

# Document Analysis

Jaehyun Park   Ahmed Bou-Rabee   Stephen Boyd

EE103  
Stanford University

August 20, 2014

# Outline

Word count vectors

Similarity measures

Topic discovery

Word clustering

Regression

Classification

## Word count vectors

- ▶ fix a dictionary containing  $n$  different words
- ▶ associate *word count vector*  $a$  with a document
- ▶  $a_i$  = number of times word  $i$  appears in the document
- ▶  $h = a/\mathbf{1}^T a$  is *word frequency* or *histogram* vector
- ▶ other normalizations and representations (e.g., tf-idf, bi-grams) are sometimes used

# Pre-processing

documents are *pre-processed* before words are counted

- ▶ stemming: remove endings from words
  - cat, cats, catty → cat
  - stemmer, stemmed, stemming → stem
- ▶ filter (remove) 'stop' words
  - short words: the, is, at
  - most common words: what, this, how
  - extremely uncommon words

## Example

Dictionary	Doc A	Doc B	Doc C
bankrupt	0	3	0
baseball	0	0	3
bat	3	0	3
harry	3	0	0
homerun	0	1	2
legendary	4	1	1
magic	3	0	1
sport	0	0	4
stock	0	3	0
⋮	⋮	⋮	⋮

can you guess the topics of each document?

## Document-term matrix

- ▶ we have a *corpus* (collection) of  $N$  documents, with word count  $n$ -vectors  $a_1, \dots, a_N$
- ▶ *document-term* matrix is  $N \times n$  matrix  $A$ , with  $A_{ij}$  = number of times word  $j$  appears in document  $i$
- ▶ rows of  $A$  are  $a_1^T, \dots, a_N^T$
- ▶  $j$ th column of  $A$  shows occurrences of word  $j$  across corpus

# Outline

Word count vectors

Similarity measures

Topic discovery

Word clustering

Regression

Classification

## Similarity measures

- ▶ two documents, with word count vectors  $a_1, a_2$ , histogram vectors  $h_1, h_2$
- ▶ *distance measure* (of dissimilarity):  $\|h_1 - h_2\|$
- ▶ *angle measure* (of dissimilarity):  $\angle(a_1, a_2) = \angle(h_1, h_2)$
- ▶ we expect these to be small when the documents have the same topics, genre, or author, and larger otherwise



## Example

- ▶ 4 chapters with histograms  $h_1, h_2, h_3, h_4$
- ▶ dictionary is 1000 most common words

**Harry Potter 1.** Harry did the best he could, trying to ignore the stabbing pains in his forehead, which had been bothering him ever since his trip into the forest . . .

**Harry Potter 2.** "Severus?" Quirrell laughed, and it wasn't his usual quivering treble, either, but cold and sharp . . .

**Foundations 1.** Gaal Dornick, using nonmathematical concepts, has defined psychohistory to be that branch of mathematics which deals with the reactions of human conglomerates to fixed social and economic stimuli . . .

**Foundations 2.** The trial (Gaal supposed it to be one, though it bore little resemblance legalistically to the elaborate trial techniques Gaal had read of) had not lasted long . . .

## Example

- $\|h_i - h_j\| (\times 100)$

	HP1	HP2	FO1	FO2
HP1	0	0.4	1.4	1.3
HP2		0	1.4	1.2
FO1			0	0.8
FO2				0

- $\angle(h_i, h_j)$  (in degrees)

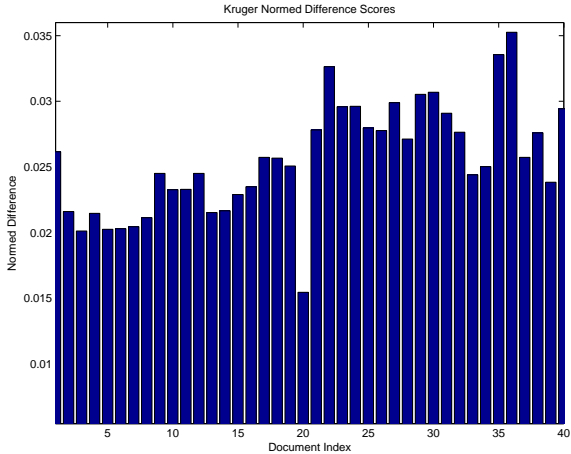
	HP1	HP2	FO1	FO2
HP1	0	40.8	84.7	84.1
HP2		0	84.1	82.0
FO1			0	74.1
FO2				0

## Example

- ▶ 40 documents, with word count histograms  $h_1, \dots, h_{40}$ 
  - 1–20 are news articles
  - 20 is by Paul Krugman
  - 21–40 are Harry Potter excerpts
- ▶ another article by Paul Krugman, with histogram  $b$
- ▶ dictionary capped at 1000 words
- ▶ let's look at  $\angle(h_i, b)$  and  $\|h_i - b\|$ ,  $i = 1, \dots, 40$

