.ones(len(scores)))
fig.update_layout(xaxis_title='PC 1',
                  yaxis_title='PC 2',
                  title='PC 2 vs. PC 1 of TF-IDF-encoded<br>Presidential Speeches',
                  width=1000, height=600)
```

$\mathbb{R}^{23998} \rightarrow \mathbb{R}^{2}$

"Principal Components Analysis"



Problem 13

Problem 13.1

Consider the following five sentences.

- `"of the college board the"`
- `"the board the board the"`
- `"board the college board of"`
- `"the college board of college"`
- `"board the college board is"`

Suppose we create a TF-IDF matrix, in which there is one row for each sentence and one column for each unique word. The value in row i and column j is the TF-IDF of word j in sentence i. Note that since there are 5 sentences and 5 unique words across all sentences, the TF-IDF matrix has 25 total values.

Is there a column in the TF-IDF matrix in which all values are 0?

- ○ Yes
- ○ No

Click to view the solution.

Problem 13.2

In which of the following sentences is “college” the word with the highest TF-IDF?

- ○ Sentence 1
- ○ Sentence 2

$$\text{idf}(\text{"the"}) = \log\left(\frac{5}{5}\right)$$

$$= \log(1)$$

$$= 0$$

Problem 13.1

Consider the following five sentences.

- `"of the college board the"`
- `"the board the board the"`
- `"board the college board of"`
- `"the college board of college"`
- `"board the college board is"`

Suppose we create a TF-IDF matrix, in which there is one row for each sentence and one column for each unique word. The value in row i and column j is the TF-IDF of word j in sentence i. Note that since there are 5 sentences and 5 unique words across all sentences, the TF-IDF matrix has 25 total values.

Is there a column in the TF-IDF matrix in which all values are 0?

- Yes
- No

Click to view the solution.

Problem 13.2

In which of the following sentences is “college” the word with the highest TF-IDF?

- Sentence 1
- Sentence 2
- Sentence 3
- Sentence 4
- Sentence 5

Click to view the solution.

Sentence 3

Sentence 4

Sentence 5

Click to view the solution.

Problem 13.3

As an alternative to TF-IDF, Suraj proposes the DF-ITF, or "document frequency-inverse term frequency". The DF-ITF of term t in document d is defined below:

$$\text{df-itf}(t, d) = \frac{\#\text{ of documents in which } t \text{ appears}}{\text{total }\#\text{ of documents}} \cdot \log\left(\frac{\text{total }\#\text{ of words in } d}{\#\text{ of occurrences of } t \text{ in } d}\right)$$

small if term occurs often in d

Fill in the blank: The term t in document d that best summarizes document d is the term with ____.

small if term is rare across all documents

the largest DF-ITF in document d

the smallest DF-ITF in document d

Click to view the solution.

Problem 14

Consider the following corpus: