Statistical Computing Final

作者: 109024701 林承慶

Introduction

這份報告所要用的資料為Amazon商品廣告的資料，從Kaggle抓下來的。這份資料是做出這份資料集的作者於2019年末從Amazon抓下來的，並且把每一個廣告是屬於哪一個分類都用資料夾分好了。資料總數為2325380，然而這次我只專注在電子產品的分類，因此實際用到的資料量是從15類產品中，挑出各1000筆資料，總共為150000筆，這份報告所用的程式語言為`python`。

在Amazon裡，電子產品(Electronics)被分為以下15種細項，同時也是我們要分類的目標:

0. Accessories-Supplies : 配件與備品
1. Camera & Photo : 相機和相片
2. Car & Vehicle : 車用電子產品
3. Cell Phones : 手機
4. Consoles : 遊戲機
5. Ebook : 電子書
6. GPS : 導航
7. Headphones : 耳機
8. Home Audio : 家庭式喇叭
9. Office : 辦公用電子產品
10. Portable Audio & Video : 可攜式影音產品
11. Projectors : 投影機
12. Security Surveillance : 監視器
13. TV & Video : 電視及影片播放機
14. Wearable Tech : 穿戴式電子產品

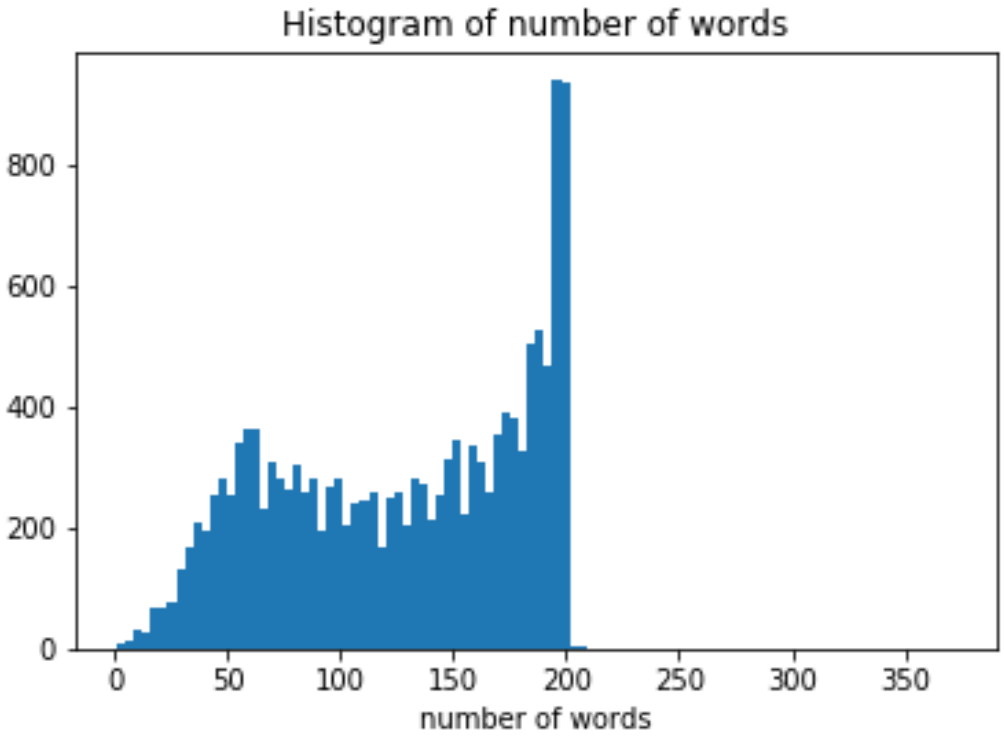
在此所說的廣告是指線上購物上面的商品名稱，線上商家為了快速增加流量，很常會出現商品名稱帶有很多的形容或是塞一堆關鍵字，例如:「53 件銀器套裝,HaWare 實心不銹鋼現代優雅餐具,包括 40 件餐具組,5 件份組,8 件牛排刀,鏡面拋光,可用洗碗機清洗」

因此，我的目標是，利用topic models / NMF等unsupervised方法，讓廣告裡面的關鍵字分群，再利用svm去分類，讓每個廣告歸類回原本的類別。

EDA 和 資料前處理

ad	label
WordForum 3.5mm 360° Stereo Conference Microphone - Omnidirectional Digital Recording for Meetings, Teleconferencing, Video Conferencing - Daisy Chain Option (4 Pack)	9
Wiresmith AC Power Adapter for Nintendo Wii U Console	4
GCION[2 Pack-Upgrade Version] Compatible with Iphone 11 Pro(5.8 inch) Iphone 11 Pro Max (6.5 inch) Screen Protector,Camera Lens Protector, Ultra Thin, High Definition, Anti-Scratch, Anti-Fingerprint-Silver	3
OtterBox COMMUTER SERIES Case for iPhone 11 Pro - BLACK	3
HD Projector - Artlii 2019 Upgraded 4000 Lumen Movie Projector, 200" HD Home Theater Projector, 1080P Support Video Projector with 2 HDMI VGA 2 USB HiFi Stereo for Movies, Sports and Video Games	11
Aiphone Corporation JK-DV Video Door Station for JK and JM Series Hands-Free Video Intercom, Zinc Die Cast, 6-13/16" x 3-7/8" x 1"	12
AV to HDMI, GANA 1080P Mini RCA Composite CVBS AV to HDMI Video Audio Converter Adapter Supporting PAL/NTSC with USB Charge Cable for PC Laptop Xbox PS4 PS3 TV STB VHS VCR Camera DVD	13
Neewer Aluminum Screw Knob Clamp Arca Swiss Compatible Mini Quick Release Clamp for QR Plate (38mm)	1
Plantronics Standard Earloop Kit Black (88814-01)	9

