livelessons
video instruction from technology experts

Lesson 3: Your First Spark Application

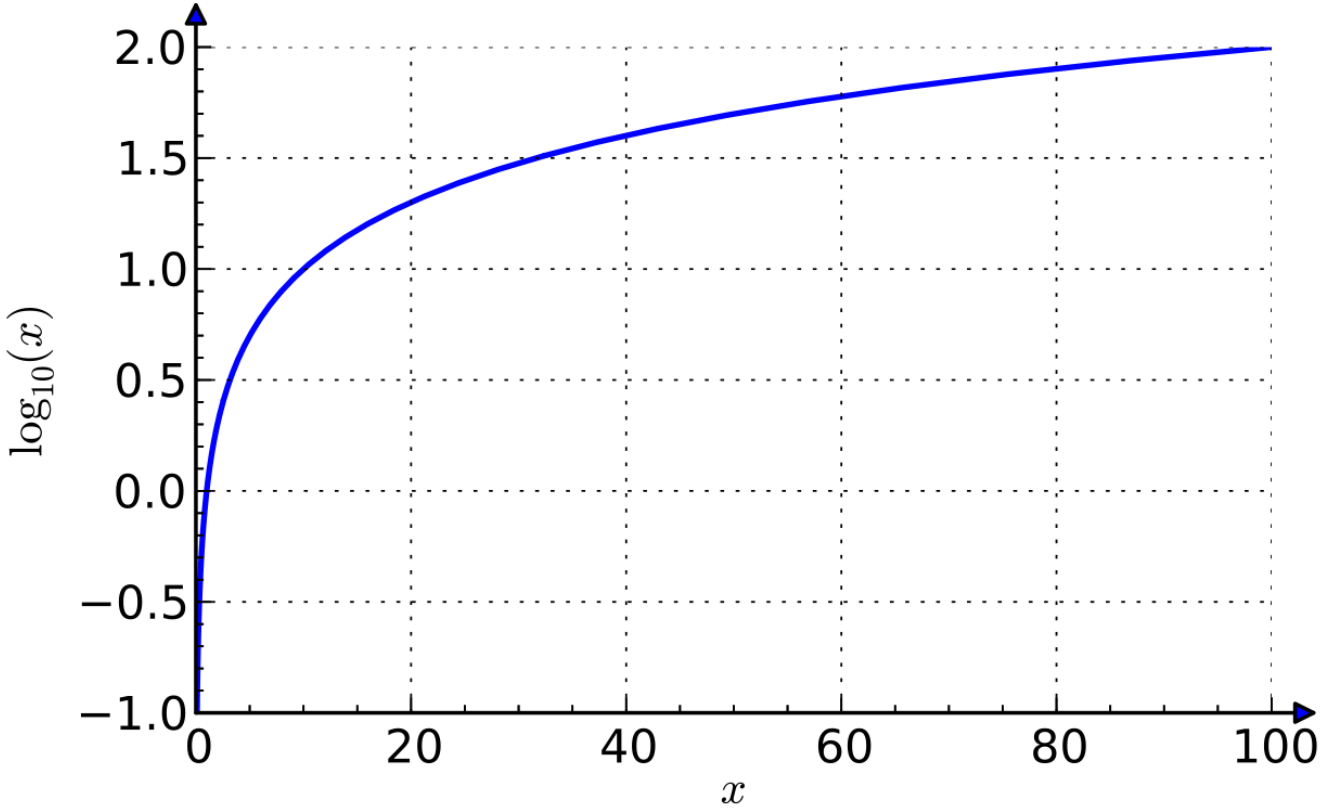
3.8 Summarization with tf-idf

TF-IDF

- Measure of discriminatory power of word (feature)
- Highest when term occurs many times in a small number of documents
- Lowest when term occurs few times in document or many times in corpus
- Useful for information retrieval (queries) and keyword extraction (among other things)

$$tf(t,d) = \frac{f_d(t)}{|d|} \qquad idf(t,D) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

