統計學習初論（105-2）

作業六

作業設計：盧信銘
國立台灣大學資管系

截止時間：2017 年 6 月 20 日上午 9 點

請至 RSAND 上批改，第一題範例命令為: sl_check_hw6q1 ./your_program，第二題為: sl_check_hw6q2 ./your_program。作業自己做。嚴禁抄襲。不接受紙本繳交，不接受遲交。請以英文或中文作答。

第一題

(50 points) 中文斷詞是處理中文文字資料的重要步驟。本題將以隱藏馬可夫模型 (Hidden Markov Model; HMM) 實做中文斷詞模組。我們使用的訓練資料為中研院製作的中文斷詞資料。本題所需的資料已經存放在 cwsas_train_v2.rdata 中。這個資料檔有四個變數，大略的說明如下:

- train_sent: 訓練資料集
- sample_sent: 小量測試資料集
- sample_sent2: 小量測試資料集
- test_sent: 大量測試資料集

本題主要會用到 train_sent，範例資料如下：

id	text2	be_tag	bmes_tag
1	時間：	BEB	BES
2	三月十日（星期四）上午十時。	BEBEBBEEBBEBEB	BEBESBMESBEBES
3	地點：	BEB	BES
4	學術活動中心一樓簡報室。	BEBEBEBEBEEB	BEBEBEBEBMES
5	主講：	BEB	BES
6	民族所所長莊英章先生。	BEEBEBEEBEB	BMEBEBMEBES

主要的欄位是：

- id: Sentence ID.
- text2: 句子。
- be_tag: 每個單字的 Tag。使用 B (單辭開始)與 E (單辭非起始字元)。

- bmes_tag: 每個單字的 Tag。使用 B(單辭開始)，M (非起始或結束字元)，E (結束字元)，S (長度為一個字的單詞)。

請寫一個名為 hmm_train 的函數，訓練一個 HMM 的斷詞模型。本題使用 BMSE 標記。這個函數不需要考慮其他種標記的處理。這個函數的傳入值為 (依此順序)：

1. sentvec: 訓練的句子。為一個 vector。
2. tagvec: 訓練句子的 BMSE 標記。長度應與 sentvec 一致。

為了方便中文的處理，我們將中文字與 BMSE 標記轉成整數。BMSE 對應的方式為 B:2, M:3, E:4, S:1。中文字轉換請使用 utf8ToInt()函數。比如說 (載入 cwsas_train_v2.rdata 之後)：

```
> print(sample_sent[12])
[1] "平常愛聽的音樂都停止放送"
> print(utf8ToInt(sample_sent[12]))
 [1] 24179 24120 24859 32893 30340 38899 27138 37117 20572 27490 25918 36865
```

在上面的例子中，"平常愛聽的音樂都停止放送"這個句子透過 utf8ToInt 函數轉換成一個整數向量，其中 24179 就是”平”的 UTF8 編碼，24120 就是”常”的編碼。你可以假設這個編碼的數值會介於 1 跟 70000 之間。

HMM 的訓練過程相當單純，大致上是計算標記與中文字的出現頻率。這個 hmm_train 函數的主要任務是計算以下事件的出現次數：

- S、B、M、E 的出現頻率。儲存在一個4×1的矩陣中。順序依照 S、B、M、E 對應的整數值。這個矩陣叫 tprior_count。
- S、B、M、E 的轉換頻率。儲存在一個4×4的矩陣中。順序依照 S、B、M、E 對應的整數值。這個矩陣叫 tseq_count。tseq_count 矩陣內的數值為由 row 至 column 的頻率。舉例而言。如果 tseq_count 的矩陣數值如下：

	S	B	M	E
S	3	7	0	0
B	0	0	0	0
M	0	5	0	0
E	0	0	0	0

  那代表在所有的標記中，如果出現 S，下一個也是 S 的次數是三次。S 接下來是 B 的次數是七次。M 接下來是 B 的次數是五次。其他的狀況都沒有發生。

- 字元與標記的聯合次數分配表。儲存在一個70000×4的矩陣中。這個矩陣叫 ct_count。每個 row 對應到一個中文字，由 utf8ToInt()決定 row index。每個 column 對應的一種標記，由 S、B、M、E 對應的整數值決定。這個矩陣中的每一個 cell 為某個字在訓練資料集中被標記為某個值的次數。舉例而言，下面的矩陣代表”放”在訓練集有 454 次被標記為 S，2 次被標記為 B，4 次被標記為 M，8 次被標記為 E。而”政”被標記為 S、B、M、E 的次數各為 4, 333, 1,1。