這是處理之前的資料，其中label就是相對於前一張的15個細項分類。在處理資料之前，我們可以去觀察大部分的廣告都是多少字。



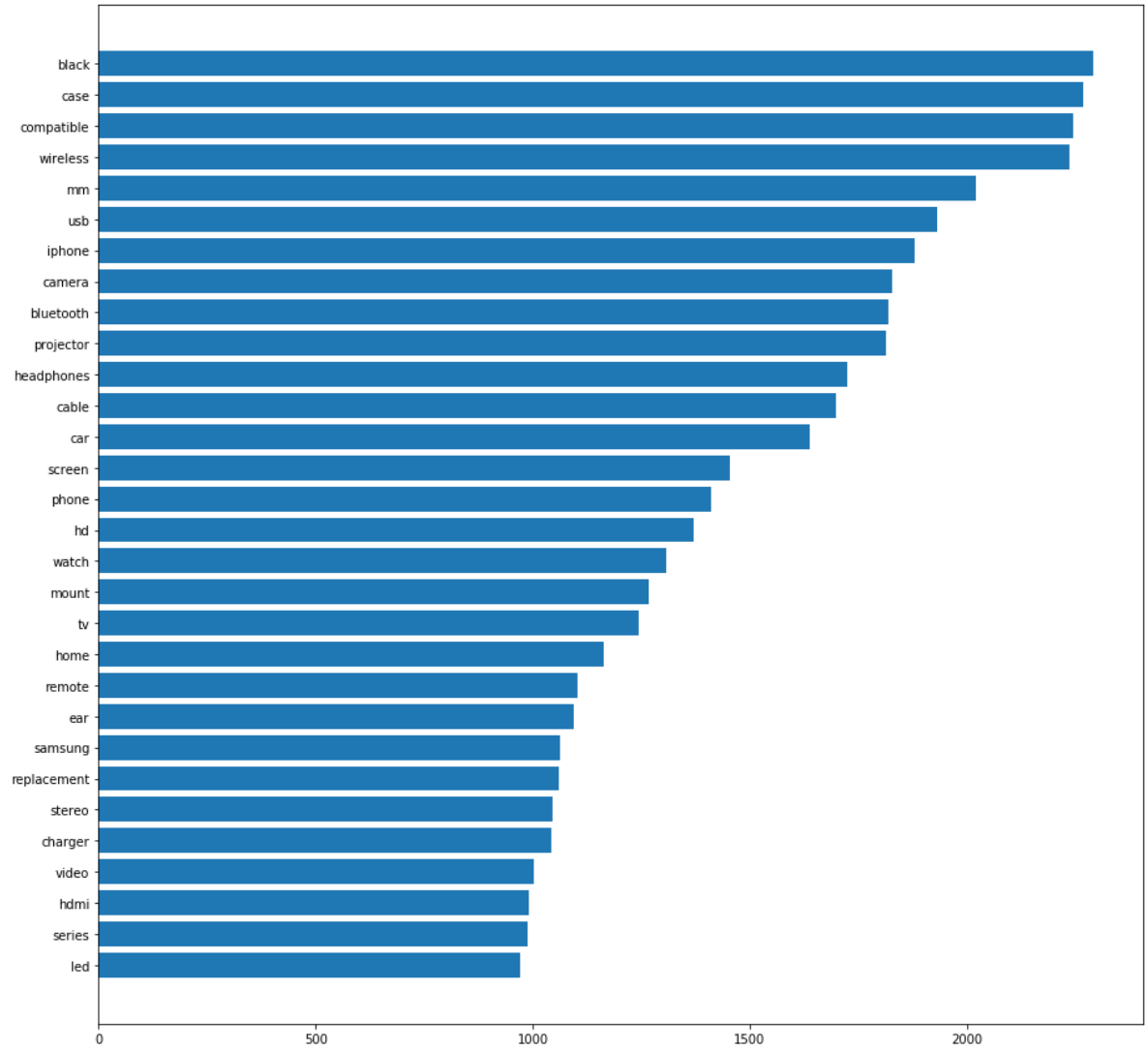
從圖中，我們可以看到說，在200字附近有個極高的peak，並且200字之後只有少數的outliers，這是因為Amazon的廣告字數限制在200字以下，因此商人們都會想在限制下塞入最多的關鍵字。

接下來，我利用`spaCy`的英文模型分析，判斷每一個英文字的詞性，並且拆解句子，這樣的程序我們叫**Tokenize**。再拆解並標籤每個字的詞性之後，我們可以排除掉停用詞(Stopwords, 會阻礙分析 的詞彙, 例如: the, is, at, want, need...)、標點符號(Punctuation)、數字(Digit)。在電子產品中，還有產品型號以及規格也會影響到我們的分析，因此也要剔除(例如: 1080p, S9)。

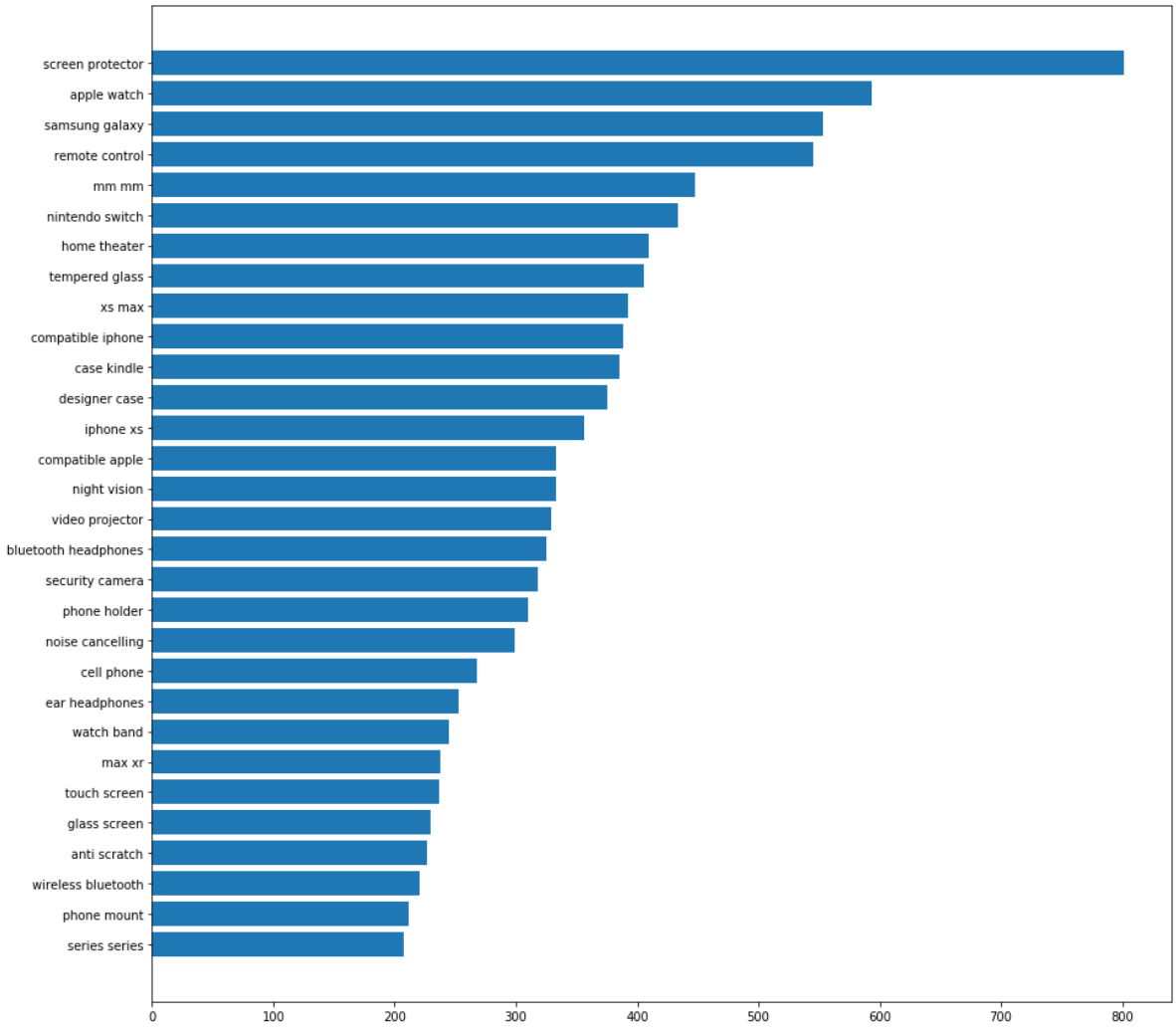
ad	label	target
WordForum 3.5mm 360° Stereo Conference Microphone - Omnidirectional Digital Recording for Meetings, Teleconferencing, Video Conferencing - Daisy Chain Option (4 Pack)	9	WordForum mm Stereo Conference Microphone Omnidirectional Digital Recording Meetings Teleconferencing Video Conferencing Daisy Chain Option Pack
Wiresmith AC Power Adapter for Nintendo Wii U Console	4	Wiresmith AC Power Adapter Nintendo Wii U Console
GCION[2 Pack-Upgrade Version] Compatible with Iphone 11 Pro(5.8 inch) Iphone 11 Pro Max (6.5 inch) Screen Protector,Camera Lens Protector, Ultra Thin, High Definition, Anti-Scratch, Anti-Fingerprint-Silver	3	Pack Upgrade Version Compatible Iphone inch Iphone Pro Max inch Screen Protector Camera Lens Protector Ultra Thin High Definition Anti Scratch Anti Fingerprint Silver
OtterBox COMMUTER SERIES Case for iPhone 11 Pro - BLACK	3	OtterBox COMMUTER SERIES Case iPhone Pro BLACK
HD Projector - Artlii 2019 Upgraded 4000 Lumen Movie Projector, 200" HD Home Theater Projector, 1080P Support Video Projector with 2 HDMI VGA 2 USB HiFi Stereo for Movies, Sports and Video Games	11	HD Projector Artlii Upgraded Lumen Movie Projector HD Home Theater Projector Support Video Projector HDMI VGA USB HiFi Stereo Movies Sports Video Games
Aiphone Corporation JK-DV Video Door Station for JK and JM Series Hands-Free Video Intercom, Zinc Die Cast, 6-13/16" x 3-7/8" x 1"	12	Aiphone Corporation JK DV Video Door Station JK JM Series Hands Free Video Intercom Zinc Die Cast x x
AV to HDMI, GANA 1080P Mini RCA Composite CVBS AV to HDMI Video Audio Converter Adapter Supporting PAL/NTSC with USB Charge Cable for PC Laptop Xbox PS4 PS3 TV STB VHS VCR Camera DVD	13	AV HDMI GANA Mini RCA Composite CVBS AV HDMI Video Audio Converter Adapter Supporting PAL NTSC USB Charge Cable PC Laptop Xbox TV STB VHS VCR Camera DVD
Neewer Aluminum Screw Knob Clamp Arca Swiss Compatible Mini Quick Release Clamp for QR Plate (38mm)	1	Neewer Aluminum Screw Knob Clamp Arca Swiss Compatible Mini Quick Release Clamp QR Plate mm
Plantronics Standard Earloop Kit Black (88814-01)	9	Plantronics Standard Earloop Kit Black

