R04922092 鄭允中

2016年6月5日

Web Mining Report : Naive Bayes Classifier and EM Algorithm

labeled and unlabeled data on text categorization

The report contain the following content:

- ○ Describe Naïve Bayes Classifier
  - (ex: parameter's meaning in text categorization, and how to estimate them)
- ○ Describe EM algorithm
  - (ex: How to derive likelihood, E-step and M-step)
- ○ Results of Experiments
- ○ Result of 2 methods.
- ○ Analysis on data's size and performance.
- ○ Some techniques in implementation and their impact.

一、執行方式

本程式為R語言撰寫，使用R interpreter 即可執行，執行前須安裝以下兩個套件：

1. SnowballC : optimize parsing text file
2. tm : for parsing text file
3. Rcpp : for C++ code

進入 R 交互式介面執行 install.packages(“PACKAGE_NAME”) 即可安裝

```
r04922092@linux2 [~/text-mining-in-EM-algorithm] R

R version 3.3.0 (2016-05-03) -- "Supposedly Educational"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R 是免費軟體，不提供任何擔保。
在某些條件下您可以將其自由散布。
用 'license()' 或 'licence()' 來獲得散布的詳細條件。

R 是個合作計劃，有許多人為之做出了貢獻。
用 'contributors()' 來看詳細的情況並且
用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。

用 'demo()' 來看一些示範程式，用 'help()' 來檢視線上輔助檔案，或
用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
用 'q()' 離開 R。

> install.packages("Rcpp")
Installing package into '/nfs/master/04/r04922092/R/x86_64-pc-linux-gnu-library/3.3'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
HTTPS CRAN mirror

 1: 0-Cloud [https]                2: Austria [https]
 3: Belgium (Ghent) [https]        4: Chile [https]
 5: China (Beijing 4) [https]      6: Colombia (Cali) [https]
 7: France (Lyon 1) [https]        8: France (Lyon 2) [https]
```

執行時使用以下格式指令執行(範例)：

sh EM.sh -i ~/text-mining-in-EM-algorithm/test -o out.txt (-n 20)

二、Describe Naïve Bayes Classifier

參考 “Text Classification from Labeled and Unlabeled Documents using EM” paper 的實作方式，實作以下公式：

1. word probability

$$\hat{\theta}_{w_t|c_j} \equiv \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1+\sum_{i=1}^{|\mathcal{D}|} N(w_t,d_i)\mathrm{P}(y_i=c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N(w_s,d_i)\mathrm{P}(y_i=c_j|d_i)}$$

意義為在某一個主題下某字出現的機率。這是有smoothing的版本，多了1和|V|兩個參數可調節 smoothing 的影響力。

R code

```
all_wordcount_in_a_topic <- apply(tdm,2,sum)
V_len <- length(all_term)
word_in_a_class_prob <- log2(t(apply(tdm,1,function(x) (1+x) / (V_len
+all_wordcount_in_a_topic))))
```

2. The class prior probabilities

$$\hat{\theta}_{c_j} \equiv \mathrm{P}(c_j|\hat{\theta}) = \frac{1+\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(y_i=c_j|d_i)}{|\mathcal{C}|+|\mathcal{D}|}$$

意義為在某一個主題出現的機率。這也是有smoothing的版本，多了1和|C|兩個參數可調節 smoothing 的影響力。

R code

```
D_len <- sum(topic_doc_count)
C_len <- length(catego)
class_prior_prob <- (1+topic_doc_count) / (C_len+D_len)
```

3. 使用訓練好的 Naive Bayes 分類器

$$\begin{aligned}\mathrm{P}(y_i = c_j | d_i; \hat{\theta}) &= \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta})}{\mathrm{P}(d_i|\hat{\theta})} \\ &= \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_r;\hat{\theta})}.\end{aligned}$$

先算某主題出現該文章的機率，再用貝氏定理反轉成該文章是某主題的機率。

注意分母不用算，因為每一個要比較的機率值除的分母都一樣，而分子取log相加，不然會有超出精度的問題。

R code

```
log_for_product <- apply(word_in_a_class_prob, 2,
                         function(x) sum(query_term_long * x) + class_prior_prob)
```

三、Describe EM algorithm

EM type I : paper version

參考 “Text Classification from Labeled and Unlabeled Documents using EM” paper 的實作方式，實作以下流程：

- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents, $\mathcal{D}^l$, only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve, as measured by the change in $l_c(\theta|\mathcal{D};\mathbf{z})$ (the complete log probability of the labeled and unlabeled data, and the prior) (see Equation 10):
  - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled document, *i.e.*, the probability that each mixture component (and class) generated each document, $\mathrm{P}(c_j|d_i;\hat{\theta})$ (see Equation 7).
  - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

其中，

E-step，是將目前的分類器對未知分類的文件做猜測。

M-step，則是根據E-step的結果重新計算分類器。

最後的輸出是一個似然率最大的分類器。

$$\text{E-step: Set } \hat{\mathbf{z}}^{(k+1)} = E[\mathbf{z}|\mathcal{D};\hat{\theta}^{(k)}].$$

$$\text{M-step: Set } \hat{\theta}^{(k+1)} = \arg\max_\theta \mathrm{P}(\theta|\mathcal{D};\hat{\mathbf{z}}^{(k+1)}).$$