TF-IDF



TF-IDF

Most Common

```
idf[:50]
```

```
[(u'students', 0.014067384597943282),
 (u'I', 0.15305316750494943),
 (u'school', 0.17010493952495984),
 (u'My', 0.3397655206814591),
 (u'The', 0.4149133167820112),
 (u'help', 0.4188088461791251),
 (u'classroom', 0.5361023876769617),
 (u'learning', 0.5748186189046272),
 (u'need', 0.5820538952580256),
 (u'They', 0.5941434194555928),
 (u'learn', 0.6187002265438729),
 (u'able', 0.7452815794748304),
 (u'use', 0.7494117483916651),
 (u"''", 0.755060153205684),
 (u'We', 0.7552806889430156),
 (u'This', 0.7749201702459683),
 (u'class', 0.7913652190100225),
 (u'would', 0.8149828303863013),
 (u'make', 0.8239845109910496),
 (u'many', 0.8273389184929604),
```

Least Common

```
idf[:-50:-1]
```

```
[(u'beer', 10.378594025517652),
 (u'worsen', 10.378594025517652),
 (u'theorist', 10.378594025517652),
 (u'Beneath', 10.378594025517652),
 (u'.how', 10.378594025517652),
 (u'unchanged', 10.378594025517652),
 (u'lessons-', 10.378594025517652),
 (u'on-stage', 10.378594025517652),
 (u'interactiveness', 10.378594025517652),
 (u'GoogleEarth', 10.378594025517652),
 (u'peers\u2019', 10.378594025517652),
 (u'pre-schools', 10.378594025517652),
 (u'PER', 10.378594025517652),
 (u'Davies', 10.378594025517652),
 (u'Spalding', 10.378594025517652),
 (u'7:15am', 10.378594025517652),
 (u'geneticists', 10.378594025517652),
 (u'20-year-old', 10.378594025517652),
 (u'inservice', 10.378594025517652),
 (u'Conquering', 10.378594025517652),
```

```
top_n = 10
summary = bag_of_words.map(lambda x: map(lambda idx: broadcast_idf.value[idx][0], np.argsort(x)[::-1][:top_n]))
```

```
summary.take(15)
```

```
[[u'science',
  u'Outreach',
  u'17-21',
  u'one-year',
  u'resource',
  u'magazine',
  u'periodical',
  u'http',
  u'York',
  u'competency'],
 [u'Worlds',
  u'Hidden',
  u'microscopes',
  u'cell',
  u'stressing',
  u'6th',
  u'single',
  u'cluster',
  u'intense',
  u'organisms'],
 [u'corner',
  u'Harlem',
  u'calming',
  u'rug',
  u'soft',
  u'world.In',
  u'began',
  u'populate',
  u'putting',
  u'stain'],
 [u'Music',
  u'music',
  u'Appreciation',
```

Summarization

Scale Up

```
import string
import json
import pickle as pkl
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import pyspark as ps
from collections import Counter
import numpy as np
```

```
sc = ps.SparkContext()
```

```
essay_rdd = sc.textFile('s3n://xxxx:xxxx@galvanize-example-data/donors_choose/essay_parse.json')
essay_rdd.first()
```

```
u'{"essay":"\\"One of my classes this year has been an 11th grade  English class.  These students will be taking the SAT next y
ear and I know that they are not ready to do well.  \\r\\\\n\\r\\\\nUnlike students in more well-financed school districts, the
y have not had access to special SAT classes or tutors.  I have been so focused on completing the curriculum and helping them t
o pass the Regents that I have not had time to do SAT preparation. They do not come from families with extra cash to pay for cl
asses on their own. \\r\\\\n\\r\\\\nI would love to be able to get them a good SAT preparation book before the end of the schoo
l term.  In this way, I can get them started so that they can review the book on their own over the summer.\\r\\\\n\\""}'
```

```
essay_rdd.count()
```

```
771929
```

Review

- We need to represent text as vectors to model documents
- The Bag-of-words model uses word counts (tf-idf improves on this)
- In vector space, we can compare documents using linear algebra
- Spark provides feature transformers to handle text input

Next Up: Implementing an Algorithm