Text Analysis II

PSC 8185: Machine Learning for Social Science

Iris Malone

April 25, 2022

Materials adapted from Rochelle Terman

Announcements

- Poster Session: April 27 (1pm-3pm)
- Problem Set 7: Due April 29 5pm ET
- Final Project: Due May 3 5pm ET

Recap

Where We've Been:

- Text is context-dependent, but computational analysis is not
- Bag of word assumption → word order doesn't matter
- Dictionary methods count frequency of words and assign weighted values

Recap

Where We've Been:

- Text is context-dependent, but computational analysis is not
- Bag of word assumption → word order doesn't matter
- Dictionary methods count frequency of words and assign weighted values

New Terminology:

- Stemming/Lemmatization
- Tokenization
- Dictionary
- Document Term Matrix

Agenda

1. Distinctive Words
2. Clustering
3. Topic Modeling
4. Course Wrap-Up

Distinctive Words

Sometimes we want custom dictionaries

Motivation:

- Want to create custom dictionaries for classification
- Feature (variable) selection: identify relevant words for subsequent analysis

Sometime we want custom dictionaries

Sometimes we are interested in specific types of words, e.g.

- Partisan Language
  e.g., compare Republican vs Democratic speeches
- Ideological Language
  e.g., compare liberal vs conservative news stories
- Gendered Language
  e.g., compare toy advertising

Finding ‘Discriminating’ Words

Main Idea: Discriminating words are the most distinctive words in each corpus, i.e. appear in 1 book, but not another.

Examples:

- Bible: ‘Jesus’ is distinct to New Testament
- Harry Potter: ‘Horcruxes’ (mostly) distinct to Deathly Hallows
- Shakespeare: ‘Oberon’ distinct to Midsummer’s Night Dream

Recap: Generating Dictionaries

1. Manual Generation
    - Careful thought (epiphanies?) about useful words
2. Crowd-Sourcing
    - M-Turk crowd assign feelings about words on scale
3. **Statistical methods**
    - Identify distinctive or separating words

Statistical Methods to Find Distinctive Words

1. Unique Usage
2. Difference in Frequencies
3. Difference in Averages
4. Term Frequency-Inverse Document Frequency

Motivating Example

Figure 1: What is each HP book about?

Option 1: Unique Usage

- **Main Idea:** Identify distinctive words based on their exclusivity to a text.

Option 1: Unique Usage

- **Main Idea:** Identify distinctive words based on their exclusivity to a text.
- **Example:** If HP Book 1 talks about a trip to the zoo and HP Book 2 never does, we should count “zoo” as distinctive

Option 1: Unique Usage

- **Main Idea:** Identify distinctive words based on their exclusivity to a text.
- **Example:** If HP Book 1 talks about a trip to the zoo and HP Book 2 never does, we should count “zoo” as distinctive
- **Limit:** These words tend to not be terribly interesting or informative

Option 2: Difference in Frequencies

- **Main Idea:** Identify distinctive words based on their difference in frequency

Option 2: Difference in Frequencies

- **Main Idea:** Identify distinctive words based on their difference in frequency
- **Method:**
  - Count the number of time each document uses a word
  - Find the words with the largest absolute difference

Option 2: Difference in Frequencies

- **Main Idea:** Identify distinctive words based on their difference in frequency
- **Method:**
  - Count the number of time each document uses a word
  - Find the words with the largest absolute difference
- **Example:** If Goblet of Fire mentions Order of the Phoenix (OOTP) once, but Book 5 talks about the OOTP a lot, then consider OOTP relatively distinct to Book 5

Option 2: Difference in Frequencies

- **Main Idea:** Identify distinctive words based on their difference in frequency
- **Method:**
  - Count the number of time each document uses a word
  - Find the words with the largest absolute difference
- **Example:** If Goblet of Fire mentions Order of the Phoenix (OOTP) once, but Book 5 talks about the OOTP a lot, then consider OOTP relatively distinct to Book 5
- **Limit:** Doesn't account for difference in total words

Option 3: Difference in Averages

Main Idea: Identify distinctive words based on difference in rates

Option 3: Difference in Averages

Main Idea: Identify distinctive words based on difference in rates

Method:

- Normalize DTM from count to proportions. For each word p in corpus c:

$$\mu_p = \frac{\sum_{i=1}^{N} p_i}{T}$$

  - p_i is the number of times a word p appears in document i,
  - N is the total number of documents in c
  - T is the total number of words in c
- Take the difference between one author's proportion of a word and another's proportion of the same word:

$$\theta_p = \mu_{p,\text{GOF}} - \mu_{p,\text{OOTP}}$$

- Find words with highest absolute difference

Option 3: Difference in Averages

Limits:

1. Favors more frequent words, e.g.
   Word 1: 30/1000 (GOF); 25/1000 (OOTP) → score 5/1000
   Word 2: 5/1000 (GOF); 1/1000 (OOTP) → score 4/1000

Option 3: Difference in Averages

Limits:

1. Favors more frequent words, e.g.
   Word 1: 30/1000 (GOF); 25/1000 (OOTP) → score 5/1000
   Word 2: 5/1000 (GOF); 1/1000 (OOTP) → score 4/1000
2. Ignores cases where one text uses a word frequently and another text barely uses it, e.g.
   Word 1: 1/1000 (GOF) and 990/1000 (OOTP)

Option 3: Difference in Averages

Limits:

