Chinese Restaurant Process
David Mareček

October 11, 2018
大颠瓦法，为佛教密宗无上瑜伽部的甚深密法。在藏传佛教各大教派中最著名的仍是直贡大颠瓦法。有许多信徒不远万里来到直贡接受颠瓦法或希望从直贡噶举派上师处得到此法。享誉海内外的直贡颠瓦法，分为口传颠瓦法和伏藏颠瓦法两类。口传颠瓦法，由初始佛持金刚传来，又分为语旨觉受传承和证语意义传承。语旨觉受传承是四大语言教授由金刚持依次传授因陀罗、龙树、玛当吉，而至底洛巴。其法有父续密集及圆满次第五道直传和颠瓦法。证悟意义传承是底洛巴亲自从金刚持处聆听，依次传授给那若巴、玛尔巴、米拉日巴、塔布拉吉、帕木珠巴，觉巴大师，止至今天其耳传甚深密法，传承从未间断。此颠瓦法称之为彩虹颠瓦法，因得此颠瓦法临终出现彩虹迷漫，故此得名。

伏藏颠瓦法，由阿弥陀佛依次传授给莲花生，赤颂的大巨尼玛。尼玛将其藏在塔拉岗位山后的黑曼陀罗湖中。后来，由大巨尼玛的转世牧羊人尼达桑杰发掘并依次传授贤士。囊卡坚赞，多旦。更堆桑布，帕果。智美洛珠，法王桑杰坚赞，门才。坚赞教授，直贡羊日岗。堪布洛。平措朗杰。此后，由直贡噶举做为甚深密法发扬光大。此颠瓦法，称之为人草颠瓦法，因其颠瓦法成功后，在头盖上能插入草，故此得名。颠瓦，意为迁识，使亡者不经过中阴，将灵魂往生净土之法。据说，接受此甚深密法颠瓦法，从人的头盖骨里出黄水或能插入茅草等现象。接受直贡颠瓦法，使人立刻感觉头疼，头顶渗出水珠，晕倒，打通梵净穴，流鼻血等现象，故此灵验而闻名于世。(欲探详情，请参阅《捷径颠瓦法彩虹修行次第摘要》。)
The task

- Let’s switch the problem to English so that we understand the results
- We delete all spaces and try to restore them back in an unsupervised manner
- We supose that we do not have any rules, any dictionary, etc.

Theresa May launches a frantic two-week campaign today to save her Brexit deal and premiership by telling MPs to do their duty and support her or face going “back to square one”. In a high-risk strategy to turn the tide of opposition in Westminster, the prime minister will then embark on a nationwide tour designed to sell her plan directly to the electorate.

theresamaylaunchesafrantictwo-weekcampaigntodaytosaveherbrexitdealandpremiershipbytellingmpstodoherduetyandsupportherorfacegoing“backtosquareone”inahigh-riskstrategytoturnthetideofoppositioninwestminster,theprimeministerwillthenembarkonanationwidetourdesignedtosellherplandirectlytotheelectorate.
1. Generate a word according to a finite table of unigram word probabilities $p(w)$.
2. Write down the word chosen.
3. With probability 0.99, go to step 1. With probability 0.01, quit.

What is the best maximum likelihood solution?

Where is the problem?

We need a better model reflecting the re-use of words in the text.
A random process, where task is analogous of seating customers in Chinese Restaurant with infinite number of tables.

First person sits at the first table.

\( n^{th} \) person can sit at a table based on following process:
- With probability \( \frac{\alpha}{n-1+\alpha} \), he chooses the first unoccupied table.
- With probability \( \frac{c}{n-1+\alpha} \), he chooses an occupied table, with \( c \) already sitting customers.

Parameter \( \alpha \) is a scalar parameter of the process.