	S	B	M	E
⋮				
25918 (放)	454	2	4	8
25919 (政)	4	333	1	1
⋮				

hmm_train 函數的傳回值為一個 list，包含以下元素 (依此順序):

- ct_count: ct_count 矩陣。
- tseq_count: tseq_count 矩陣。
- tprior_count: tprior_count 矩陣。

以下為範例輸出入：

```
> #Sample 1
> model_s1=hmm_train(train_sent$text2[1:100],
train_sent$bmes_tag[1:100])
> print(model_s1$tprior_count)
     [,1]
[1,]  501
[2,]  504
[3,]  245
[4,]  504
> print(model_s1$tseq_count)
     [,1] [,2] [,3] [,4]
[1,]  188  213    0    0
[2,]    0    0  126  378
[3,]    0    0  119  126
[4,]  273  231    0    0
> print(model_s1$ct_count[65290:65300,])
      [,1] [,2] [,3] [,4]
 [1,]    0    0    0    0
 [2,]    0    0    0    0
 [3,]   61    0    0    0
 [4,]    0    0    0    0
 [5,]    0    0    0    0
 [6,]    0    0    0    0
 [7,]    0    0    0    0
 [8,]    0    0    0    0
 [9,]    1    2    0    1
[10,]    0    0    1    1
[11,]    0    0    0    1
> print(colSums(model_s1$ct_count))
[1] 501 504 245 504

> #Sample 2
> model_s1=hmm_train(train_sent$text2[101:500],
train_sent$bmes_tag[101:500])
> print(model_s1$tprior_count)
     [,1]
[1,] 2128
[2,] 2041
```

```
[3,]  301
[4,] 2041
> print(model_s1$tseq_count)
     [,1] [,2] [,3] [,4]
[1,]  794  934    0    0
[2,]    0    0  246 1795
[3,]    0    0   55  246
[4,] 1195  846    0    0
> print(model_s1$ct_count[65290:65300,])
      [,1] [,2] [,3] [,4]
 [1,]    0    0    0    0
 [2,]    0    0    0    0
 [3,]  253    0    0    0
 [4,]    0    0    0    0
 [5,]    0    0    0    0
 [6,]    0    0    0    0
 [7,]    0    1    2    1
 [8,]    0    2    1    0
 [9,]    0    1    2    0
[10,]    0    0    3    0
[11,]    0    0    0    0
> print(colSums(model_s1$ct_count))
[1] 2128 2041  301 2041
```

Evaluation: All credits will be given based on the correctness of 10 testing cases. Correct output in a case is worth 5 points.

第二題

(50 points) 有了前一題的訓練結果，我們在這一題針對輸入的句子進行標記的預測，並輸出分詞結果。請為這個任務設計 hmm_predict 函數。這個函數的傳入值為(依此順序)：

- model: 前題訓練完所輸出的資料結構。
- allsent: 要處理的句子，為一個 vector。
- sepchar: 分詞插入字元，預設是(半形)空白(" ")。
- addsmooth: Additive smoothing 所要加上的值。預設是 1。

針對 allsent 中的每一個句子，hmm_predict 會利用 Max-Sum Algorithm 計算最佳標記的預測值，並依此對這個句子斷詞，在適當的位置插入 sepchar。詳細的過程參照上課討論與投影片 。

製作函數輸出時請注意：

- 如果句子的長度是零 (也就是空字串)，斷詞的結果與標記請傳回空字串("")。
- 句子的最後不應有分隔符號 (sepchar)。

- ct(字與標記)的機率值計算法，請將 ct_count 矩陣所有元素加上 addsmooth 之後，除以所有 count 的合。這與投影片中的做法有些許不同。投影片中是先取有出現的字做一個小的 ct_count 矩陣之後才算機率。這個做法會讓斷詞結果有些微不同。

此函數的輸出為一個 list，包含以下元素 (依此順序):

- outsent: 為一個 vector。包含所有在適當位置插入 sepchar 之後的句子。順序應與 allsent 一致。
- outtag: 為一個 vector。包含所有句子的 BMSE 最佳標記。順序應與 allsent 一致。

