how to *do* computationally assisted research
digital literacy @ comwell

Kristoffer L Nielbo
knielbo@sdu.dk
knielbo.github.io/

March 22, 2018
```python
class Person(object):
    def __init__(self, name):
        self.name = name
    def says_hello(self):
        print 'Hello, my name is', self.name

class Researcher(Person):
    def __init__(self, title=None, areas=None, **kwargs):
        super(Researcher, self).__init__(**kwargs)
        self.title = title
        self.areas = areas

KLN = Researcher(name = 'Kristoffer L Nielbo',
                 title = 'Associate professor',
                 areas = ['Humanities Computing', 'Culture Analytics', 'eScience'])

KLN.says_hello()
```
evolution of workflows
data
Data objects that are described over a set of (qualitative or quantitative) features

- Fundamental difference between structured data and unstructured* data
- Word processing files, pdfs, emails, social media posts, digital images, video, and audio
- Today > 80% of all data are unstructured
- Unstructured data require expertise in culture, media, linguistic ...
data access and sampling

select (sample*) a set of documents (target data) relevant to your research question from a data collection

>>> online databases and research libraries are excellent resources

- proprietary issues
- data protection acts
- ethical concerns
- availability (e.g., historical sources)

>>> sample requirements

- “all the data”
- balancing and stratification
- bias reduction
we will focus on documents stored locally in a plain text without markup

```html
"""The First Book of Moses, called Genesis

    {1:1} In the beginning God created the heaven and the earth. {1:2}
And the earth was without form, and void; and darkness was upon the
face of the deep. And the Spirit of God moved upon the face of the
waters.

    {1:3} And God said, Let there be light: and there was light. {1:4}
And God saw the light, that it was good: and God divided the light""

BUT with a bit of code everything is possible

```python
import urllib2
from HTMLParser import HTMLParser

class html_parser(HTMLParser):
    def handle_starttag(self, tag, attrs):
        print "start tag:" , tag
    def handle_endtag(self, tag):
        print "end tag :" , tag
    def handle_data(self, self, data):
        print "data :" , data

url = "https://knielbo.github.io/"
response = urllib2.urlopen(url)
webpage = response.read()
parser = html_parser()
parser.feed(webpage)```
preprocessing
proprocessing|language normalization

- To prepare a document we need to parse, slice and split it at the relevant level(s).

- Unstructured data are very noisy, so to increase the signal, we therefore remove irrelevant data through preprocessing.

- A range of text normalization techniques to preprocess the data:
  - Casefolding
  - Removal of non-alphanumeric characters (punctuation, blanks) and numerals
  - Vocabulary pruning
  - Identification of parts of speech
  - Reduction of inflectional forms through stemming and lemmatization
  - Disambiguation
  - Synonym substitution

... one man’s rubbish may be another’s treasure
- normalization by reducing inflected words to their stem, base or root form
- the stem need *not* be identical to the morphological root
- sufficient that related words map to the same stem (stem ≠ valid root)
- search engines treat words with the same stem as synonyms (*conflation*)

**Porter stemming algorithm - step 1a**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSES -&gt; SS</td>
<td>caresses -&gt; caress</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IES -&gt; I</td>
<td>ponies -&gt; poni</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ties</td>
<td>-&gt; ti</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SS -&gt; SS</td>
<td>caress -&gt; caress</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>cats -&gt; cat</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
proprocessing | structuring words

– selecting the right **formalism** for representing a problem over a data set
– many techniques rely on basic probabilistic or geometric properties of the data set

**example**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>1 59</th>
<th>0.073</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>am</td>
<td>1 16</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>and</td>
<td>1 24</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>anywhere</td>
<td>1 1</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>anywhere</td>
<td>1 7</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>you</td>
<td>1 34</td>
<td>0.042</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td></td>
<td>55 804</td>
<td>1.0</td>
</tr>
</tbody>
</table>
example

any collection of $m$ documents can be represented in the vector space model by a document-term matrix of $m$ documents and $n$ terms.

A vector space model is a basic modeling mechanism for a word- or document-space (whether we look at rows or columns).

- a document vector with only one word is collinear to the vocabulary word axis.
- a document vector that does not contain a specific word is orthogonal/perpendicular to the word axis.
- two documents are identical if they contain the same words in a different order (BOW assumption).
analysis
analysis|basic properties

– describe basic properties of the data, e.g., simple distributions and relations
– result in themselves or input to more advanced analysis
– the value depends critically on domain knowledge

beauty lies in simplicity
The joint distribution defines a posterior probability: $P(\theta, z, \phi)$
use posterior to:

**Train on a corpus:** Bayesian inference on $\theta$ and $\phi$
**Train on a new documents $d$:** fix $P(w \mid z)$ to infer $P(z \mid d)$
- Multiple inference algorithms available (expectation-maximization/VEM and Gibbs sampling/GIBBS)

---

**Procedure 1 Generative Model**

1. **choose** $\theta_i \sim \text{Dir}(\alpha)$, $i$ is a document
2. **choose** $\phi_k \sim \text{Dir}(\beta)$, $k$ is a topic
3. **for** each word position **do**
4.  **choose** a topic $z_{ij} \sim \text{Multinomial}(\theta_i)$
5.  **choose** a word $w_{ij} \sim \text{Multinomial}(\phi_{z_{ij}})$
6. **end for**
interpretation and evaluation
evaluation: is our model valid?

\[ \begin{align*}
\text{\rightarrow relevant objects (e.g., ham)} \\
\text{\leftarrow irrelevant objects (e.g., spam)} \\
\circ \text { objects classified with relevant class label}
\end{align*} \]

\begin{enumerate}
  \item \text{ERROR}
  \item \text{CORRECT}
\end{enumerate}

Precision: fraction of retrieved instances that are relevant

\[ P = \frac{TP}{TP + FP} \]

Recall: fraction of relevant instances that are retrieved

\[ R = \frac{TP}{TP + FN} \]

\[ P \text{ and } R \text{ are inversely related. Identify balance through a Precision-Recall curve.} \]
interpretation | what does our model mean?

- philosophers and sinologists have been debating the existence of mind-body dualism in classical Chinese philosophy
- with domain experts, unsupervised learning was used to identify a multi-level dualistic semantic space
- one model (LDA) was further utilized to predict class of origin for controversial texts slices
knowledge
cases
— historians and media researchers theorize about the causal dependencies between public discourse and advertisement

— time series analysis of keyword frequencies (from seedlists) indicated that for some categories ‘ads shape society’, while other categories merely ‘reflect’

— advertisements show a faster decay (on-off intermittent behavior) than public discourse (long-range dependencies)