L90 Practical: Part II, Continued

Simone Teufel

Michaelmas 2018/2019
Practical Session Nov 21: How to analyse the doc2vec system

Before end of term: I will give you TICK plus feedback to your report by email

Jan 15 2019: Submit 4,000-word report on the doc2vec system + your analysis
What you should have by now

- A NB classifier that can run on BOW
- An SVM classifier than can run on both BOW and document embeddings
- A methods for training document embeddings
- A simple statistical test
Some numerical sanity tests

- NB could be around 75-80%
- SVM-BOW can be made to go to 84%
- Doc2Vec could be around 88%
What we will add now

- A more powerful statistical test
- How to do error analysis (in general and specific to embedding space interpretation)
A more powerful test: Permutation test

- Paired samples: two systems are run on identical data
- Tests whether the population mean is different ($H_1$) or the same ($H_0$)
- Non-parametric tests: no assumptions about distribution in your underlying data

\[ \alpha = P(\text{Type I Error}) = P(\text{Reject } H_0 | H_0 \text{ is True}) \]

$\alpha$ is the probability of a false positive (significance level).

\[ \beta = P(\text{Type II Error}) = P(\text{Do Not Reject } H_0 | H_1 \text{ is True}) \]

$\beta$ is the probability of a false negative. $1-\beta$ is the power of the test.
Consider the \( n \) paired results of System A and B.

You will observe a difference \( d \) between the means of system A and B.

If there is no real difference between the systems (and they come from one and the same distribution), it should not matter how many times I swap the two results, right?

There are \( 2^n \) permutations (each row can be 0 or 1; swapped or not).

How many of these permutations result in a difference \( d \) as high as the unpermuted version, or higher?

That proportion is your \( p \)

Final twist: If you cannot test \( 2^n \) resamplings, test a large enough random subset.
More formally

- The Permutation test evaluates the probability that the observed difference in mean $M$ between the runs has been obtained by random chance.
- If the two runs are indeed the same, then the paired re-assignments should have no impact on the difference in $M$ between the samples.
- Re-sampling: For each paired observation in the original runs, $a_i$ and $b_i$, a coin is flipped. If 1, then swap the score for $b_i$ with $a_i$. Otherwise, leave the pair unchanged.
- Repeat $R$ times; compare differences in $M$. 
The probability of observing the difference between the original runs by chance approximated by:

\[ p = \frac{s + 1}{R + 1} \]  \hspace{1cm} (1)

- \( s \): number of permuted samples with difference in \( M \) higher than the one observed in the original runs
- If \( R < 2^n \) because of size, we call this a **Monte Carlo Permutation test**.
### Permutation Test: Example with Real-Valued Results

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>One permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>System A</td>
<td>System B</td>
</tr>
<tr>
<td>Item 1</td>
<td>0.01</td>
<td>0.1</td>
</tr>
<tr>
<td>Item 2</td>
<td>0.03</td>
<td>0.15</td>
</tr>
<tr>
<td>Item 3</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>Item 4</td>
<td>0.01</td>
<td>0.08</td>
</tr>
<tr>
<td>Item 5</td>
<td>0.04</td>
<td>0.3</td>
</tr>
<tr>
<td>Item 6</td>
<td>0.02</td>
<td>0.4</td>
</tr>
<tr>
<td>Observed MAP</td>
<td>0.0267</td>
<td>0.205</td>
</tr>
<tr>
<td>Absolute Observed Difference</td>
<td>0.178</td>
<td>0.0017</td>
</tr>
</tbody>
</table>

- $2^6$ possible permutations for coin throws over 6 items
- Exhaustive resampling: 2 out of 64 permutations are equal or larger than the observed difference in MAP, 0.178.
- $p$-value = $\frac{2}{64} = 0.0462$.
- Reject Null hypothesis at confidence level $\alpha = 0.05$. 

Simone Teufel

L90 Practical: Part II, Continued
What you should do

- Implement Monte Carlo Permutation test
- Use it in the future for all stat. testing where possible
- Use $R=5000$
Three types of analysis

- How could we practically improve the system?
- Deployment test
- What does the Embedding space “encode”? 
SVM performance error analysis

- Which documents does the system make the most catastrophic errors for? (And why?)
- Goal: changes in algorithm or parameters to improve results
- Consider a sizable amount of errors
- Try to classify them:
  - Likelihood of fixing them (low-hanging fruit vs. holy grail of NLP)
  - Frequency of this error
  - Source of this error
- Very typical thing done after achieving results or milestones – deciding where to go next.
Deployment test

- After submission: test your system on new, real data.
- Why a deployment test if you are satisfied with your system’s performance on the 2000 articles?
- Because any of the following may have happened:
  - Wrong assumptions about data (type of films, language, . . .)
  - Unrepresentative sampling
  - Model over- or underfitting
  - Taste and fashion over time
- Say, choose some IMDB reviews for movies from 2017 or 2018 you liked or disliked.
- The data you will test on is then really new, unseen and real.
Finding out what the model is really doing

E.g., see Lau and Baldwin (2016), and Li et al. (2015):

- Are similar documents close to each other in Doc2Vec space?
- Are similar words close together in Doc2Vec space?
- Are document embeddings close in space to their most critical content words?
Start from known similar groupings of reviews, then look at their distance in Embedding space.

Not the other way round.

Similarity must be defined before you measure angles between embeddings.

There are many ways to do this.

We recommend this website for visualisation: https://projector.tensorflow.org (due to Paula C.)
Thank you!