以下為範例輸出入：

```
> #Sample 1
> load(file="cwsas_train_v2.rdata")
> model1=hmm_train(train_sent$text2, train_sent$bmes_tag)
> print(model1$tprior_count)
        [,1] radixsort
[1,] 2992288
[2,] 2454416
[3,]  464053
[4,] 2454416
> print(model1$tseq_count)
        [,1]    [,2]   [,3]    [,4]
[1,] 1216459 1069985      0       0
[2,]       0       0 324834 2129582
[3,]       0       0 139219  324834
[4,] 1448315 1005869      0       0
> print(model1$ct_count[65290:65300,])
        [,1] [,2] [,3] [,4]
 [1,]   86   22   10   21
 [2,]    2    4   19    4
 [3,] 456609    1    5    2
 [4,]   49  107 1381  105
 [5,] 6830    4  480   31
 [6,]    9   27  200    6
 [7,]   67  560 4286 2285
 [8,]  945 4148 1605  509
 [9,]  875 1415 1456  613
[10,]  756 1002 1046  370
[11,]  534  657 1093  440
> print(colSums(model1$ct_count))
[1] 2992288 2454416  464053 2454416
>
> out1=hmm_predict(model1, sample_sent)
> print(out1)
$outsent
 [1] "紀惠 容舉 例"
 [2] "某 大學 發生 狼 師性 侵女 學生"
 [3] "婦女 團體 在 協助 女學 生時"
 [4] "狼師 的 妻子 知道"
```

```
 [5] "馬上 告女 學生 通姦"
 [6] "最 後女 學生 不 但 沒有 找 回 公義"
 [7] "還 要 賠償 3 0萬 元"
 [8] "女 學生 沒錢"
 [9] "最後 協助 的 婦女 團體 一 人出 2萬 元"
[10] "女人 為 難女 人"
[11] "他 這 幾 天 病房 異常 安靜"
[12] "平常 愛 聽 的 音樂 都 停止 放送"
[13] "疑似 因為 豬 哥亮 身體 虛弱 說 話 音量 也 不 高"

$outtag
 [1] "BEBEB"              "SBEBESBEBEBE"       "BEBESBEBEBE"
 [4] "BESBEBE"            "BEBEBEBE"           "SBEBESSBESSBE"
 [7] "SSBESBES"           "EBEBE"              "BEBESBEBESBEBES"
[10] "BESBES"             "SSSSBEBEBE"         "BESSSBESBEBE"
[13] "BEBESBEBEBESSBESSB"

#Sample 2
> out2=hmm_predict(model1, sample_sent2)
> print(out2)
$outsent
 [1] "今" "。" "婦" "十" "馬" "最" "3"  "女" "最" "人" "他" "平" ","

$outtag
 [1] "B" "S" "B" "B" "B" "S" "S" "B" "S" "S" "S" "B" "S"

> #Sample 3
> out3=hmm_predict(model1, test_sent[851:900])
> print(out3)
$outsent
 [1] "美國 即 將 恢復 科技 進口 ，"
 [2] "可望 誘發 亞洲 出口 成長 的 第二 個 上升 波段 。"
 [3] "報告 表示 ，"
 [4] "亞洲 企業 獲利 尚 未 納入 科技 出口 加速 復甦 考量 ；"
 [5] "同時 ，"
 [6] "出口 復甦 將 可 支撐 亞洲 主要 科技 出口 國家 的 貨幣 升值 。"
 [7] "所 羅門 美邦 亞洲 經濟 學家 Cl i ff T an 在 報告 中 表示 ，"
 [8] "即 便 沒有 美國 與 日本 的 進口 需求 支撐 ，"
 [9] "亞洲 出口 成長 仍 於 二o o 一年 第四 季 反彈 回升 。"
[10] "他 表示 ，"
[11] "亞洲 出口 的 第一 波 反彈 波 反映 出新 的 區域 間 貿易 ，"
[12] "其 原因 可能 是 過去 兩 年 來 ，"
[13] "流入 中國 大陸 的 龐大 外國 直接 投資 （ ＦＤ Ｉ） ，"
[14] "可能 已 開始 改變 亞洲 地區 貿易 與 生產 模式 。"
[15] "報告 表示 ，"
[16] "所 羅門 美邦 的 分析 師 看法 、 半 導體 價格 與 股票 市場 俱已 提供 線
索 ，"
[17] "顯示 科 技業 已 在 反彈 。"
[18] "除 了 個體 經濟 與 市場 證據 之外 ，"
[19] "總體 經濟 亦 出現 美國 終於 開始 重新 回補 科技 產品 庫存 跡象 。"
