國立交通大學

資訊科學與工程研究所

碩士論文

基於中國餐廳過程之在線學習方法

Online Chinese Restaurant Process

研 究 生：蔡宗勳

指導教授：李嘉晃　教授 、 劉建良　博士

中 華 民 國　一〇二　年 九 月

基於中國餐廳過程之在線學習方法

Online Chinese Restaurant Process

研 究 生：蔡宗勳　　Student：Tsung-Hsun Tsai

指導教授：李嘉晃　　Advisor：Prof. Chia-Hoang Lee

共同指導：劉建良　　Co-Advisor：Dr. Chien-Liang Liu

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

September 2013

Hsinchu, Taiwan, Republic of China

中華民國一〇二年九月

基於中國餐廳過程之在線學習方法

學生：蔡宗勳　　指導教授：李 嘉 晃　教授

共同指導：劉 建 良　博士

國立交通大學

資訊科學與工程研究所碩士班

摘　要

目前各領域的資料已經漸漸成長為巨量資料，許多傳統的機器學習方法已經無法處理這些巨量資料。在線學習方法具備動態模型更新特性且一次只需將一筆資料載入記憶體做處理，可即時處理大量資料，因此為解決巨量資料的一個方法。此外，處理巨量資料時，要在訓練模型之前就事先決定參數是一件困難的事，往往只能透過專家經驗或實驗測試以得到模型參數；貝氏無母數模型提供了一個使群數參數能夠依資料特性自行決定的方法，適合用於巨量資料上。

中國餐廳過程早期是機率論上用來描述空間中一群切割之分佈的隨機過程，若將其對應至從 Dirichlet Process 取樣的一個過程，則可以從一個分佈取樣出多組參數，每一組參數又分別代表一個分佈。本論文提出的方法為將在線學習的概念擴展於中國餐廳過程上，並利用在線學習過程中的每一筆訓練資料來影響機率模型中參數的估計，進而建立出整個模型。在實驗中，當資料量大時，我們提出的 Online CRP 不僅在分類的效能上能夠達到監督式學習方法的標準，且在執行時間也比很多方法快速，驗證本方法可準確並有效率的處理巨量資料問題。

Online Chinese Restaurant Process

Student：Tsung-Hsun Tsai Advisor：Prof. Chia-Hoang Lee

Co-Advisor ： Dr. Chien-Liang Liu

Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

Abstract

The rise of big data provides an opportunity for the enterprises to use data analytics to gain competitive advantage, but it also brings challenges to process, manage and analyze the large data sets. One typical challenge is to process large volumes of streaming data in real time. Online machine learning allows the model to learn one instance at a time, in which the model is updated according to the prediction result and the true label of the instance. Compared with batch machine learning algorithms, online machine learning is more appropriate to process streaming data, and it can adjust learning model as receiving more new unknown data. Besides online processing, parameter selection is an important task in machine learning in dealing with model selection, but the task is generally achieved by heuristic rules or cross-validation technique with a validation set. In big data process, parameter should be adapted as with data rather than a fixed one. Nonparametric Bayesian model provides a means for the model to adapt parameters with the data. This study proposes an online Chinese Restaurant Process algorithm, which extended from Chinese Restaurant Process (CRP). The proposed algorithm is an online and nonparametric parameter algorithm, so it can process streaming data efficiently and the parameters are adapted with the data. Compared with CRP, the proposed algorithm is an online algorithm, in which we use regret theory to design a new prior knowledge and likelihood function based on the consistence between the real label information and prediction result. In the experiments, the proposed algorithm works well in large data set, and generally outperform the other online machine learning algorithms.

誌謝

首先，感謝指導教授李嘉晃老師以及共同指導教授劉建良博士對我的悉心指導，才能有今日的成果。接著要感謝兩位辛苦的口試委員，許尚華教授、周澤民博士，謝謝口委們的建議，讓本論文的內容能夠更加完整。

同時，我亦感謝這兩年來陪伴在我身邊的實驗室同學們、學長、學姊以及學弟。

最後，我要感謝我的爸爸、媽媽、哥哥，感謝你們對我的愛護和包容。謝謝你們在背後默默的支持，使我能夠順利的完成碩士學位。

心中有太多的感謝不知道如何表達，在此僅以本篇論文表示我對你們最誠摯的感謝，並祝福你們身體健康、萬事如意，謝謝。

蔡宗勳　謹誌
資訊科學與工程研究所
智慧型系統實驗室
中華民國一百零二年九月

目 錄

圖目錄

表目錄

第一章 緒論

1.1 研究動機

隨著資訊與科技的蓬勃發展，人們每天皆可在網路上得到相當多資訊，這些資訊已經漸漸成長為巨量資料。要對龐大的資料集做處理、分析及分類，已經不能以人工方式處理。目前許多領域都在研究如何使用機器以自動化方式處理龐大的資料，例如機器學習（Machine Learning）、資訊擷取（Information Retrieval）、圖形識別（Pattern Recognition）和自然語言處理（Natural Language Processing）等等，不管是對文字、圖片、聲音、或是影像，目的是希望透過電腦自動分析資料，得到準確的結果，也藉此節省人工處理所需之成本與時間。因此如何讓使用者快速和正確的得到所需資訊，是一項重要的研究議題。

近十幾年來，已經出現許多監督式學習(Supervised Learning)分類的方法，如 SVM[1]、Ada-boost[2]、Random Forest[3]、Logistic Regression[4]和 Naïve Bayes[5]等等，這些方法需要足夠標記資料訓練分類器，同時必須將所有標記資料一次載入機器做運算和處理；這些方法在有限運算能力的機器上，往往無法有效率處理這些無限且快速膨脹的資訊。

在機器學習方法中，機率模型是用來從觀察到的資料建立學習模型。傳統的參數模型(parametric model)使用固定且有限的參數，容易在模型(通常可用一些參數來表示)與資料之間產生 overfitting 或 underfitting 的問題，例如 Kmeans[6]、GMM[7]等方法，在分群或分類之前我們都必須先給定一個群數或類別數參數，但是此參數要設為多少，是一項困難的問題；而設定此參數又是一件很重要的事，就算我們先將訓練資料切一小塊交叉驗證測試資料集合來測試哪個參數較好，還是不容易找到一個最佳的參數(不會造成 overfitting 或 underfitting 的參數)。

參數模型中，通常針對想要解的問題，只能觀察到部分資料，若想估計資料的分佈，貝氏方法提供一個給定事前機率，計算這些資料產生分佈之事後機率的方法

來估計分佈參數。傳統上，在分佈上面的事前機率通常是從一定的範圍內給定，這限定了推論(Inference)的範疇。無母數方法提升了事前機率所給定的空間，通常是包含所有分佈的空間。因為所有分佈空間都已考慮在模型中，貝氏無母數(Bayesian Non-parametric)方法可避免參數選擇(parameter selection)和挑選模型(model selection)問題，而且參數可隨資料自行變動，可降低 overfitting 和 underfitting 的狀況，基於 Dirichlet Process[8] 的 Dirichlet Process Mixture Model 即是一個經典的貝氏無母數模型。中國餐廳過程(Chinese Restaurant Process)[9,10,11]描述了從一個 Dirichlet Process 中取樣出分佈參數的過程，以中國餐廳過程建構出來之混合模型(Mixture Model)又可以稱為中國餐廳過程混合模型(Chinese Restaurant Process Mixture Model)。

實務上，很多問題具備 Online 特性，在我們做完分析後給了預測，系統馬上可以得到正確的標記資訊(Label Data)，例如線上廣告點擊率、每天的氣候預測、streaming data 等等；傳統的監督式演算法，並不適合處理此類問題，因為監督式演算法需批次處理訓練資料，若訓練資料持續累積，模型必須重新訓練，需要花費大量時間，因此不適用於學習模型需持續根據資料作變動的應用上，甚至訓練資料量也會漸漸增多，終將會超過單機系統負荷能力。而目前為了處理上述問題主要有三個研究方向:

(1) 演算法平行化:將原本單機的演算法改為可分散式平行處理的演算法，例如將循序式演算法引入 MPI(Message Passing Interface)和 OpenMP 等概念。或是像由 NVIDIA 所推出之 CUDA(Compute Unified Device Architecture)，以及 Google 所提倡的 Map-Reduce[12]技術。

(2)隨機化:採用隨機的概念，針對原資料集一次隨機抽取一個來處理，直到達到收斂條件，因此，最後真正處理的資料量通常會只是全部資料的一個子集合，例如 Stochastic Gradient Descent[13]。

(3) 在線學習(online learning)[14]:一次只針對一筆資料來學習與調整模型參數，例如 online perceptron[15]。

本論文提出的方法保留了中國餐廳過程混合模型(Chinese Restaurant Process Mixture Model)的無母數特性，也保留了 Online learning 的特性，可有效處理巨量資料。

1.2 研究目的

傳統的許多監督式學習分類方法已經無法處理這些龐大的巨量資料，面臨的問題除了隨著訓練資料持續的新增，必須持續更新模型參數外，要在訓練模型之前，事先決定類別數參數也是一件困難的事，因此我們希望可以發展一個可以即時針對新增的訓練資料更新模型參數外，同時使模型依資料特性自行決定類別數參數，且能有效處理巨量資料的方法。

本論文的主要目的即是提出一個基於傳統中國餐廳過程(Chinese Restaurant Process)的全新 online learning algorithm。此演算法可以即時性的針對新增的訓練資料來更新模型，更可以用來處理巨量資料。

此外，對於新到的訓練資料，若以往沒有此類別的訓練資料，運用無母數方法的特性，本論文的演算法可以即時性的開拓訓練一個新類別並加入原來的模型，使之更具彈性。

1.3 論文架構

1. 緒論
2. 相關研究
3. Online Chinese Restaurant Process
4. 實驗結果
5. 未來展望
6. 參考文獻

第二章 相關研究

2.1 Online Learning

在線學習(Online Learning)是一種機器學習方法，其特性是每次只針對一筆資料即時地做預測分析，在取得該筆資料正確資訊後，利用預測的結果和正確的資訊對模型做參數調整。

傳統監督式學習方法(Supervised Learning)以批次(Batch)方式訓練學習模型，若加進新進資料調整模型(Model)，必須連同舊的訓練資料以及新進的訓練資料全部一起重新訓練出一個模型；因此當訓練資料數量越來越多，更新模型的時間也會隨之越來越長；除此之外，監督式學習方法還存在著一個問題，當訓練資料達到一定數量，必定會有單機硬體負載不了而無法訓練出模型的問題。

使用在線學習方法可以解決上面提到的兩個問題: 1.不須使用全部資料重新訓練模型。2.在線學習其一次只需要處理一筆資料，因此只要一筆資料的維度是機器可以負荷的，不會有單機硬體無法負載的問題。

由於近年來在線(Online)的概念非常的熱門，也因此很多論文的題目都會以在線(Online)命名，根據本研究之觀察，有些演算法通常是用 Online 來突顯模型參數會隨時間演變，但是並無關於用標記資料來對模型參數做更新的特性，例如 Online Latent Dirichlet Allocation [16]、Online Hierarchical Dirichlet Process [17]等等；而有些方法符合 Generic Online Learning Process 的精神，每次只針對一筆資料即時做預測分析，取得正確的資訊後，利用預測結果和正確資訊來對模型調整參數，例如前面提到的 Online Perceptron 以及 Online SVM[18]、Online Logistic Regression[19] 等等方法。Online Perceptron 是在空間中建立一條直線(分類器)，並利用其法向量和資料的內積將資料分成+1 和-1 兩個類別，其直線會隨著預測錯誤的向量做調整。Online SVM 衍生自傳統的 SVM，容許資料一筆一筆加入來修改 support vector 的反矩陣，並利用此反矩陣來將資料分成+1

和-1 二個類別；Online Logistic regression 衍生自 Logistic Regression，容許一次針對一筆訓練資料，對迴歸方程式的係數做蒙特卡羅取樣更新，使方程式可用來預測下一筆資料，其預測值會落在[0,1]之間。本論文所提之方法亦是基於 Generic Online Learning Process 原則下設計的。Generic Online Learning 的標準過程如下：

Generic Online Learning Algorithm

---

1. for t = 1,2,...
2. Get $x_t \in R^n$
3. Learner predicts x_t and attains prediction result $p_t \in R$
4. Receive correct answer $y_t \in R$
5. Update learning model base on p_t and y_t
6. end for

Line 1 代表資料一筆一筆進入迴圈中，Line 2 取得資料 x_t 的資訊，Line3 為訓練出來之模型對資料做預測，且得到預測結果 p_t ，Line 4 接收到相對於 x_t 之正確答案 y_t ， Line 5 使用預測結果 p_t 和正確答案 y_t 來更新學習模型。

2.2 貝氏定理與貝氏網路

2.2.1 貝氏定理

假設C_1，C_2，…，C_n是樣本空間(sample space) Θ 的分割，且有一觀察到的事件 A，則根據貝氏定理(Bayes theorem)；

$$P(C_i|A) = \frac{P(C_i)P(A|C_i)}{P(A)} = \frac{P(C_i)P(A|C_i)}{\sum_{i=1}^{n} P(C_i)P(A|C_i)} \propto P(C_i)P(A|C_i)$$

其中

A　　：觀察到的事件(Evidence)。

$P(C_i)$　　：事前機率(Prior Probability)

$P(A|C_i)$　　：可能性(Likelihood)

$P(C_i|A)$　　：事後機率(Posterior Probability)

事前機率(Prior)又稱為古典機率，其代表在還未觀察到其他事件之前，因某些事前知識(Prior Knowledge)所認知的某一事件 C_i 所發生之機率；事後機率(Posterior)則假設我們觀察到某一事件 A 發生後，產生某一事件 C_i的機率;可能性估計(Likelihood)又稱為樣本機率(Sample Probability)，由於 A 是我們觀察到的事件，因此可以計算事件 C_i 產生此事件的機率。

在可能性估計中，針對每一個$P(A|C_i)$，我們通常用概似函數(likelihood function) $L(A,C_i)$來表示$P(A|C_i)$，估算概式函數的方法常利用對概似函數微分求極值來得到此機率值,此方法稱作最大可能性估計(Maximum Lilelihood Estimate，簡稱 MLE)；當使用 MLE 算出$P(A|C_i)$之後，且對某一事件C_i 所發生之機率有足夠事前知識來定義事前機率(Prior)時(也就是已知$((P(C_i))_{i=1}^{n}$)，我們便可以計算觀察到事件 A，發生某事件的事後機率(Posterior)。由於樣本空間中，共有 C_1，C_2，…，C_n個事件，因此我們會算出 n 個 $((P(C_i|A))_{i=1}^{n}$ ，直接挑機率最大之

事件的方法，又稱作最大事後機率(Maximum a Posterior，簡稱 MAP)。

若不採用 MAP，也可以搭配取樣(Sample)方法，從 n 個 $C_i|A$ 事件來挑一個事件。取樣方法又可分為很多種，例如簡單隨機抽樣 (Simple Random Sampling)、系統抽樣(systematic sampling)、分層抽樣(stratified sampling)、群集抽樣(cluster sampling)等等。

2.2.2 貝氏網路

貝氏網路(Bayesian network)，又可以稱信念網路(belief network)或有向非循環圖形模型(directed acyclic graphical model)。貝氏網路是一種機率圖型模型(Probability Graph Model)[20]，藉由有向非循環圖形(directed acyclic graphs)中得知隨機變數(random variable)及其相對應的條件機率分配(conditional probability distributions)。