1. Favors more frequent words, e.g.
   Word 1: 30/1000 (GOF); 25/1000 (OOTP) → score 5/1000
   Word 2: 5/1000 (GOF); 1/1000 (OOTP) → score 4/1000
2. Ignores cases where one text uses a word frequently and another text barely uses it, e.g.
   Word 1: 1/1000 (GOF) and 990/1000 (OOTP)
3. Biased towards differences in rates of frequent words > differences in rates of rare words

Option 3: Difference in Averages

Limits:

1. Favors more frequent words, e.g.
   Word 1: 30/1000 (GOF); 25/1000 (OOTP) → score 5/1000
   Word 2: 5/1000 (GOF); 1/1000 (OOTP) → score 4/1000
2. Ignores cases where one text uses a word frequently and another text barely uses it, e.g.
   Word 1: 1/1000 (GOF) and 990/1000 (OOTP)
3. Biased towards differences in rates of frequent words > differences in rates of rare words

Solution: Divide the difference in texts' average rates by the pooled average rates

Option 4: TF-IDF

Main Idea: Identify distinctive words based on their relative frequency and unique usage.

Option 4: TF-IDF

Main Idea: Identify distinctive words based on their relative frequency and unique usage.

- Term Frequency (TF): number of times word p appears in document i
- Inverse Document Frequency (IDF): measure of exclusive usage across documents i

Option 4: TF-IDF

Main Idea: Identify distinctive words based on their relative frequency and unique usage.

- Term Frequency (TF): number of times word p appears in document i
- Inverse Document Frequency (IDF): measure of exclusive usage across documents i

TF-IDF Equation:

$$w_{ip} = tf_{ip} \times log \frac{N}{df_p}$$

- tf: number of occurrences of word p in document i
- df_p: number of documents containing word p
- N: total number of documents

Option 4: TF-IDF

Example:

- Harry, Hermione, and Ron appear in all HP books so tf is high and IDF score is approximately zero ($log\frac{7}{7} = log(1) = 0$) → low TF-IDF scores
- Viktor Krum, Fleur Delacour, and Cedric Diggory are relatively unique to GOF so will have **higher IDF scores**
- Cedric Diggory is mentioned more in GOF so he will have a higher TF-IDF score than Krum and Delacour
- Ranking TF-IDF scores:
  Diggory > Delacour > Krum > ...> Quidditch > Harry

Option 4: TF-IDF

Main Takeaways:

- Common words have lower TF-IDF scores
- Higher TF-IDF scores reflect more exclusive usage adjusting for frequency

Choosing Best Metric

Which metric best distinguishes texts?

- Depends on context and goal
- Classification → accuracy, precision, recall
- Qualitative inference:
  - Face Validity: Do these results make sense?
  - Convergence: Do different metrics lead to the same results?
  - "Gold standard:" Do human inter-reliability tests match up?

Clustering

Motivation

We just learned what makes documents distinct ...
Now, what make documents similar?

- Simple Answer: Similar word count vectors
- Complicated Answer: Similar use of language

Motivating Example: Turn It In

Germany was the first to employ area bombing tactics during its assault on Poland in September 1939. In 1940, during the Battle of Britain, the Luftwaffe failed to bring Britain to its knees by targeting London and other heavily populated areas with area bombing attacks. Stung but unbowed, the Royal Air Force (RAF) avenged the bombings of London and Coventry in 1942 when it launched the first of many saturation bombing attacks against Germany. In 1944, Hitler named the world's first long-range offensive missile V-1, after "vergeltung," the German word for "vengeance" and an expression of his desire to repay Britain for its devastating bombardment of Germany.