注意，這個EM公式其實是從M步驟開始，因為要先用已標記的數據訓練一個最原初的naive bayes 分類器。

[note] 本paper的EM公式 與 教授授課投影片的EM公式 不同

投影片公式沒有用到未標記數據，且EM是在執行分類階段才做，在標記的數據量充足時，type II EM公式有較好的效果，但在標記數據量不足時，type I EM公式有良好的優化效果。

EM type II : slide version

不使用未標記文件而將主題視為 M step 調整的參數，得到的EM公式如下：

$$Log-Likelihood: \log L(\lambda) = \sum_{w \in V} c(w,d) \log[\lambda p(w|\theta_1) + (1-\lambda) p(w|\theta_2)]$$

$$E-step: \quad p(z_w = 1|w) = \frac{\lambda p(w|\theta_1)}{\lambda p(w|\theta_1) + (1-\lambda) p(w|\theta_2)}$$

$$M-step: \quad \lambda^{new} = \frac{\sum_{w \in V} c(w,d) p(z_w = 1|w)}{\sum_{w \in V} c(w,d)}$$

Word	#	$P(w\|\theta_1)$	$P(w\|\theta_2)$	Init	Iteration 1		Iteration 2	
				$\lambda^{(0)}$	$P(z=1\|w)$	$\lambda^{(1)}$	$P(z=1\|w)$	$\lambda^{(2)}$
The	4	0.5	0.2	0.5	0.71	0.46	0.68	0.43
Paper	2	0.3	0.1		0.75		0.72	
Text	4	0.1	0.5		0.17		0.14	
Mining	2	0.1	0.3		0.25		0.22	
Log-Likelihood				-15.45	-15.39		-15.35	

四 、Results & Analysis

1. Precision

以下是 naive bayes 和 EM 的 Precision 比較

labeled count \ method	Naive bayes	EM
5	0.277	0.331
10	0.395	0.488
20	0.505	0.597
50	0.624	0.691
All	0.709	0.721, 0.740 *

[註] All labeled count 使用 slide version EM, 有較好的 Precision

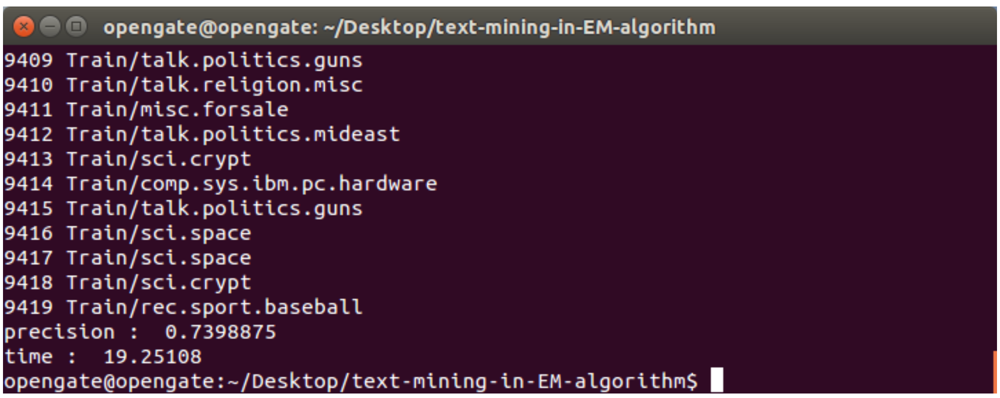
當labeled count 很少時，用EM來標記unlabeled data有很好的提高precision的效果。但當labeled data超過一定的量，已充足時，引進unlabeled的方式就沒有很好的提升效果，而type II 純使用labeled data的EM 反而效果更好 (如EM-All labeled count 欄位)。

2. Performance

當labeled數量過少時，EM可能因為迭代次數過多導致難以收斂，如 labeled count = 1 時。而當 labeled count 過大時，也可能因為 labeded + unlabeled term引進過多而使記憶體無法容納造成系統遲緩。

所有的data size 都能在20分鐘內跑完。

最慢的選項：all labeled/unlabeled data, (due to long term vector) : 19 min



五、Other techniques in implementation

使用低階語言提升效能

由於R語言的效率不彰，以下兩段程式碼port到c++實作再把結果傳回

1. 詞頻統計 : tm.cpp
2. type II EM : EM.cpp

成功將時間縮短至20分鐘內，效能大約差3倍。下面以 EM Algorithm 的 R 和 C++ 版本跑benchmark。

```
                                                             test
2 EM_AlgorithmCpp(word_freq = ShortVectoLong(all_term, query_term),
tau = rep(1/length(catego), length(catego)), tdm = tdm)
1    EM_Algorithm(word_freq = ShortVectoLong(all_term, query_term),
tau = rep(1/length(catego), length(catego)), tdm = tdm)
  replications elapsed relative
2          100   8.781    1.000
1          100  27.999    3.189
```