CRP generates a probability distribution.
Chinese restaurant process (CRP)

Demo:
http://topicmodels.west.uni-koblenz.de/ckling/tmt/crp.html?parameters=2&dp=1#
CRP - properties

- Denote $z_i$ the table occupied by the $i$-th customer.
- A possible arrangement of 10 customers:

```
1, 3, 8  2, 5, 9, 10  4, 6, 7
```

$$P(z_1, \ldots, z_{10}) = P(z_1) P(z_2|z_1) \cdots P(z_{10}|z_1, \ldots, z_9) =$$

$$= \frac{\alpha}{\alpha + 1} \frac{\alpha}{\alpha + 2} \frac{1}{\alpha + 3} \frac{\alpha}{\alpha + 4} \frac{1}{\alpha + 5} \frac{1}{\alpha + 6} \frac{2}{\alpha + 7} \frac{2}{\alpha + 8} \frac{2}{\alpha + 9} \frac{3}{\alpha}$$
The probability of a seating is invariant under permutations.
Permuting the customers permutes the numerators in the above computation, while the denominators remains the same.
This property is known as **exchangeability**.
CRP for Chinese segmentation

1. \( i = 0 \) number of words generated.
2. \( i = i + 1 \)
3. With probability \( \frac{\alpha}{\alpha + i - 1} \), generate a word according to base probability \( P_0 \).
4. With probability \( \frac{i - 1}{\alpha + i - 1} \), repeat one word that was already generated before.
5. Write down the word chosen.
6. With probability \( p_{cont} \), go to step 2. With probability \( 1 - p_{cont} \), quit. (\( p_{cont} = 0.99 \))

Probability of the whole generated text is:

\[
\prod_{i=1}^{n} \frac{\alpha P_0(w_i) \text{ count}(w_i \text{ in previous words selections})}{\alpha + i - 1} \cdot p_{cont}^{n-1} \cdot (1 - p_{cont})
\]
How to set the base distribution for words?

1. Word = empty
2. Pick a character from the uniform distribution (1/C)
3. Add the chosen character to the end of Word.
4. With probability $p_c$, go to 2, with probability $1 - p_c$, output the Word. ($p_c = 0.5$)

This base distribution prefers short words to long words, though it assigns positive probability to an infinity of arbitrarily long words.
1. Imagine there are only three characters $a$, $b$, and $c$ and $p_c = 0.5$. What is the base probability of words $a$, $aa$, $aaa$, $bcb$?

2. We observe a character sequence $ab$. What is more probable segmentation? $a\ b$ or $ab$?

3. We observe a character sequence $aa$. What is more probable segmentation? $a\ a$, or $aa$?

4. We observe a character sequence $abab$. What is the most probable segmentation?
There is $2^{n-1}$ possible segmentations for $n$-characters long data.

Exchangeability: if we reorder the words in the sequence, overall probability is the same.

$n - 1$ latent binary variables $s_i$: denoting whether there is or isn’t a separator between two characters.

Collapsed Gibbs sampling. Sample one variable conditioned by all the others.

We can virtually move the changed words at the end of the sequence, compute the overall probability of the two possibilities and then move the words virtually back.
\[ \forall i \in \{1 \ldots n - 1\} : \text{initialize } s_i \text{ randomly; } \]

compute initial counts of words \( \text{count[word]} \) and total word-count \( t \);

for \( \text{iter} \leftarrow 1 \text{ to } \text{numiter} \) do

\[
\text{for } i \leftarrow \text{randomPermutation}(1 \text{ to } n - 1) \text{ do}
\]

prev = previous word; next = next word; joined = prev + next;

if \( s_i == 0 \) then \( \text{count[joined]}--; t--; \)

else \( \text{count[prev]}--; \text{count[next]}--; t -= 2; \)

\[
\begin{align*}
\text{p[0]} &= \frac{\alpha \text{P}_0(\text{joined}) + \text{count[joined]}}{\alpha + t}, \\
\text{p[1]} &= \frac{\alpha \text{P}_0(\text{prev}) + \text{count[prev]}}{\alpha + t} + \frac{\alpha \text{P}_0(\text{next}) + \text{count[next]}}{\alpha + t + 1};
\end{align*}
\]

\( s_i = \text{sample 0 or 1 from with weights } \text{p[0]} \text{ and } \text{p[1]} \);

if \( s_i == 0 \) then \( \text{count[joined]}++; t ++; \)

else \( \text{count[prev]}++; \text{count[next]}++; t += 2; \)

end

end
Annealing

- When sampling, we can regulate the speed of convergence using so-called temperature $T \in (0, \infty)$.
- Instead of sampling with weights $(w_1, w_2, w_3, \ldots, w_n)$ we can use $(\frac{1}{T} w_1, \frac{1}{T} w_2, \frac{1}{T} w_3, \ldots, \frac{1}{T} w_n)$.
- The higher $T$, the larger searching area and the slower convergence.
- For $T \to \infty$, the sampling is uniform.
- For $T \to 0$, we always choose the best option.