LEARNING PROBABILISTIC MODELS

20

學習機率模型

本章中我們把學習視為從觀察中進行不確定推理的一種形式。

第 13 章指出過在現實環境中不確定性是很常見的。代理人可以透過機率理論以及決策理論來處理不確定性問題，但是首先它們必須根據經驗學習關於世界的機率理論。本章將解釋它們如何藉著將學習任務本身進行形式化後成爲一個機率推理過程(第 20.1 節)，來達成。我們將看到學習的貝氏觀點是極其強有力的，它能夠爲雜訊、過適配(或過擬合)和最佳化預測等問題提供通用的解決辦法。同時，它還考慮了一個事實：對於一個非全知的代理人來說，它可能永遠無法確定哪一種關於世界的理論是正確的，卻又必須透過某種關於世界的理論進行決策。

在第 20.2 節和第 20.3 節中，我們將描述學習的機率模型方法——主要是貝氏網路。本章的一些材料相當數學，儘管對於一般課程的理解可以不用深入其中的細節。讀者不妨回顧一下第 13 章和 14 章，並且查看一下附錄 A，會比較有益處。

20.1 統計學習

如同在第 18 章中一樣，本章的關鍵概念是**資料**與**假設**。在這裡，資料就是**證據**(evidence)——也就是，對部分或者全部用於對域進行描述的隨機變數的實例化。「假設」則是關於域如何起作用的機率理論，包括了作爲特例的邏輯理論。

考慮一個簡單的例子。大家都喜歡吃的 Surprise 糖果有兩種口味：一種是櫻桃味(眞好吃呀！)，另一種是酸橙味(難吃透頂！)。糖果生產商有著特別的幽默感，他們用同樣的不透明糖果紙包裝糖果，而不管是什麼口味的。然後糖會被裝進很大的包裝中再賣出去，這種大包裝根據元素比例可以分爲 5 類——但是從外表分辨不出它們的區別。

h_1: 100% 櫻桃味
h_2: 75% 櫻桃味 + 25% 酸橙味
h_3: 50% 櫻桃味 + 50% 酸橙味
h_4: 25% 櫻桃味 + 75% 酸橙味
h_5: 100% 酸橙味

新打開一包糖果，用隨機變數 H(作爲假設)來代表這包的類型，它的可能取值是從 h_1 到 h_5。當然，H 無法直接觀察到。隨著糖果一顆顆被剝開並品嘗，資料逐漸顯現出來——$D_1, D_2, \ldots, D_N$ ，其中每個 D_i 都是一個隨機變數，可能的取值爲 *cherry*(櫻桃)或 *lime*(酸橙)。代理人面臨的基本任務是預測下一顆糖果的口味[1]不考慮表面的細節，這種模式可以用於引入許多主要的問題。代理人確實需要推斷出關於它的世界中一個理論，雖然這個理論可能很簡單。

貝氏學習根據給定的資料簡單地計算每種假設的可能性，然後在此基礎上進行預測。也就是說，預測是利用所有的假設完成的，根據機率分別給予每個假設相對應的權值，而不是只使用單一「最佳」假設。這樣，學習簡化爲機率推理。令 **D** 代表所有的資料，它的觀察值記作 **d**；根據貝氏法則，可以得到每個假設的機率：

$$P(h_i \mid \mathbf{d}) = \alpha\, P(\mathbf{d} \mid h_i)\, P(h_i) \tag{20.1}$$

現在，設想我們要得到關於一個未知量 X 的預測。那麼我們有：

$$\mathbf{P}(X \mid \mathbf{d}) = \sum_i \mathbf{P}(X \mid \mathbf{d}, h_i)\mathbf{P}(h_i \mid \mathbf{d}) = \sum_i \mathbf{P}(X \mid h_i)P(h_i \mid \mathbf{d}) \tag{20.2}$$

這裡，我們假定每個假設都決定了 X 上的一個機率分佈。這個公式顯示了預測是對每個單獨假設中預測的加權平均。這些假設本身本質上是原始資料和預測結果之間的「媒介」。貝氏方法的關鍵量是**假設的事前機率**(hypothesis prior)——$P(h_i)$，以及在每種假設下資料的**概似**(likelihood)——$P(\mathbf{d} \mid h_i)$。

對於我們的糖果例子，我們暫時先假定 $h_1, \ldots, h_5$ 的一個事前分佈是給定爲(0.1, 0.0, 0.2, 0.0, 0.1)，如製造商的宣傳所言。資料的概似率，是在觀察過程爲 **i.i.d.(**參見 18.4 節)的假設上所得到的計算，所以有

$$P(\mathbf{d} \mid h_i) = \prod_j P(d_j \mid h_i) \tag{20.3}$$

例如，假如一包糖果實際上是全酸橙的包裝(h_5)，而且前 10 顆糖都是酸橙味的，那麼 $P(\mathbf{d} \mid h_3)$是 0.5^{10}，這是因爲在 h_3 包中，有半數的糖果是酸橙味的[2]。在圖 20.1(a)顯示了 5 種假設事後機率在觀察到連續取出 10 顆酸橙糖的過程中是如何變化的。注意，機率是從它們的事前值開始的，所以一開始 h_3 是最可能的選擇並且保持到第 1 顆酸橙糖被剝開。在第 2 顆酸橙糖被剝開後，h_4 是最可能的；在第 3 顆之後以及更多糖果被剝開後，h_5(可怕的全酸橙包) 成爲最可能的。當一連出現 10 顆酸橙糖之後，我們相當確定我們的命運了。基於公式(20.2)，圖 20.1(b)給出了下一顆糖是酸橙味的預測機率。如我們所預期的一樣，它單調遞增地趨向於 1。

這個例子顯示，貝氏預測最終會與眞的那個假設一致。這就是貝氏學習的特性。對於任何沒有排除掉眞實假設值的固定事前機率，任何爲假的假設值的事後機率都會最終消退。這種情況的發生是因爲產生「不典型」資料的機率會不確定地逐漸減小。(這一點和在第十八章中關於 PAC 學習的討論相仿)。更重要的是，貝氏預測是最佳的，且與資料集的大小無關。只要給定了假設的事前機率，任何另外一種預測一般都不會比這個更正確。

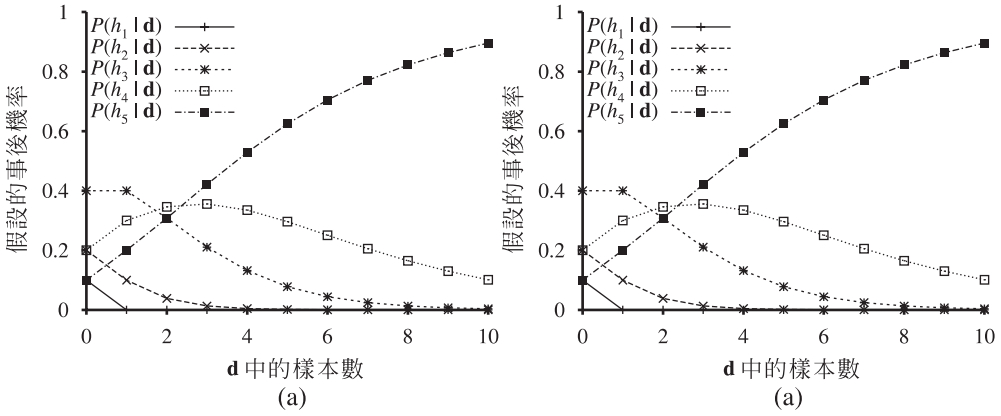

圖 20.1 (a) **根據公式**(20.1)**得到的事後機率** $P(h_i \mid d_1, \ldots, d_N)$。**觀察的數量** N **從** 1 **到** 10，**每個觀察結果都是一塊酸橙味糖果**。(b) **根據公式**(20.2)**得到的貝氏預測** $P(d_{N+1} = lime \mid d_1, \ldots, d_N)$

當然，貝氏的最佳特性的代價也很高。對於眞實的學習問題，假設空間通常很大甚至是無限的，如我們在第十八章中看到的那樣。有某些情況下，公式(20.2)中的求和(或者在連續的情況下的積分)能夠很容易地進行，但是大多數情況下我們必須求助於近似方法或經過簡化的方法。

一種很常見的近似方法——這種方法常用於科學計算——是基於單一的最可能假設進行預測的——也就是，h_i 使 $P(h_i \mid \mathbf{d})$最大化。這種方法經常被稱爲**最大事後假設**或縮寫爲 MAP 假設。根據一個 MAP 假設 h_{MAP} 做出的預測在 $\mathbf{P}(X \mid \mathbf{d}) \approx \mathbf{P}(X \mid h_{\text{MAP}})$的意義上來說近似於貝氏方法。在我們的糖果例子中，當連續取出 3 顆酸橙味糖之後，$h_{\text{MAP}} = h_5$，進而 MAP 方法預測第 4 顆糖果爲酸橙味的機率是 1.0——這是一個比圖 20.1(b)中所示的機率爲 0.8 的貝氏預測要危險很多的預測。隨著得到更多的資料，由於 MAP 假設的競爭者會變得越來越不可能，MAP 和貝氏預測就會更加接近。

因爲 MAP 假設需要解決的是一個最最佳化問題而非一個大的求和(或積分)問題，所以它常常比貝氏學習更容易，雖然在我們的例子中顯示不出這一點。不過，在本章後面的部分我們能夠看到相關的例子。