我們可以看到右邊是原始文字，左邊是處理完的文字。可以觀察到前面所述的停用詞、標點符號、數字、型號及規格都不見了。處理完之後的文字，我們可以用**文字雲**去描繪字被使用的頻率多寡。

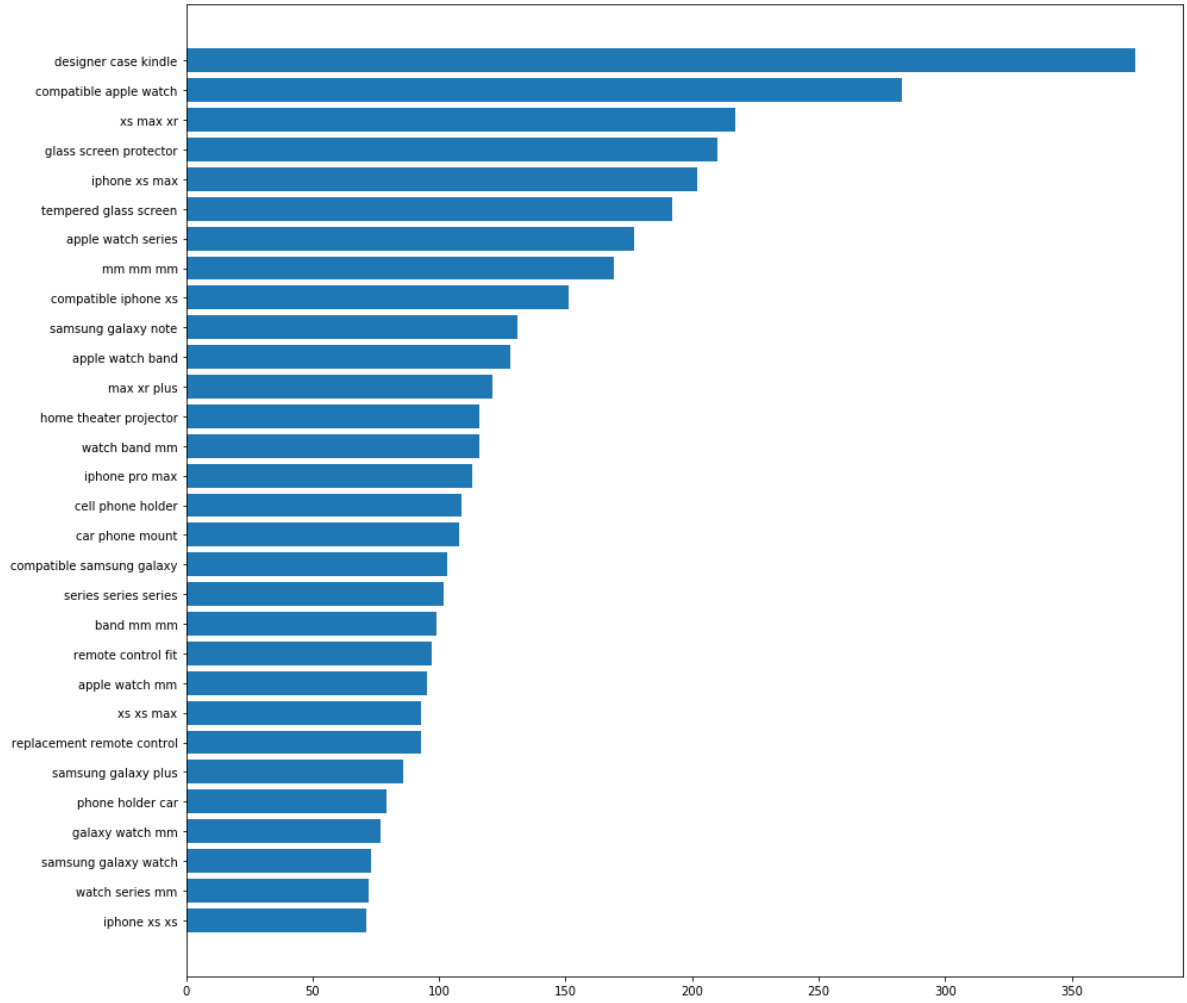
我們能夠看到Iphone、Projector和Phone是最多的，其次是Mount，可以大約看得出來和手機相關的手機殼或手機架，是最多商品會提到的。如果是要看文字被使用的總數，則是利用傳統的柱狀圖，就如以下所示



前面4個為black、case、compatible、wireless，這四個是最多被使用的字彙。同時我們可以大致推估說，黑色是較多商品會有的顏色，保護套類型的商品很多，相容性是大家很注重的地方，無線是當年的趨勢。我們上面所看到的柱狀圖為1-gram的分析。**N-gram**分析指的是，我們把句子拆成N個字的詞，例如我想要分析"I love you too"，那麼1-gram就是拆成{"I", "love", "you", "too"}，而2-gram則是{"I love", "love you", "you too"}。這樣的拆法的用意是，有些字相遇的時候，會出現另一種意思，為了抓出這種詞彙，因此採用這種方式。例如:"Night"是夜晚，"Vision"是視野，但是"Night Vision"是夜視功能。接下來我們看2-gram以及3-gram。



這張就顯示了與上面不一樣的結果。前四個為"screen protector"、"apple watch"、"samsung galaxy"以及"remote control"，可以看到螢幕保護貼是Amazon最多的產品，也有可能是有很多商家標榜在Amazon上賣手機，都會送螢幕保護貼。再來看3-gram的結果。



前兩名為"designer case kindle"以及"compatible apple watch"。前面是Amazon的閱讀器收納袋。後者則是看得出來，在1-gram中的compatible大部分是指對於apple watch的相容性，而2019年最熱門的電子產品就是apple watch了。

最後，電腦以及之後要用的演算法仍然只能處理數字，所以我們需要把文字轉成數字，而這邊有兩種方法: **Countvectorize** 以及 **TF-IDF**。