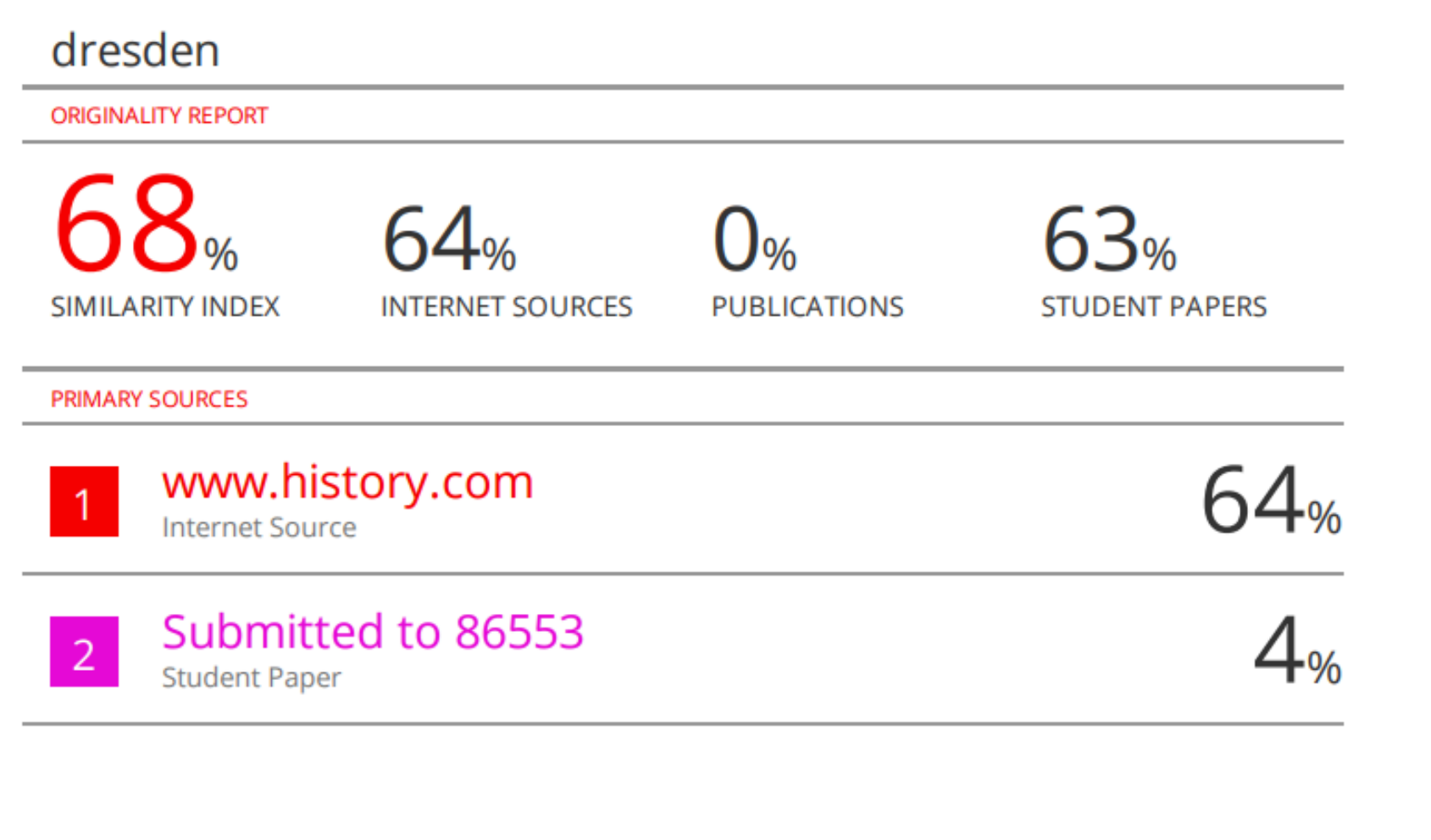
The Allies never overtly admitted that they were engaged in saturation bombing; specific military targets were announced in relation to every attack. However, it was but a veneer, and few mourned the destruction of German cities that built the weapons and bred the soldiers that by 1945 had killed more than 10 million Allied soldiers and even more civilians. The firebombing of Dresden would prove the exception to this rule.

Figure 2: History.com Essay on Dresden Bombing

I think the bombing of Dresden is controversial, because it was not important to German wartime or industrial center. All enemy industries, not just munitions, were targeted. It is different because the bombs are not just aimed at military bases nor military related places. Instead it bombs an entire place including civilians and troop areas. For an example all enemy industry, not just war mutants, were attacked, and civilians proportions of cities are obliterated along troop areas. Causing wreaking on the German economy it then betraying the German people's moreal, and forcing an early surrender. The purpose was to terrorize the German population by forcing and early surrender, but they were disrupting important lines of communication that would have hired the Soviet offensive. In 1944, Hitler named the world's first long range offensive missile V-1 after vergeltung. The German word for "vengeance" and an expression of his desire to repay Britain for devastating bombardment of Germany. Dresden's contribution to the war effort was minimal compared with other german cities. In February 1945, refugees fleeing the Russian advance in the east took refuge there. As Hitler had thrown much of his surviving forces into defense of Berlin in the north, city defences were minimal, and the Russians would have had little trouble capturing Dresden. It seemed an unlikely target for a major Allied air attack. There is no disputing that the British incendiary attack on the night of February 13 to February 14 was conducted, if not primarily, for the purpose of terrorizing the German population and forcing early surrender.

Figure 3: Student Essay on Dresden

Motivating Example: Turn It In



dresden

ORIGINALITY REPORT

68%	64%	0%	63%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.history.com Internet Source	64%
2	Submitted to 86553 Student Paper	4%

Figure 4: Turnitin Similarity Report

Types of Text Clustering

1. **Single Membership Models**:

2. **Mixed Membership Models**:

Types of Text Clustering

1. **Single Membership Models**:
    - Each document assigned to one cluster
    - Example: Doc1 is about the Sorcerer's Stone; Doc4 is about the Goblet of Fire
2. **Mixed Membership Models**:

Types of Text Clustering

1. **Single Membership Models**:

2. **Mixed Membership Models**:
   - Each document assigned to multiple cluster
   - Interested in $P(\text{topic} \mid \text{document})$
   - Example: Doc1 and Doc7 are about the hero's journey; Doc4, Doc5, and Doc6 are about social injustice

Texts and Geometry

Consider a document term matrix:

$$X = \begin{pmatrix} & Word1 & Word2 & Word3 & \dots & WordP \\ Doc1 & 1 & 0 & 0 & \dots & 0 \\ Doc2 & 0 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ DocN & 0 & 0 & 1 & \dots & 3 \end{pmatrix}$$

Texts and Geometry

Consider a document term matrix:

$$X = \begin{pmatrix} & Word1 & Word2 & Word3 & \dots & WordP \\ Doc1 & 1 & 0 & 0 & \dots & 0 \\ Doc2 & 0 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ DocN & 0 & 0 & 1 & \dots & 3 \end{pmatrix}$$

By transforming text into a word count matrix, we represent each document as point in multi-dimensional space:

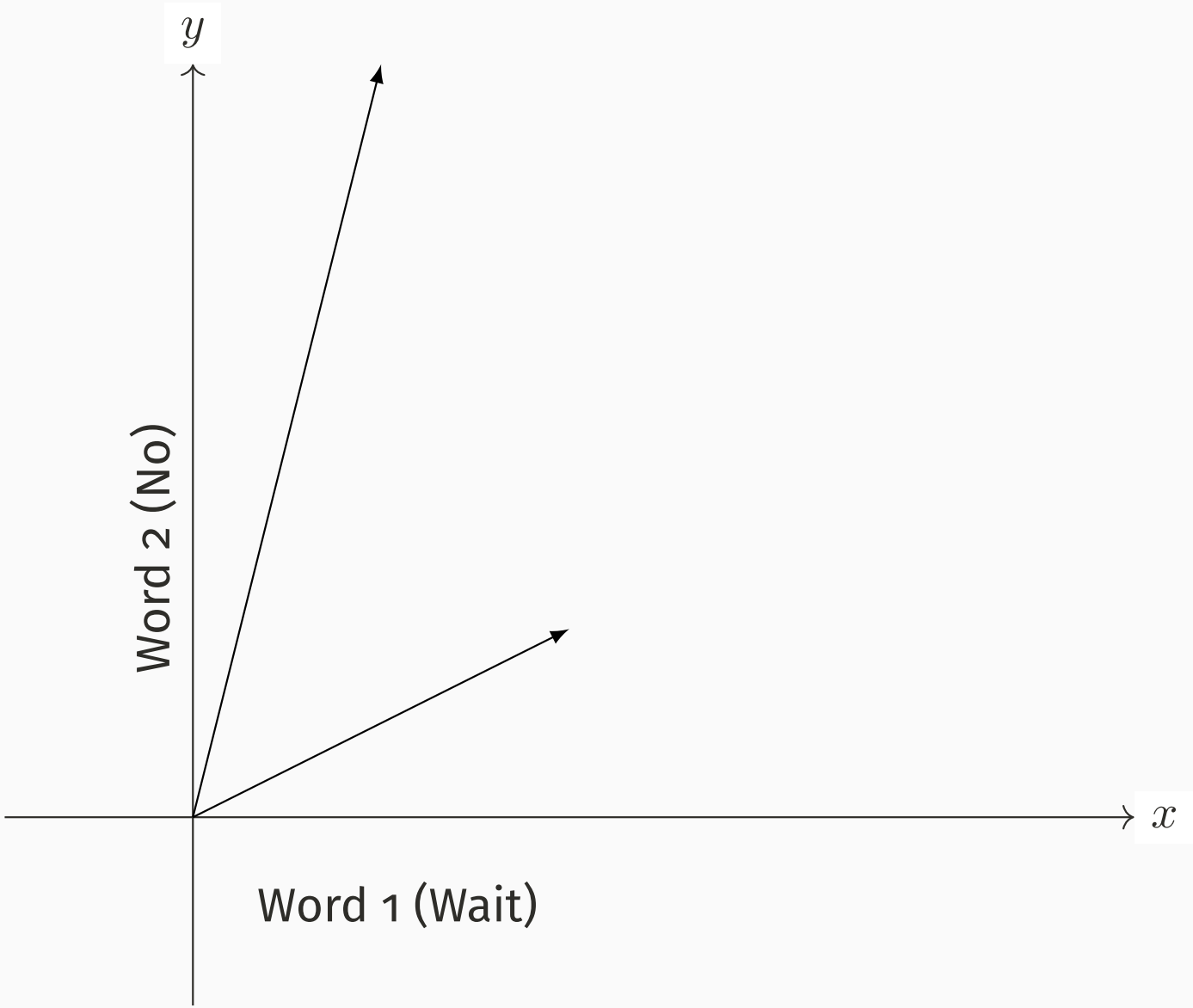
- Provides a geometry
- natural notions of distance and similarity
- apply linear algebra to calculate distance mathematically