一般而言，貝氏網路中的節點表示隨機變數，它們可以是可觀察到的變量、潛在變量，或是未知參數等等。連結兩節點的箭頭代表兩個隨機變數彼此間具有因果關係或是非條件獨立的；若節點中變數間不存在一條路徑，其隨機變數彼此間不存在因果關係或是條件獨立的。若兩個節點間以一個單箭頭連接在一起，表示其中一個節點是 parents，另一個是 descendants 或 children，兩節點就會產生一個條件機率值。下圖為一個貝氏網路的例子

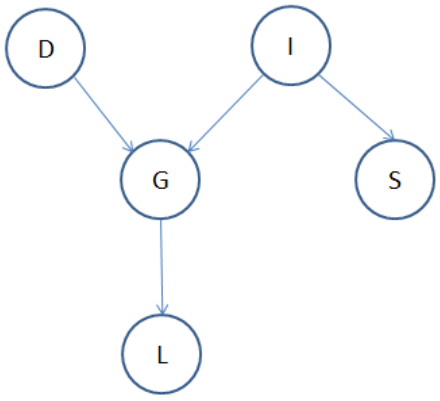


圖　2.2-1: 一個貝式網路例子，每個圓圈內的字母代表一個隨機變數

上圖中，每個圓圈內的字母代表一個隨機變數，由此圖我們可以輕易的推導出這些事件的聯合分佈機率如式子(2.1):

$$P(D,I,G,S,L) = P(D)P(I)P(G|D,I)P(S|I)P(L|G) \qquad (2.1)$$

2.3 Dirichlet Process

2.3.1 Dirichlet Distribution

在瞭解 Dirichlet Process 之前，本論文先介紹 Dirichlet Distribution，其定義如下:

若 $\sum_{i=1}^{k}\pi_i = 1$ ，且對所有的 i=1, 2, ⋯, k ，$\alpha_i > 0$ ，$\pi_i > 0$，且 $\pi_1,\dots,\pi_k$的聯合機率密度函數(joint probability density function)為:

$$p(\pi_1,\dots,\pi_k) = \frac{\Gamma\left(\sum_{i=1}^{k}\alpha_i\right)}{\prod_{j=1}^{k}\Gamma(\alpha_j)}\prod_{j=1}^{k}\pi_j^{\alpha_j-1}$$

稱$(\pi_1,\dots,\pi_k)$是一組 Dirichlet Distribution，記作$(\pi_1,\dots,\pi_k)$~ $\mathrm{Dir}(\alpha_1,\dots,\alpha_k)$。

令 $\alpha_0 = \sum_{i=1}^{k}\alpha_i$ ， 其期望值和變異數如下:

$$\mathrm{E}[\pi_i] = \frac{\alpha_i}{\alpha_0}$$

$$\mathrm{Var}[\pi_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$$

Dirichlet Distribution 的主要性質如下:

1. 結合性(Merge): 假設$(A_1, \ldots, A_k)$ 是 (1, ⋯, n)的切割，若$(\pi_1, \ldots, \pi_k)$ ~ $\mathrm{Dir}(\alpha_1, \ldots, \alpha_k)$ ，則$(\sum_{i\in A_1}\pi_i, \ldots, \sum_{i\in A_k}\pi_i)$ ~ $\mathrm{Dir}(\sum_{i\in A_1}\alpha_i, \ldots, \sum_{i\in A_k}\alpha_i)$

2. 可切割性(Split): 假設$(\pi_1, \ldots, \pi_k)$ ~ $\mathrm{Dir}(\alpha_1, \ldots, \alpha_k)$，若 (p_1, p_2)~$\mathrm{Dir}(\alpha_1\beta_1, \alpha_1\beta_2)$，$\beta_1 + \beta_2 = 1$ ，則$(\pi_1 p_1, \pi_2 p_2, \ldots, \pi_k)$ ~ $\mathrm{Dir}(\alpha_1\beta_1, \alpha_1\beta_2, \ldots, \alpha_k)$

3. Multinomial 分佈的共軛先驗分佈(Conjugate prior)

   因此若$(\pi_1, \ldots, \pi_k)$ ~ $\mathrm{Dir}(\alpha_1, \ldots, \alpha_k)$，且 z ~ $\mathrm{Multinomial}(\pi_1, \ldots, \pi_k)$

   則 p(π | z) ~ $\mathrm{Dir}(\alpha_1 + \delta_1(z), \ldots, \alpha_k + \delta_k(z))$，其中$\delta_i(z)$代表若 z=i 時，

   $\delta_i(z)$=1，否則$\delta_i(z)$=0。

2.3.2 Dirichlet Process

2.3.2.1.定義

Let Θ be a measurable space, with a probability measure G_0 on the space. Let α be a positive real number. A *Dirichlet process* DP(α, G_0) is defined to be the distribution of a random probability measure G over Θ such that, for any finite

measurable partition $(A_1, A_2, \ldots, A_n)$ of Θ, the random vector ($G(A_1)$, ... , $G(A_n)$) is distributed as a finite-dimensional Dirichlet distribution with parameters ($\alpha G_0(A_1)$, ... , $\alpha G_0(A_n)$):

$$(G(A_1) , \ldots , G(A_n)) \sim \mathrm{Dir}(\alpha G_0(A_1) , \ldots , \alpha G_0(A_n))$$

We write $G \sim DP(\alpha, G_0)$ if G is a random probability measure with distribution given by the Dirichlet process.

此定義為Ferguson於1973年提出，其主要是用來描述分佈的分佈，後來陸續有學者使用了不同的觀點來看DP，如Sethuraman (1994)使用 stick-breaking construction[21] 來建構一個DP；Blackwell and MacQueen 在1973年時提出 Pòlya's urn scheme[22]來描述從DP中取樣一個分佈參數的過程等等。其中Pòlya's urn scheme又可以對應成隨機過程(stochastic process)中的中國餐廳過程。下面幾個小節將會介紹這些方法。

2.3.2.2. stick-breaking construction

stick-breaking construction為 Sethuraman 於1994年提出，其建構DP方法如下:

假設$(\beta_k)_{k=1}^{\infty}$ 和 $(\phi_k)_{k=1}^{\infty}$ 為彼此獨立的隨機變數，且

$$\boldsymbol{\beta} \,|\alpha_0 \text{，} G_0 \sim \mathrm{Beta}(1, \alpha_0)$$

$$\phi_k|\alpha_0 \text{，} G_0 \sim G_0$$

那麼就可以定義G如下:

$$\pi_k = \beta_k \prod_{l=1}^{k-1}(1-\beta_l) \qquad 2.2$$

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, where\ \delta_{\phi_k} = \begin{cases} 1, if\ \phi_k = k \\ 0, other \end{cases} and \sum_{k=1}^{\infty} \pi_k = 1 \qquad 2.3$$

式子 2.2 是由 Griffiths, Engen, and McCloskey 三個人所定義的，因此由此

式建構出來的 π 又可直接寫作 π ~ GEM(α_0)，GEM 為 Griffiths, Engen, and McCloskey 三人的名字縮寫。

2.3.2.3. Pòlya's urn scheme

Pòlya's urn scheme 為 Blackwell and MacQueen 在 1973 提出。假設 G~DP(G_0, α)，此方法對應的不是 G 本身，而是描述從 G 取樣的一個過程。其詳細描述如下:

假設 G~DP(G_0, α)，且 θ_1，θ_2，…，θ_{i-1} 是從 G 中取樣出來的彼此獨立的隨機變數，在給定θ_1，θ_2，…，θ_{i-1} 的情況下求 θ_i 的條件機率為

$$\theta_i|\,\theta_1,\theta_2,\ldots,\theta_{i-1},\alpha,G_0 \sim \sum_{l=1}^{i-1}\frac{1}{i-1+\alpha}\delta_{\theta_l}+\frac{\alpha}{i-1+\alpha}G_0$$

$$where\ \delta_{\theta_l}=\begin{cases}1, if\ \theta_l=l\\ 0, else\end{cases} \tag{2.4}$$

式子(2.4)可以想像成從一個箱子抽球，第 i 次抽出的球便是 θ_i，可以將 θ_i想像成顏色。抽到什麼顏色的球，我們就多放一顆同顏色的球到箱子裡，同時也有一定的機率抽一種新的顏色的球放入箱子。

若不計算球是第幾次被抽到，只計算抽到的顏色的次數，從式子 2.1 可以知道，假設目前箱子裡面有 K 種顏色，m_k是箱子中每種顏色球的數量，ϕ_k*代表一種顏色*，我們可以將式子簡化如下:

$$\theta_i|\,\theta_1,\theta_2,\ldots,\theta_{i-1},\alpha,G_0 \sim \sum_{k=1}^{K}\frac{m_k}{i-1+\alpha}\delta_{\phi_k}+\frac{\alpha}{i-1+\alpha}G_0$$

$$where\ \delta_{\phi_k}=\begin{cases}1, if\ \theta_i=\phi_k\\ 0, else\end{cases} \tag{2.5}$$

由式子 2.5 可以看出，若箱子內某種顏色的球較多，抽中此顏色的球機率就比較大，而且抽中後又會增加下一次抽中此顏色的機率，造成了大者恆大的現象。式子 2.5 也展示了 DP 是帶有群的性質的，此式子與 2.4 要介紹的中國餐廳過程有著

極大的關聯。

2.3.3 Dirichlet Process Mixture Model

Dirichlet process mixture model (DPMM)可以想成是將Finite Mixture model(FMM)延伸成Infinite Mixture Model。因此本論文先介紹FMM。

一個FMM可用graphical model 來表示，其表示法如下圖:

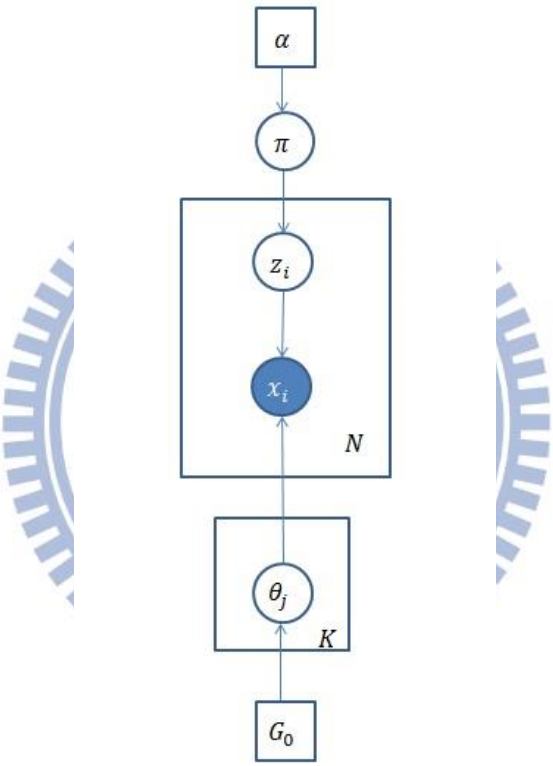


圖 2.3-1 : Finite Mixture Model

上圖中，若以生成模型的角度來看，在給定G_0, α 後，我們可以先取樣出$\boldsymbol{\pi}$，得到$\boldsymbol{\pi}$之後可以取樣出一個 z_i，假設取樣到的z_i值為j，接著我們可以從 G_0取樣出θ_j，由於在此我們沒有限定θ_j是什麼分佈的參數，因此不失一般性用$F(\theta_j)$來代表其分佈，因此給定 θ_j 後，x_i 將服從$F(\theta_j)$。完整取樣與生成過程可用以下式子代表。

$$\boldsymbol{\pi}\,|\alpha_0 \sim \mathrm{Dir}(\frac{\alpha}{K},\ldots,\frac{\alpha}{K})$$

$$z_i|\boldsymbol{\pi} \sim \boldsymbol{\pi}$$

$$\theta_{z_i} | G_0 \sim G_0$$

$$x_i | z_i, \{\theta\}_{k=1}^{K} \sim F(\theta_{z_i})$$

當我們假設K是無窮大的時候，FMM就變成了DPMM，其graphical model如下圖：

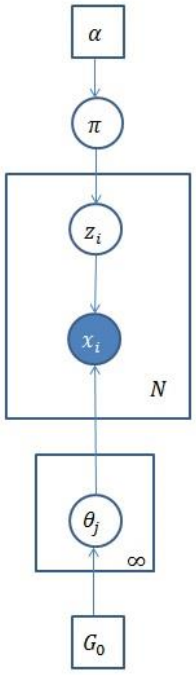


圖　2.3-2：Dirichlet Process Mixture Model-Infinite Mixture Model 表示法

DPMM的生成過程和FMM是完全一樣的，只有在取樣θ_j時原本有K種可能，現在變成無窮多種可能。因此，在DPMM中，將FMM中 π 的 Dirichlet prior，改成服從 stick-breaking construction 中 π 的建構方法，也就是說，$\pi \sim GEM(1,\alpha)$。完整取樣與生成過程可用以下式子代表。

$$\boldsymbol{\pi} \sim GEM(1,\alpha)$$

$$z_i | \boldsymbol{\pi} \sim \boldsymbol{\pi}$$

$$\theta_k | G_0 \sim G_0$$

$$x_i | z_i, \{\theta\}_{k=1}^{K} \sim F(\theta_{z_i})$$

如果我們沒有使用indicator變數z_i的話，且沒有想要明顯的表達一個群的參數，

我們也可以將 DPMM 轉換成 Pòlya’s urn 的表示法，如下圖:

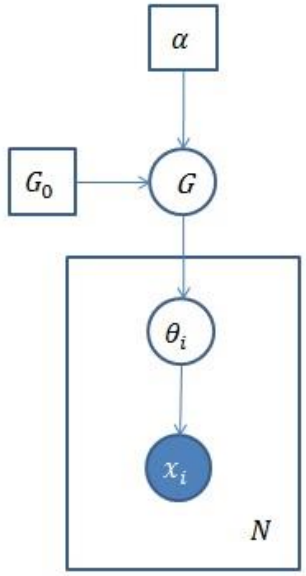


圖 2.3-3: DPMM – Pòlya’s urn 表示法

上圖中，若以生成模型的角度來看，在給定G_0, α 後，我們可以取樣出一個滿足DP的G，得到G之後，我們可以根據G來取樣θ_i，得到θ_i後，由於在此我們沒有限定θ_i是什麼分佈的參數，因此不失一般性用$F(\theta_i)$來代表其分佈，因此給定 θ_i 後，x_i 將服從$F(\theta_i)$。完整取樣與生成過程可用以下式子代表。

$$G \sim DP(G_0, \alpha)$$

$$\theta_i | G \sim G$$

$$x_i | \theta_i \sim F(\theta_i)$$

若我們觀察到一些資料$x_1, x_2, \ldots, x_N$，可以利用這些資料來做 inference。DP 無法直接使用 EM 演算法[23]來估算參數；因此 Neal(2000)[24]針對八種情形，包含直接對參數做 Gibbs Samling、G_0和取樣出來θ_i互相共軛(Conjugate)、以及互相不是共軛等等情形，提出了八個 algorithm，詳細的介紹如何使用 Gibbs Sampling 的方法來估計模型參數。後續還有 Michael Jordan(2004)[26]提出使用 Variational 方法來估計 DP 的模型參數。