Countvectorize

首先先把所有文本中的字都收集起來，製作出詞袋(Word of Bags)，通常是以1維的array表現。再來回頭分析每個句子裡面字所出現的次數，並做成向量表示。

例如: "庭院深深深幾許"，詞(字)袋就會是{"庭"、"院"、"深"、"幾"、"許"}，所以Count vector產生的向量為(1,1,3,1,1)。

TF-IDF

這個方法其實包含兩個部分: **詞頻**(TF, Term Frequency)和**逆向文件頻率**(IDF, Inverse Document Frequency)。其中詞頻是指該詞在句子裡面所出現的頻率，第t個詞出現在第d篇的文件頻率為$tf_{t,d}$。舉例來說，"庭院深深深幾許"，其中第3個字(詞)，"深"，出現在第一個句子的頻率為$tf_{3,1} = 3/7$。

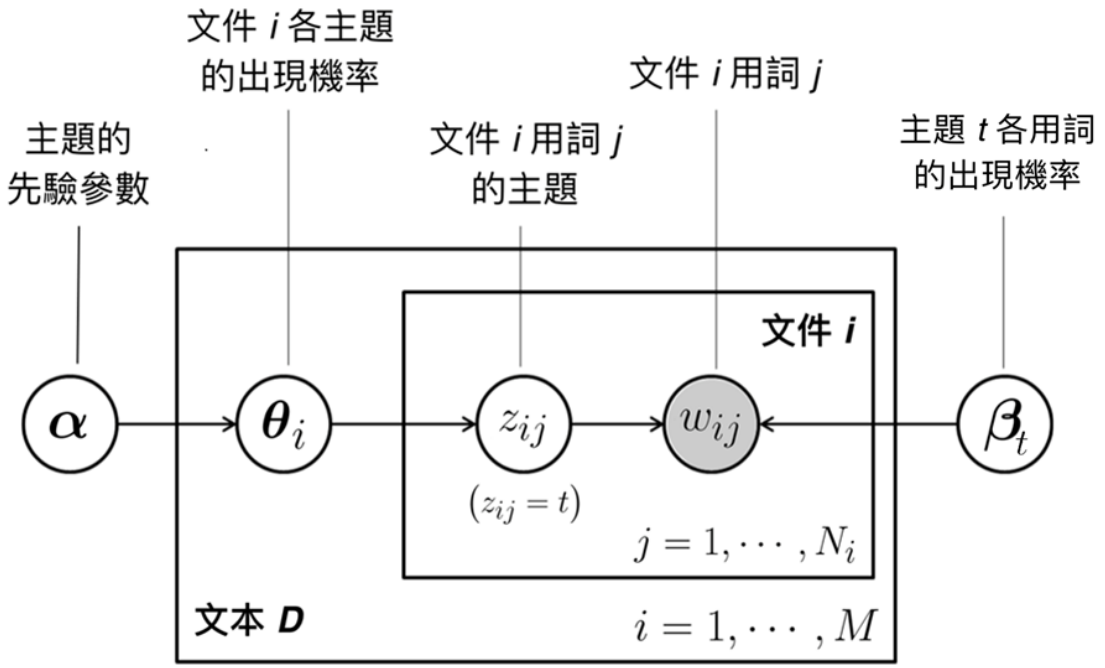
逆向文件頻率則是處理常用字問題。假設詞彙t在所有文本數量為D篇文章中出現於d_t篇文章裡，那逆向文件頻率就是$idf_t = \log_{10}(D/d_t)$。因此，我們用這兩個指標來做出某個字對文章的重要性的分數$w_{t,d} = tf_{t,d} \times idf_t$。

舉例來說，如果我們想要探索2020年出版的所有碩博士論文，"感謝" 這個詞被使用頻率tf很高，但每篇文章都有這個詞的逆向頻率idf就很低，那"感謝"的分數$w_{t,d}$就很低；"無母數"這個詞在統計領域被使用頻率tf很高，同時每篇文章都有這個詞的逆向頻率idf也很高，那"無母數"的分數$w_{t,d}$就很高。

Unsupervised Learning

Latent Dirichilet Allocation

簡稱**LDA**。LDA是對於每個文件做三層的Mixture models，我們直接來看示意圖:



圖中，灰色區塊是我們所能觀測到的，也就是文本本身；白色部分則是隱含在背後的模型。LDA假設每個文本產生，隱含著數個主題，而每個主題有會有數個詞彙來表示，因此就會產生上面的示意圖。裡面的參數個別為:

- α是topics的先驗參數
- $\theta_i \sim Dir(\alpha)$是第$i$個文件的主題出現的機率
- $z_{ij} \sim Multinomial(\theta_i)$為給定第$i$個文件，第$j$個詞彙，所抽出來的主題
- $\beta_t \sim Dir(\eta)$在η的先驗參數下，給定t主題，每個詞彙受出現的機率
- $w_{ij} \sim Multinomial(\beta z_{ij})$就是給定$t$主題，給定第$i$個文件下，第$j$個詞彙下，所抽出來的詞彙

因此，我們可以把所抽出來的機率寫成以下的形式:

$$
\begin{aligned}
P(\vec{w}|\vec{\alpha},\beta) &= \sum_{\vec{z}}\prod_{n=1}^{N} P(w_n|z_n,\beta)P(\vec{z}|\vec{\alpha}) \\
&= \sum_{\vec{z}}\prod_{n=1}^{N} P(w_n|z_n,\beta)\int\prod_{n=1}^{N} P(z_n|\vec{\theta})P(\vec{\theta}|\vec{\alpha})d\vec{\theta} \\
&= \int\sum_{\vec{z}}\prod_{n=1}^{N} P(w_n|z_n,\beta)P(z_n|\vec{\theta})P(\vec{\theta}|\vec{\alpha})d\vec{\theta} \\
&= \int\prod_{n=1}^{N}\sum_{z_n} P(w_n|z_n,\beta)P(z_n|\vec{\theta})P(\vec{\theta}|\vec{\alpha})d\vec{\theta}
\end{aligned}
$$

然而，這個公式並沒有解析解，更別說如果想要從文本D中，利用求log likelihood的最大值來估計$\vec{\alpha},\beta$，則

$$\ln P(D|\vec{\alpha},\beta)=\sum_{d=1}^{D}\ln P(\vec{w}^{(d)}|\vec{\alpha},\beta)$$

是無法計算出來結果的。因此有兩種估計的方式來解決這個問題: Variational Inference 和 Collapsed Gibbs Sampling。

Variational Inference

在**Variational Inference**裡，我們利用Jensen's Inequality去找出log likelihood的下界:

$$\ln P(\vec{w}|\vec{\alpha},\beta)=\ln\int\sum_{\vec{z}}\frac{P(\vec{w},\vec{z},\vec{\theta}|\vec{\alpha},\beta)}{q(\vec{z},\vec{\theta})}q(\vec{z},\vec{\theta})d\vec{\theta}\geq\int\sum_{\vec{z}}q(\vec{z},\vec{\theta})\ln\frac{P(\vec{w},\vec{z},\vec{\theta}|\vec{\alpha},\beta)}{q(\vec{z},\vec{\theta})}d\vec{\theta}=L(\vec{\alpha},\beta)$$

其中，$L(\vec{\alpha},\beta)$ 叫就 Evidence Lower Bound (ELBO)。我們可以繼續拆解ELBO

$$\begin{aligned}L(\vec{\alpha},\beta)&=\int\sum_{\vec{z}}q(\vec{z},\vec{\theta})\ln\frac{P(\vec{w},\vec{z},\vec{\theta}|\vec{\alpha},\beta)}{q(\vec{z},\vec{\theta})}d\vec{\theta}\\&=\int\sum_{\vec{z}}q(\vec{z},\vec{\theta})\ln\frac{P(\vec{z},\vec{\theta}|\vec{w},\vec{\alpha},\beta)P(\vec{w}|\vec{\alpha},\beta)}{q(\vec{z},\vec{\theta})}d\vec{\theta}\\&=-\int\sum_{\vec{z}}q(\vec{z},\vec{\theta})\ln\frac{q(\vec{z},\vec{\theta})}{P(\vec{z},\vec{\theta}|\vec{w},\vec{\alpha},\beta)}d\vec{\theta}+\int\sum_{\vec{z}}q(\vec{z},\vec{\theta})\ln P(\vec{w}|\vec{\alpha},\beta)d\vec{\theta}\\&=-KL\{q(\vec{z},\vec{\theta})\|P(\vec{z},\vec{\theta}|\vec{w},\vec{\alpha},\beta)\}+\ln P(\vec{w}|\vec{\alpha},\beta)\end{aligned}$$