無論對於貝氏學習還是 MAP 學習，假設事前 $P(h_i)$都扮演了非常重要的角色。在第 18 章中，我們看到當假設空間表達能力過強時，會出現**過適配**的現象，以至於包含了許多能夠和資料集符合得很好的假設。與對被考慮的假設加上一個任意限制的方法不同，貝氏和 MAP 學習方法用事前機率來使複雜度高的假設處於不利地位。典型的情況是，越複雜的假設具有越低的事前機率——其中一部分原因在於複雜的假設通常多於簡單的假設。另一方面，更複雜的假設有更強的適配資料的能力。(極端的情況是使用尋找表，能夠機率爲 1 地重新產生資料)。因此，假設事前表現了在假設的複雜度和它對資料的適配程度之間的一種折衷。

我們可以在邏輯的情況中更清晰地看到這種折衷的效果，其中 H 只包括確定性的假設。這種情況下，如果 h_i 是一致的，則 $P(\mathbf{d} \mid h_i)$取值爲 1，否則取值爲 0。看看公式(20.1)，我們會發現 h_{MAP} 就是與資料一致的最簡單的邏輯理論。因此，讓事後學習取得最大值就提供了奧卡姆剃刀的一個自然表現。

另一種關於在複雜度和資料的適配度之間取得折衷的見解，是透過對公式(20.1)取對數得到的。選擇 h_{MAP}，最大化 $P(\mathbf{d} \mid h_i)P(h_i)$就等效於最小化

$$-\log_2 P(\mathbf{d} \mid h_i) - \log_2 P(h_i)$$

利用我們在 18.3.4 中介紹過的消息編碼與機率之間的關係，可以看出，$-\log_2 P(h_i)$項等於指定假設 h_i 所需的位數。而且，$-\log_2 P(\mathbf{d} \mid h_i)$在給定假設下，就是指定資料所需的附加位元數。(爲了理解這一點，考慮如果假設可以準確地直接預測資料，則一位元都不需要——如在 h_5 的情況下，並且出現一連串酸橙味的糖果——$\log_2 1 = 0$。)因此，MAP 學習就是要選擇提供最大的資料壓縮的假設。同樣的任務透過**最小描述長度**(或縮寫爲 MDL)學習演算法，能夠更直接地處理來處理。儘管 MAP 學習是透過賦予較簡單的假設擁有更高的機率來表達簡單性，MDL 表達方式更直接，其係透過計算二元編碼後的假設與資料所含的位元數。

最後一個簡化是利用於假設空間上假定一個**均勻**事前機率而得。這種情況下，MAP 學習過程退化爲選取一個可以使得 $P(\mathbf{d} \mid H_i)$最大化的 h_i 值。這被稱爲**最大概似**(ML)假設，h_{ML}。最大概似學習在統計學領域很常見，在該學科中許多研究者不信任假設事前的主觀本質。當沒有理由透過推理優先選擇一個假設而不是另一個時——例如，所有的假設複雜度都相同時——最大概似的方法是很合理的。當資料集很大時，由於資料會淹沒假設的事前分佈，所以這種方法爲貝氏和 MAP 學習提供一個很好的近似，但是它對於小資料集是有問題的(後面我們會看到這種情況)。

20.2 完整資料下的學習

機率模型——假設資料是從此模型產生出來的前提下——學習的一般任務，被稱作**密度估計**。(原來是應用在機率密度函數中連續的變數的一個項，在這兒也可以被用在離散分佈中)。

這個章節涵括了最簡單的例子，即我們擁有**完整資料**。當每個資料點包含了待學習的機率模型中每個變數的值的時候，資料就是完整的。我們將專注在**參數學習**上面——爲結構固定的機率模型尋找數值參數。例如，我們可能對學習一個給定了結構的貝氏網路中的條件機率感興趣。我們也將簡要地介紹一下結構學習上的問題與無參數的密度估計。

20.2.1 最大概似參數學習：離散模型

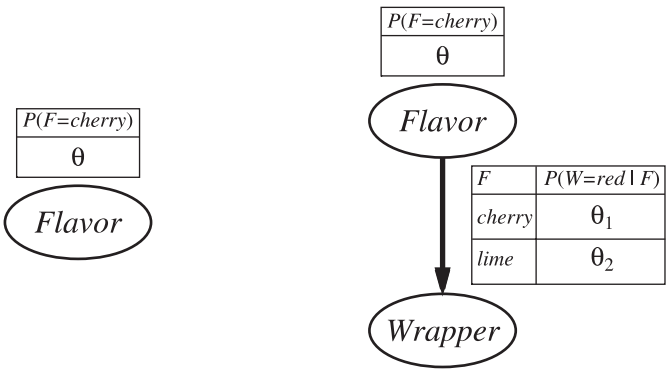

圖 20.2

(a) **針對未知櫻桃與酸橙比例的糖果情況的貝氏網路模型；**

(b) **針對糖紙顏色(機率地)取決於糖果口味的模型**

設想我們從一個新的糖果商那裡買了一包酸橙糖與櫻桃糖，其中酸橙糖和櫻桃糖的比例是完全未知的——也就是說，可能是 0 到 1 之間的任意值。在這種情況下，我們擁有了一組連續假設。此例中我們稱爲 θ 的**參數**是櫻桃糖的比例，假設則記爲 h_θ (酸橙的比例就是 $1-\theta$)。如果我們假定所有的比例都有相同的事前，那麼最大概似方法是合理的。如果我們用貝氏網路對這種情景模式化，我們只需要一個隨機變數，*Flavor*(口味)(從糖果包裡隨機選取出來的一顆糖的口味)。它的取值可以爲 *cherry*(櫻桃)和 *lime*(酸橙)，其中取 *cherry* 的機率爲 θ [參見圖 20.2(a)]。現在設想我們剝開了 N 顆糖，其中有 c 顆是櫻桃味的，有 $\ell = N - c$ 顆是酸橙味的。根據公式(20.3)，這個特定資料集的概似機率爲

$$P(\mathbf{d} \mid h_\theta) = \prod_{j=1}^{N} P(d_j \mid h_\theta) = \theta^c \cdot (1-\theta)^\ell$$

利用使得這個運算式最大化的 θ 值，可以得到最大概似假設。透過使**對數概似機率**最大化，也可以得到同樣的值，

$$L(\mathbf{d} \mid h_\theta) = \log P(\mathbf{d} \mid h_\theta) = \sum_{j=1}^{N} \log P(d_j \mid h_\theta) = c \log\theta + \ell \log(1-\theta)$$

(透過取對數，我們將資料的乘積簡化爲對資料求和，通常更容易運算)。爲了找到 θ 的最大概似值，我們把 L 對 θ 求導並令結果運算式等於 0：

$$\frac{dL(\mathbf{d} \mid h_\theta)}{d\theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \qquad \Rightarrow \qquad \theta = \frac{c}{c+\ell} = \frac{c}{N}$$

那麼，用自然語言表達就是，最大概似假設 h_{ML} 斷言糖果包中櫻桃糖的實際比例和到目前爲止所觀察到的已剝開糖果中的比例相等！

看來我們做了大量工作來發現一個顯然的事實。儘管如此，實際上我們已經展示了進行最大概似參數學習的一種標準方法：

1. 寫出資料的概似運算式，它是待學習參數的一個函數。
2. 對每個參數的對數概似進行求導。
3. 找到滿足導數爲 0 的對應參數值。

比較需要技巧的通常是最後一步。在我們的例子中這一點並不明顯，但是在很多情況下，我們可能會需要求助於疊代求解演算法或者其他數值最佳化技術，如在第四章中所述。這個例子還描述了總體而言最大概似學習的一個重要問題：當資料集足夠小，以致有些事件沒有被觀察到時——例如，沒有櫻桃糖——最大概似假設會爲這些沒有被觀察到的事件賦予 0 機率。各種技巧被用來避免這個問題，例如將每個事件的計數初始化爲 1 而不是 0。

讓我們來看另一個例子。設想這個新的糖果商打算給顧客一點兒提示，使用了紅色的和綠色的糖紙。每顆糖果的 *Wrapper*(糖紙)的選取是有一定機率的，依據一個依賴於口味的未知條件分佈。對應的機率模型如圖 20.2(b)所示。注意，這裡有 3 個參數：θ、θ_1 和 θ_2。利用這幾個參數，觀察到例如說一顆櫻桃糖包著綠糖紙的概似可以從貝氏網路的標準語義得到(第 14.2 節)：

$$\begin{aligned}&P(Flavor = cherry, Wrapper = green \mid h_{\theta,\theta_1,\theta_2})\\&= P(Flavor = cherry \mid h_{\theta,\theta_1,\theta_2}) P(Wrapper = green \mid Flavor = cherry, h_{\theta,\theta_1,\theta_2})\\&= \theta \cdot (1-\theta_1)\end{aligned}$$

現在我們剝開 N 顆糖，其中 c 顆是櫻桃味的，ℓ 顆是酸橙味的。糖紙的計數是這樣的：r_c 顆櫻桃糖是紅色的糖紙，g_l 顆是綠色的糖紙，而 r_ℓ 顆酸橙糖是紅色的糖紙，g_ℓ 顆是綠色的糖紙。則資料的概似可以由下式給出

$$P(\mathbf{d} \mid h_{\theta,\theta_1,\theta_2}) = \theta^c (1-\theta)^\ell \cdot \theta_1^{r_c} (1-\theta_1)^{g_c} \cdot \theta_2^{r_\ell} (1-\theta_2)^{g_\ell}$$

這個公式看起來非常可怕，不過取對數是有幫助的：

$$L = [c \log\theta + l \log(1-\theta)] + [r_c \log\theta_1 + g_c \log(1-\theta_1)] + [r_l \log\theta_2 + g_l \log(1-\theta_2)]$$