2.4 Chinese Restaurant Process

中國餐廳過程(CRP)是一種隨機過程(stochastic process)，主要是用顧客到餐廳用餐選桌入座的情景,描述了一個空間上的切割(partition) $A_1, A_2, \ldots, A_K$的分佈，下面幾個小節將有詳細的介紹。

2.4.1 符號定義

在本小節我們先定義一下符號及其對應項目，以方便後面介紹中國餐廳過程(Chinese Restaurant Process)，其符號對應表如下：

表 1：CRP 符號對應表

符號	意思
x_i	第 i 個顧客
z_i	第 i 個顧客坐在第幾桌
m_k	第 k 桌的人數
θ_j	第 j 桌的菜色
α	調控開新桌比例之參數
G_0	base distribution
k	目前桌子有人的桌數
K	餐廳總桌數，趨近於 ∞
$H(x_i, \theta_j)$	第 i 個人對第 j 桌菜色的偏好程度

2.4.2 中國餐廳過程

中國餐廳過程(Chinese Restaurant Process)最早源自於對舊金山的中國餐廳

的假想：假設某餐廳裡面，有無限個桌子，而每個桌子有無限個座位，可以坐無限多個顧客。根據這個假設，又可以分成兩種情景。

情景(1)：

第一個顧客進餐廳之後選定某桌子。之後的每個顧客進餐廳後，會去看看自己是不是要加入已經有人坐的桌子，顧客也可以選擇自己開新桌。顧客選擇已經有人坐的桌子之機率會正比於該桌的已就座人數(m_k)，而顧客自己開新桌的機率會正比於某定值 α，其機率公式如下：

$$P(z_i = j \mid z_{-i}, \alpha) \propto \begin{cases} \frac{m_j}{i-1+\alpha}, & if\ j \le k \\ \frac{\alpha}{i-1+\alpha}, & if\ j = k+1 \end{cases} \qquad (2.6)$$

其示意圖如下圖:

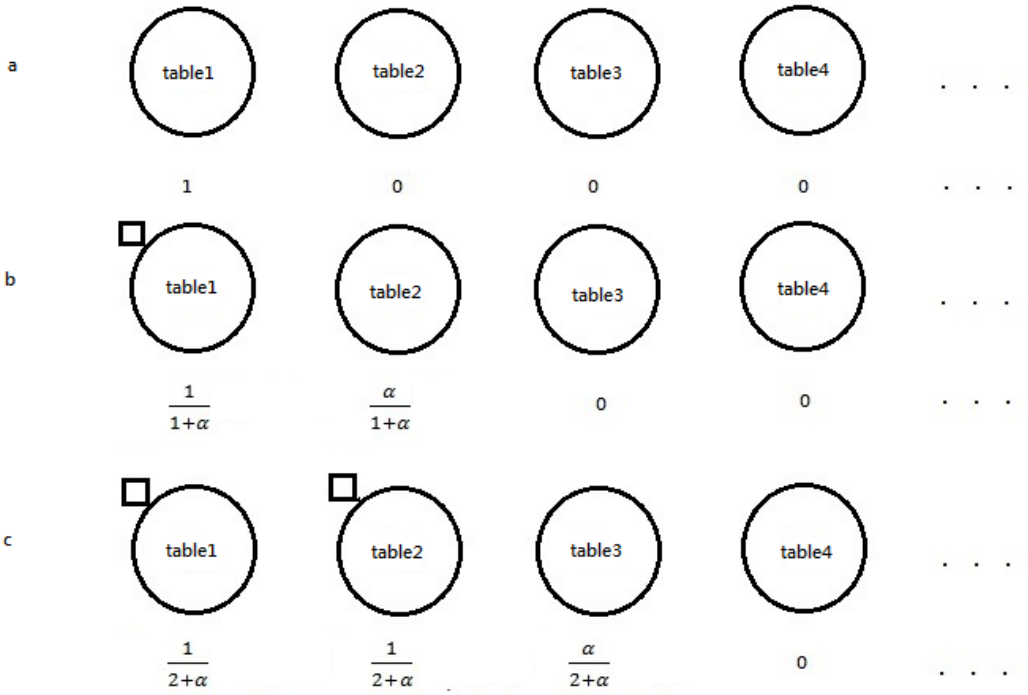


圖　2.4-1: CRP 情景(1) 示意圖：

不失一般性假設開桌順序為 table1、table2、…。 (a) 在都還沒有人進餐廳時，顧客開新桌的機率

為 1。(b) 在(a)之後，顧客坐於第一桌，下一位顧客選擇 table1 的機率為 1/ (1+α)， 選擇開新桌的機率為 α / (1+α) 。(c) 假設在(b)，第二位顧客選擇了 table2，下一位顧客選擇 table1 的機率為 1/(2+α)，選擇 table2 的機率為 1/(2+α)，選擇開新桌的機率為 α/(2+α)。

假設顧客群是 x_1、x_2、…、x_N，中國餐廳過程情景(1)最主要目的便是提供一個顧客選擇進入各桌的機率，它間接地說明了這 N 個人分散於各桌的情形。而當有新的顧客進來時，便可以參考此分佈來考慮要選擇哪一個桌子。

此過程同時也是一個無母數的演算法，也就是分群的時候不必限定群數 k，讓資料自己判斷有幾群，且不管目前資料有幾群，永遠存在一個選擇全新類別的機率。

中國餐廳過程又可以更進一步的假設，若顧客入座後，這些彼此同一桌的顧客便會依照他們自己的特徵來形成一個特徵分佈，此時，可以把此分佈比喻為每桌的菜色。也就是接下來要談到的第二個情景。

情景(2)：

第一個顧客進餐廳之後選定某桌子，並點菜。之後的每個顧客進餐廳後，會去看看已經有人在的桌子，如果那桌的菜自己喜歡吃，就選擇坐那一桌，如果每桌的菜都不是自己喜歡吃的，顧客也可以選擇自己開新桌。另外顧客選擇已經有人坐的桌子之機率會正比於該桌的已就座人數(m_k)及對該桌菜色的偏好程度，顧客自己開新桌的機率會正比於某定值α及對平均菜色的偏好程度。

$$P(z_i = j \mid z_{-i}, x_i, \theta, G_0, \alpha) \propto \begin{cases} \frac{m_j}{i-1+\alpha} H(x_i, \theta_j), & if\ j \le k \\ \frac{\alpha}{i-1+\alpha} \int H(x_i, \theta_j) dG_0(\theta_j), & if\ j = k+1 \end{cases} \quad (2.7)$$

其示意圖如下圖:

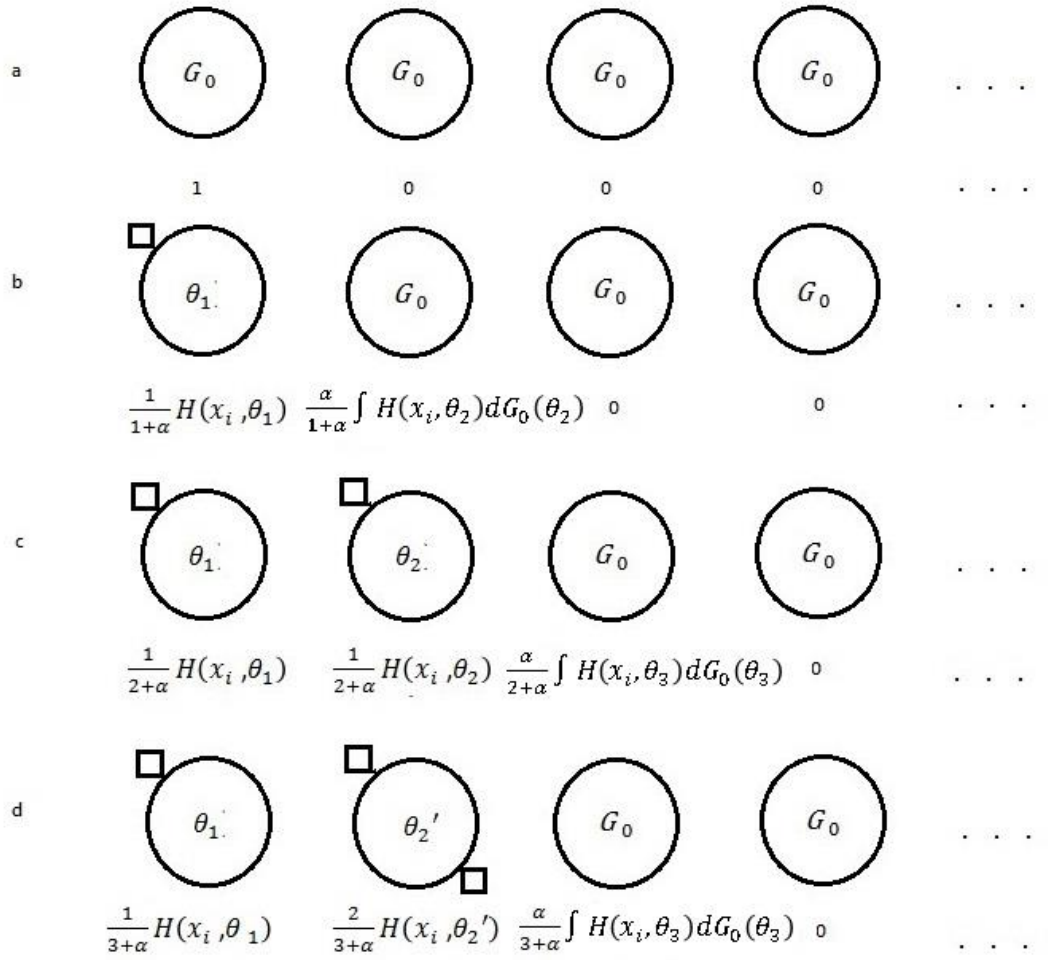


圖 2.4-2 : CRP 情景(2) 示意圖:

不失一般性假設開桌順序為 table1、table2、…。 (a) 在都還沒有人進餐廳時，顧客開新桌的機率為 1。(b) 在(a)之後，顧客坐於第一桌，下一位顧客選擇 table1 的機率為(1/ (1+α)) * H(x_2,θ_1)， 選擇開新桌的機率為 (α / (1+α)) * $\int H(x_2,\theta_2)dG_0(\theta_2)$ 。(c) 假設在(b)，第二位顧客選擇了 table2，下一位顧客選擇 table1 的機率為 (1/ (2+α)) * H(x_3,θ_1)，選擇 table2 的機率為(1/ (2+α)) * H(x_3,θ_2)，選擇開新桌的機率為(α / (1+α)) * $\int H(x_3,\theta_3)dG_0(\theta_3)$。 (d) 假設在(c)，第三位顧客選擇了 table2，下一位顧客選擇 table1 的機率為 (1/ (3+α)) * H(x_4,θ_1)，選擇 table2 的機率為(2/ (3+α)) * H(x_4,θ_2')，選擇開新桌的機率為(α / (3+α)) * $\int H(x_4,\theta_3)dG_0(\theta_3)$，此時的$\theta_2'$是根據$x_2$、$x_3$和$G_0$的事後機率所取樣得到的。

同樣的，假設顧客群是 x_1、x_2、…、x_N，中國餐廳過程情景(二)的目的，也是提供顧客到各桌的機率。此機率算法除了考量各桌顧客的分佈以外，也需要考量每一桌的特徵分佈，也就是每一桌的菜色。如此一來，新的顧客進入餐廳後，選定桌子前，便會把顧客入桌後的機率分佈當成事前機率(prior)來考慮，除此之外，

還會衡量自己喜歡該桌的菜色的程度，換句話說，就是會依該桌特徵分佈，衡量一下可以滿足該顧客所擁有的特徵的可能性(likelihood)。

2.4.3 Graphical Model and Gibbs Sampling

CRP 在情景(2)的過程，在數學上可以建構出一個和 Dirichlet process mixture model 等價的分佈，我們稱此為 CRP Mixture Model。

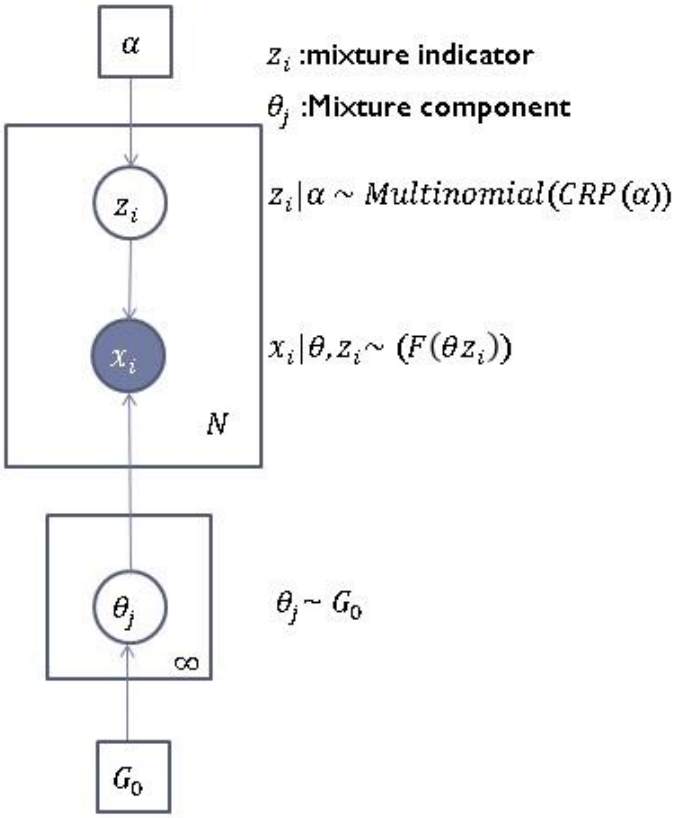


圖　2.4-3 : Graphical Model: CRP Mixture Model

圖 2.4-3 是 CRP Mixture Model 的 graphical model 表示法，其參數估計可以使用 Gibbs Sampling 來實作。Gibbs Sampling 的詳細步驟如下：

假設任意初始化 z_1、z_2 、 … 、z_N

step1: For i = 1 ,…, N :將 x_i 移出 z_i 桌，若 z_i桌只有 x_i 自己，則從當前的狀態移除整個 θ_{z_i}，移除後，再重新根據式子 2.7 取樣 z_i，若 z_i桌只有 x_i 自己，

則從 G_0 當中取樣出一個 θ_{z_i}

step2: For i = 1,…,k :重新根據 prior G_0 和 第 i 桌的資料分佈來取樣一個新的 θ_i

做完步驟 1、2，即代表完成了一回合的 Gibbs Sampling，若要取得較可靠的隨機變數估計值，須至少重複步驟 1、2 到達一定回合數量才可，此即 Neal(2000) 的 Algorithm 2[24]。

2.4.4 Collapsed Gibbs Sampling

由於我們在取樣一個分佈的時候，本身就必須是從一個分佈所取樣出來的，因此若我們取樣出來的分佈和我們從分佈中取樣時的分佈剛好有共軛性(Conjugate)的話，我們便可以將式子 2.7 的 likelihood 部分對θ做積分變成：

$$P(z_i = j \mid z_{-i}, x_i, \theta, G_0) \propto \begin{cases} \frac{m_j}{i-1+\alpha} \int H(x_i, \theta_j) dF(\theta_j)_{-i}, if\ j \le k \\ \frac{\alpha}{i-1+\alpha} \int H(x_i, \theta_j) dG_0(\theta_j),\ if\ j = k+1 \end{cases} \quad (2.8)$$