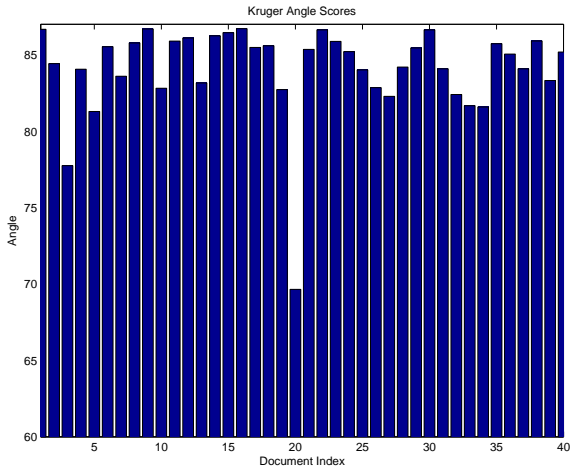
## Example

$$\|h_i - b\|, i = 1, \dots, 40$$



## Example

$$\angle(h_i, b), i = 1, \dots, 40$$



# Outline

Word count vectors

Similarity measures

Topic discovery

Word clustering

Regression

Classification

## $k$ -means on histogram vectors

- ▶ start with corpus of  $N$  documents with histograms  $h_1, \dots, h_N$
- ▶ use  $k$ -means algorithm to cluster into  $k$  groups of documents
- ▶ groups usually have similar topics, genre, or author
- ▶ this is sometimes called (automatic) *topic discovery*
- ▶ centroids  $z_1, \dots, z_k$  are also histograms

## Example

- ▶ corpus of 555 documents, dictionary capped at 1000 most common words
  - 185 Harry Potter excerpts
  - 185 education articles
  - 185 sports articles
- ▶ use  $k$ -means with  $k = 3$ ; best of 10 random initializations
- ▶ results:

Cluster	Sports	Education	Harry Potter
1	183	39	19
2	2	146	0
3	0	0	166



## Example

words associated largest coefficients of centroid vectors:

Cluster 1	player	year	league	football	team
Cluster 2	student	education	school	university	college
Cluster 3	harry	hermione	ron	eye	said

## Example

- ▶ let's use our three cluster centroids to classify *new* documents:
  - 15 Harry Potter excerpts
  - 15 education articles
  - 15 sports articles
- ▶ results (in a *confusion matrix* or table):

predicted ↓    true →	Sports	Education	Harry Potter
Sports	15	0	0
Education	0	15	0
Harry Potter	0	0	15

# Outline

Word count vectors

Similarity measures

Topic discovery

**Word clustering**

Regression

Classification

## Document count vectors

- ▶ we have a corpus of  $N$  documents
- ▶ associate with a word its *document count vector*  $b$
- ▶  $b_i$  = number of times word appears in document  $i$
- ▶  $b_i$  are columns of document-term matrix  $A$   
(word count vectors are rows of  $A$ )
- ▶ normalized document count (histogram) vector is  $g = b/\mathbf{1}^T b$
- ▶ words that appear in similar ways across the corpus have close document count or histogram vectors

## Word clustering

- ▶ use  $k$ -means algorithm on histograms  $g_i$  to partition words into  $k$  groups
- ▶ words in same cluster tend to co-appear in the same documents in the corpus

## Example

- ▶ same example as above (555 documents, 1000 words)
- ▶ run  $k$ -means word clustering with  $k = 50$
- ▶ words from some of the clusters:

investigate	charge	lawsuit	allege	title
concuss	injury	draft	retire	brain
gryffindor	firebolt	slytherin	broom	penalty
world	team	soccer	game	brazil

# Outline

Word count vectors

Similarity measures

Topic discovery

Word clustering

Regression

Classification

## Regression model

- ▶ goal: predict a number  $y$  (e.g., grade, score, rating) from a document's word count vector  $a$
- ▶ regression model:

$$\hat{y} = w^T a + v$$

- $\hat{y}$  is predicted value of  $y$
  - $a$  is a document word count vector
  - $w$  is weight vector;  $v$  is offset
  - we are to choose  $w$  and  $v$  so  $y \approx \hat{y}$
- ▶ we have a *training set* of  $N$  documents and their ('true')  $y$  values

$$(a_1, y_1), \dots, (a_N, y_N)$$



## Regression

- ▶ want  $w, v$  for which  $y_i \approx \hat{y}_i = w^T a_i + v$
- ▶ we'll judge regression prediction error via its RMS value

$$\left( \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \right)^{1/2}$$

- ▶ choose  $w, v$  to minimize

$$\sum_{i=1}^N (\hat{y}_i - y_i)^2 + \lambda \|w\|^2 = \|Aw + v\mathbf{1} - y\|^2 + \lambda \|w\|^2$$

- $\lambda > 0$  is regularization parameter
- first term is RMS prediction error (squared, times  $N$ )