其中，KL就是KL divergence的意思。因此，Variational Inference的目標就是找到$q(\vec{z},\vec{\theta})$(的參數)，使得KL divergence最小。因此，要找到最大的ELBO: $L(\phi,\gamma;\vec{\alpha},\beta)$，其中$\phi$是Multinomial parameter，$\gamma$是Dirichlet parameter。接下來的演算法，使用EM algorithm去迭代出我們所想要的答案:

0. 先找一組初始值$(\vec{\alpha},\beta)$
1. E-step: 給定$(\vec{\alpha},\beta)$，找一組(ϕ,γ)使得$L(\phi,\gamma;\vec{\alpha},\beta)$最大。$\Psi(\cdot)$是digamma function

$$\phi_i^{(n,d)}\propto\beta_{i,w_n^{(d)}}exp\{\Psi(\gamma_i^{(d)})-\Psi(\sum_{j=1}^{K}\gamma_j^{(d)})\}$$

$$\gamma_i^{(d)}=\alpha_i+\sum_{n=1}^{N_d}\phi_i^{(n,d)}$$

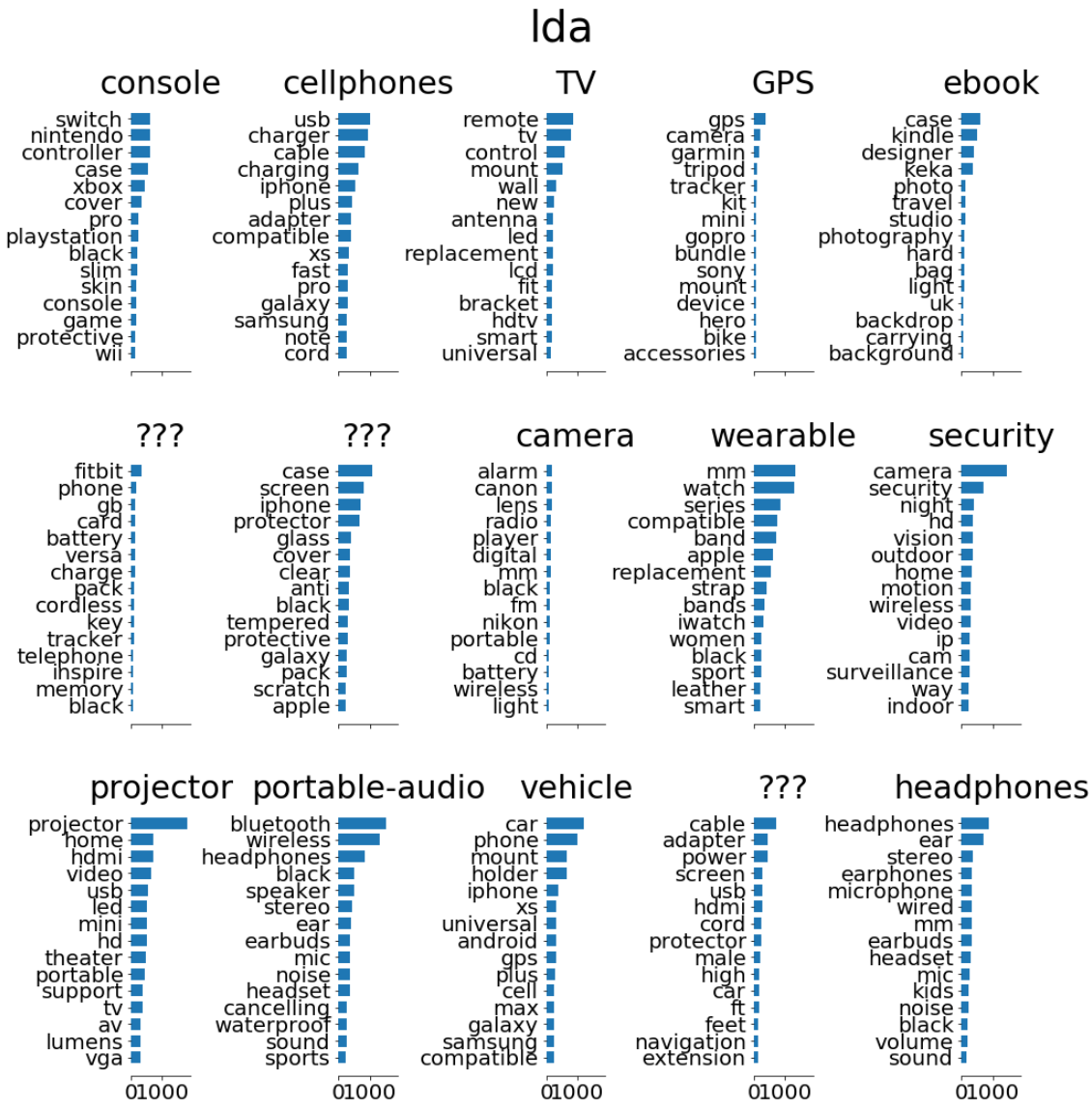
2. M-step: 用從1.找到的(ϕ,γ)，找一組$(\vec{\alpha},\beta)$使得$L(\phi,\gamma;\vec{\alpha},\beta)$最大。因為$\alpha$本身沒有解析解，因此只能用牛頓法去迭代尋找。

$$\beta_{i,j}\propto\sum_{d=1}^{D}\sum_{n=1}^{N_d}\phi_i^{(n,d)}I(w_n^{(d)}=j)$$

$$L_\alpha=\sum_{d=1}^{D}[\ln\Gamma(\sum_{i=1}^{K}\alpha_i)-\sum_{i=1}^{K}\ln\Gamma(\alpha_i)+\sum_{i=1}^{K}(\alpha_i-1)\{\Psi(\gamma_i^{(d)})-\Psi(\sum_{j=1}^{K}\gamma_j^{(d)})\}]$$

3. 持續1.2步驟，直到$L(\phi,\gamma;\vec{\alpha},\beta)$收斂。

在python中，`sklearn`套件裡面的LDA是利用Variational Inference，我們利用前面所述的Countvectorize，代入LDA得出的結果:



x軸是LDA算出來詞彙在特定主題下出現的機率。標題部份是我看完分布之後，再每一個topic去給標題。其中標題為"???"的，就是我無法判別出是哪一類別的分組。

Collapsed Gibbs Sampling

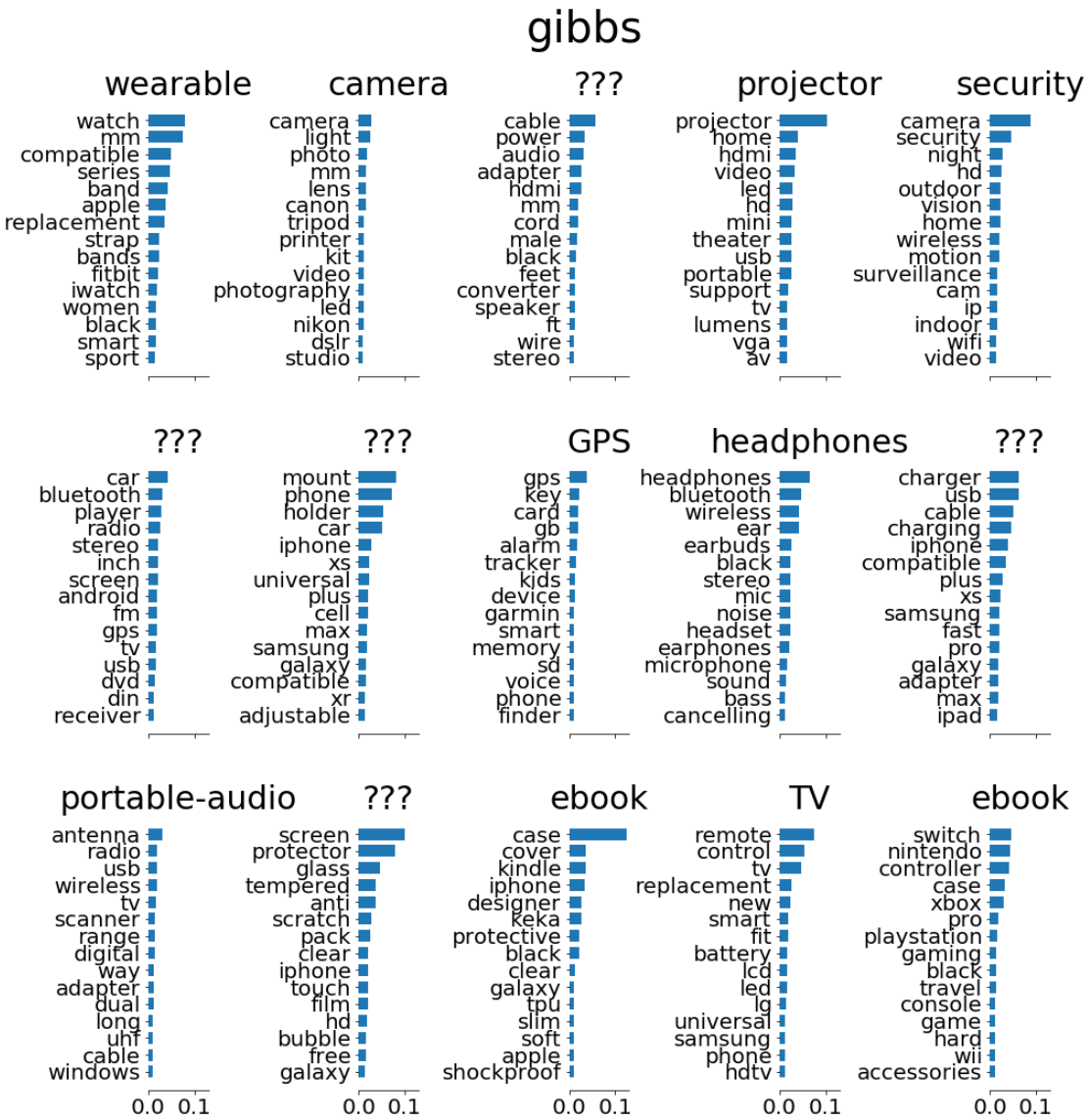
從名稱就可以了解，我們將要用Gibbs Sampling來找出我們想要的。令$k = z_n^{(d)}$和$\nu = w_n^{(d)}$，並且$c_i^{(d)}$是有多少主題i在文件d裡面，c_{ij}是多少主題i字彙j在所有文件裡面，則

$$P(k|Z\backslash k, D, \vec{\alpha}, \eta) \propto P(Z, D|\vec{\alpha}, \eta) \propto \frac{(\eta_\nu^{(k)} + c_{k\nu} - 1)(\alpha_k + c_k^{(d)} - 1)}{\sum_{j=1}^{V} \eta_j^{(k)} + \sum_{j=1}^{V} c_{kj} - 1}$$

因此，演算法為

0. 尋找$\vec{\alpha}, \eta, Z$的初始值
1. 抽取$z_n^{(d)}$，使得更新Z
2. 找到$\vec{\alpha}, \eta$，使得聯合log likelihood $P(Z, D|\vec{\alpha}, \eta)$最大。回到第1步驟。

在python中，我使用的是`lda`套件，裡面就是用Collapsed Gibbs Sampling，我們利用前面所述的Countvectorize，代入LDA得出的結果，接下來的報告中，我會用"Gibbs"來代稱這個方法，以區別Variational Inference:

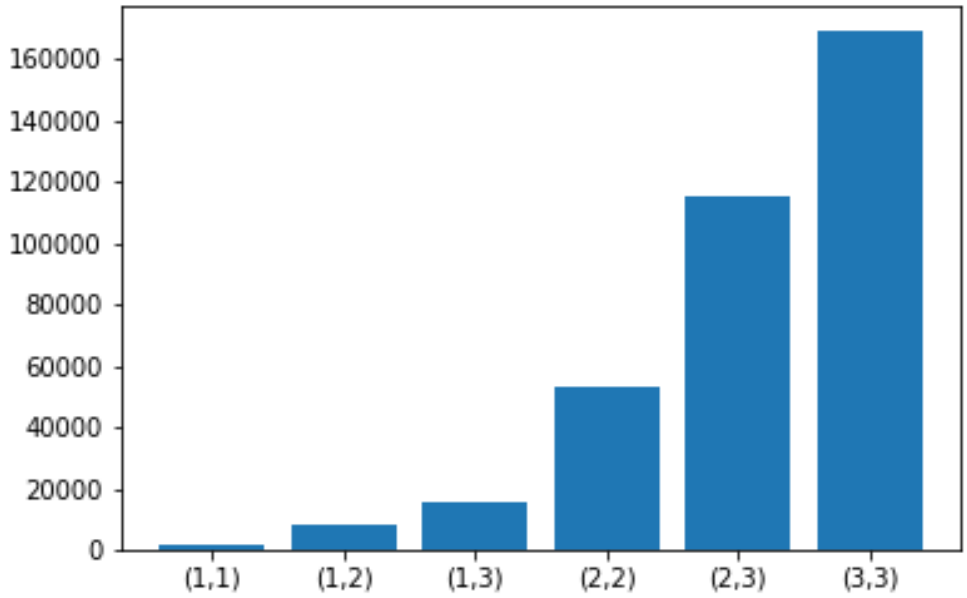


x軸是LDA算出來詞彙在特定主題下出現的機率。標題部份是我看完分布之後，再每一個topic去給標題。其中標題為"???"的，就是我無法判別出是哪一類別的分組。我們可以看到說，相較於利用Variational Inference的方式，用Collapsed Gibbs Sampling多了兩個我無法判別的topic。但除此之外，並沒有看到有什麼差異。

在這邊特別說明一下，我們在使用LDA時，會使用perplexity在計算LDA模型是否貼合資料的分布，公式如下:

$$perplexity = \exp(\frac{-\sum_{d=1}^{D}\log(p(w_d))}{\sum_{d=1}^{D}N_d})$$

分子部分就是生成整個文本的log likelihood，而分母則是整個文集裡所有字的個數。所以若LDA模型越不能生成出這個文本，perplexity會變高。因此我們期望perplexity越低越好，通常來說，perplexity是用來找要有多少個topics，但是因為我已經明確直到原本是15個主題了，因此在這邊我想要用來說明為何我們只用1-gram來分析:



x軸是N-gram的範圍，例如來說(1,3)的意思是模型裡面有1-gram、2-gram和3-gram。y軸是perplexity。可以明確看到說perplexity最低的是(1,1)，也就是只用1-gram的LDA模型。也因此雖然我們在觀察的時候用到了2-gram和3-garm，但是在使用LDA和NMF時，卻只用1-gram。