式子 2.8 中，$F(\theta_j)_{-i}$是計算根據觀察到 G_0，$\sum_{a:(z_a=j)\wedge(a\neq i)} x_a$ 這些資訊所產生 θ_j 的事後機率，此即相當於對圖 2.4.3 的 graphical model 做 Collapsed Gibbs Sampling，原來的 graphical model 便可以簡化成下圖:

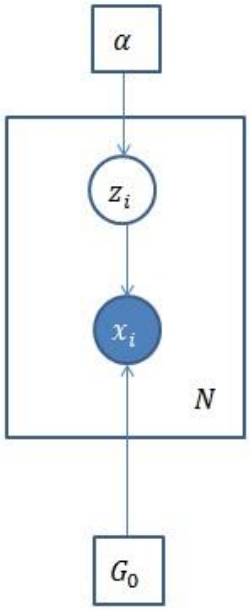


圖　2.4-4 : Collapsed Gibbs Sampling on CRP mixture model

Collapsed Gibbs Sampling 的詳細步驟將會變動如下:

任意初始化 z_1、z_2 、 … 、z_N

step1: For i = 1 ,…, N :將 x_i 移出 z_i 桌，若 z_i桌只有 x_i 自己，則從當前的狀態移除整個 θ_{z_i}，移除後，再重新根據式子 2.8 取樣 z_i，若 z_i桌只有 x_i 自己，則從 G_0 當中取樣出一個 θ_{z_i}

做完步驟 1，即代表完成一回合的 Collapsed Gibbs Sampling，若要取得較可靠的隨機變數估計值，須至少重複步驟 1 到達一定回合數量才可，此即 Neal(2000) 的 Algorithm 3[24]。

2.5 Distance Dependent Chinese Restaurant Process

近年以 CRP 為基礎深入研究的有美國普林斯頓大學的教授 David Blei 及其學生 Frazier 所發表的 Distance Dependent Chinese Restaurant Process (DDCRP)[26]，該概念是從傳統的 CRP 做衍生，傳統的 CRP 為顧客選桌子時和每桌顧客人數成比例，而在 DDCRP 中，顧客選擇考量的是和其他顧客同桌的機率，並使用顧客間之距離當成 prior，其機率公式如下:

$$P(z_i = z_j \mid z_{-i}) \propto \begin{cases} f(d_{ij}), & if\ j \le k \\ \alpha, & if\ j = k+1 \end{cases} \quad (2.9)$$

其中 $f(d_{ij})$ 是顧客 i 和 j 的距離函式(decay function)，例如$f(d_{ij})$ = exp(-(d_{ij}/a))。

後續有 Christopher D. Manning 及其學生 Richard Socher 提出 Spectral Chinese Restaurant Process[27]，其概念主要是結合了 Spectral clustering [28] 中的 Laplacian matrix 來對資料先做降維，再以 DDCRP 的概念實作 gibbs sampling 來估算模型參數。

但是今天若是以巨量資料的觀點來看，針對已就坐的顧客做 Sampling 來看是要跟誰坐同一桌，等於是對全部的 Training Data 都要計算出一個機率值，若講求計算效率更必須預先建立好距離關係矩陣，然後再做一遍最大可能性估計，由於資料量為巨量資料，計算量也將成為巨量計算量，因此 DDCRP 是不適合處理巨量資料的

2.6 Topic Model

在機器學習和自然語言處理研究領域中，主題模型(Topic Model)是一種用來發現收集到的文集裡面之主題(此主題可以是抽象的)，直覺上，給一份文件特定主題，我們可以期望一些和特定主題相關或不相關的字出現在文件的頻率是高或低；例如假設有一份文件主題為電腦，那麼”hardware”可能就比”tea”更有機會出現在這份文件。目前已經發展出許多的主題模型，其中以 Latent Dirichlet Allocation(LDA)[29]、hierarchical latent dirichlet allocation(hLDA)[30]最具代表性，許多主題模型皆是從這兩個模型衍伸的模型，因此本論文在下面幾個小節裡介紹這兩種主題模型。

2.6.1 Latent Dirichlet Allocation

LDA 將每一份文件的 Topic 分佈視為一個服從 Dirichlet 分佈的隨機變數，此設定使模型更具彈性，對於未見過的文件，皆可以從 Dirichlet 分佈中取樣出文件的 Topic 分佈參數，並在此分佈參數下，考慮產生文件中的每一個詞。其 graphical model 如下:

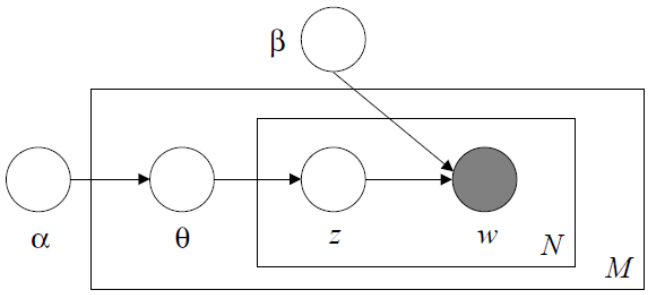


圖 2.6-1:2003 年 David Blei 提出的 LDA 模型，本圖擷取至原著[29]

上圖中 α 是 Dirichlet 分佈的參數，β 是一個 V*K 的矩陣，V 代表全部的字，K 代表全部的 topic，因此，$\beta_{i,k}$ 代表詞 w_i 在第 k 個 topic 下的機率值，即

$P(w_i \mid \theta_k)$，θ 則是一個元素皆大於 0 的 K 維向量，其代表了 K 個 topic 的機率。若以生成模型的角度來看，其生成過程如下:

1. Choose $\theta \sim Dir(\alpha)$

2. For each of the N words w_i:

   Choose a word w_i from $p(w_i|\theta, \beta)$

若我們觀察到一些資料$w_1, w_2, .. w_N$，在 2003 年 Davad Blei 原文中，是用 Variational inference 來估計參數，其主要概念是找一個可以使用 EM 演算法來估算模型參數之近似模型；後來 Griffiths 和 Steyvers 提出以 Gibbs Sampling 的方式來估算模型參數[31]。

值得一提的是，機率式潛藏語意分析(PLSA)[32]是 LDA 在參數α為 1 時，最大化事後機率(MAP)和最大可能性估計(MLE)下的一個特例[33]。

2.6.2 Hierarchical Latent Dirichlet Allocation

傳統的 LDA 中，Topic 的個數參數必須先給定，這使得模型較不彈性，且如同本論文在 1.1 中所談的，容易有 overfitting 或 underfitting 的問題。在”Hierarchical topic models and the nested chinese restaurant process”一文中，先提出一個 Nested Chinese Restaurant Process 的概念，再將此概念引進傳統 LDA 中，改善了 Topic 個數參數必須先給定的問題，使得整個模型更具彈性。接下來幾個小節，我們會簡單介紹此模型。

2.6.2.1. Nested Chinese Restaurant Process

Nested Chinese Restaurant Process (nested CRP)延伸自 CRP，其情景如下:

假設在一個城市中有無限多個餐館，其中一個餐館被定義為 root。每個餐館中有無限多張桌子，每張桌子上都有一張卡片，上面寫著其他餐館的名字，其中，每個餐館的名字都會至少在其中一個餐館的其中一個桌子的卡片上出現一次，因此，這個城市中的餐館被組織成一個無限分支和深度的樹結構。如下圖:

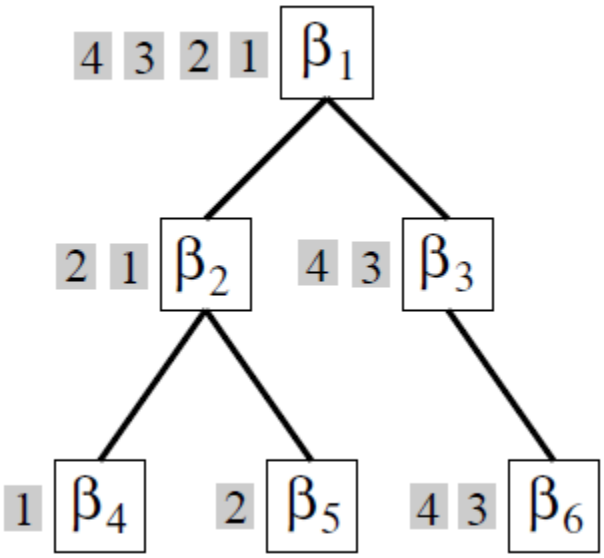


圖　2.6-2: David Blei 等人提出的 nested Chinese Restaurant Process 示意圖
其中每個 β 為一個餐館，每個路徑為一個顧客的旅遊路徑。本圖擷取自原著[30]。

一個遊客到這個城市度假，第一天晚上，他進入 root 餐廳並且採用 CRP 分佈選擇桌子。第二天晚上，他去第一天晚上的餐館桌子上卡片標示的餐館，這樣一直重複下去。M 個遊客在這個城市旅遊 L 天後，路徑的集合就描述了一個 L 階層的無限樹的隨機子樹。

2.6.2.2. Hierarchical latent dirichlet allocation

hLDA 中將 nested CRP 的概念加入原本的 LDA 中，利用 nested CRP 當作 prior 來建構 topic 的深度，其每一個節點中，又採用傳統的 CRP 來建構他的廣度，其

graphical model 如下圖：

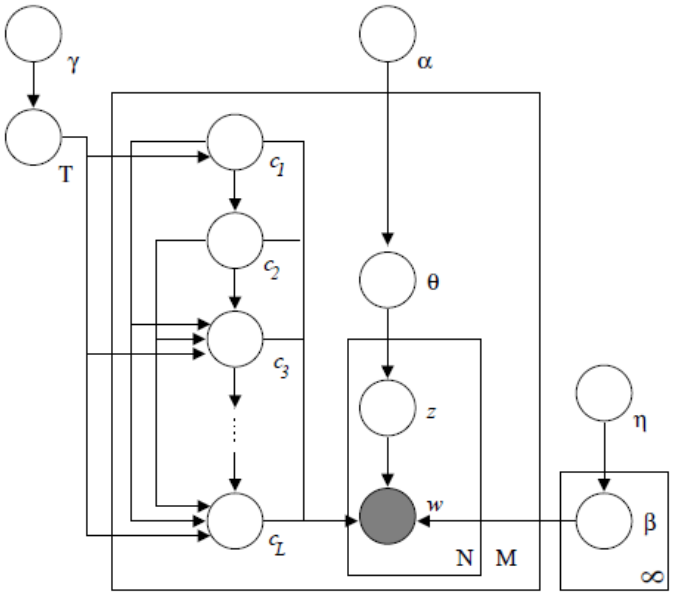


圖　2.6-3 : David Blei 等人提出的 hLDA graphical model，本圖擷取自原著[30]

圖中 γ 是 T 的參數，T 代表了所有無限樹的路徑集合，每個 c_i則對應為一個餐館，共有 L 個，其中c_1為 root，而每個 β 會對應一個c_i，η 是 β 的參數。假設我們今天觀察到一些資料$w_1, w_2, .. w_N$，可以使用 gibbs sampling 的方式來估計模型參數。若以生成模型的角度來看，其生成過程如下:

1. Let c_1 be the root restaurant
2. For each level l ∈ {2, … , L}

   Draw a table from restaurant c_{l-1} using CRP Equation. Set c_l to the restaurant referred to by that table

1. Draw an L-dimention topic proportion vector θ from Dir(α)
2. For each word n ∈ {1, … , N}

   (a) Draw z ∈ {1, … , L} from Mult(θ)

   (b) Draw w_n from the topic associated with restaurant c_z

hLDA利用nested CRP來建構topic的深度，改善了topic個數必須先給定的狀況，取而代之的必須先給定深度。

第三章 Online Chinese Restaurant Process

本章節將介紹本論文所提出的基於中國餐廳過程的在線學習方法(Online Chinese Restaurant Process)，此方法是一個在線學習(Online Learning)方法。目前訓練資料已經漸漸成長為巨量資料，取得龐大的訓練資料後需要一個有效且快速的方法來訓練分類模型，除此之外，對於如何利用持續取得的訓練資料來更新模型，也是一個值得研究的問題。我們提出的方法不但可以對測試資料作即時的預測，也能針對新進訓練資料即時更新模型。

此外，在線中國餐廳過程也保留了傳統中國餐廳過程之特性，以無母數(Non-Parametric)方式讓資料本身決定模型類別數參數，不需事先給定類別數參數。將此特性與 Online Learning 結合，對於動態資料，本論文提出之方法可以持續自動調整模型參數，以及類別數參數。

本章節的架構如下：3.1 將介紹整個系統的架構，3.2 會定義一些符號，3.3 為在線中國餐廳過程的假想情景描述與介紹，3.4 為從傳統的 CRP Mixture Model 衍生的 Online CRP Mixture Model ，3.5 說明此模型也適用 Collapsed Sampling ，3.6 將介紹 Online CRP 之演算法，3.7 為演算法分析與探討。

3.1 系統架構

本章節所提出的 Online CRP 是一個在線學習演算法(Online learning algorithm)，因此整個系統架構遵守 2.1 中 Generic Online Learning Algorithm，本論文的系統架構如下圖表示。

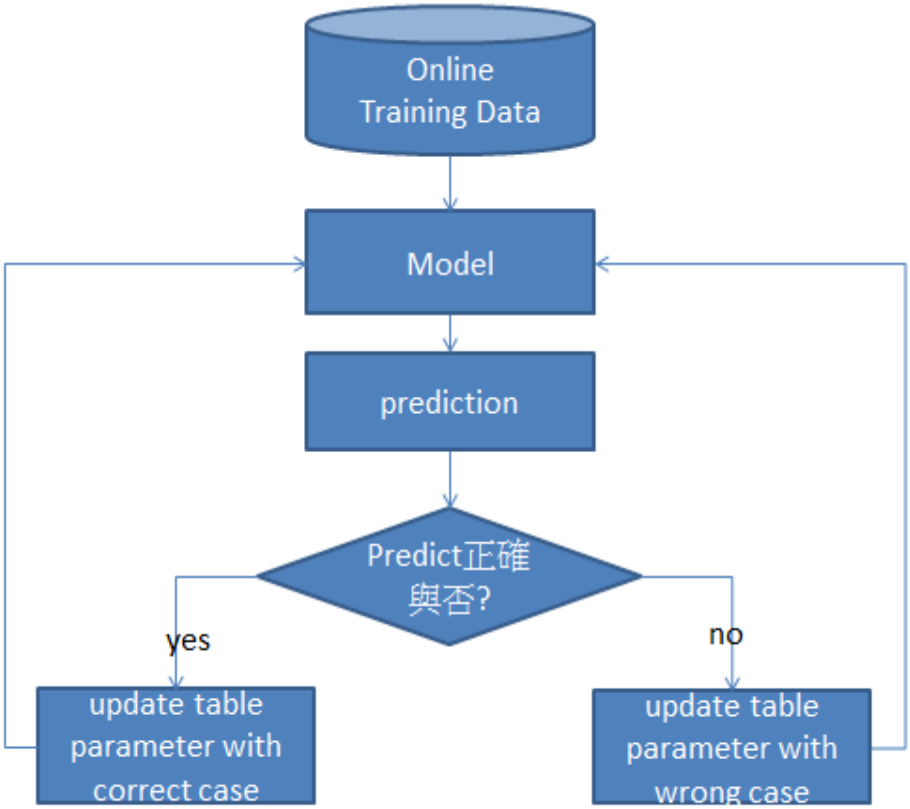


圖　3.1-1：系統架構圖

上圖中，系統一次看一筆訓練資料，並使用前一次所建好的模型來預測當前這一筆訓練資料的類別，接著對照正確答案，並依照預測的結果對錯而分成兩種更新模型的情況來調整模型，以利對下一筆資料的預測。

