LAB #5: PROJECT RELATED SKILLS

CS 109A, STAT 121A, AC 209A: Data Science

Fall 2016

Harvard University

FUNCTIONAL PROGRAMMING ELEMENTS OF PYTHON

LAMBDA FUNCTIONS

For defining short functions, we need not use the `def`...`return` protocol. Instead, we can encode a function "anonymously" using the following format

```
lambda <input>: <do something to input and return>
```

For example,

```
lambda x: x**2
```

encodes a function which takes as input some value stored in the parameter `x` and **returns** the value in `x` squared.

FUNCTIONS AS PARAMETERS

Functions are first-class objects in Python. This means that if you want to pass a function to another function, you can just treat it as any other object.

```
In [6]: def apply_it(f, x):
            return f(x)

        def square(x):
            return x**2

In [7]: x = 2
        apply_it(square, x)

Out[7]: 4
```

FUNCTIONS AS PARAMETERS

Functions are first-class objects in Python. This means that if you want to pass a function to another function, you can just treat it as any other object.

```
In [6]: def apply_it(f, x):
            return f(x)

In [8]: x = 2
        apply_it(lambda x: x**2, x)

Out[8]: 4
```

FUNCTIONS AS PARAMETERS

Functions are first-class objects in Python. This means that if you want to pass a function to another function, you can just treat it as any other object.

```
In [15]: import pandas as pd

In [16]: series = pd.Series([1, 2, 3, 4])
         series

Out[16]: 0    1
         1    2
         2    3
         3    4
         dtype: int64

In [17]: series.apply(lambda x: x + 1)

Out[17]: 0    2
         1    3
         2    4
         3    5
         dtype: int64
```

TEXT ANALYSIS

REPRESENTING TEXTUAL DATA

Comparing the content of the following two sentences is easy for an English speaking human (clearly both are discussing the same topic, but with different emotional undertone):

1. Linear R3gr3ssion is very very cool!
2. What don't I like it a single bit? Linear regressing!

REPRESENTING TEXTUAL DATA

Comparing the content of the following two sentences is easy for an English speaking human (clearly both are discussing the same topic, but with different emotional undertone):

1. Linear R3gr3ssion is very very cool!
2. What don't I like it a single bit? Linear regressing!

But a computer doesn't understand

- which words are nouns, verbs etc (grammar)
- how to find the topic (word ordering)
- feeling expressed in each sentence (sentiment)

We need to represent the sentences in formats that a computer can easily process and manipulate.

PREPROCESSING

If we're interested in the topics/content of text, we may find many components of English sentences to be uninformative.

1. Word ordering
2. Punctuation
3. Conjugation of verbs (go vs going), declension of nouns (chair vs chairs)
4. Captitalization
5. Words with mostly grammatical functions: prepositions (before, under), articles (the, a, an) etc
6. Pronouns?

These uninformative features of text will only confuse and distract a machine and should be removed.

BAG OF WORDS

After preprocessing our sentences:

1. (S_1) linear regression is very very cool
2. (S_2) what don't like single bit linear regression

We represent text in the format that is most accessible to a computer: numeric. We simply make a vector of the **counts** of the words in each sentence.

	linear	regression	is	very	cool	what	don't	like	single	bit
S_1	1	1	1	1	1	0	0	0	0	0
S_2	1	1	0	0	0	1	1	1	1	1

Turning a piece of text into a vector of word counts is called **Bag of Words**.

Now that our sentences $\mathbf{S_1}$ and $\mathbf{S_2}$ are vectors, we can use a huge number of mathematical tools to compare and contrast them.

For example,

1. (Edit distance) We compare the similarity of $\mathbf{S_1}$ and $\mathbf{S_2}$ by computing
$$\mathrm{sum}(S_1 - S_2)$$
2. (Cosine Similarity) We compare the similarity of $\mathbf{S_1}$ and $\mathbf{S_2}$ by computing
$$\mathrm{cosine\,angle}(S_1, S_2) = \frac{S_1 \cdot S_2}{\|S_1\|\|S_2\|}$$

Based on a chosen similarity measure, we can break up a **corpus of text** into **cluster** of documents that are the most similar to each other.