Non-negative Matrix Factorization

非負矩陣分解，簡稱**NMF**。想法其實非常簡單，假設有一個$n \times m$的非負矩陣V，那麼我們想要把V矩陣拆成兩個非負矩陣W和H，其中W是$n \times r$，而H是$r \times m$，通常$(n+m)r < nm$。用數學表示就是

$$V_{n\times m} \approx W_{n\times r} H_{r\times m}$$

從這邊，我們可以知道這個方法只能用在矩陣裡面元素皆為正的或是0，並且也很明顯的發現這種分解方式跟PCA比分解的不太好。但是就是因為他最大的限制就有是非負，因此非負矩陣分解會拆解出目標的部分零件，並且聚集這些相近的零件。例如把NMF用在臉部照片，就可以把鼻子、眼睛、嘴唇...等臉部零件拆解出來。利用在文本上，就會是把重要且類似topic的詞抓出來。

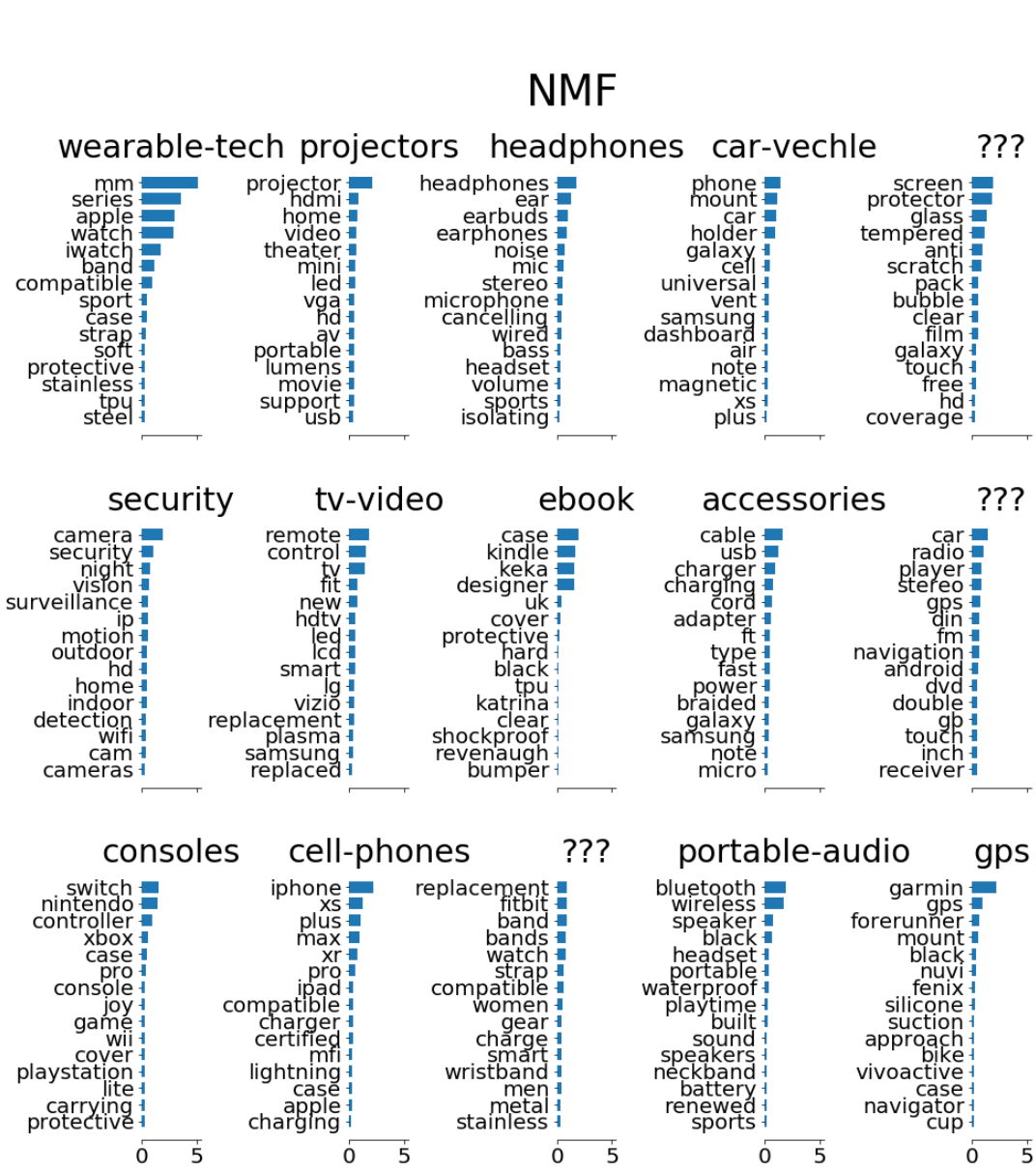
然而，由於NMF是近似分解的方式，因此我們必須定義Cost functions，用來評估我們的分解夠不夠好。這邊所使用的Cost function是一種 divergence，定義如下:

$$D(V\|WH) = \sum_{ij}(V_{ij}\log\frac{V_{ij}}{(WH)_{ij}} - V_{ij} - (WH)_{ij})$$

如果$\sum_{ij} V_{ij} = \sum_{ij}(WH)_{ij} = 1$，那麼這個式子就會變成Kullback-Leibler divergence或是relative entropy。這個function有相對應的迭代方式，這個方法叫Multiplicative update rules，因為推導不是此篇重點，因此我在下面直接寫答案:

$$H_{rm} \leftarrow H_{rm}\frac{\sum_i W_{ir}V_{im}/(WH)_{im}}{\sum_k W_{kr}}$$
$$W_{nr} \leftarrow W_{nr}\frac{\sum_\mu H_{r\mu}V_{n\mu}/(WH)_{n\mu}}{\sum_\nu H_{r\nu}}$$

接下來我就利用前面所處理TF-IDF的資料，代入NMF模型去fit，並且得出以下結果



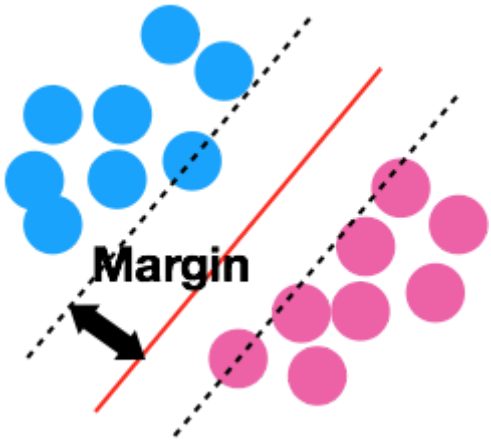
x軸為詞的TF-IDF分數，而每一個小圖的標題都是我看完分群狀況後，把標題填上去的。其實相較於LDA，NMF所得出來的結果，前10個字會比較貼合topic一點，也就是說他裡面比較沒有混雜的詞彙。但同時，仍然還是有"???"的主題發生

Discussion

- 由於LDA模型使用機率模型所產生的，因此Countvectorize比較符合，TF-IDF則是因為轉換時出現的是score，所以不適用。
- 由於NMF就是單純的矩陣分解，因此前處理複雜一點也能接受，因此我們所使用的是TF-IDF模型。
- 兩者概念上最大的不同是，LDA在資料生成上面加上了Dirichlet prior，而NMF沒有。這也表示LDA的主題和字的機率是可以變動的，但NMF不行。
- 因此，在我們確信主題的機率是固定的時候，或是hyperparameter的變異程度對於資料來說太大的時候，NMF才佔有優勢；要不然LDA會比較好。

Support Vector Machine

Support Vector Machine，簡稱**SVM**。基本概念就是想在資料中，找到一個超平面(hyperplane)，使得response被明確分出來，我們可以看下圖範例:



中間紅線是我們的分類器，黑線則是分類的邊界。為了把讓分類器分得夠好，我們會期望紅線到邊界的距離越遠越好。因此我們可以把這類的問題寫成公式，假設我們的分類器為$f(x) = w^T x + b$，訓練資料為(x_i, y_i)，並且定義slack variables ξ，那麼公式為:

$$\min_{w,b} \frac{\|w\|^2}{2} + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \text{ for } i = 1, \ldots, n$$

這邊的公式叫做Soft margin的SVM。若沒有slack variables ξ，那我們會稱為Hard margin的SVM。slack variable，主要是增加我們分類的容錯狀況。因為有時候資料本身是難以找出一條線切開的，那時候我們就只要「大部分」是對的就好，此時就需要ξ進來。而C是調控ξ的參數，若C越小，容錯程度越大；若C越大，則容錯程度越小，越像Hard Margin.