3.2 符號定義

在更詳細的介紹 Online CRP 之前，我們先在此定義一些變數的符號及其對應之物理意義，以方便介紹之後的小節。

表 2 : Online CRP 符號定義與對應表

符號	意義
x_i	第 i 個顧客
z_i	模型判斷第 i 個顧客應坐在第幾桌
y_i	實際上第 i 個顧客坐在第幾桌
m_j	第 j 桌的人數
θ_j	第 j 桌的菜色
$H(x_i, \theta_j)$	第 i 個人對第 j 桌菜色的偏好程度
K	餐廳總桌數，趨近於 ∞
k	目前桌子有人的桌數
G_0	base distribution
α	調控開新桌比例參數
$g(\gamma_1, \gamma_2, j)$	第 j 桌之 Relaxing function
γ_1, γ_2	Relaxing rate
e_j	因y_i影響而離開第 j 桌的次數
f_j	因y_i影響而加入第 j 桌的次數

3.3 Online CRP

Online CRP 是延伸 2.2 小節所提及之中國餐廳過程情景(二)，詳細的情景如下:

假設:某間餐廳裡面，正要舉行一場婚宴，此餐廳有無限個桌子，而每個桌子有無限個座位，婚宴總招待會替每位顧客安排座位。

第一個顧客進餐廳之後，向婚宴總招待說明自己想吃的菜色，婚宴總招待安排他到某桌就位。之後的每個顧客進餐廳後，皆會向婚宴總招待說明自己想吃的菜色，婚宴總招待會去看看已經有人在的桌子，若那桌預訂菜色是顧客喜歡吃的，就安排顧客去那一桌，如果每桌的預訂菜色都不是顧客喜歡吃的，也可以安排顧客開新桌。每個顧客有可能因故離開一桌而加入另外一桌，婚宴總招待不希望此情形發生的頻率太高，因此他記錄了每桌有顧客因故離開或是因故加入的次數，並將此情形納入為下一位顧客安排座位的考量。另外，婚宴總招待於顧客進來時也記錄了顧客們喜歡吃的菜色，他會依照最終顧客入座情形來決定每桌的菜色內容。

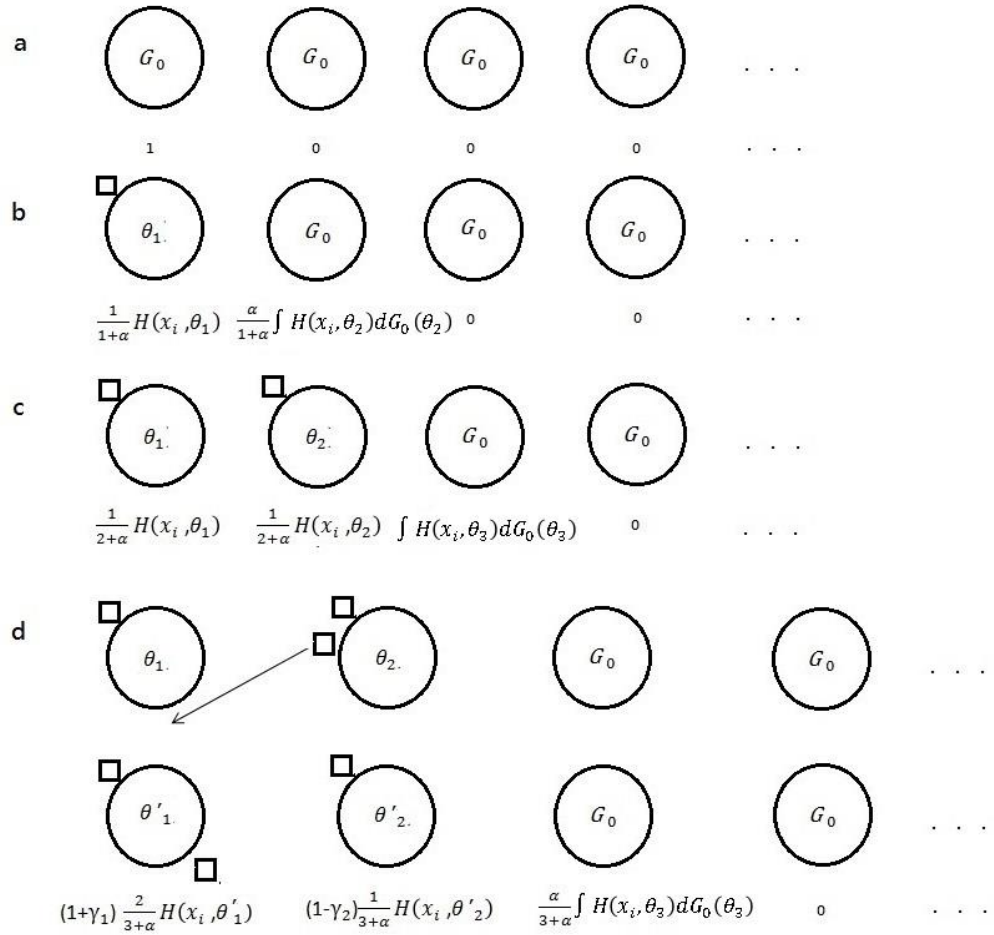


圖　3.3-1: Online CRP 情景示意圖: 不失一般性假設開桌順序為 table1、table2、…。 (a) 在都還沒有人進餐廳時，婚宴總招待安排顧客開新桌的機率為 1。(b) 在(a)之後，顧客坐於第一桌，婚宴總招待安排下一位顧客於第一桌的機率為(1/ (1+α)) *　$H(x_2,\theta_1)$， 安排開新桌的機率為 (α / (1+α)) *　$\int H(x_i,\theta_2)dG_0(\theta_2)$ 。(c) 假設在(b)，婚宴總招待安排第二位顧客到第二桌，其安排下一位顧客到第一桌的機率為 (1/ (2+α)) *　$H(x_3,\theta_1)$，到第二桌的機率為(1/ (2+α)) *　$H(x_3,\theta_2)$，開新桌的機率為(α / (1+α)) *　$\int H(x_3,\theta_3)dG_0(\theta_3)$。 (d) 假設在(c)，婚宴總招待安排第三位顧客到第二桌，但是顧客因故離開第二桌到了第一桌。婚宴總招待看到這個狀況，安排下一位顧客到第一桌的機率為 $(1+\gamma_1)$ *(2/ (3+α)) *　$H(x_4,\theta'_1)$，安排到第二桌的機率為$(1-\gamma_2)$ *(1/ (3+α)) *　$H(x_4,\theta'_2)$，開新桌的機率為(α / (3+α)) *　$\int H(x_3,\theta_4)dG_0(\theta_4)$，此時的$\theta_1'$是根據$x_1$、$bx_3$和$G_0$的事後機率所取樣得到的，$\theta_2'$是根據$x_1$、$-bx_3$和$G_0$的事後機率所取樣得到的。

婚宴總招待看到一位顧客離開一桌加入另外一桌時，會猜想他為何要離開那一桌，直覺上會覺得若安排下個顧客到他離開的那桌，很可能會造成下個顧客也離開的情形；同時也會覺得若直接將下位顧客安排到他加入的那一桌的話，可能比較不會造成顧客又離開的情形。當婚宴總招待考量到這一點，造成了原本安排座位方式

的機率鬆弛(probability relaxing)現象，因此我們在此引進經濟學上面的反悔理論(regret theory)[34]精神並結合傳統中國餐廳定理的原則，設計出一個鬆弛函式(relaxing function) g(γ_1, γ_2, j)如方程式 3.1 所示，來達到使顧客選擇每桌的事前機率達到機率鬆弛的效果。

$$g(\gamma_1, \gamma_2, j) = (1+\gamma_1)^{f_j}(1-\gamma_2)^{e_j} \quad (3.1)$$

式子 3.1 之意義為當婚宴總招待看到有人離開一桌加入另外一桌時，將對之後的顧客安排到那桌的意願減少 γ_2，而對安排到他加入的那一桌意願增加 γ_1。這個考量是因為婚宴總招待害怕再度做錯決定，安排錯位置，造成婚宴的混亂，因此他參考了之前錯誤決策之結果，來擬訂下一次安排座位之策略。婚宴總招待安排顧客到各桌的機率，除了會和該桌人數m_j成正比，還會和每桌非經過他安排而增加的人數 f_j 成正比，同時也會和每桌他排定後卻因故離開的人數 e_j 成反比。

在每一桌上的第一次顧客因故離開原桌 z_i要加入新桌 y_i時，針對 z_i桌的鬆弛函式 $g(\gamma_1, \gamma_2, z_i) = (1+\gamma_1)^0(1-\gamma_2)^1 = (1-\gamma_2)$，針對 y_i桌的鬆弛函式，$g(\gamma_1, \gamma_2, y_i) = (1+\gamma_1)^1(1-\gamma_2)^0 = (1+\gamma_1)$，我們利用此反悔精神將之與各桌人數相乘當成婚宴總招待的全新的事前機率背景知識(Prior Knowledge)。因此，假設婚宴總招待即將安排第 i 個顧客入桌，我們重新定義其選擇桌子的機率式如下:

$$P(z_i=j \mid z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha) \propto \begin{cases} g(\gamma_1,\gamma_2,j)\frac{m_j}{i-1+\alpha}H(x_i,\theta_j) & , if\ j \le k \\ \frac{\alpha}{i-1+\alpha}\int H(x_i,\theta_j)dG_0(\theta_j), & if\ j = k+1 \end{cases} \quad (3.2)$$

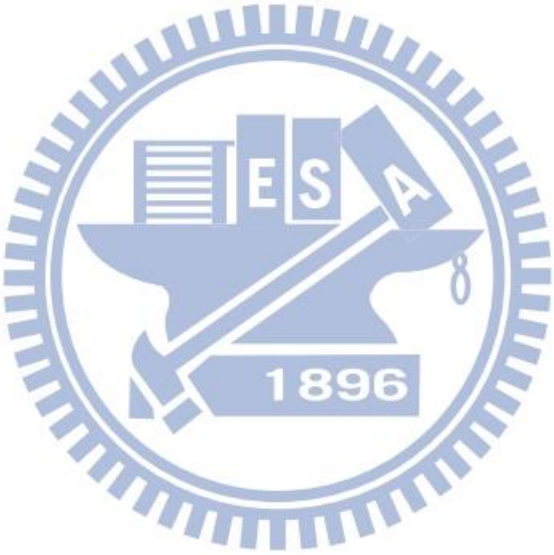
$$= \begin{cases} (1+\gamma_1)^{f_j}(1-\gamma_2)^{e_j}\frac{m_j}{i-1+\alpha}H(x_i,\theta_j), & if\ j \le k \\ \frac{\alpha}{i-1+\alpha}\int H(x_i,\theta_j)dG_0(\theta_j), & if\ j = k+1 \end{cases}$$

$$where\ \ f_j = \sum_{a=1}^{i-1} I[(y_a = j \wedge z_a \neq j)]$$

$$and \quad e_j = \sum_{a=1}^{i-1} I[\,(z_a = j \wedge y_a \neq j] \tag{3.3}$$

將本論文所定義之鬆弛函式代入式子 3.2，即可得式子 3.3；當f_j 和 e_j 皆為 0 時，在線中國餐廳定理便會退化成傳統的中國餐廳過程，式子 3.3 將等同於式子 2.7。

另外，婚宴總招待也希望喜歡同菜色的顧客們盡量坐在同一桌，因此他會根據最後顧客就定位後的情形來更動每桌預訂菜色 $\{\theta_j\}_{j=1}^{k}$，這部分我們將在下一小節有更詳細的說明。



3.4 Graphical Model and Sampling

我們把正確的標記訓練資料當成一個觀察到的變數 y_i，用來間接影響我們每次對其他隨機變數的取樣結果，本論文所提及之 Online CRP，其 graphical model，如下圖所示:

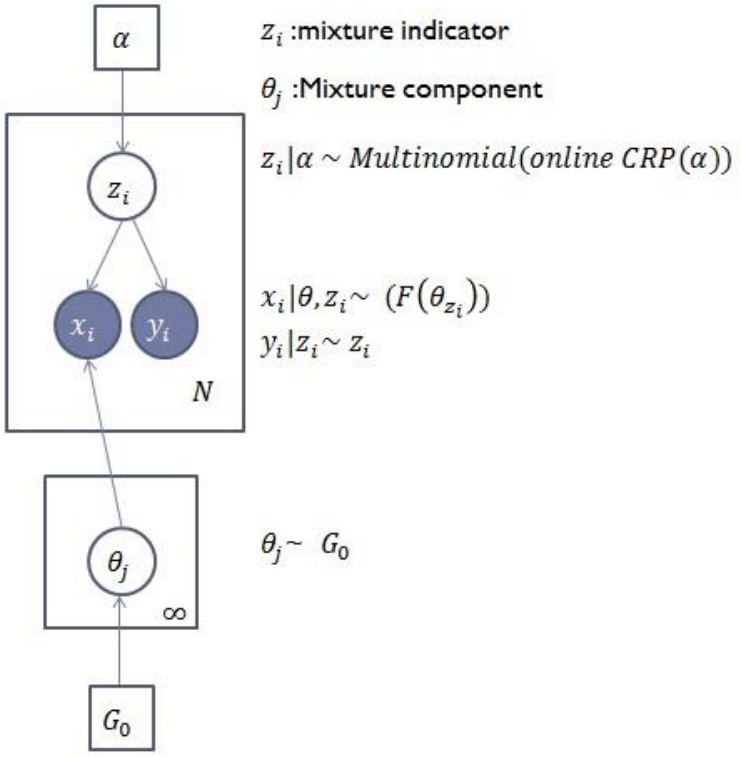


圖　3.4-1 : Graphical Model: Online CRP Mixture Model

對於每一筆資料 x_i ，當我們取樣完z_i後之後，我們需要針對每一桌更新它們的參數 θ_j(預訂菜色)。Online CRP 與傳統的 CRP 最大的不同點在於 Online CRP 已經引入了標記資料 y_i，每回合完成取樣(Sample)並得到模型預測的 z_i 後，必須將其與正確的標記資訊 y_i 做比較。若 $z_i = y_i$，代表資料的預測類別是正確的，此時可利用其他在z_i桌上的資料以及G_0和x_i 計算 θ_{z_i}的事後機率並取樣出新 θ_{z_i}；若 z_i 不等於 y_i，則代表資料的預測類別是錯誤的，這就表示，此筆資料所擁有的特徵在

z_i 桌出現的可能性估計被高估；因此雖然 z_i 桌最終沒有x_i這一點，但我們假設我們在 z_i桌觀察到一點 x_i'= - b * x_i (b 是一個大於 0 之參數，控制該桌預訂菜色與顧客x_i喜歡之菜色偏離之強度)，並利用其他在z_i桌上的資料以及G_0和x_i' 計算 θ_{z_i} 的事後機率並取樣出新θ_{z_i}，因此，x_i 屬於 z_i 桌的可能性估計將會下降；另一方面，此筆資料的特徵在 y_i桌出現的機率被低估，因此我們必須加強 x_i 特徵在y_i桌出現的分佈比例。因此我們假設我們觀察到一點 x_i''= b * x_i (b 是一個大於 0 之參數，控制該桌預訂菜色與顧客x_i喜歡之菜色相近之強度)，並根據在z_i桌上的資料以及 G_0和x_i'' 計算θ_{y_i}的事後機率並取樣出新 θ_{y_i}來達到機率鬆弛，因此，x_i 屬於 y_i桌的可能性估計將會上升。