# Validation

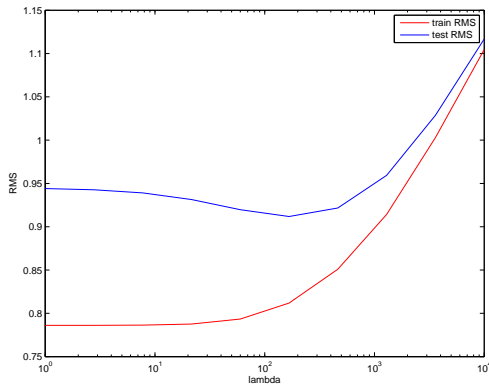
- ▶ we are interested on  $w, v$  that give good predictions on *new, unseen* documents
- ▶ so we *test* or *validate*  $w, v$  on a different set of documents, the *test set*
- ▶ we choose  $\lambda$  so that the RMS prediction error *on test set* is small

## Example

- ▶ set of 8884 Yelp reviews with at least 50 words
- ▶ dictionary is 1000 most common words, e.g., place, great, food, good
- ▶ reviews  $y_i$  have values in  $\{1, 2, 3, 4, 5\}$ 
  - $\text{avg}(y) = 3.56$ ,  $\text{std}(y) = 1.28$
  - so always guessing  $\hat{y} = 3.56$  gives RMS error 1.28
- ▶ divide documents into training set (6218) and test set (2666)

## Example

RMS error versus  $\lambda$



using  $\lambda = 150$  gives RMS test error  $\approx 0.92$

## Example

weights with largest values

word	weight
perfect	0.196
best	0.178
five	0.164
fantastic	0.160
amazing	0.158
awesome	0.157
⋮	⋮
terrible	-0.231
rude	-0.280
horrible	-0.281
worst	-0.284
bland	-0.298

## Example

- ▶ now let's take prediction  $\hat{y}$  and round it to  $\{1, 2, 3, 4, 5\}$
- ▶ results:

Prediction error	Train	Test	Predicting 4
perfect	47%	41%	33.3%
off by one	47%	50%	44%
off by two	5.6%	8.9%	12%
off by three	0.20%	0.53%	9.7%
off by four	0%	0%	0%

## Example

confusion matrix on training set

predicted ↓    true →	1	2	3	4	5
1	95	308	183	10	0
2	34	261	441	44	0
3	2	94	594	327	10
4	0	10	403	1392	250
5	0	2	104	1072	582

## Example

confusion matrix on test set

predicted ↓ true →	1	2	3	4	5
1	37	108	111	10	0
2	14	89	202	35	0
3	2	46	224	162	4
4	0	7	224	554	135
5	0	4	76	429	193



# Outline

Word count vectors

Similarity measures

Topic discovery

Word clustering

Regression

**Classification**

# Document classification

- ▶ documents have *labels* from a finite set, e.g.,
  - *email* or *spam*
  - *excerpt from Harry Potter* or *not*
  - *about sports* or *news* or *neither*
- ▶ divides documents into *classes*
- ▶ we'll focus on binary case, with two labels
- ▶ *document classification*: given word count vector  $a$ , guess which class the document is in
- ▶ judge classification performance by error rate on test set

## Least squares classification

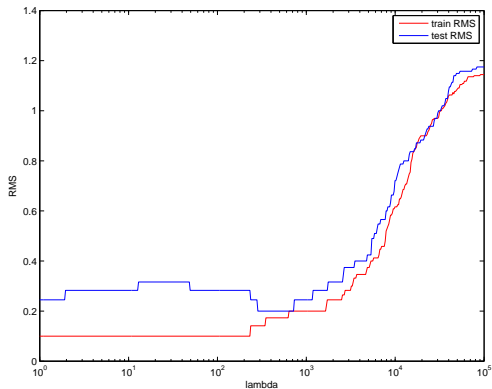
- ▶ we use label  $y_i = 1$  for one class and  $y_i = -1$  for the other
- ▶ find regression model  $\tilde{y} = w^T a + v$
- ▶ guess (classify) document using  $\hat{y} = \mathbf{sign}(\tilde{y})$
- ▶ choose regularization parameter  $\lambda$  by error rate on test set

## Example

- ▶ same corpus of 555 documents: sports, education, and Harry Potter
- ▶ split into training set (370 documents) and test set (185 documents)
- ▶ predict *sports* articles versus *not sports*
- ▶ label sports articles with 1, others  $-1$

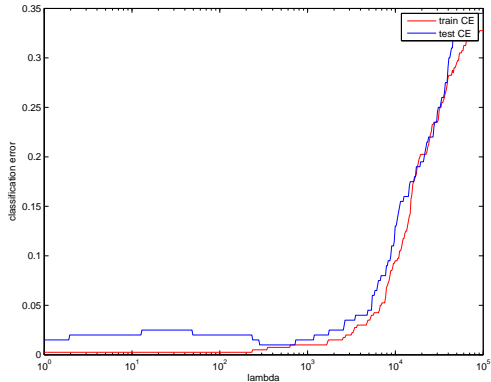
## Example

RMS prediction error versus  $\lambda$



## Example

classification error versus  $\lambda$



choosing  $\lambda = 285$  gives test set error rate around 2%

## Example

weights with largest values

word	weight
olympics	0.0532
play	0.0491
football	0.0464
player	0.0402
final	0.0359
committee	0.0341
⋮	⋮
school	-0.0230
get	-0.0249
read	-0.0269
campus	-0.0320
harry	-0.0360