[20] "報告 指出 ，"
[21] "去 年 十二 月 至 今年 一月 的 三 個 月間 ，"
```

```
[22] "美國 科技 產品 庫存 年率 開始 翻揚 ，"
[23] "而 在 同時 ，"
[24] "科技 產品 出貨 則 持續 大幅 成長 。"
[25] "這 意味 著 ，"
[26] "科技 出貨 反彈 的 初期 可能 是 由庫 存供 應 ，"
[27] "但 是 庫存 水 準現 在 已 消耗 殆盡 。"
[28] "而 且 ，"
[29] "所 羅門 美邦 所作 分析 顯示 ，"
[30] "美國 科技 庫存 趨勢 並 不 是 完全 由 半 導體 左右 ，"
[31] "其 他 科技 部門 庫存 可能 也 已 觸底 。"
[32] "所 羅門 美邦 表示 ，"
[33] "對 亞洲 而 言 ，"
[34] "重要 的 是 ，"
[35] "隨著 美國 科技 業重 新 建立 庫存 ，"
[36] "他們 將 必須 進口 科技 零組 件 。"
[37] "而 在 幾 個 月前 ，"
[38] "美國 科 技業 並 未 感受 到 這 種 急 迫性 。"
[39] "報告 表示 ，"
[40] "美國 科技 恢復 進口 需求 ，"
[41] "將 可 嘉惠 南韓 、 新 加坡 、 馬來 西亞 與 菲律 賓等 亞洲 科技 出口 國
家 。"
[42] "台灣 雖然 亦 可 受惠 ，"
[43] "惟 因 生產 作業 轉進 中國 大陸 ，"
[44] "刺激 效果 可能 要 打 折扣 。"
[45] "年輕 的 中華 男籃 隊率 先 飛抵 韓國 後 ，"
[46] "連日 來 繼續 接受 韓 國代 表隊 和 ＫＢ Ｌ職 籃勁 旅 的 震撼 教育 ，"
[47] "中韓 兩 國男 籃隊 在 第一 場 熱 身賽 短兵 相接 ，"
[48] "差距 高達 廿多 分 ，"
[49] "韓 國籃 球界 人士 建議 ，"
[50] "中華 男 籃隊 經驗 不足 ，"

$outtag
 [1] "BESSBEBEBES"                      "BEBEBEBEBESBESBEBES"
 [3] "BEBES"                            "BEBEBESSBEBEBEBEBEBES"
 [5] "BES"                              "BEBESSBEBEBEBEBEBESBEBES"
 [7] "SBEBEBEBEBEBESBESBESBESBES"       "SSBEBESBESBEBEBES"
 [9] "BEBEBESSBESBEBESBEBES"            "SBES"
[11] "BEBESBESBESBEBESBESBES"           "SBEBESBESSSS"
[13] "BEBEBESBEBEBEBESBEBES"            "BESBEBEBEBEBESBEBES"
[15] "BEBES"                            "SBEBESBESBESSBEBESBEBEBEBEBES"
[17] "BESBESSBES"                       "ESBEBESBEBEBES"
[19] "BEBESBEBEBEBEBEBEBEBEBES"         "BEBES"
[21] "SSBESSBEBESSSBES"                 "BEBEBEBEBEBEBES"
[23] "SSBES"                            "BEBEBESBEBEBES"
[25] "SBESS"                            "BEBEBESBEBESBEBESS"
[27] "SSBESBESSBEBES"                   "SSS"
[29] "SBEBEBEBEBES"                     "BEBEBEBESSSBESSBEBES"
[31] "SSBEBEBEBESSBES"                  "SBEBEBES"
[33] "SBESSS"                           "BESSS"
[35] "BEBEBEBESBEBES"                   "BESBEBEBEBESS"
[37] "SSSSBES"                          "BESBESSBESSSSBES"
[39] "BEBES"                            "BEBEBEBEBES"
```

```
[41] "SSBEBESSBESBEBESBEBEBEBEBEBES" "BEBESSBES"
[43] "SSBEBEBEBEBES"                 "BEBEBESSBES"
[45] "BESBEBEBESBEBESS"              "BESBEBESBEBESBEBEBESSBEBES"
[47] "BESBEBESBESSBEBEBES"           "BEBEBESS"
[49] "EBEBEBEBES"                    "BESBEBEBES"
```

Evaluation: All credits will be given based on the correctness of 10 testing cases. Correct output in a case is worth 5 points.