依照以上之原則，每一個 θ_j'便是根據在第 j 桌上的資料和我們假設存在第 j 桌上的虛擬資料以及G_0計算θ_j的事後機率而取樣出來的。由於我們取樣出來的那些值其實際物理意義為一群特徵發生的機率值，因此在此必須限制我們所假設的虛擬資料不能使特徵發生的機率值為負。

另外原本 CRP 的 prior 是正比於顧客人數，若 $z_i = y_i$，代表當前的比率不需要做任何改變；若 $z_i \neq y_i$，代表當前的比率需要調整，因此我們在 3.3 小節中引入了鬆弛函式(relaxing function) $g(\gamma_1, \gamma_2, j)$ 來達到機率鬆弛，如此便可以依照每次和 y_i 比較的結果，來幫助模型重新調整事前機率(prior)。根據圖 3.4-1，可得到取樣z_i的公式如式子 3.4 所示:

$$P(z_i = j \mid z_{-i}, y_{-i}, x_i, \theta_j, \alpha, G_0) \propto P(z_i = j \mid z_{-i}, \alpha) \mathrm{P}(x_i \mid \theta_j) \qquad (3.4)$$

$P(z_i = j \mid z_{-i}, \alpha)$即為我們所定義之 Online CRP prior，$\mathrm{P}(x_i \mid \theta_j)$即為概似函式，我們使用$\mathrm{H}(x_i, \theta_j)$表示，式子 3.4 即可化成式子 3.2。

這邊需要注意的是，由於此 graphical model 會受正確資料的影響來調整每個模型的參數，因此訓練資料的先後順序將不再具有原來 CRP 的可交換性(exchangeable)，且此 graphical model 所建立出來的分佈和 Dirichlet process

mixture model 所建立出來的分佈不相同。

3.5 Collapsed Sampling

如同 2.2.4 所說,若我們取樣的分佈與此分佈的分佈有共軛性,根據 Neal2000 的 algorithm 3,對於 G_0 和 θ_j,我們可以使用 Collapsed Sampling, 將 graphical model 簡化成下圖:

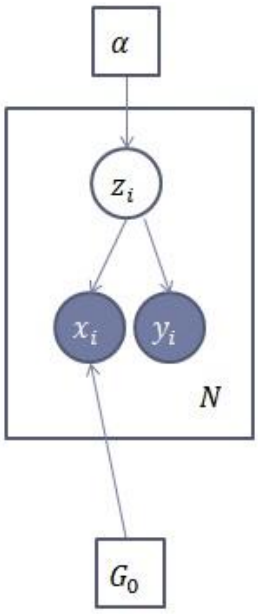


圖　3.5-1 : Online CRP Mixture Model : collapsed sampling

因此，針對每一筆資料 x_i 便只需要取樣 z_i，如式子 3.5 所示:

$$P(z_i = j \mid z_{-i}, y_{-i}, x_i, , \theta_j, \alpha, G_0) \propto \begin{cases} (1+\gamma_1)^{f_j}(1-\gamma_2)^{e_j}\frac{m_j}{i-1+\alpha}\int H\left(x_i, \theta_j\right) dF(\theta_j)_{-i}, if\ j \le k \\ \frac{\alpha}{i-1+\alpha}\int H(x_i, \theta_j) dG_0(\theta_j), \qquad if\ j = k+1 \end{cases} \quad (3.5)$$

式子 3.5 中$F(\theta_j)_{-i}$的定義，和 3.4 小節所提及的一樣，$F(\theta_j)_{-i}$是代表根據在第 j 桌上的資料(不含 x_i)和我們假設存在第 j 桌上的虛擬資料以及G_0，產生 θ_j的事後機率。

3.6 演算法

以 3.3 小節為基礎，我們可以寫出一個 Online Chinese Restaurant Process training algorithm 如下:

Algorithm 3.6.1 : Online CRP

Input: α、G_0 、γ_1、γ_2、b

Initialize: k=0 、for all s $\in \mathbb{N}$, m_s = 0、f_s=0、e_s=0、

1. **for** i = 1 to ∞ **do**
2. Get a data x_i
3. **if** k==0 **then**
4. z_i = 1;
5. **else**
6. sample z_i by P ($z_i = j | z_{-i}, y_{-i}, x_i, \theta_j, \alpha, G_0, \gamma_1, \gamma_2$) $\propto$ $\begin{cases} (1+\gamma_1)^{f_j}(1-\gamma_2)^{e_j}\frac{m_j}{i-1+\alpha}H(x_i, \theta_j) & , if\ j \le k \\ \frac{\alpha}{i-1+\alpha}\int H(x_i, \theta_j)dG_0(\theta_j) & , \quad if\ j = k+1 \end{cases}$
7. Receive a correct answer y_i
8. **if** z_i==y_i **then**
9. Put x_i into table z_i , sample a new θ'_{z_i} from the posterior of θ'_{z_i} , based on given G_0 and all the real and pseudo data associated with the table z_i , $\theta_{z_i} := \theta'_{z_i}$
10. $m_{z_i} = m_{z_i} + 1$;
11. **else**
12. Assume a pseudo data $-bx_i$ join table z_i, sample a new θ'_{z_i} from the posterior of θ'_{z_i} , based on given G_0 and all the real and pseudo data associated with the table z_i , $\theta_{z_i} := \theta'_{z_i}$
13. Assume a pseudo data bx_i join table y_i, sample a new θ'_{y_i} from the posterior of θ'_{y_i} , based on given G_0 and all the real and pseudo data associated with the table y_i , $\theta_{y_i} := \theta'_{y_i}$
14. $m_{y_i} = m_{y_i} + 1$;
15. $e_{z_i} = e_{z_i} + 1$;
16. $f_{y_i} = f_{y_i} + 1$;
17. **end if**
18. **end if**
19. **end for**

在演算法 3.6.1 中，我們不失一般性假設每一筆資料的 y_i 和桌號順序是相對應的。Line 2 取得資料 x_i ，Line 3-4 代表若目前開桌數為 0，此顧客一定會開新

桌，因此直接指定 z_i 為 1；Line 6 主要是要取樣出 z_i，由於 z_i 值有 k+1 種可能性，因此 Line 6 必須計算 $P(z_i=1|\ z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，$P(z_i=2|z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，…，$P(z_i=k|z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，$P(z_i=k+1|z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，然後從中取樣出一個情況。Line 7 得到相對於 x_i 的正確解答 y_i，Line 8-10 若取樣出來的 z_i 和 y_i 做比較，結果兩者值一樣的話，在 Line 9 會為 z_i 桌取樣出一個新的 θ_{z_i}，Line 10 為將該桌人數加 1。Line 11-17 為取樣出來的 z_i 和 y_i 做比較，結果兩者值不同的情形，Line 12-13 分別對 z_i 桌和 y_i 桌取樣出一個新 θ_{z_i} 和 θ_{y_i}，並於 Line 14 將 y_i 桌人數加 1，於 Line 15-16 分別紀錄 e_{z_i} 和 f_{y_i}。

Algorithm 3.6.2 : Collapsed Online CRP

Input: α、G_0 、γ_1、γ_2、b

Initialize: k=0 、m_j = 0、f_j=0、e_j=0、for all $j \in \mathbb{N}$

1. **for** i = 1 **to** ∞ **do**
2. Get a data x_i
3. **if** k==0 **then**
4. z_i = 1;
5. **else**
6. sample z_j by $P(z_i = j|z_{-i}, y_{-i}, x_i, \theta_j, \alpha, G_0, \gamma_1, \gamma_2) \propto \begin{cases} (1+\gamma_1)^{f_j}(1-\gamma_2)^{e_j}\frac{m_j}{i-1+\alpha}\int H(x_i, \theta_j)\, dF(\theta_j)_{-i}, & if\ j \le k \\ \frac{\alpha}{i-1+\alpha}\int H(x_i, \theta_j) dG_0(\theta_j) & ,\quad if\ j = k+1 \end{cases}$
7. Receive a correct answer y_i
8. **if** z_i==y_i **then**
9. Update the sufficient statistic of table z_i due to x_i join table z_i
10. $m_{z_i} = m_{z_i} + 1$;
11. **else**
12. Update the sufficient statistic of table z_i due to $-bx_i$ join table z_i
13. Update the sufficient statistic of table y_i due to bx_i join table y_i
14. $m_{y_i} = m_{y_i} + 1$;
15. $e_{z_i} = e_{z_i} + 1$;
16. $f_{y_i} = f_{y_i} + 1$;
17. **end if**
18. **end if**
19. **end for**

在演算法 3.6.1 中，我們不失一般性假設每一筆資料的 y_i 和桌號順序是相對應的。Algorithm 3.6.2 是以 3.4 小節為基礎，Line 2 取得資料 x_i，Line 3-4 代表若目前開桌數為 0，此顧客一定會開新桌，因此直接指定 z_i 為 1；Line 6 主要是要取樣出 z_i，由於 z_i 值有 k+1 種可能性，因此 Line 6 必須計算 $P(z_i = 1 \mid z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，$P(z_i=2 \mid z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，…，$P(z_i=k \mid z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，$P(z_i=k+1 \mid z_{-i}, y_{-i}, x_i, \theta, G_0, \alpha)$，然後從中取樣出一個情況。Line 7 會得到相對於 x_i 的正確解答，Line 8-10 將取樣出來的 z_i 和 y_i 做比較，若結果兩者值一樣的話，在 Line 9 會為更新 z_i 桌加入 x_i 的充分統計量(sufficient statistic)，Line 10 為將該桌人數加 1。Line 11-17 為取樣出來的 z_i 和 y_i 做比較，結果兩者值不同的情形，Line 12-13 分別對 z_i 桌和 y_i 桌更新其充分統計量，並於 Line 14 將 y_i 桌人數加 1，於 Line 15-16 分別紀錄 e_{z_i} 和 f_{y_i}。Algorithm 3.6.2 與和 Algorithm 3.6.1 最大的不同是，我們針對每筆資料 x_i 加入某桌，不必重新取樣該桌參數，取而代之的是直接更新其充分統計量即可。

3.7 演算法分析

在 Algorithm 3.6.1 和 Algorithm 3.6.2 中，對於每一筆訓練資料，我們只需要使用每桌的參數來計算一筆訓練資料屬於每一桌的機率歸屬值，每次選擇最大的機率歸屬值當成預測的類別 z_i，當正確的標記資料 y_i 來到時，會比對此資訊，並依比較結果更新桌子參數，每一筆資料更新完後，代表系統已經將模型更新完畢，可以接受下一筆測試資料做預測或接受下一筆訓練資料做模型更新的動作

在整個過程中，不需將整個訓練資料集的資訊存進記憶體，這些資訊已經隱含在各桌子的參數裡。因此，我們只需把每個桌子的參數存在記憶體中，並使用這些參數來估計每筆資料對每個類別的機率歸屬值，如此可節省記憶體空間。舉例來說，如果我們桌子的菜色代表的是一個高斯分佈，那麼我們每桌只需儲存位於該桌資料

分佈的平均值(Mean)和變異數(Variance)，當有新資料要進來時，我們其實只需要對這兩個值做一些計算更新而已。

而在 Algorithm 3.6.1 第 5 行的取樣過程中，假設目前系統中已經訓練了 k 類，這就代表著每筆資料必定能夠在 O(k)時間裡面取樣出一個類別[35]。而 3.6.1 第 6 行到第 15 行更新模型參數的地方，最差的情形是當 $z_i \neq y_i$ 的時候，我們必須更新預測錯誤的 z_i 和正確標記的 y_i 兩個桌子之參數，因此更新的時間複雜度為 O(1)，因此，若訓練的資料有 n 筆，此演算法整體的時間複雜度為 O(n*k)，通常，n>>k，k 可被視為常數，因此，O(n*k)≈ O(n)；同理 Algorithm 3.6.2 也是如此。

另外和 Algorithm 3.6.1 和 Algorithm 3.6.2 也保留了傳統中國餐廳過程中無母數的特性，可針對資料特性，自行決定類別數，當某個時間點訓練資料出現一個新類別時，本方法可直接取樣出一個新桌子和其參數給新類別，所以該方法是非常實用且富彈性的。

3.8 取樣 hyperparameter α

無母數方法並非完全沒有參數，因此還是存在必須決定參數問題(無母數方法只是提供其下層之參數一個更寬廣的空間與選擇)，本小節將探討兩種根據資料本身特性決定參數 α 之方法。

3.8.1 根據事後機率決定參數 α

和傳統的 CRP mixture 一樣，我們可以更進一步取樣 α (Antoniak 1974)[36]。給定 z，產生 α 的事後機率如式子 3.6 所示：

$$P(\alpha \mid \mathbf{z}) \propto P(\mathbf{z} \mid \alpha) P(\alpha) \qquad (3.6)$$

給定 α，產生 z 的的機率分佈如式子 3.7 所示:

$$P(\mathbf{z} \mid \alpha) = \prod_{i=1}^{t} \frac{I\left[m_{z_i}=0\right]\alpha + I\left[m_{z_i}\neq 0\right] g(\gamma_1,\gamma_2,z_i)\, m_{z_i}}{\alpha + \sum_{j=1}^{k} g(\gamma_1,\gamma_2,j)\, m_j}$$

Where t represents the number of data points, $g(\gamma_1,\gamma_2,z_l)$ denotes the current relaxing function corresponding to the moment x_i joining to table z_i , m_{z_i} denotes the number of data points in the table z_i corresponding to the moment x_i joining to table z_i , I is an indicator function .