Example: Comparing documents

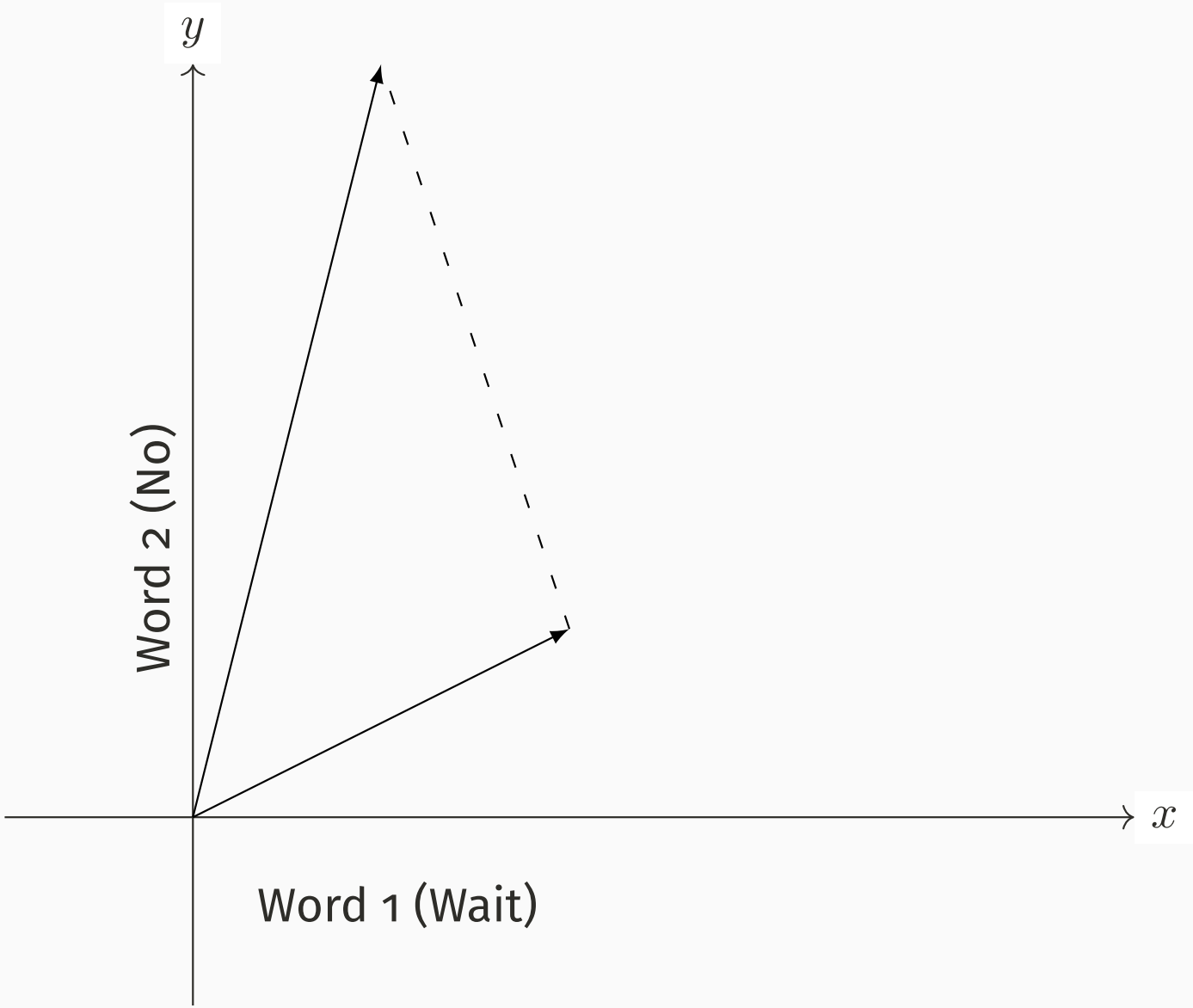
Document 1: “Wait? No wait.” → (2 Waits, 1 No) → (2, 1)

Document 2: “No, wait! No, no, no! → (1 Wait, 4 No) → (1,4)

Example: Comparing documents



Example: Comparing documents



Recap: Euclidean Distance

We can measure the similarity of language based on Euclidean distance:

$$d\left(x_{i'j}, x_{ij}\right) = \sqrt{\sum_{i=1}^{n}\left(x_{ij} - x_{i'j}\right)^2}$$

- When distance is small, observation pairs (language vectors) are more similar
- When distance is larger, observation pairs (language vectors) are more dissimilar

Recap: Euclidean Distance

Document 1: $\mathbf{X}_1 = (2, 1)$

Document 2: $\mathbf{X}_2 = (1, 4)$

$$d(\textbf{doc1, doc2})) = \sqrt{(1-2)^2 + (4-1)^2} = \sqrt{10} = 3.16$$

Limits to Euclidean Distance

Euclidean distance depends on document length.

Example:
Document 1: “Wait? No wait.” → (2, 1)

Document 2: “No, wait! No, no, no! → (1,4)

Document 3: “Wait? No wait. Wait, wait, no! …” → (4, 2)

$$d(\text{doc2}, \text{doc3})) = \sqrt{(4-1)^2 + (2-4)^2} = \sqrt{13} = 3.60$$

Limits to Euclidean Distance

Euclidean distance depends on document length.

Example:
Document 1: “Wait? No wait.” → (2, 1)

Document 2: “No, wait! No, no, no! → (1,4)

Document 3: “Wait? No wait. Wait, wait, no! …” → (4, 2)

$$d(\text{doc2}, \text{doc3})) = \sqrt{(4-1)^2 + (2-4)^2} = \sqrt{13} = 3.60$$

This makes document 3 seem more different than document 1. But it’s the same content!

Limits to Euclidean Distance

Euclidean distance depends on document length.

Example:
Document 1: “Wait? No wait.” → (2, 1)

Document 2: “No, wait! No, no, no! → (1,4)