SVM除了可以處理簡單的線性分類以外，也可以處理複雜的分類狀況，此時的分類器會變成:

$$f(x) = w^T \Phi(x) + b$$

其中，Φ是個讓x從原本的空間$\mathbb{R}^d$轉換到空間$\mathbb{R}^D$，並且$D >> d$。也就是說我們把原本的分類問題，打到更高維度的空間來解決。除此之外，我們定義kernel function為$k(x_j, x_i) = \Phi(x_j)^T \Phi(x_i)$，利用kernel function，我們可以延伸出更多的分類方法，假設γ是常數:

- Linear kernel $k(s, z) = s^T z$
- Polynomial kernel $k(s, z) = (1 + \gamma s^T z)^d$ for any $d > 0$
- Gaussian kernels (rbf) $k(s, z) = \exp(-\gamma\|s - z\|^2)$ for $\gamma > 0$

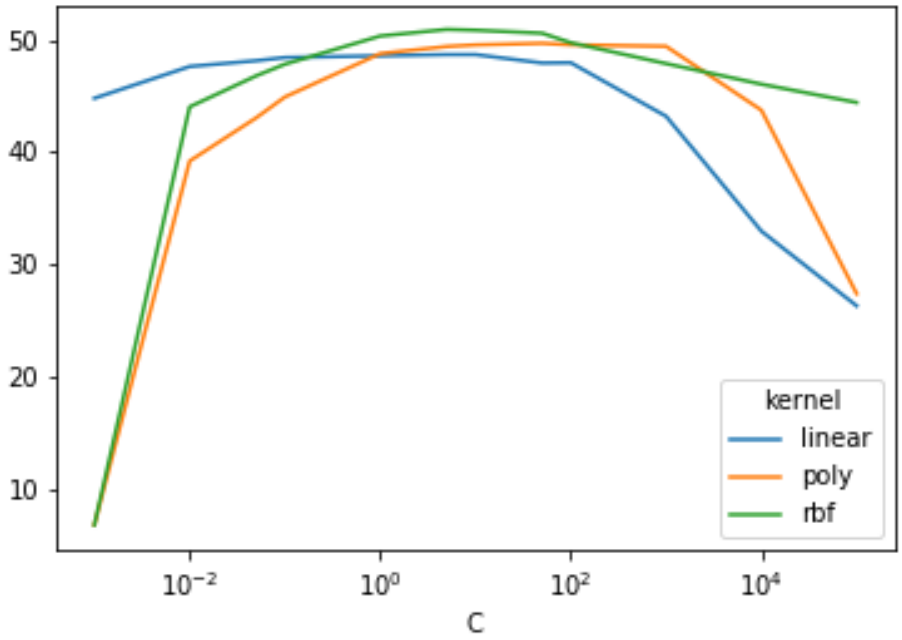
有了這些kernel function，我們更加能靈活的使用svm來分類。

然而，SVM其實是屬於一個二元分類器，也就是它原本的功用是在資料中間畫出分類的界線，並且說明在界線兩邊的資料是不一樣的。但是在我們先前討論的資料中，是有多達15個類別的，因此為了要讓SVM能夠處理多類別的資料分類問題，通常會使用兩種解法:

- One-against-Rest (One-against-All, OvA, OvR): 我們把題目變成分為A類別以及非A類別，那麼就會使得多元類別的問題降到二元類別。但是這種方法很容易產生資料不平衡。
- One-against-One (OvO): 我們就讓類別兩兩一組，並且做分類。假設我們有n個類別，則我們就需要做$C_2^n = \frac{n(n-1)}{2}$次的分類，也就是會有至少$O(n^2)$的時間複雜度，會增加計算的複雜度。但同時就不太會發生資料不平衡的問題。

在python中的sklearn裡面的SVM所用的都是OvO的策略，因此以下也都是用這個策略。

接下來我來講解我做SVM的過程，首先我把從unsupervised的三種方法中所做出來的結果，當作SVM的input。我把資料分成75%的training data和25%的test data，並且因為nmf中做出來的數值相差很大，所以就把所有的資料都標準化。再來我利用GridSearchCV去找出在polynomial kernel中，d 選多大最好，進而找哪個kernel和C是最佳的SVM參數。以下是:



由於三種資料得出的參數很類似，所以我以一張作為代表。x軸是C，y軸是從5次的cv所得到的accuracy rate。可以看到說，linear是平穩的在上面，直到100開始往下掉。poly和rbf(Gaussian)的變化很相近，都是從0.01開始升上去，在大約0.1-100之間有最高值，之後隨著C越大而慢慢下降。也因此SVM不能調C太大會太小的。

最後我們看到三種Unsupervised的方法下，並且SVM在經過GridSearchCV挑選最佳參數後，進而利用test data得出預測的accuracy rate.

	NMF	**LDA**	**Gibbs**
kernel	rbf	rbf	rbf
C	10	5	10
accuracy rate	0.605067	0.515733	0.5432

有趣的是，所有的方法都挑出了rbf (Gaussian)的方法來當作最佳的kernel，並且C也都使用5-10接近個位數的數值。所以其實對於SVM來說，在這個資料下，用哪一個unsupervised方法去做處理是沒差的。我們可以看到在accuracy rate中，NMF和先前所提到看到的結果一樣，由於TF-IDF的關係，所以NMF會相較於LDA盡量把相關且對topic重要的字集合在一起，同時也有可能是因為廣告的主題機率不太會變動，因此NMF較LDA好。而意外的是，利用Collapsed Gibbs Sampling的LDA，雖然在解釋topic上面比其他兩種都還要弱，但在預測能力上卻是居中。

Conclusion

這次報告主題是利用Amazon的商品廣告去幫助商品分類，我所使用的方法是利用LDA和NMF去建立topics，接著利用SVM去對topics產生的向量做分類。由於文字資料相較於數字資料的前處理更多也更複雜，因此我多用了Tokenize的方式，刪除了Stopwords，利用文字雲和N-grams的方式去做EDA，並且利用Countvectorize和TF-IDF的方式讓文字轉換成數字。經由此次報告，我得出以下的結論:

1. 解釋能力、建模能力和預測能力在這個例子中是不太相關的事: 2-gram在EDA中最好解釋，但perplexity並沒有比較好；Gibbs出來的結果是最難以分辨主題的，但預測能力中卻不是最差的。
2. NMF因為概念單純，所以可以接受較複雜的前處理，同時也表示topic的機率是固定的；LDA因為是3層模型，所以在前處理中就需要簡單一點，但同時能處理topic的機率一直變動的問題。在這個資料中，我認為因為每個廣告都只對應一個topic，所以topic的機率會是固定的，也會導致在這份分析中，NMF較LDA表現得好。
3. SVM的C值是個很難挑選的參數，C太大或太小都不行。
4. SVM在這個例子中，表現的並不太好，有可能是因為資料裡面許多的類別互相交錯，例如: 車用電子產品中有很多手機架，會和手機配備搞混，所以辨認的時候就算打到了高維度空間，也難以找到分界的超平面。