(3.7)

將式子 3.7 代回式子 3.6，可得到式子 3.8

$$P(\alpha \mid \mathbf{z}) \propto \prod_{i=1}^{t} \frac{I\left[m_{z_i}=0\right]\alpha + I\left[m_{z_i}\neq 0\right] g(\gamma_1,\gamma_2,z_i)\, m_{z_i}}{\alpha + \sum_{j=1}^{k} g(\gamma_1,\gamma_2,j)\, m_j} P(\alpha) \qquad (3.8)$$

假設我們知道 α 本身所服從的機率分佈，依照式子 3.8，本論文所提出之方法也可以依造我們觀察到的資料，取樣出適當的 α 。

3.8.2 利用 Online Learning 特性每回合決定參數 α

由於 alpha 決定後，在我們做預測時，真正有影響的是對於新類別之預測，因此可以利用 Online Learning 特性，來決定適當的參數 α 。假設我們對於 x_i 預測是新類別，實際上不是新類別，那麼代表 α 的值太大，因此 α 應該調小一點；假設我們對於 x_i 預測不是新類別，實際上是新類別，那麼代表 α 的值太小，因此 α 應該調大一點。此方法較為簡單，且決定速度快，適合實務上的應用。

第四章 實驗

由於以下的實驗為將 Online CRP 用來作文件分類的實驗，我們先將文件分群與中國餐廳的意義作對應，其詳細的對應表如下表:

表 3:Online CRP 與文章分類實驗之符號對應表

符號	意義
x_i	第 i 篇文章
z_i	模型判斷第 i 篇文章屬於第幾類
y_i	實際上第 i 篇文章屬於第幾桌
m_j	目前第 j 類的文章數
θ_j	第 j 類各項 feature 發生的機率分佈
$H(x_i, \theta_j)$	第 i 篇文章屬於第 j 類的最大可能性估計
K	文章總類別數，趨近於 ∞
k	目前偵測到的類別數
G_0	base distribution
α	調控開新類別的比例參數
g(γ_1, γ_2 , j)	第 j 類的機率鬆弛函式
γ_1 , γ_2	鬆弛率
e_j	模型預測是第 j 類，但此判斷是錯的次數。
f_j	模型預測不是第 j 類，但結果是第 j 類的次數。

在實作最大可能性估計的時候，本論文是假設文件的分佈是服從 Multinomial 分佈，因此，每一類的 Multinomial 參數都是從一個 Dirichlet 分佈(參數 **β**)中取樣而得到的，這兩個分佈是共軛的(Conjagate)，因此我們採用 collapsed sampling (Algoritm 3.6.2)，其積分(integrate)的詳細過程如下:

$$P(X|\boldsymbol{\beta}) = \int P(X|\theta)P(\theta|\boldsymbol{\beta})d\theta \tag{4.1.1}$$

$$= \int (\prod_{j=1}^{m} \theta_j^{N_j(X)})(\frac{1}{C(\alpha)} \prod_{j=1}^{m} \theta_j^{\beta_j - 1})\, d\theta \tag{4.1.2}$$

$$= \frac{1}{C(\alpha)} \int (\prod_{j=1}^{m} \theta_j^{N_j(X)+\beta_j - 1})\, d\theta \tag{4.1.3}$$

$$= \frac{C(N(X) + \boldsymbol{\beta})}{C(\boldsymbol{\beta})} \tag{4.1.4}$$

其中 $$C(\boldsymbol{\beta}) = \int \prod_{j=1}^{m} \theta_j^{\beta_j - 1} d\theta = \frac{\prod_{j=1}^{m} \Gamma(\beta_j)}{\Gamma(\beta_{j\bullet})}$$

且 $$\beta_{j\bullet} = \sum_{j=1}^{m} \beta_j$$

按照定義,對參數 θ 做積分可以得到式子 4.1.1,由於 θ 是一組 Multinomial 參數，而 α 是一組 Dirichlet 參數，將其分佈的定義代入式子 4.1.1，我們可以得到式子 4.1.2 ，由於第一項和第二項是一樣的形式，因此指數部分可以直接相加，可以得到式子 4.1.3，最後將 C 函式的定義代入，可將式子化簡成 4.1.4。

因此當我們使用 collapsed sampling 時，條件機率式的計算方法如下:

$$P(z_i = j \mid z_{-i}, x_i, \alpha, \beta) \propto \begin{cases} (1+\gamma_1)^{f_j}(1-\gamma_2)^{e_j} \frac{m_j}{i-1+\alpha} \int H(x_i, \theta_j)\, dF(\theta_j)_{-i}, & if\ j \le k \\ \frac{\alpha}{i-1+\alpha} \int H(x_i, \theta_j)\, dF(\theta_j)_{i-} & ,if\ j = k+1 \end{cases} \tag{4.2.1}$$

$$= \begin{cases} (1+\gamma_1)^{f_j}(1-\gamma_2)^{e_j} \frac{m_j}{i-1+\alpha} \frac{C(N(x_i)+N_{z=j}(x_{-i})+\beta)}{C(\beta+N_{z=j}(x_{-i}))} & ,if\ j \le k \\ \frac{\alpha}{i-1+\alpha} \frac{C(N(x_i)+\beta)}{C(\beta)} & ,\quad if\ j = k+1 \end{cases} \tag{4.2.2}$$

根據原始定義，我們可以得到式子 4.2.1，再將式子 4.1.1~4.1.4 的過程帶入化簡，可得到式子 4.2.2。其中C(β)需用到Gamma function，基於電腦計算於階

層上有其計算範圍限制的原因，我們在此採用替代算法來估算$\frac{C(N(x_i)+N_{z=j}(x_{-i})+\beta)}{C(\beta+N_{z=j}(x_{-i}))}$ 與 $\frac{C(N(x_i)+\beta)}{C(\beta)}$ 。由於式子 4.2.1 中積分之物理意義為每看到一個字就產生一個新的第 j 桌參數θ_j，並利用此參數來計算該字產生之機率，再將這些字發生之機率作連乘之動作；我們直接假設整篇文章屬於第 j 桌，然後直接用此時第 j 桌的字詞分佈來計算第 j 桌產生x_i這篇文章的機率，此計算方法為直接計算對 θ_j 積分之上限。

4.1 實驗資料集介紹

本論文使用的資料集共有三種，每種的資料集有其不同的特性，其類別與文章數也不太相同，以下介紹這三種資料集。

1.20newsgroup

20 Newsgroups是一個常被用在文件分群或分類的資料集，此資料集從網路新聞討論串取出，包含了將近20000篇的文章，共有20個類別，包含電腦類、運動類、政治類等類別，每個類別都是1000篇文章；因每篇文章的標頭會有一些不必要的標頭資訊，故本實驗將其標頭過濾掉。

2.RCV1

RCV1 是 Ruters 釋出的新 dataset，全資料集依 David D. Lewis 的分類[37]，共可 103 類以及約 80 萬篇文章，本實驗的資料集為原資料集之子集，本論文實驗之主要文章為比照 Lewis 論文的分類樹(category tree)，並取其第二層節點(node)當成是每篇文章的標記資訊，屬於第三層或第四層的文章的標記資率全部皆對應到它們在第二層的父點(parent node)，在這邊我們移除了只屬於第一層的文件以及在第二層同時屬於兩個類別的文件。

經過上述處理後，共剩下 53 類，以及 15564 篇測試資料和 518571 篇訓練資料，feature 數有 47236 個。

3.Wikipedia

我們使用 Wikipedia 網站提供的 dump 資訊，全部大約三百萬篇文章，我們選取用字出現頻率前五萬高的字詞當作特徵，若文章的用字皆未達頻率前五萬高，則移除此文章，經過這樣的處理後，大約剩下兩百萬篇文章，並用原始標示的主題當作標記資料，共有 24 類，詳細的資訊如下表

表 4:Wikipedia 資料集

類別名稱	資料數量	類別名稱	資料數量
Mathematics	9683	Applied_sciences	15869
People	108175	Health	15060
Science	32113	Business	36167
Law	19304	Humanities	20897
Geography	442786	Belief	1679
History	79121	Chronology	820576
Culture	137074	Society	59006
Agriculture	26746	Life	42265
Politics	59883	Computers	6949
Nature	14655	Environment	13341
Technology	58406	Arts	17907
Education	38675	Language	5346

4.2 實驗步驟

4.2.1 前處理

雖然前一節在介紹各個文章集時，已經有根據不同的特性過濾掉一些沒有用的資訊，或是會提供答案的資訊，但還不夠完整，以下列舉出本論文對文章集的前置處理。

(1) 只保留英文字母、大小寫一律改小寫

文章中包含許多標點符號及數字等沒有用的資訊，本實驗只保留英文字母；而一個字如果出現在句首或句中，會因為大小寫的問題被判斷為不同的字，所以字母大小寫一律改小寫。

(2) 去除 Stop Words

文章中會出現一些常用字，像是代名詞、介係詞、助動詞、連接詞等，它們基本上對於分類是沒有幫助的，所以會先行過濾掉。

(3) 進行 Stemming 處理

英文字上名詞會有單複數之分、動詞會根據不同的時態而有不同的拼法，但是它們是同樣意思的字，所以對於每篇文章會進行Stemming 的處理，把這些不同拼法但意思同樣的字統一成原形。

4.2.2 實驗方法

本實驗的機器設備與環境如下表:

表 5: 實驗機器設備與環境

CPU	Intel(R) Core(TM) i7 CPU 3770 3.4GHz 3.4GHz
RAM	16GB
Language	Matlab

本實驗在資料集資料量為兩萬筆以下的實驗，是採用 5-fold 的實驗方法，其步驟為先統一先隨機地打亂資料集的排列順序，再把資料集分成5 等分和五個 fold，每一個 fold 皆分成 80%之 Training Data 和 20%的 Testing Data，其意義在於使每一個等分都能有成為 Testing Data 的機會，以避免一次實驗中 20%的 Testing Data 皆為原本就是好分地那些 Data Point。

在資料集 RCV1 的實驗上，是採用如 4.2.1 所述之 training data 和 testing data。

而在資料量達兩百萬筆的 Wikipedia 資料集，是採用隨機將資料分成 80%之 Training Data 和 20%的 Testing Data 之作法。

我們用來與本論文提出的方法 Online CRP 的比較方法有 Online Perceptron、Online Logical Regression 、Online SVM、Logical Regression、SVM、Naive Bayes。

4.3 效能評估方式

4.3.1 F1 cluster evaluation

雖然此實驗是分類之實驗，但由於本論文所提出之方法 Online CRP，於預測時容許預測都不是我們訓練資料裡面擁有的類別，因此若使用 F1 class evaluation，會有計算定義混淆之問題。因此本論文一律採用 F1 cluster evaluation 來做效能測量指標。以下為 F1 cluster evaluation 四種比較的方法:

1. True Positives(TP):

系統將兩篇文章分在同一群，而這兩篇文章實際上也是在同一個類別裡。

2. False Positives(FP):

系統將兩篇文章分在同一群，但這兩篇文章實際上不是在同一個類別裡。

3. True Negatives(TN):

系統將兩篇文章分在不同群，而這兩篇文章實際上也不在同一個類別裡。

4. False Negatives(FN):

系統將兩篇文章分在不同群，而這兩篇文章實際上卻是在同一個類別裡。

而 F1 cluster evaluation measure 之定義如下

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

4.3.2 Error rate

錯誤率即為計算預測的類別和實際的類別不相等的次數再除以總預測次數。

4.3.3 Execution time

此部分為測量整個程式從模型開始訓練到預測完所有測試資料的時間。本實驗以秒數為計量單位。

4.4 實驗結果

20newsgroup:

表 6：20newsgroup 實驗結果(5-fold 之平均值 ± 2 倍標準差)

	Error Rate	Cluster F1 Value	Execution time(s)
Online CRP	0.200980+-0.010022	0.679006+-0.022357	312.818196+-8.920450
Online Last Perceptron	0.516627+-0.039270	0.213904+-0.032936	1075.408193+-103.908960
Vote Perceptron	0.443566+-0.013335	0.252989+-0.022854	36970.677459+-2111.787757
Avg Perceptron	0.363954+-0.024588	0.433337+-0.031250	37482.347879+-3988.263142
kernel Last Perceptron	0.371754+-0.021299	0.422586+-0.030748	384.324955+-103.357849
kernel Vote Perceptron	0.352151+-0.045431	0.463472+-0.049069	14799.319818+-1688.544731
kernel avg Perceptron	0.375507+-0.023233	0.424742+-0.029058	13423.765702+-199.894576
Online LR	0.317697+-0.016178	0.500984+-0.019566	113526.530614+-2755.268273

RCV1:

表 7：RCV 實驗結果

	Error Rate	Cluster F1 Value	Execution time(s)
Online CRP	0.147713	0.851655	54958
Online Last Perceptron	0.117643	0.890532	259512

Wikipedia:

表 8：Wiki 實驗結果

	Error Rate	Execution time(s)
Online CRP	0.3602	112030
Online Last Perceptron	0.3888	2249000

4.5 實驗結果討論

1.與其他 Online Learning Algorithm 之比較

於 20newsgroup 兩萬篇文章的實驗中，可以從表 6 明顯看出，本論文提出之方法之效能贏過其他方法，且執行時間也比其他方法快很多。其中 Online SVM 更

是無法順利於理想時間(一個月)內跑完實驗。

於 RCV1(表 7)和 Wikipedia(表 8)的實驗中，其訓練資料量分別達到 50 萬和 160 萬，使用 kernel method [38] 之方法需計算 kernel matrix，我們使用的機器之記憶體已無法建造此矩陣。另外由於訓練資料量大，其他方法都須要跑非常久，唯有 Online CRP 與 Online Last Perceptron 可以在理想時間(一個月)內跑完，而 Online CRP 又比 Online Last Perceptron 快了許多。而且 Online Last Perceptron 之模型必須先給定類別數參數，Online CRP 則不必，兩者的實驗效果在訓練資料量大的情況下是差不多的。本論文提出的 Online CRP，不僅不須事先決定類別數參數，且效能甚至可贏過須決定類別數參數的其他 Online Learning 方法。

此外，20newsgroup 是每個類別文章數皆為 1000 篇，而 RCV1 和 Wikipedia 兩個資料集裡面，每個類別之文章數是極度不平均的，由實驗中亦可看出，本論文提出之方法，在類別資料量平均和不平均的情況下，皆能有不錯的效能。

2. 與著名 superviesd 方法比較

我們更進一步的將監督式學習方法也依同樣的實驗步驟作實驗比較，得到以下的結果：

20newsgroup

表 9: 20newsgroup 資料集與 supervised 方法效能比較表(5-fold 之平均值 ± 2 倍標準差)

20news	Error Rate	F1 value	Execution time(s)
Online CRP	0.200980+-0.010022	0.679006+-0.022357	312.818196+-8.920450
SVM	0.203030+-0.009245	0.651600+-0.010391	5931.925530+-56.237568
LR	0.179275+-0.019520	0.692891+-0.028493	8576.386376+-315.337173
Naive Bayes	0.227584+-0.011982	0.613931+-0.021737	2909.171871+-30.390296

RCV

表 10 : RCV1 資料集與 supervised 方法更新模型比較表

RCV	Error Rate	F1 value	Execution time(s)
Online CRP	0.147713	0.851655	7023
SVM	0.0748	0.9318	346112
LR(lambda=30)	0.1409	0.8394	12367
Naive Bayes	0.18	0.7850	143831

Wikipedia

表 11 :Wikipedia 資料集與 supervised 方法更新模型比較表

	Error Rate	Execution time(s)
Online CRP	0.3602	7112
LR(lambda=30)	0.3488	40952
Naive Bayes	0.3887	389209

從實驗中可看出，本論文提出之 Online CRP 的效能表現逼近監督式學習方法。在 20newsgroup 的實驗中(表 9)，就已經可明顯看出 Online CRP 比其他方法快了許多，這是因為監督式學習方法，都有需要測試的參數，例如 Logical Regression 必須挑選 regularization 參數 λ，SVM(linear)中必須挑選 penalty 參數 C，Naïve Bayes 中必須挑選 smoothing 參數 β，這些參數皆是用來避免訓練模型造成 overfitting 或 underfitting，且若隨便給定，效能變動將很大，因此實驗過程中，皆必須將原本的訓練資料再度切割出一小塊交叉驗證測試資料集來供參數測試使用，並從中挑選適當的參數。