取對數的好處很明顯：對數概似是 3 項的和，每一項只包含單一的參數。當我們對每一個參數進行求導並令其等於 0 時，可以得到 3 個獨立的方程，每個只包含一個參數：

$$\frac{\partial L}{\partial\theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \qquad \Rightarrow \quad \theta = \frac{c}{c+\ell}$$

$$\frac{\partial L}{\partial\theta_1} = \frac{r_c}{\theta_1} - \frac{g_c}{1-\theta_1} = 0 \qquad \Rightarrow \quad \theta_1 = \frac{r_c}{r_c+g_c}$$

$$\frac{\partial L}{\partial\theta_2} = \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1-\theta_2} = 0 \qquad \Rightarrow \quad \theta_2 = \frac{r_\ell}{r_\ell+g_\ell}$$

θ 的求解和前面相同。θ_1 的解，即櫻桃糖包著紅色糖紙的機率，就是實際觀察到的櫻桃糖包著紅色糖紙的比例，θ_2 也是類似的。

這些結果是非常鼓舞人心的，而且容易看出，它們可以擴展到任何條件機率能夠用表格表示的貝氏網路。最重要的一點是，在完整資料的條件下，一個貝氏網路的最大概似參數學習問題被分解爲幾個單獨的學習問題，每個參數對應於一個。參見習題 20.7，一種非表格的情形，其中每個參數影響若干個條件機率。第二點是，已知一個變數的母變數，則它的參數值就是在每種母變數取值設置下所觀察到的該變數值的頻率。如前所述，當資料集很小時我們必須小心避免零值的問題。

20.2.2 原始貝氏模型

或許用於機器學習的最常見的貝氏網路模型是在 13 章初次介紹的原始貝氏模型。在這個模型中，「類」變數 C(需要進行預測)是根節點，而「屬性」變數 X_i 是葉節點。之所以這個模型是「原始的」，是因爲對於給定的類，它假定了各個屬性彼此是條件獨立的。(圖 20.2(b)中的模型是只有一個屬性的原始貝氏模型)。假設使用布林型變數，則參數爲

$$\theta = P(C = true),\ \theta_{i1} = P(X_i = true \mid C = true),\ \theta_{i2} = P(X_i = true \mid C = false)$$

最大概似參數值透過與圖 20.2(b)中完全相同的方式得到。一旦模型經過這種方式的訓練，它就可用於對類變數 C 尚未觀察到的新實例進行分類。根據觀察到的屬性值 $x_1, \ldots, x_n$，可給出每類機率如下：

$$\mathbf{P}(C \mid x_1, \ldots, x_n) = \alpha\, \mathbf{P}(C) \prod_i \mathbf{P}(x_i \mid C)$$

透過選取最可能的類可以得到確定性的預測。圖 20.3 顯示了當這種方法用於第十八章中的餐館問題時得到的學習曲線。該方法論的學習效果很好，儘管比不上決策樹學習方法。這大概是因爲眞實假設——是一棵決策樹——無法透過原始貝氏模型精確地表示。當然，事實上原始貝氏學習在很寬範圍的應用中都有出人意料的好效果，它的經過 boost 改進的版本(習題 20.5)是最有效的通用學習演算法之一。原始貝氏學習可以很好地擴展到超大規模的問題：如果有 n 個布林屬性，就有 $2n + 1$ 個參數，並且不需要透過搜尋來尋找最大概似的原始貝氏假設 h_{ML}。最後，原始貝氏學習系統可輕鬆應付有雜訊或遺失的資料，並在適當的時候給出機率預測。

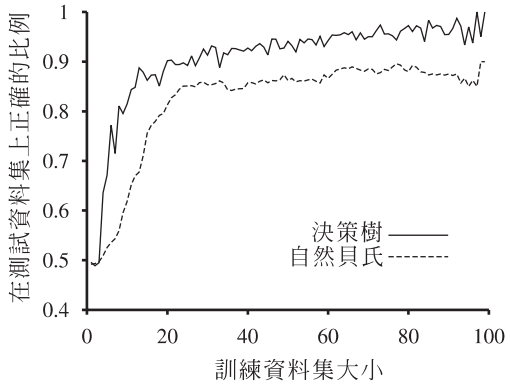

圖 20.3　把原始貝氏學習方法應用於第 18 章的餐館問題得到的學習曲線；同時顯示了使用決策樹學習得到的學習曲線，作為對比。

20.2.3 最大概似參數學習：連續模型

在第 14.3 節中介紹了連續機率模型，例如**線性高斯**模型。因爲在現實世界的應用中連續變數的情況是非常普遍的，所以瞭解如何從資料中學習連續模型是非常重要的。最大概似學習的原理在連續及離散情況中均同。

讓我們從一個非常簡單的例子開始：學習一個單一變數高斯密度函數的參數。也就是說，資料根據下式產生：

$$P(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

這個模型的參數是平均值 μ 和標準差 σ。(注意，正規化「常數」依賴於 σ，所以我們不能忽略它)。令觀察值爲 $x_1, \ldots, x_N$。則對數概似爲

$$L=\sum_{j=1}^{N}\log\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x_j-\mu)^2}{2\sigma^2}}=N(-\log\sqrt{2\pi}-\log\sigma)-\sum_{j=1}^{N}\frac{(x_j-\mu)^2}{2\sigma^2}$$

如往常一樣，令導數爲 0，我們得到

$$\begin{aligned}\frac{\partial L}{\partial\mu}&=-\frac{1}{\sigma^2}\sum_{j=1}^{N}(x_j-\mu)=0 &\Rightarrow\quad \mu&=\frac{\sum_j x_j}{N}\\ \frac{\partial L}{\partial\sigma}&=-\frac{N}{\sigma}+\frac{1}{\sigma^3}\sum_{j=1}^{N}(x_j-\mu)^2=0 &\Rightarrow\quad \sigma&=\sqrt{\frac{\sum_j(x_j-\mu)^2}{N}}\end{aligned} \tag{20.4}$$

也就是說，平均值的最大概似值就是樣本的平均值，而標準差的最大概似值就是樣本變異數的平方根。我們再一次得到了令人滿意的結果，和「常識性」經驗相吻合。

現在，考慮一個線性高斯模型，具有一個連續的母變數 X 和一個連續的子變數 Y。如 14.3 節所解釋，Y 具有高斯分佈，其平均會線性取決於 X 之值，而其標準差爲固定。爲了學習條件分佈 $P(Y \mid X)$，我們可以最大化條件概似：

$$P(y\mid x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y-(\theta_1x+\theta_2))^2}{2\sigma^2}} \tag{20.5}$$

這裡，參數爲 θ_1，θ_2 和 σ。資料是二元組(x_j, y_j)的一個集合，如圖 20.4 所示。應用通常的方法(習題 20.6)，我們可以找到各個參數的最大概似值。此處所述的論點不同。如果我們只考慮參數 θ_1 和 θ_2，它們定義了 x 和 y 之間的線性關係，則很明顯地，最大化關於這些參數的對數概似，等效於最小化公式(20.5)中冪的分子部分$(y-(\theta_1x+\theta_2))^2$：這就是 L_2 損失，即實際值 y 和預測值 $\theta_1x+\theta_2$ 之間的誤差平方。這個量值即是透過 18.6 節中所介紹的標準**線性回歸**過程進行最小化所得。現在我們可以理解原因：倘若資料產生時帶有固定變異數的高斯雜訊，最小化誤差平方和就提供了最大概似的直線模型。

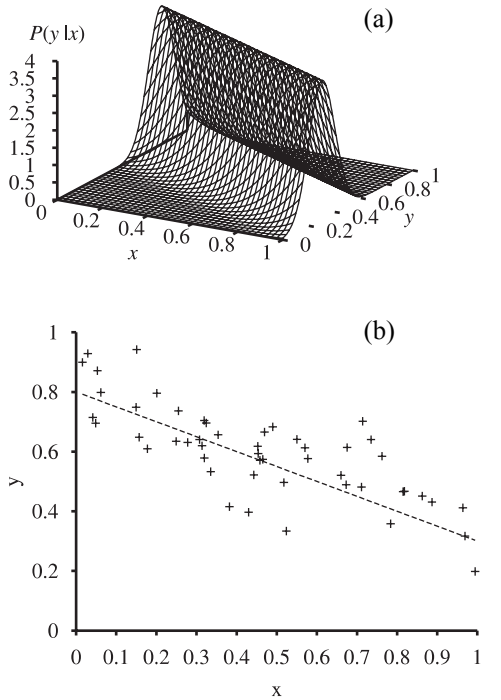

圖 20.4 (a) **一個線性高斯模型為** $y=\theta_1x+\theta_2$ **加上具固定變異數的高斯雜訊。**(b) **一個根據該模型產生的** 50 **個資料點集合**

20.2.4 貝氏參數學習

最大概似學習引出了某些很簡單的過程，但是對於小資料集它有一些嚴重的不足。例如，在看到一顆櫻桃糖後，最大概似假設的結果爲這個包是 100%的櫻桃糖(也就是 $\theta = 1.0$)。除非有一個假設事前指出糖果包裡要麼都是櫻桃糖要麼都是酸橙糖，否則這就不是一個合理的結論。這袋糖果更有可能是酸橙糖跟櫻桃糖的綜合口味。參數學習的貝氏方法開始先在可能的假設之上定義事前機率分佈。我們稱其爲**事前假設**。隨著資料的到達，事後機率分佈也隨之更新。

圖 20.2(a)中的糖果例子有一個參數 θ：隨機選取的一顆糖是櫻桃口味的機率。從貝氏的角度看，θ 是一個隨機變數 Θ(其定義假設空間)的(未知)取值；而事前假設就是事前分佈 **P**(Θ)。這樣，$P(\Theta = \theta)$ 就是包裡櫻桃糖的比例爲 θ 的事前機率。