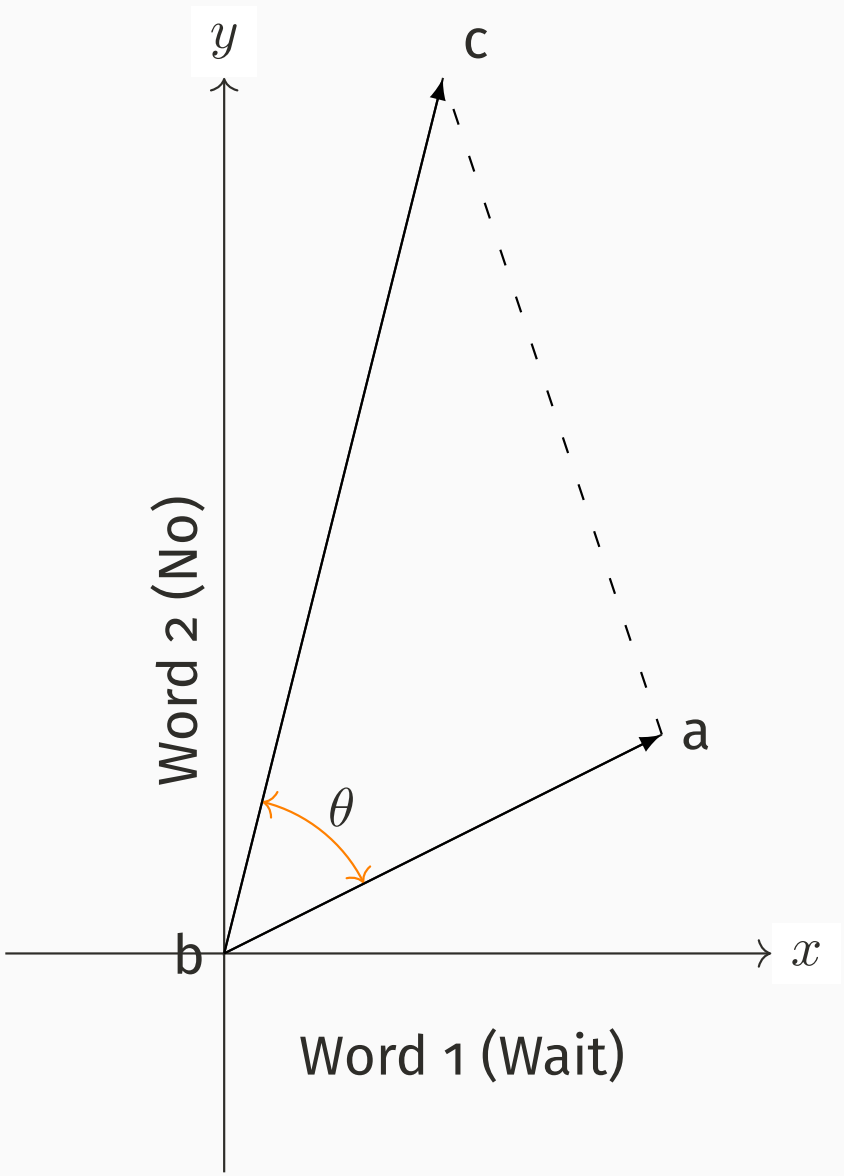
Document 3: “Wait? No wait. Wait, wait, no! ...” → (4, 2)

$$d(\text{doc2}, \text{doc3})) = \sqrt{(4-1)^2 + (2-4)^2} = \sqrt{13} = 3.60$$

This makes document 3 seem more different than document 1. But it’s the same content!

Solution: **Cosine similarity**

Cosine Similarity



- Cosine Similarity
- Adjusts for document length
- Measures **cosine of the angle** θ between vectors
- Measure of similarity rather than distance between 0, 1
- Convert to distance (or dissimilarity): $1 - cos(\theta)$

Clustering Documents

Procedure:

- Measure similarity different documents
- Define number of clusters
- Apply k-means clustering to partition documents

K-means clustering is **single membership** because it considers each document to fall under one topic.

Recall: K-Means Clustering

Main Idea: Partition observations into pre-set number of clusters

Recall: K-Means Clustering

Main Idea: Partition observations into pre-set number of clusters

- K is the number of clusters and must be fixed in advance
- Goal of this method is to maximize the similarity of samples within each cluster
- Good cluster has smallest within-cluster variation
- $W(C_l)$ is measure of similarity between pairs of observations

$$min_{C_1,\ldots,C_k} \sum_{l=1}^{K} W(C_l)$$

- Distance$^2(x_i, x_j)$ is Euclidean distance between observations
- Cluster centroid is mean value of observations (μ)

K-Means Clustering Example

Goal: Cluster the following texts:

1. I like to eat broccoli and bananas
2. I ate a banana smoothie for breakfast
3. Hamsters and kittens are cute
4. My sister adopted a kitten

K-Means Clustering Example

Inputs

- Document Term Matrix

Doc	adopt	banana	breakfast	broccoli	cute	ate	hamster	kitten	like	smoothie
1	0	1	0	1	0	1	0	0	1	0
2	0	1	1	0	0	1	0	0	0	1
3	0	0	0	0	1	0	1	1	0	0
4	1	0	0	0	1	0	0	1	0	0

K-Means Clustering Example

Inputs

- Document Term Matrix

Doc	adopt	banana	breakfast	broccoli	cute	ate	hamster	kitten	like	smoothie
1	0	1	0	1	0	1	0	0	1	0
2	0	1	1	0	0	1	0	0	0	1
3	0	0	0	0	1	0	1	1	0	0
4	1	0	0	0	1	0	0	1	0	0

- $K = 2$ (food and pets)

K-Means Clustering Example

Outputs

- Go through each word and assign p(word | topic) for k clusters

K-Means Clustering Example

Outputs

- Go through each word and assign p(word | topic) for k clusters
- μ_k: Cluster means/centroids
  - Topic A centroid centered around ‘ate’, ‘banana, breakfast’, ‘broccoli’
  - Topic B centroid centered around ‘adopt’, ‘cute’, ‘hamster’
  -
- K-means algorithm will assign each observation to the cluster with the closest mean

K-Means Clustering Example

Outputs

- Go through each word and assign p(word | topic) for k clusters
- μ_k: Cluster means/centroids
  - Topic A centroid centered around ‘ate’, ‘banana, breakfast’, ‘broccoli’
  - Topic B centroid centered around ‘adopt’, ‘cute’, ‘hamster’
  -
- K-means algorithm will assign each observation to the cluster with the closest mean
- C_k Cluster assignment:
  - $C_1 : [\text{Doc}1, \text{Doc}2]$
  - $C_2 : [\text{Doc}3, \text{Doc}4]$

Topic Modeling

What is Topic Modeling?