在 RCV 實驗中(表 10)，我們假設目前模型為從 40 萬筆訓練資料中訓練出來的模型，若此時另外新增十萬筆訓練資料，欲做模型更新，整體效能最好的是 SVM(linear)，但是更新時間需較久，這是因為大多數時間都花在挑選參數上(在本實驗中，我們使用台大林智恆教授網站上的 svm 程式，並仿照其子程式 grid.py 裡挑參數的方法來挑參數，共測試五個參數，平均測試一個參數約需 69000 秒)；而

Online CRP 的表現介於 SVM 與 Naïve Bayes 之間，但是 Online CRP 更新十萬筆資料只需約 7000 秒，比其他監督式學習方法更新模型時間皆快速。

在 Wikipedia 實驗中(表 11)，我們假設目前模型為從 150 萬筆訓練資料訓練出來之模型，若此時另外新增十萬筆訓練資料，欲做模型更新的動作，可以發現 Online CRP 在 features 數 5 萬個的情況下，針對十萬筆新進的訓練資料作模型更新，只需約 7000 秒，Logical Regression 則必須花費 40952 秒；此外，SVM 在 Wikipedia 實驗中，已無法在理想時間內跑完實驗。

監督式學習的方法，在 Matlab 裡面皆須使用稀疏矩陣存取法，才可運算，否則在資料量達到 60000 筆之後，就會出現記憶體不足之錯誤訊息，而 Online CRP 可用完全矩陣表示法來運算，這是因為系統一次針對一筆資料更新模型參數。由此可見，當資料量達到稀疏矩陣表示法所能負載的瓶頸，或是資料並非在很多項是 0 的情況下，使用監督式學習方法將會面臨極大的困難。

另一方面，在實驗過程中，我們發現了在資料量達到某一個量後，本論文提出之 Online CRP 的效能會漸趨穩定，以 RCV 資料集為例，針對本論文所提出之方法，每隔一萬筆資料對相同的測試資料作測試並計算一次錯誤率，可得到下圖:

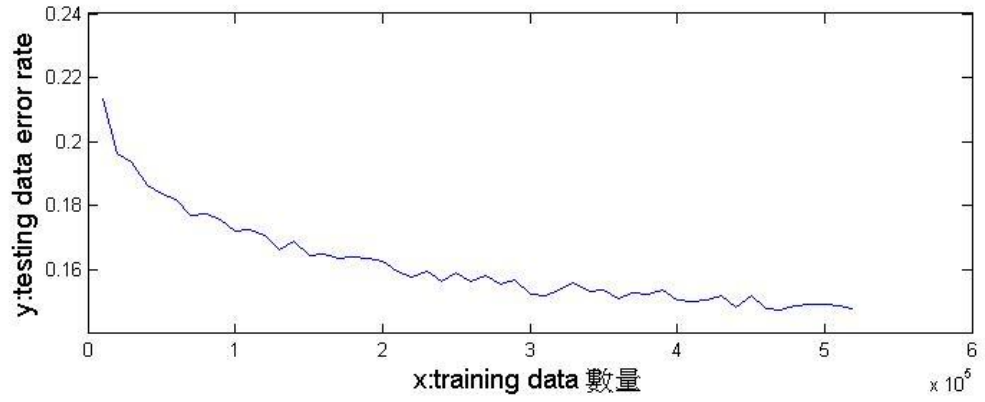


圖　4.5-1 : RCV1 資料集：Online CRP 每訓練一萬筆資料對一萬五千筆測試資料做預測之錯誤率。

1. Online CRP with Stochastic Gradient Descent trick

由 2.的討論中，我們知道了當訓練資料量達到某個量，我們的模型就已

經可以有不錯的效能表現，因此我們參考了 Léon Bottou 所整理的 Stochastic Gradient Descent Tricks[39]之概念，每隔一千點當成一個 period，定期計算 training error，並與上一次計算之 training error 相減，若此差值若低於某一個定值則收斂並停止訓練，得到結果如下:

RCV1:

表 12: RCV 資料集：Online CRP 與使用 SGD trick 之 Online CRP 效能之比較

	Error Rate	F1 value	Execution time(s)
Online CRP	0.147713	0.851655	54958
Online CRP with SGD trick	0.1775	0.8118	1312

Wikipedia:

表 13:Wikipedia 資料集： Online CRP 與使用 SGD trick 之 Online CRP 效能之比較

	Error Rate	Execution time(s)
Online CRP	0.3602	112030
Online CRP with SGD trick	0.3807	594

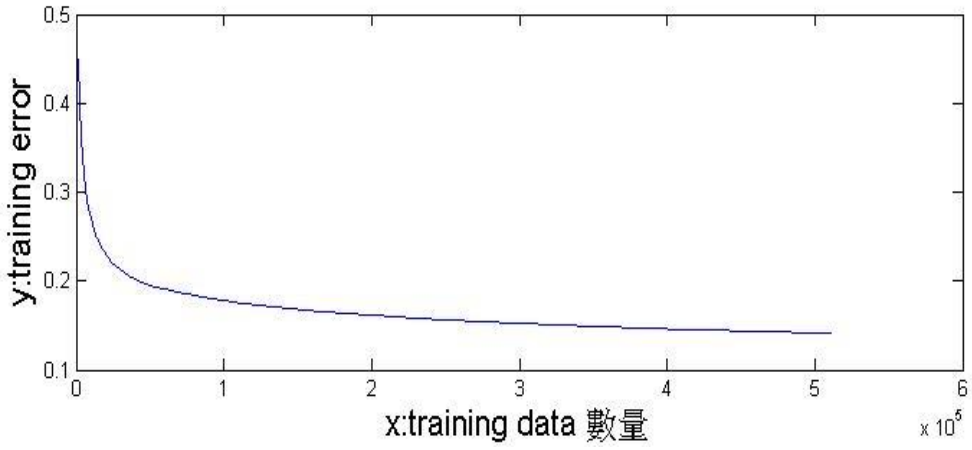


圖 4.5-2:RCV1 資料集：Online CRP 每隔一千筆資料紀錄一次 training error。

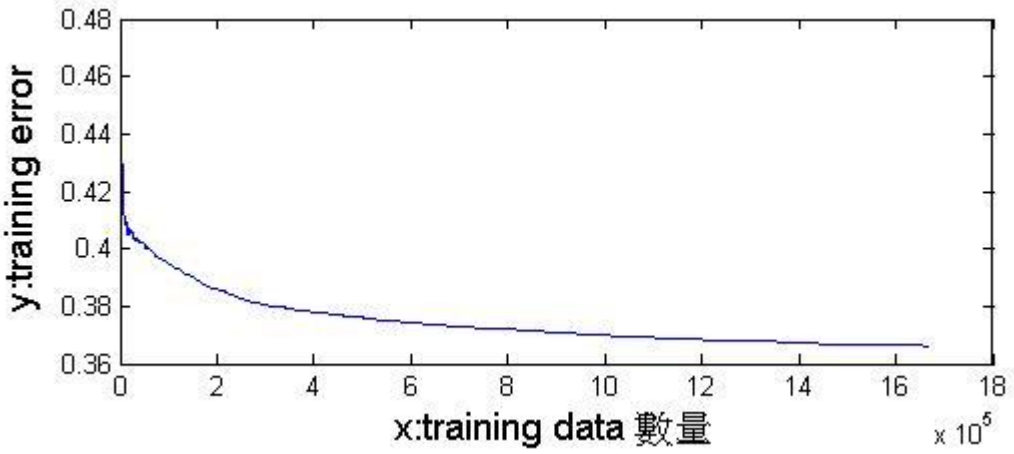


圖　4.5-3:Wikipedia 資料集：Online CRP 每隔一千筆資料記錄一次 training error。

由圖 4.5-2 和圖 4.5-3 可看出，Online CRP 之 training error 在資料量到達一個數量後，最後此值會趨近於穩定。而由表 12 和表 13 中可以看出，本實驗提出之方法，若加入 Stochastic Algorithm 的一些 trick，雖然效能比將全部訓練資料用來訓練差了 2%~3%，但是模型訓練時間卻比原先快了好幾十倍。

2. 實驗總結

我們提出的 Online CRP 是無母數方法，因此模型是當遇到新類別的時候，才會開始針對其類別參數做訓練，和其他本論文所提及之 Online Learning 方法都需事先給定類別數參數之模型有很大的差異，然而 Online CRP 的表現卻不比其他 Online Learning 方法的差，且 Online CRP 之效能可以達到和監督式學習方法差不多的水準，執行時間也比其他方法快很多，甚至可以進一步的引進 Stochastic 演算法常用的技巧，可提早訓練出一個趨近穩定的模型，適合應用於巨量資料上。

第五章 結論與未來展望

5.1 研究總結

在大多的情況下，資料類別及個數資訊在巨量資料問題下是未知的，而且也無法透過專家經驗或實驗方式得知，因此無母數方法比固定參數的機器學習方法更適合處理巨量資料。在實驗中，當資料量大時，我們提出的 Online CRP 不僅在分類的效能上能夠達到監督式學習方法的標準，且在執行時間也比很多方法快速，驗證本方法可準確並有效率的處理巨量資料問題。

5.2 未來展望

目前有 online 概念的 graphical model 研究，大部分只研究主題隨時間的演變，較少探討使用標記資料來對模型的參數估計做調整的問題，因此未來可能可以利用標記資料來探討其他 graphical model 和 Online Learning 的結合。

標記資料引入 graphical model 在此論文是使用一個隨機變數來表示標記資料，未來的方向可以在 graphical model 中多設計一些隨機變數，例如加入 Universum 的資料等等，來影響整個系統的機率參數之估計。

本論文提出的方法具動態的自我成長和自我訓練功能，並不排斥於對新領域的即時學習與擴展，因此很適合整合成一套實務上的應用系統，也是未來可以考慮的發展方向。此外，針對新訓量資料引進，參數一直在變動，可以由圖 4.5-1 中看出，新訓練資料加進來，參數變動並不一定造成效能提升，因此未來可以考慮借用 Average Perceptron 之概念，將全部或部分參數紀錄下來，並使用所有參數預測之平均的機率值來當預測值。

參考文獻

[1] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features”, ECML, Berlin: Springer, pp. 137–142, 1998.

[2] R. E. Schapire and Y. Singer, “Boostexter: A boosting-based system for text categorization”, Machine Learning, vol. 39, no. 2/3, pp. 135–168, 2000.

[3] LEO BREIMAN , ”Random Forest” , Machine Learning, 45, 5–32, 2001, 2001 Kluwer Academic Publishers. Manufactured in The Netherland

[4] D. W. Hosmer and Stanley Lemeshow, Applied Logistic Regression, 2nd ed., Wiley, 2000.

[5] A. McCallum and K. Nigam, “A comparison of event models for naïve bayes text classification”, in IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION. AAAI Press, pp. 41–48, 1998.

[6] MacQueen, J. B. (1967). "Some Methods for classification and Analysis of Multivariate Observations". Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability

[7] McLachlan, G., " Mixture Models." Marcel Dekker, New York, NY (1988)

[8] Ferguson, Thomas (1973). "Bayesian analysis of some nonparametric problems". Annals of Statistics 1 (2): 209–230.

[9] Aldous, D. J. (1985). "Exchangeability and related topics". École d'Été de Probabilités de Saint-Flour XIII — 1983. Lecture Notes in Mathematics1117. pp. 1–1.

[10] Pitman. Combinatorial Stochastic Processes. Lecture Notes for St. Flour Summer School. Springer-Verlag, New York, NY, 2002.

[11] Navarro, D, & Perfors. A. “The Chinese restaurant process” University of Adelaide

[12] Kunle Olukotun, “Map-Reduce for Machine Learning on Multicore” in NIPS, 2006

[13] Léon Bottou “Large-Scale Machine Learning with Stochastic Gradient Descent” Proceedings of COMPSTAT'2010, pp 177-186

[14] Shalev-Shwartz “Online Learning: Theory, Algorithms, and Applications” Foundations and TrendsRin Machine Learning Vol. 4, No. 2 (2011) 107–194

[15] ROBERT E. SCHAPIRE “Large Margin Classification Using the Perceptron Algorithm” 1999 Kluwer Academic Publishers, Machine Learning, 37, 277–296 (1999)

[16] M. Hoffman, D. Blei, and F. Bach. Online learning for latent dirichlet allocation. Advances in Neural Information Processing Systems, 23:856–864, 2010.

[17] Chong Wang, John Paisley, David M. Blei. Online variational inference for the hierarchical dirichlet process. In Proc. of the 14th Int'l. Conf. on Artificial Intelligence and Statistics (AISTATS), Vol. 15 (2011), pp. 752-760.

[18] Cauwenberghs and T. Poggio, "Incremental and Decremental Support Vector Machine Learning," in Adv. Neural Information Processing Systems (NIPS*2000), Cambridge MA: MIT Press, vol. 13, 2001

[19] Fedor Zhdanov and Vladimir Vovk. “Competitive online generalized linear regression under square loss” , ECML 2010

[20] Daphne Koller, Nir Friedman, “Probabilistic Graphical Models: Principles and Technique” ,MIT Press, 2009

[21] Sethuraman, J. (1994), “A Constructive Definition of Dirichlet Priors,” Statistica Sinica, 4, pp. 639–650.29

[22] Blackwell, D. and MacQueen, J. (1973), “Ferguson Distributions via Polya ´ Urn Schemes,” Annals of Statistics, 1, pp. 353–355.

[23] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from

Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society, Series B 39 (1): 1–38

[24] Neal "Markov chain sampling methods for Dirichlet process mixture models." Journal of Computational and Graphical Statistics, 9(2):249–265, 20001

[25] Blei, D., M. Jordan. Variational methods for the Dirichlet process. In 21st International Conference on Machine Learning. 2004.

[26] D. M. Blei and P. I. Frazier. Distance dependent Chinese restaurant processes. In ICML, 2010.

[27] Richard Socher,Andrew Maas,Christopher D. Manning," Spectral Chinese Restaurant Processes: Nonparametric Clustering Based on Similarities", AISTATS,2011

[28] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In NIPS 14. MIT Press, 2001

[29] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, January 2003.

[30] D. Blei, T. Griths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. Advances in Neural Information Processing Systems, 16, 2003.

[31] Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. Proceedings of the National Academy of Science, 101, 5228-5235

[32] Thomas Hofmann, " Learning the Similarity of Documents : an information-geometric approach to document retrieval and categorization" ,Advances in Neural Information Processing Systems 12, pp-914-920, MIT Press, 2000

[33] Mark Girolami, and Ata Kaban, "On an equivalence between PLSI and LDA", SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference

on Research and development in informaion retrieval, page 433--434.

[34] Loomes, G. and Sugden, R. (1982), “Regret theory: An alternative theory of rational choice under uncertainty”, Economic Journal, 92(4), 805–24.

[35] Erik B. Sudderth, “Graphical Models for Visual Object Recognition and Tracking”, Submitted to the Department of Electrical Engineering and Computer Science on May 26, 2006 in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Electrical Engineering and Computer Science

[36] Charles E. Antoniak, ” Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems”, Ann. Statist. Volume 2, Number 6 (1974), 1152-1174.

[37] Lewis, D. D.; Yang, Y.; Rose, T.; and Li “F. RCV1: A New Benchmark Collection for Text Categorization Research” Journal of Machine Learning Research, 5:361-397, 2004.

[38] J. Shawe-Taylor and N. Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, 2004.

[39] Léon Bottou, “Stochastic Gradient Descent Tricks “, Neural Networks: Tricks of the Trade , Lecture Notes in Computer Science Volume 7700, 2012, pp 421-436