如果參數 θ 可以取 0 到 1 之間的任意值，則 **P**(Θ)必須是一個連續分佈，它的取值只在 0 到 1 之間且不爲 0，該分佈的積分爲 1。均勻密度 $P(\theta) = Uniform\ [0, 1](\theta)$是一個候選(參見第 13 章)。已知均勻密度是 ***β* 分佈(**beta distribution**)**家族的一個成員。每個 β 分佈由兩個**超參數**[3] a 和 b 來定義，如下式：

$$\beta\,[a, b](\theta) = \alpha\,\theta^{a-1}(1-\theta)^{b-1} \tag{20.6}$$

θ 的取值範圍是[0, 1]。使得分佈的積分爲 1 的正規化常數 α 是取決於 a 和 b。(參見習題 20.8)。圖 20.5 顯示了對於 a 和 b 的各種值的分佈情況。分佈的平均值爲 $a / (a + b)$，所以較大的 a 值暗示著一種信念，即 Θ 更接近於 1 而不是 0。較大的 $a+b$ 的值使得分佈的峰值更突顯，暗示著關於 Θ 值的更大把握。因此，β 函數族就提供了一系列有用的可能事前假設。

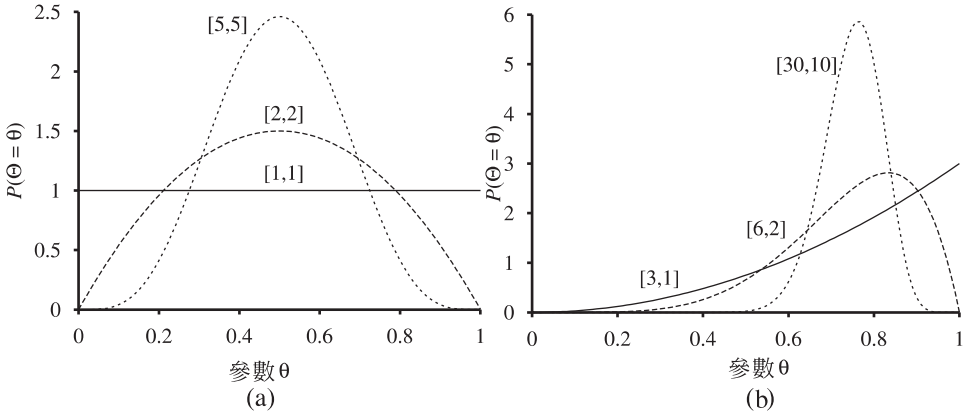

圖 20.5　當[*a*, *b*]取不同值時的 β [*a*, *b*]分佈的例子

在 β 函數族的靈活性之外，它還具有另一個很好的特性：如果 Θ 有一個事前 β [*a*, *b*]，那麼在觀察到一個資料點之後，Θ 的事後分佈也是一個 β 分佈。換言之，β 在更新後就封閉了。β 函數族被稱爲布林變數分佈族的**共軛事前**[4]。讓我們看一下它是如何起作用的。假設我們觀察到一顆櫻桃糖，則：

$$P(\theta \mid D_1 = cherry) = \alpha P(D_1 = cherry \mid \theta) P(\theta)$$
$$= \alpha' \theta \cdot \beta [a, b](\theta) = \alpha' \theta \cdot \theta^{a-1}(1-\theta)^{b-1}$$
$$= \alpha' \theta^{a}(1-\theta)^{b-1} = \beta [a+1, b](\theta)$$

因此，在看到一顆櫻桃糖後，我們簡單地增加參數 a 的值得到事後機率；同樣，在看到一顆酸橙糖後，我們增加參數 b 的值。這樣，可以將超參數 a 和 b 視爲**虛擬計數**，在某種意義上說事前 $\beta [a, b]$ 的行爲表現正如我們是從均勻事前 $\beta[1, 1]$開始的，並且實際看到了 $a-1$ 顆櫻桃糖及 $b-1$ 顆酸橙糖。

透過考察 a 和 b 按照固定比例遞增的一系列 β 分佈，我們可以生動地看到參數 Θ 的事後分佈如何隨著資料的到達而變化的。例如，假如實際的糖果包中有 75%爲櫻桃味的。圖 20.5(b)顯示了序列 $\beta [3, 1]$，$\beta [6, 2]$，$\beta [30, 10]$。顯然，分佈收斂到眞實值 Θ 附近的一個狹窄峰值。另外，對於大資料集，貝氏學習(至少在這個例子中)的收斂給出與最大概似學習相同的結果。

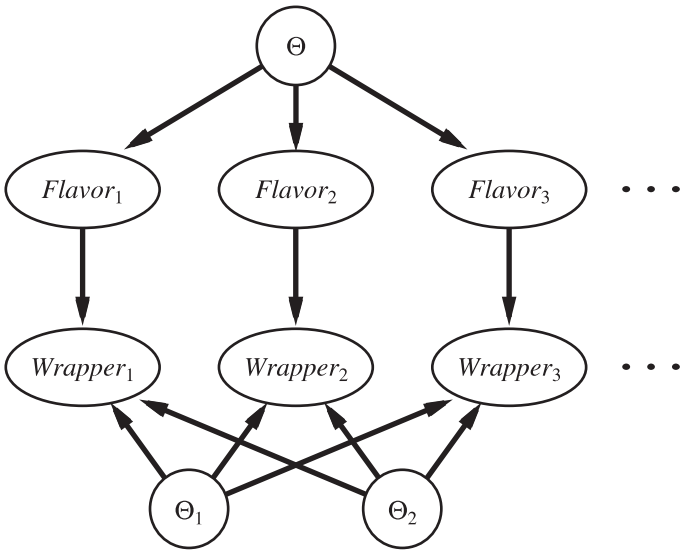

圖 20.6 與一個貝氏學習過程相對應的貝氏網路。參數變數 Θ，Θ_1 和 Θ_2 的事後分佈可以從它們的事前分佈及變數 $Flavor_i$ 和 $Wrapper_i$ 中的證據來導出

現在讓我們來看一個更複雜的例子。圖 20.2(b)中的網路有 3 個參數：θ，θ_1 和 θ_2，其中 θ_1 是紅色糖紙包著櫻桃糖的機率，θ_2 是紅色糖紙包著酸橙糖的機率。對於貝氏假設事前而言，必須涵蓋所有的 3 個參數——也就是說，我們需要指定 $\mathbf{P}(\Theta, \Theta_1, \Theta_2)$。通常，我們假定**參數獨立性**：

$$\mathbf{P}(\Theta, \Theta_1, \Theta_2) = \mathbf{P}(\Theta)\,\mathbf{P}(\Theta_1)\,\mathbf{P}(\Theta_2)$$

在這個假定條件下，每個參數都可以有自己的 β 分佈，並可以隨著資料的到達分別進行更新。從圖 20.6 可以看出我們是怎麼將事前假設與任意資料合併到一個貝氏網路中。節點 Θ，Θ_1 與 Θ_2 並沒有母節點。然而每次我們觀察糖紙與一塊糖果的相對應的口味，我們就增加一個 $Flavor_i$ 節點，其係基於口味參數 Θ：

$$P(Flavor_i = cherry \mid \Theta = \theta) = \theta$$

同時，我們也會增加一個結點 $Wrapper_i$，其取決於 Θ_1 與 Θ_2：

$$P(Wrapper_i = red \mid Flavor_i = cherry, \Theta_1 = \theta_1) = \theta_1$$

$$P(Wrapper_i = red \mid Flavor_i = lime, \Theta_2 = \theta_2) = \theta_2$$

現在，整個貝氏學習過程可以被形式化表示爲一個推理問題。我們新增了一些證據節點，然後接著查詢未知的節點(在這個例子中指的是 Θ，Θ_1，Θ_2)。這種學習和預測的形式化方法明確了一個事實：貝氏學習不需要額外的「學習理論」。並且，本質上也只有一種學習演算法，即貝氏網路的推理演算法。當然啦，這些網路的本質與 14 章提到的網路有些不同，原因在於潛在的大量的證據變數所代表的訓練集和常見的連續值參數變數。

20.2.5 學習貝氏網路的結構

到目前爲止，我們一直假定貝氏網路的結構是已知的，而我們只是試圖學習參數。網路的結構表示了一個域的基本因果知識，往往專家甚至新手都很容易提供。然而，在某些情況下，因果模型可能無法得到或者有爭議——例如，有的公司一直宣稱吸煙不會導致癌症——所以，重要的是理解如何能夠從資料中學習貝氏網路的結構。這個章節針對主要的想法進行一個簡要的概述。

最顯而易見的方法是搜尋一個好的模型。我們可以從不包含任何有連結的模型入手，然後開始爲每個節點添加母節點，要符合我們剛剛談論方法中的參數，並且估量結果模型的準確性。此外，我們還可以先對模型的結構有一個初始猜測，然後使用爬山法或者模擬退火搜尋進行修正，隨著結構的每次變化調整參數。修正方法可以包括對連接進行反轉、添加、或刪除。在這個過程中我們必須避免引入循環，所以很多演算法假定變數是有順序的，而一個節點的母節點必須是在排序中先出現的節點(正如第十四章的構造過程一樣)。爲了完全的一般性，我們還需要尋找可能的排序。

還有兩種可選方法用於判斷何時一個合適的結構已經被找到。首先測試隱含於結構中的條件獨立性斷言是否確實能滿足資料。例如，針對餐館問題的原始貝氏模型就假定了

$$\mathbf{P}(Fri/Sat, Bar \mid WillWait) = \mathbf{P}(Fri/Sat \mid WillWait)\ \mathbf{P}(Bar \mid WillWait)$$

然後我們可以檢驗資料，在對應的條件頻率之間上式成立與否。現在，即使這個結構描述了域的眞實因果本質，資料集中的統計波動意味著該公式永遠無法得到嚴格的滿足，所以我們需要執行一個合適的統計測試，來看看是否有足夠的證據顯示獨立性假設被破壞了。結果網路的複雜度將取決於測試中使用的臨界值——獨立性測試越嚴格，加入的有連結越多，過適配的風險也就越大。

