Intro:

這次作業要用 Naive Bayes 去 train Language Model, 再用 EM Algorithm 算 mixture 的期望值
程式部分用 python2.7 + sqlite 完成, 並使用 pypy 編譯

Naïve Bayes :

假設每個 Unigram 彼此間都是獨立的, 所以可以直接拿每個 Unigram 的 probability 計算

$$P(w_i|w_1 \ldots w_{i-1}) \approx P(w_i) = \frac{c(w_i)}{\sum_{\tilde{w}} c(\tilde{w})}$$

在 train 的時候我沒有做 smooth 而是等到對 test data Predict 時才做
Predict 時對 test data 的每個 Unigram 求 Likelihood

$$P(W_{test}|M) = \prod_{w \in W_{test}} P(w|M)$$

這時 P 會加入 smooth 避免沒出現過的字為 0

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (\theta = \mathrm{M})$$

由於機率過小不夠 float 存(-inf), 因此用了 log

$$\log P(W_{test}|M) = \sum_{w \in W_{test}} \log P(w|M)$$

而每個 doc 裡面的所有 unigram 所算出的 Log Likelihood 中最大者即為該 model 的 class
求出 maxima Likelihood, 完成 predict
命中率約 64%

補充:

取出 Unigram 時, 會濾掉一些 Stop Words, 另外加入一些 counts 過高者,
並對@還有.,符號做置換

EM Algorithm:

主要針對 UnLabel data, 假設每個 doc 中的每個 word 都是由已知的 20 個 topics 之 mixture language mode 生成

E-step:

$$P(z_{j=1}|w) = \frac{\lambda_j P(w|\theta_j)}{\sum_{i=1}^{20} \lambda_i P(w|\theta_i)} \quad \text{j:[1,20]}$$

即為求 Zj = 1 時的期望值

M-step:

Estimate λ

$$\lambda_j^{new} = \frac{\sum_w c(w,d) P(z_{j=1}|w)}{\sum_w c(w,d)} \quad \text{j:[1,20]}$$

Estimate θ

結論:

這次作業又被 python 錶了一次,前面在寫 Naive Bayes 時因為 file list 沒有照順序讀進來造成結果不管怎麼 tune 都是 0.05(用猜的差不多),後來是有人建議用寫好的 NB 驗證一次,確認問題點,雖然花了不少時間搞 svm format 不過也算是上了一課

在看 EM 的時候與同學討論一直誤會是要用上課教的那種情況估,先 validation 取得 labeled 和 mixture 之間的最佳比例, 再用得到的 λ 計算 expectation, 這樣每次 iterative 都要暴力 validation 且程式的 spec 有要能更動 input data size, 最後還好有寄信問助教,才避免真的去搞一個推不大出來的式子

這次寫 IR 作業由於 python 比較熟了一些,寫起來問題比較少,不會像前一次那樣 list 滿天飛 performance 又差到爆,不過 debug 還是有些障礙, 數學的部分因為這次用 Naïve 和 mixture model 所以不用搞到複雜的 distribution 且 M-step 有已經導好的公式所以很弱的微積分也避掉了