- Unsupervised algorithm
- Assigns corpus elements to substantively meaningful categories or topics using the statistical correlations between words

What is Topic Modeling?

- Unsupervised algorithm
- Assigns corpus elements to substantively meaningful categories or topics using the statistical correlations between words
- **Mixed membership:** each document is a **mixture** of different topics

What is Topic Modeling?

- Unsupervised algorithm
- Assigns corpus elements to substantively meaningful categories or topics using the statistical correlations between words
- **Mixed membership:** each document is a **mixture** of different topics
- Uses **Latent Dirichlet Allocation** algorithm

Latent Dirichlet Allocation

- Initially randomly assign word to k topic
- For each document, go through each word and compute:

Latent Dirichlet Allocation

- Initially randomly assign word to k topic
- For each document, go through each word and compute:
  - $P(\textbf{word } w \mid \textbf{topic } t)$: proportion of assignments to topic t over all documents that come from this word w. Tries to capture how many documents are in topic t because of word w.
  - $P(\textbf{topic } t \mid \textbf{document } i)$: proportion of words in document i that are assigned to topic t. If a lot of words from document belongs to t, it is more probable that word w belongs to t.

Latent Dirichlet Allocation

- Initially randomly assign word to k topic
- For each document, go through each word and compute:
  - $P(\textbf{word } w \mid \textbf{topic } t)$: proportion of assignments to topic t over all documents that come from this word w. Tries to capture how many documents are in topic t because of word w.
  - $P(\textbf{topic } t \mid \textbf{document } i)$: proportion of words in document i that are assigned to topic t. If a lot of words from document belongs to t, it is more probable that word w belongs to t.

Procedure

- Apply Bayes to calculate:

$$\pi_k = P(\text{word with topic } t)$$

$$P(\text{word with topic } t) = P(\text{topic } t \mid \text{document } i) \times P(\text{word } w \mid \text{topic } t)$$

Procedure

- Apply Bayes to calculate:

$$\pi_k = P(\text{word with topic } t)$$

$$P(\text{word with topic } t) = P(\text{topic } t \mid \text{document } i) \times P(\text{word } w \mid \text{topic } t)$$

- Assign probability score π_k as topic distribution over words

Procedure

- Apply Bayes to calculate:

$$\pi_k = P(\text{word with topic } t)$$

$$P(\text{word with topic } t) = P(\text{topic } t \mid \text{document } i) \times P(\text{word } w \mid \text{topic } t)$$

- Assign probability score π_k as topic distribution over words

- Assign probability score θ_i as document distribution over words ($P(\text{document i} \mid \text{topic } t)$)
  - θ_i tells us probability document i belongs to a given topic cluster t

Key Modeling Considerations: Preprocessing

1. Topic models are sensitive to feature selection
2. Common to remove sparse words, stop words, etc., but can affect modeling

Key Modeling Considerations: Hyperparameters

1. Number of K topics

2. Alpha α Parameter

3. Beta β Parameter

Key Modeling Considerations: Hyperparameters

1. Number of K topics
    - Identify optimal K using cross-validation
    - Different K produce different topic **coherence** scores
2. Alpha α Parameter

3. Beta β Parameter

Key Modeling Considerations: Hyperparameters

1. Number of K topics

2. Alpha α Parameter
    - Document Topic Density
    - Higher α means document has more potential topics
3. Beta β Parameter

Key Modeling Considerations: Hyperparameters

1. Number of K topics

2. Alpha α Parameter

3. Beta β Parameter
    - Topic Word Density
    - Higher β means topic has larger number of words

Key Modeling Considerations: Hyperparameters

1. Number of K topics

2. Alpha α Parameter

3. Beta β Parameter

Rule of Thumb: $K = 10$ or $K = 20$, $\alpha = 0.1$, $\beta = 0.05$

What makes a good topic model?

Rule of thumb: A “good’ topic model is one where...

- Topics are semantically interpretable
- Topics are substantively relevant
- Topics have relatively good **coherence scores**

Main Takeaway: Use your social science expertise to validate topic models.

LDA Modeling Example

Goal: topic model the following documents

- I like to eat broccoli and bananas
- I ate a banana smoothie for breakfast
- Hamsters and kittens are cute
- My sister adopted a kitten yesterday
- Look at this hamster munching on broccoli

LDA Modeling Example

Inputs

- Document Term Matrix
- $K = 2$ (food and pets)

LDA Modeling Example

Outputs

- Topic distribution over words π_k

Topic	broccoli	banana	breakfast	kitten	cute	hamster	like	yesterday	Total
A	0.30	0.25	0.20	0.01	0.01	0.01	0.12	0.10	1
B	0.01	0.01	0.01	0.35	0.24	0.25	0.08	0.05	1

LDA Modeling Example

Outputs

- Topic distribution over words π_k

Topic	broccoli	banana	breakfast	kitten	cute	hamster	like	yesterday	Total
A	0.30	0.25	0.20	0.01	0.01	0.01	0.12	0.10	1
B	0.01	0.01	0.01	0.35	0.24	0.25	0.08	0.05	1

- Document Distribution over topics θ_i

Document	Topic A Weight	Topic B Weight	Total
1	0.99	0.01	1
2	0.99	0.01	1
3	0.01	0.99	1
4	0.01	0.99	1
5	0.60	0.40	1

Conclusion

- Clustering uses Euclidean distance or cosine similarity to measure similarity
- K-Means clustering for documents may work. or may not.
- LDA topic modeling powerful tool to group documents together
- You now have the basic tools for ML!