一種和本章主旨更一致的方法來自於所提出模型(在機率意義下)對資料的解釋程度。不過，須非常注意如何測量解釋程度。如果只是嘗試找到最大概似假設，則我們最終會得到一個全連接網路，這是因爲給一個節點添加更多的母節點不會減少概似程度(習題 20.9)。我們被迫以某種方式對模型的複雜度加以懲罰。MAP(或 MDL)方法只是在比較不同的結構之前簡單地根據結構的概似度(在參數調整之後)減去懲罰值。貝氏方法是在結構和參數之上設置一個聯合的事前機率。通常情況下由於有過多的結構而無法求和(是變數數目的超指數量級)，所以大部分研究者採用 MCMC 對結構進行取樣。

懲罰複雜度(無論透過 MAP 還是貝氏方法)在最佳化結構與網路中條件分佈的表示本質之間引入了一個重要的聯繫。對於表格化表示的分佈，一個節點的分佈複雜度懲罰隨著母節點數目的增加呈指數級增長，但是，對於例如說「雜訊或」分佈，則只是線性增長。這意味著透過雜訊或模型(或其他簡潔的參數化模型)進行學習傾向於比透過表格化分佈進行學習產生包含更多母節點的結構。

20.2.6 無參數模型的密度估計

可以藉由採取 18.8 節的無參數方法，從而在不對其結構及參數化做任何假設之情況下，學習一機率模型。進行**無參數密度估計**的工作通常是在連續域中完成，正如圖 20.7(a)所示。圖中可以看到由兩個連續變數所定義出空間中的機率密度函數。圖 20.7(b)我們看到的樣本是從密度函數取得的一些資料點。問題來了，我們是否有辦法從這些取樣來回復一個模型呢？

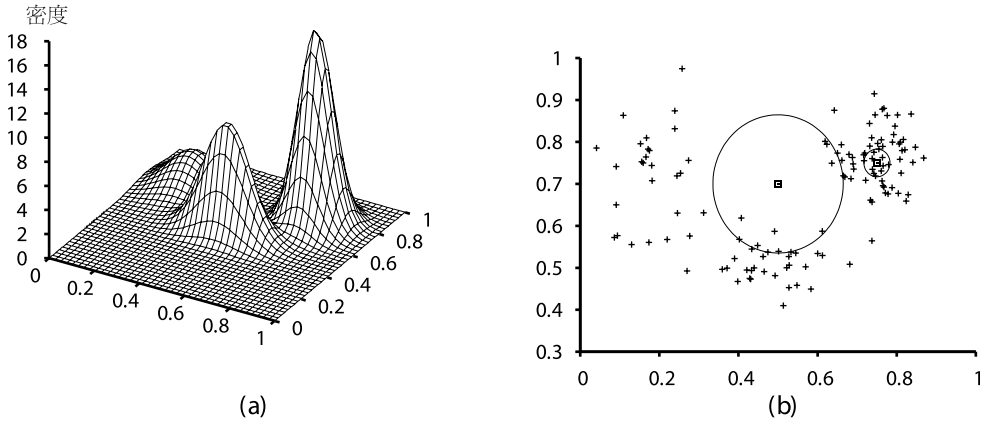

圖 20.7 (a) **由圖** 20.11(a)**所得的一個高斯混合的** 3D **圖示**

(b) **從該混合中取的** 128 **點樣本，連同兩個查詢點(小正方形)其及** 10-**最近鄰(中圓圈與大圓圈)**

一開始我們會先考慮 *k*-**最近鄰**模型。(在第 18 章中我們看到最近鄰模型用在分類與回歸；而這邊我們看到它用在密度估計)。給予一個資料點的樣本，在查詢點 **x** 估計未知的機率密度我們可以簡單的量測 **x** 的近鄰周邊資料點的密度。圖 20.7(b)可以看到兩個查詢點(小正方形)對於每一個查詢點我們已經畫出圍繞 10 個臨近點的最小的圓圈——10 個最近鄰。我們可以看到中間的圓很大，表示這裡是低密度，而右邊的圓較小，表示這裡是高密度。在圖 20.8 中，我們看到三張使用 *k*-最近鄰模型的密度估計圖，分別針對不同的 *k* 值。看來(b)應該是對的，(a)有太多突尖(*k* 值太小)，(c)太過平緩(*k* 值太大)。

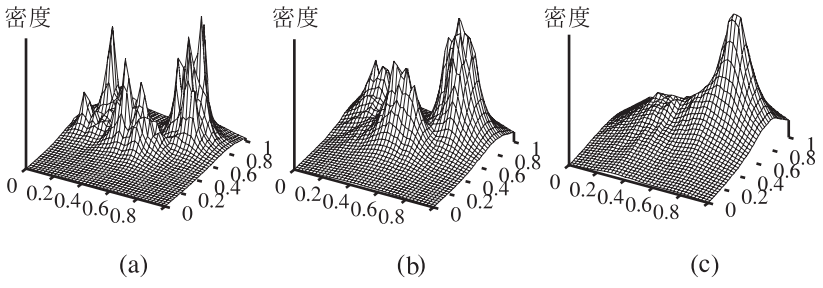

圖 20.8 **使用** *k*-**最近鄰的密度估計，可應用到圖** 20.7(b)**中的資料，依序分別是** *k* =3，10 **和** 40。*k* = 3 **會產生太多突尖，**40 **太過平緩，**10 **就差不多正確。***k* **的最佳值可以透過交叉驗證進行選取。**

另一個可能性是使用**核函數**，如同我們在局部加權回歸所做的。要將核模型應用到密度估計，先使用高斯核假設每個資料點會產生自己的小密度函數。查詢點 **x** 上的估計密度於是就成爲每個核函數給予的平均密度：

$$P(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{x}, \mathbf{x}_i)$$

我們將假定是球狀高斯，每個軸的標準差是 w：

$$K(\mathbf{x}, \mathbf{x}_i) = \frac{1}{(w^2\sqrt{2\pi})^d} e^{-\frac{D(\mathbf{x}, \mathbf{x}_i)^2}{2w^2}}$$

其中 d 是 **x** 的維數且 D 是歐式距離函數。選取一個合適的核寬度 w 值對我們來說仍然是一個問題；圖 20.9 可看出太小、剛好與太大的值。可以透過交叉驗證選取一個合適的 w 值。

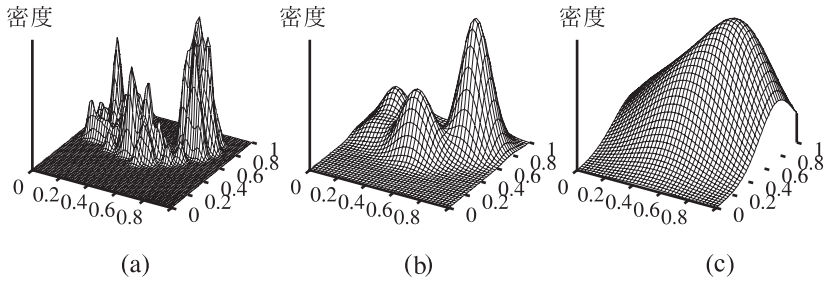

圖 20.9　圖 20.7(b)中資料的核密度估計，使用了高斯核，其中 w 分別取 0.02，0.07 和 0.20。w = 0.07 是差不多正確

20.3 隱變數學習：EM 演算法

前面的章節處理的是完全可觀察的情況。而許多現實世界的問題在可得到的學習資料中有不可觀察的**隱變數**(有時稱爲**潛變數**)。例如，醫療記錄通常包括表現症狀、醫師診斷、所使用的治療方式、可能還包括治療結果等，但是很少包含對疾病本身的直接觀察！(注意到診斷並非疾病；其係表觀症狀的因果結論，而症狀是由疾病引起的)。有人也許會問，「既然疾病無法觀察到，那麼爲什麼不構造一個不包含它的模型呢？」答案出現在圖 20.10 中，圖中顯示了關於心臟病的一個小的、假想的診斷模型。有 3 個可以觀察的誘病因素和 3 種可以觀察的症狀(其名稱太沉悶，在這裡就不寫了)。設每個變數有 3 個可能的取值[例如，*none*(無)，*moderate*(中等)，和 *severe*(嚴重)]。從圖(a)的網路中去掉隱變數，產生圖(b)的網路；參數的總數從 78 增加到 708。因此，潛變數可以急劇地減少指定一個貝氏網路所需參數的數目。接著，這還可以急劇減少學習參數所需的資料量。

隱變數是很重要的，不過它們確實會使學習問題複雜化。以圖 20.10(a)爲例，給定父節點，如何學習 *HeartDisease*(心臟病)的條件分佈並不是顯而易見的，因爲我們不知道在每種情況下 *HeartDisease* 的值；相同的問題在學習症狀的分佈時同樣會出現。本節描述了一種被稱爲**期望最大化**的演算法，或縮寫爲 EM，它可以用一種非常通用的方式解決這個問題。我們將展示 3 個實例，然後提供通用的描述。演算法初看上去像魔法，不過一旦發展出那種直覺知識，就可以在學習問題的巨大範圍內爲 EM 找到用武之地。

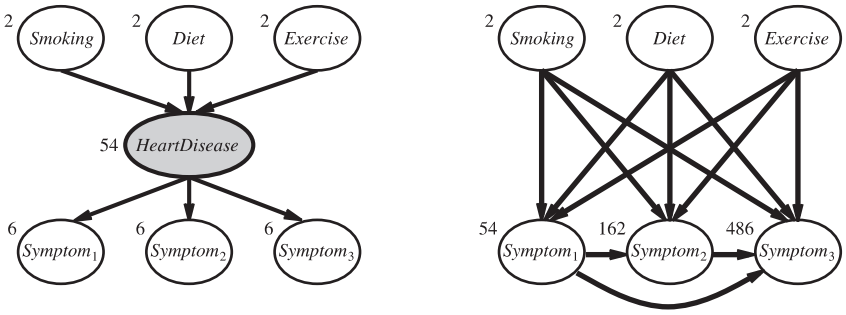

圖 20.10 (a) 一個心臟病的簡單診斷網路，假定 *HeartDisease*(心臟病)為一個隱變數。每個變數有 3 個可能的取值，標出的數字是在其條件分佈下獨立參數的數目；一共有 78 個

(b) 去掉 *HeartDisease* 後的等效網路。注意，給定父節點下，症狀變數不再是條件獨立的。這個網路需要 708 個參數

20.3.1 無監督群集：學習混合高斯分佈

無監督群集是一個在物件集上辨識多種類別的問題。這個問題是無監督的，因爲沒有給出類別的標記。例如，設想我們記錄了 10 萬顆恒星的光譜，透過光譜能揭示出恒星的不同類型嗎？如果可以，它們有多少特性，以及特性有哪些？我們都聽說過諸如「紅巨星」和「白矮星」這樣的術語，但是恒星不會把這些標籤寫在自己的帽子上——天文學家們必須執行無監督的群集來辨識它們的類別。其他例子還包括在林奈(Linnæan)生物分類學中對種、屬、目等等的辨識，以及建立自然種類對普通物體劃分類別(參見第 12 章)。

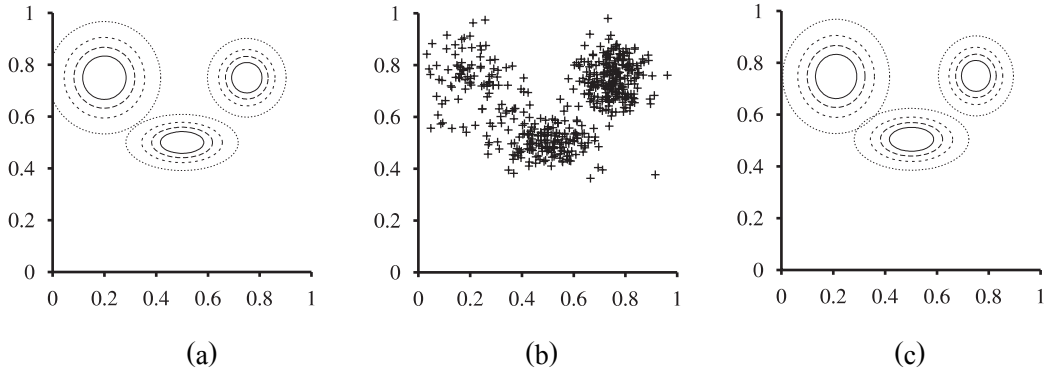

圖 20.11 (a) 有 3 種組成元素的高斯混合模型；權重(從左到右)為 0.2，0.3 和 0.5

(b) 取樣自(a)圖模型的 500 個資料點

(c) 透過由(b)資料而得之 EM 所重建之模型

無監督群集從資料入手。圖 20.11(b)中顯示了 500 個資料點，每個點指定兩個連續屬性的值。這些資料點與恒星相對應，而屬性可以與在兩個特定頻率上的光譜強度相對應。下一步，我們需要理解何種可能的分佈會產生這些資料。群集方法假定資料是由一個**混合分佈** P 產生的。這樣的分佈有 k 種**元素**，每種元素本身是一個分佈。資料點的產生是透過先選擇一種元素、然後根據該元素產生一個實例而完成的。令隨機變數 C 代表元素，取值爲 1, ... , k，則混合分佈由下式給出：

$$P(\mathbf{x}) = \sum_{i=1}^{k} P(C = i)P(\mathbf{x} \mid C = i)$$

其中 $\mathbf{x}$ 代表一個資料點的屬性值。在連續型資料的情況下，對於元素的分佈的一個自然選擇就是多元高斯分佈，也就是所謂的**混合高斯**分佈族。混合高斯分佈的參數是 $w_i = P(C = i)$(即每種元素的權值)，$\boldsymbol{\mu}_i$(每種元素的平均值)，以及 $\boldsymbol{\Sigma}_i$(每種元素的共變異數)。圖 20.11(a)所示的是 3 個高斯分佈的混合;這個分佈實際上是圖(b)中的資料來源，亦是圖 20.7(a)所示之模型。

於是，無監督群集問題就是要從類似於圖 20.11(b)中的原始資料恢復出一個類似於圖 20.11(a)中所示的混合模型。很明顯，如果我們知道哪種元素產生了每個資料點，那麼就更容易恢復出每種元素的高斯分佈：我們可以從給定的元素中選出所有的資料點，然後用公式(20.4)(的一個多元版本)來適配資料集的高斯分佈的參數。另一方面，如果我們知道每種元素的參數，那麼我們至少可以在一定的機率意義下將每個資料點分配到某種元素。問題在於我們既不知道分配也不知道參數。

在這種背景下 EM 方法的基本觀念是，假裝我們知道模型的參數，然後推斷出每個資料點屬於每種元素的機率。接下來，我們用元素對資料進行重新適配，每種元素都針對整個資料集進行適配，根據每個資料點屬於每種元素的機率對其加權。這個過程疊代進行，直到收斂爲止。本質上，我們是基於當前的模型，透過推斷隱變數的機率分佈——每個資料點屬於哪種元素——的方式對資料進行「完備化」。對於混合高斯模型，我們可以任意初始化混合模型的參數，然後按照下面的兩步過程進行疊代：

1. E 步驟：

計算機率 $p_{ij} = P(C = i \mid \mathbf{x}_j)$，資料 $\mathbf{x}_j$ 由元素 i 產生的機率。根據貝氏法則，我們得到 $p_{ij} = \alpha\, P(\mathbf{x}_j \mid C = i)\, P(C = i)$。$P(\mathbf{x}_j \mid C = i)$項就是第 i 個高斯分佈中 $\mathbf{x}_j$ 的機率，而 $P(C = i)$項就是第 i 個高斯分佈的加權參數。定義 $p_i = \boldsymbol{\Sigma}_j\, p_{ij}$，表示目前被指定到元素 i 的有效資料點數目。

2. M 步驟：

依序使用下列步驟來計算新的平均值、共變異數和元素權重：

$$\mu_i \leftarrow \sum_j p_{ij}\mathbf{x}_j / n_i$$

$$\Sigma_i \leftarrow \sum_j p_{ij}(\mathbf{x}_j - \mu_i)^{\mathsf{T}} / n_i$$

$$w_i \leftarrow n_i / N$$

其中 N 是資料點的總數。E 步驟，或稱爲期望步驟，可以被視爲計算隱含的**指示變數** Z_{ij} 的期望值 p_{ij}，其中如果資料 $\mathbf{x}_j$ 是由第 i 個元素產生的，則 Z_{ij} 取 1，否則取 0。M 步驟，或稱爲最大化步驟，在給定隱含指示變數的期望值的條件下，尋找使資料的對數概似最大化的新參數值。

當 EM 學習用於圖 20.11(a)中資料時，其最後一個模型如圖 20.11(c)所示；它在視覺上很難和產生資料的原始模型區分開。圖 20.12(a)在 EM 進展中根據當前模型繪製了資料的對數概似。

這裡有兩點需要注意。第一，最後學習到的模型的對數概似稍微超過了產生資料的原始模型。這看起來可能有些奇怪，不過它恰好反映了資料是隨機產生的，因而可能無法準確反映底層模型這一事實。第二點是，*EM* 在每次疊代中增加了資料的對數概似。這個事實很容易作一般性證明。此外，在一定條件下，可以證明 EM 能夠達到概似的局部極大值。(在很少的情況下，它可以到達一個鞍點或者甚至一個局部極小值)。在這種意義上，EM 類似於一個基於梯度的爬山法演算法，但是要注意它沒有「步長」這個參數！

實際情況並不總是像圖 20.12(a)中那樣好的。有時候，會發生例如一個高斯元素縮小的情況，以至於只涵蓋了一個單個的資料點。那麼它的變異數會等於 0，而概似機率成爲無窮大！另一個問題是，兩個元素可能發生「合併」，得到相同的平均值和變異數，並共用它們的資料點。這類退化的局部極大的情況是很嚴重的問題，尤其在高維度的時候。一種解決方案是在模型參數上設置事前，並且應用 MAP 版本的 EM 演算法。另一種方案是，如果一種元素太小或者太接近於另一種元素，就使用新的隨機參數重新開始一種元素。合理的初始值也會很有幫助。