Course Wrap-Up

ML is powerful tool for prediction problems

Main Idea: ML is a form of artificial intelligence increasingly used in social science to solve complex prediction problems. It involves a set of computer algorithms which 'learn' patterns in existing data to assist in prediction and inference.

What We've Covered

1. Supervised Learning

2. Unsupervised Learning

What We've Covered

1. Supervised Learning
    - 1.1 Regression and Classification
    - 1.2 Cross-Validation
    - 1.3 Regularization and Feature Selection
    - 1.4 Random Forests, Boosting, Bagging
2. Unsupervised Learning

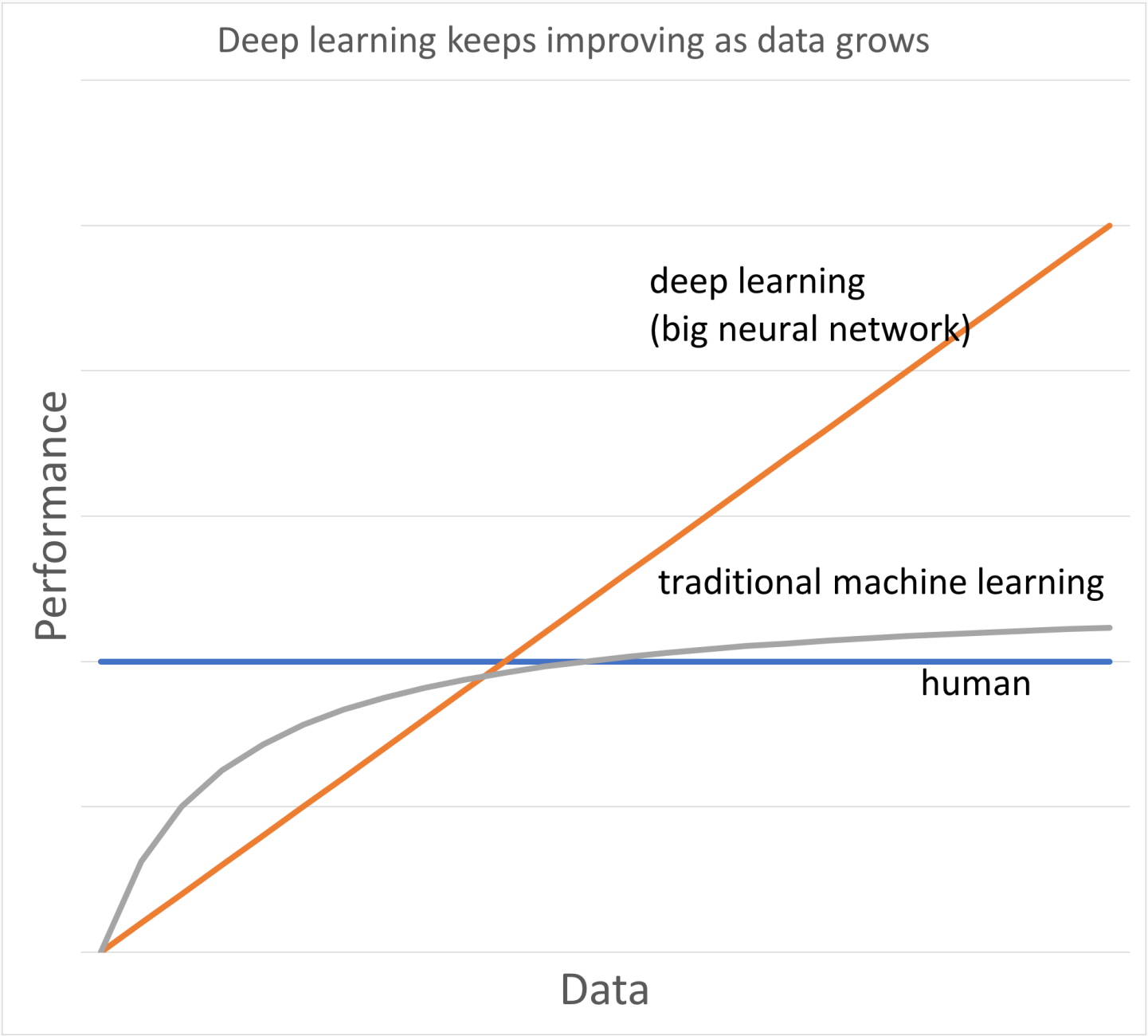
What We've Covered

1. Supervised Learning
    - 1.1 Regression and Classification
    - 1.2 Cross-Validation
    - 1.3 Regularization and Feature Selection
    - 1.4 Random Forests, Boosting, Bagging
2. Unsupervised Learning
    - 2.1 Principal Component Analysis
    - 2.2 Clustering
    - 2.3 Topic Models (Text Analysis)

ML is still a black box sometimes...

- No p-values or hypothesis testing
- No estimable f
- Non-parametric modeling
- Lots of hyperparamaters

...but it can perform really well



Social Science is Key to ML

Black box means we need social science expertise (more than ever) to...

- ask the right question
- assemble the right inputs
- validate the outputs
- refine and reiterate the model

You've got this!