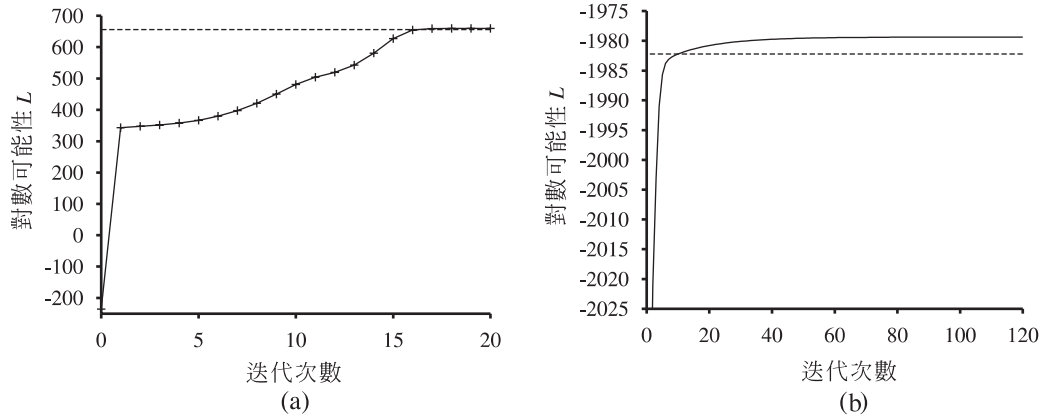

圖 20.12　圖中顯示了資料的對數概似 L，並表示為 EM 疊代次數的一個函數。水平線顯示了根據真實模型的對數概似。

(a) **圖 20.11 中的高斯混合模型**

(b) **圖 20.13(a)中的貝氏網路**

20.3.2 學習含有隱變數的貝氏網路

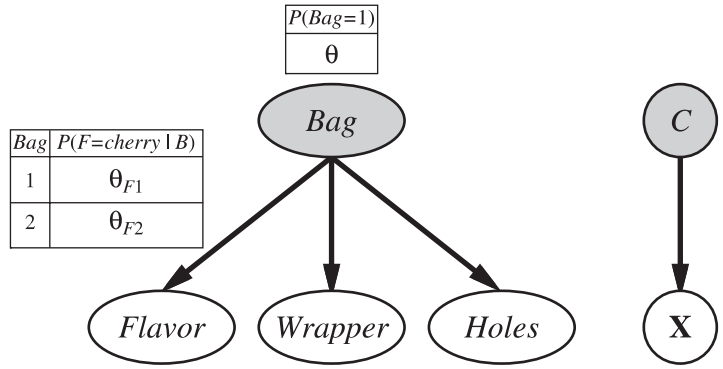

圖 20.13 (a) **糖的混合模型。不同口味、糖紙、及洞等等的比例取決於糖果袋子，而無法觀察到。**
(b) **高斯混合模型的貝氏網路。可觀察變數 X 的平均值和共變異數取決於元素** *C*

要學習含有隱變數的貝氏網路，我們運用對於混合高斯分佈有效的同樣見解。圖 20.13 表示了一種情形，有兩包糖果，混合在一起。糖果可以用 3 種特徵描述：除了 *Flavor*(口味)和 *Wrapper*(糖紙)外，有些糖在中間還有 *Hole*(洞)，而有些沒有。每包中糖果的分佈透過一個**原始貝氏**模型進行描述：對於給定的袋子，各個特徵彼此獨立，但是每個特徵的條件機率分佈依賴於袋子。參數如下：θ 是糖來自袋子 1 的事前機率；θ_{F1} 和 θ_{F2} 是糖的口味爲櫻桃味的機率，且分別已知糖果來自袋子 1 或袋子 2；θ_{W1} 和 θ_{W2} 是糖紙爲紅色的機率；而 θ_{H1} 和 θ_{H2} 是糖有洞的機率。注意，整個模型是一個混合模型。(事實上，我們也可以把混合高斯模型建立爲一個貝氏網路模型，如圖 20.13(b)所示。)在圖中，袋子是一個隱變數，因爲一旦糖果被混合到一起，我們就再無法知道每顆糖來自哪個袋子了。在這樣的例子中，我們能夠透過觀察混合在一起的糖果恢復出關於兩個袋子的描述嗎？我們執行一次這個問題的 EM 疊代。首先，讓我們看看資料。從眞實參數值如下的模型中產生 1000 個實例：

$$\Theta = 0.5, \theta_{F1} = \theta_{W1} = \theta_{H1} = 0.8, \theta_{F2} = \theta_{W2} = \theta_{H2} = 0.3 \tag{20.7}$$

也就是說，糖從任何一個袋子中以同樣的機率規律取出；第一袋大多數是櫻桃味、紅色糖紙、有洞的，第二袋大多數是酸橙味、綠色糖紙、無洞的。8 種可能種類的糖的計數分別如下：

	$W=$ 紅色		$W=$ 綠色	
	$H=1$	$H=0$	$H=1$	$H=0$
$F=$ 櫻桃味	273	93	104	90
$F=$ 酸橙味	79	100	94	167

我們從對參數進行初始化開始。爲了讓數值較簡單，我們將如下選取

$$\theta^{(0)} = 0.6, \quad \theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{H1}^{(0)} = 0.6, \quad \theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0.4 \tag{20.8}$$

首先，我們從參數 θ 入手。在完全可觀察的情況下，我們可以根據已觀察到的來自袋子 1 和袋子 2 中的糖果計數進行直接估計。因爲袋子是隱變數，所以我們替代地計算期望的計數值。期望計數 $\hat{N}(Bag = 1)$是對所有糖而言來自袋子 1 的機率之和：

$$\theta^{(1)} = \hat{N}(Bag = 1) / N = \sum_{j=1}^{N} P(Bag = 1 \mid flavor_j, wrapper_j, holes_j) / N$$

這些機率可以透過貝氏網路的任何推理演算法計算出來。對於諸如我們的例子中那樣的原始貝氏模型來說，我們可以利用貝氏法則和條件獨立性「手工」地進行推導：

$$\theta^{(1)} = \frac{1}{N} \sum_{j=1}^{N} \frac{P(flavor_j \mid Bag = 1) P(wrapper_j \mid Bag = 1) P(holes_j \mid Bag = 1) P(Bag = 1)}{\sum_i P(flavor_j \mid Bag = i) P(wrapper_j \mid Bag = i) P(holes_j \mid Bag = i) P(Bag = i)}$$

把這個公式應用於例如說 273 顆紅色糖紙包裝的有洞的櫻桃味糖，我們可以得到一個貢獻值：

$$\frac{273}{1000} \cdot \frac{\theta_{F1}^{(0)} \theta_{W1}^{(0)} \theta_{H1}^{(0)} \theta^{(0)}}{\theta_{F1}^{(0)} \theta_{W1}^{(0)} \theta_{H1}^{(0)} \theta^{(0)} + \theta_{F2}^{(0)} \theta_{W2}^{(0)} \theta_{H2}^{(0)} (1 - \theta^{(0)})} \approx 0.22797$$

繼續處理計數表中的其他 7 種糖，我們可以得到 $\theta^{(1)} = 0.6124$。

現在讓我們考慮其他參數，例如 θ_{F1}。在完全可觀察的情況下，我們可以根據已觀察到的來自袋子 1 和袋子 2 中的糖果計數進行直接估計。櫻桃糖來自袋子 1 的期望計數爲：

$$\sum_{j:Flavor_j = cherry} P(Bag = 1 \mid Flavor_j = cherry, wrapper_j, holes_j)$$

再一次，這些機率可以用任何貝氏網路的演算法計算。完成這個過程，我們得到所有參數的新值：

$$\begin{aligned} &\theta^{(1)} = 0.6124, \quad \theta_{F1}^{(1)} = 0.6684, \quad \theta_{W1}^{(1)} = 0.6483, \quad \theta_{H1}^{(1)} = 0.6558, \\ &\theta_{F2}^{(1)} = 0.3887, \quad \theta_{W2}^{(1)} = 0.3817, \quad \theta_{H2}^{(1)} = 0.3827 \end{aligned} \tag{20.9}$$

在第一次疊代之後資料的對數概似從初始的−2044 左右增長到大約−2021，如圖 20.12(b)所示。也就是說，更新改進了概似度本身，大約有 $e^{23} \approx 10^{10}$ 倍。經過第 10 次疊代，學習到的模型是比原始模型更好的適配($L = -1982.214$)。接著，進展變得很慢。這在 EM 中並不罕見，而許多實際系統將 EM 與一個基於梯度的演算法諸如牛頓-拉夫森方法(參見第四章)相結合，完成學習的最後階段。

來自這個例子的一般經驗是，參數的更新對包含隱變數的貝氏網路學習來說，可以從每個取樣的推理結果中直接得到。而且，對於每一個參數而言需要的只是局部的事後機率。在這邊「局部」指的是對於每一個的變數 X_i，CPT 可以僅從 X_i 與其母變數 $\mathbf{U}_i$ 涉及的事後機率來學得。定義 θ_{ijk} 爲 CPT 參數 $P(X_i = x_{ij} \mid \mathbf{U}_i = \mathbf{u}_{ik})$，則參數更新可以由下面的正規化期望技術給出：

$$\theta_{ijk} \leftarrow \hat{N}(X_i = x_{ij}, \mathbf{U}_i = \mathbf{u}_{ik}) / \hat{N}(\mathbf{U}_i = \mathbf{u}_{ik})$$

期望計數可以透過對取樣進行求和得到，運用任何一個貝氏網路推理演算法計算出機率值 $P(X_i = x_{ij} \mid \mathbf{U}_i = \mathbf{u}_{ik})$。對於精確演算法——包括消元法——所有的這些機率值都可以作爲一個標準推理的副產品直接得到，不需要針對學習的額外計算。另外，對每一個參數，學習所需的資訊可以在局部獲得。

20.3.3 學習隱馬爾可夫模型

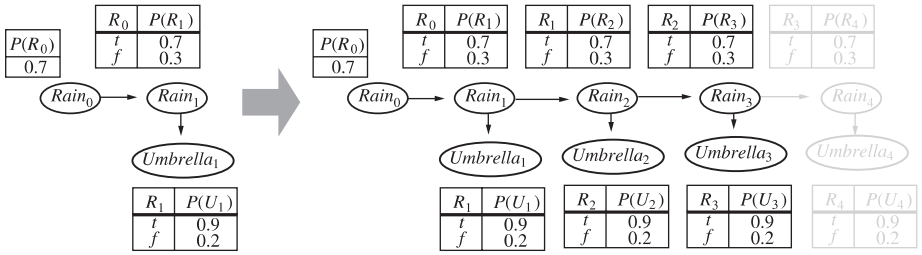

圖 20.14　表示一個隱馬爾可夫模型的展開的動態貝氏網路(圖 15.16 的重複)

我們的最後一個 EM 的應用涉及學習隱馬爾可夫模型(HMM)中的轉移機率。回憶一下在第十五章中，一個隱馬爾可夫模型可以表示爲一個動態的貝氏網路，它有一個離散的狀態變數，如圖 20.14 所示。每個資料點由一個有限長度的觀察序列組成，要解決的問題就是從一組觀察序列(也可能只是一個長序列)中學習轉移機率。

我們已經知道如何學習貝氏網路，但是有一個複雜因素：在貝氏網路中，每個參數是不同的；而另一方面，在隱馬爾可夫模型中，在時刻 t 從狀態 i 到狀態 j 的單獨的轉移機率 $\theta_{ijt} = P(X_{t+1} = j \mid X_t = i)$在時間中是重複的——也就是說，對於所有的 t，有 $\theta_{ijt} = \theta_{ijt}$。爲了估計從狀態 i 到狀態 j 的轉移機率，我們可以簡單地計算系統在狀態 i 時轉移到狀態 j 的次數的期望比例：

$$\theta_{ij} \leftarrow \sum_t \hat{N}(X_{t+1} = j, X_{t=i}) / \sum_t \hat{N}(X_{t=i})$$

再一次，期望計數可以透過任何 HMM 推理演算法計算得到。圖 15.4 中所示**前向–後向**演算法可以修改得很容易計算必要機率。重要的一點是，所需機率是透過**平滑**而不是**過濾**獲得的；也就是說，我們需要注意在估計一個特定轉移發生的機率時的後續證據。謀殺案中的證據通常是在犯罪發生之後(即從狀態 i 轉移到狀態 j)獲得的。

20.3.4 EM 演算法的一般形式

我們已經看到了 EM 演算法的幾個例子。每個例子都涉及到對每個實例計算隱變數的期望值，然後重新計算參數值，這裡把期望值當作觀察到的值來使用。令 $\mathbf{x}$ 爲在所有實例中的所有觀察值，$\mathbf{Z}$ 代表所有實例的所有隱變數，θ 是機率模型的所有參數。那麼 EM 演算法可以表示爲

$$\theta^{(i+1)} = \underset{\theta}{\operatorname{argmax}} \sum_{\mathbf{Z}} P(\mathbf{Z} = \mathbf{z} \mid \mathbf{x}, \theta^{(i)}) L(\mathbf{x}, \mathbf{Z} = \mathbf{z} \mid \theta)$$

這個公式是一個簡約表示的 EM 演算法。E 步驟是求和計算，也就是「完整」資料關於分佈 $P(\mathbf{Z} = \mathbf{z} \mid \mathbf{x}, \theta^{(i)})$的對數概似期望值，該分佈是在給定資料下隱變數的事後。M 步驟是關於參數的期望對數概似的最大化過程。對於混合高斯模型，隱變數是 Z_{ij}，其中如果實例 j 由元素 i 產生，則 Z_{ij} 是 1。對於貝氏網路，Z_{ij} 是實例 j 中未觀測變數 X_i。對於 HMM，Z_{jt} 是實例 j 在時間 t 時序列的狀態。一旦確認出合適的隱變數，從一般形式出發，爲特定的應用推導出一個 EM 演算法是可能的。

只要我們理解了 EM 的一般想法，就很容易推導出各種變形和證明。例如，在許多情況下，E 步驟——計算隱變數的事後——是不可操作的，如同大型貝氏網路一樣。研究顯示可以用一個近似的 E 步驟，仍然可以得到有效的學習演算法。透過一個取樣演算法，例如 MCMC(參見第 14.5 節)，學習過程是非常直觀的：MCMC 存取的每個狀態(隱變數和觀測到的變數的配置)可以當作一個完整的觀察進行精確的處理。這樣，參數可以在每次 MCMC 的轉移之後直接進行更新。另一個近似的推理形式，諸如變形的和迴圈的方法，也被證實對學習超大規模網路是有效的。

20.3.5 學習含有隱變數的貝氏網路結構

在第 20.2.5 節中，我們討論了用完全的資料學習貝氏網路結構的問題。當隱變數可能會影響被觀察的資料，事情變得更困難了。在最簡單的情況下，一位人工專家可能會告訴學習演算法說某些特定引變數是存在的，造成演算法會在網路架構中騰出空間給他們。例如，某個演算法也許試圖學習圖 20.10(a)中所示結構，已經知道 *HeartDisease*(一個三值變數)應該被包含在模型中這個資訊。和完全資料的情況一樣，整體的演算法有一個外部的迴圈在整個結構中尋找與內部迴圈去適配網路參數到結構中。

如果學習演算法未被告知哪個隱藏變數存在，則有兩個選擇：要麼假裝資料實際是完全的——強制演算法去學習圖 20.10(b)中的參數密集模型——要麼建立新的隱變數以便簡化模型。後一個方案可以透過把新的修正選擇包含在結構搜尋中實作：除了修正連接關係，該演算法能夠添加和刪除隱變數或者改變它的數量。當然，這個演算法並不知道它建立的新變數被稱爲 *HeartDisease*；它也不會給相對應的值起一個有意義的名字。幸運的是，新建立的隱變數通常與一個已有變數聯繫在一起，所以人類專家往往可以審查新變數涉及的局部條件分佈，從而推斷出它的含義來。

和完全資料的情況一樣，純粹的最大概似結構學習會產生一個全連接的網路(而且，是沒有隱變數的網路)，所以需要某種形式的複雜性懲罰手段。我們也可以應用 MCMC 對很多可能的網路結構做取樣，因而會近似於貝氏學習。例如，我們可以透過對元素的數目進行取樣來學習包含未知數目元素的混合高斯分佈；透過 MCMC 過程的取樣頻率提供關於高斯分佈的數目的近似事後分佈。

對於完全資料的情況，內部迴圈很快——只是一個從資料集裡抽取條件頻率的問題。當存在隱變數時，內部迴圈可能涉及大量 EM 的疊代或者一個基於梯度的演算法，而每次疊代又涉及到計算貝氏網路中的事後機率分佈，這本身就是一個 NP 難題。現在，已經證實這種方法對於學習複雜模型是不實用的。一個可能的改進是所謂**結構化 EM** 演算法，它運轉的方式和普通的(參數)EM 演算法大體相同，除了它在更新參數的時候也同時更新結構。正如普通 EM 演算法在 E 步驟中使用當前的參數計算期望的計數、然後在 M 步驟中把這些計數用於選取新的參數一樣，結構化 EM 使用當前的

結構計算期望的計數，然後將這些計數應用於 M 步驟，對潛在的新結構的概似率進行評價。(這與外部迴圈/內部迴圈方法不同，後者計算的是對每種潛在結構的新的期望計數。)這樣，結構化 EM 演算法可以產生多個網路的結構選擇，而不需要重新計算期望的數值，同時，它還有能力學習非平凡的貝氏網路結構。雖然如此，在我們可以說完全解決結構學習問題之前，還有許多工作要做。

20.4 總結

統計學習方法的範圍包括從簡單的平均值計算到構造諸如貝氏網路以及類神經網路這樣的複雜模型的方法。它們有許多應用，遍及電腦科學領域、工程領域、神經生物學領域、心理學領域以及物理學領域等。本章提出了一些基本想法，並給出了一些數學基礎的分析。要點如下：

- 貝氏學習方法把學習形式化地表示爲機率推理的一種形式，利用觀察結果更新在假設上的事前分佈。這種方法爲實現奧卡姆剃刀提供了一種很好的方式，但是它對於複雜的假設空間很快會變成不可操作的。
- 最大事後(MAP)學習方法選擇給定資料上的單一最可能假設。它仍然使用假設事前，而此方法往往比完全貝氏學習更可操作一些。
- 最大概似學習方法簡單地選擇使得資料的概似度最大化的假設，它等價於使用均勻事前的 MAP 學習。在諸如線性回歸以及完全可觀察的貝氏網路這樣的簡單情況下，可以很容易地找到近似形式的最大概似解。**原始貝氏**學習是一種非常有效的方法，它具有很好的擴展能力。
- 當一些變數是隱變數時，局部最大概似解可以透過 EM 演算法找到。這樣的應用包括使用混合高斯模型的群集、對貝氏網路進行學習，以及對隱馬爾可夫模型進行學習等。
- 學習貝氏網路的結構是**模型選擇**的一個特例。它通常涉及到結構空間的一個離散搜尋過程。需要有某種方法在模型複雜度與適配度之間取得一個折衷。
- **無參數模型**使用資料點集合來表示一個分佈。因此，參數的數目隨著訓練集而增長。最近鄰方法關注於距離問題點最近的實例，而**核**方法構造了一個所有實例的加權距離組合。

統計學習方法一直是個很活躍的研究領域。在理論和實踐方面都取得了許多巨大的進步，已經達到這樣的程度：對於準確或近似推理可行的幾乎任何模型，進行學習都是可能的。

參考文獻與歷史的註釋 BIBLIOGRAPHICAL AND HISTORICAL NOTES

人工智慧中統計學習技術的應用在早年間(參見 Duda 和 Hart，1973)是一個活躍的研究領域，但是當主流 AI 領域開始專注於符號方式時，它漸漸從主流人工智慧中獨立出來。20 世紀 80 年代後期，緊隨著貝氏網路模型的引入，對於它的興趣再次興起；大約在同一時期，類神經網路的統計學觀點開始湧現出來。在 20 世紀 90 年代後期，研究興趣又值得注意地集中到機器學習、統計學和類神經網路上，中心是根據資料建立大規模機率模型的方法。

原始貝氏模型是一種最古老且最簡單形式的貝氏網路模型，它的歷史可以追溯到 20 世紀 50 年代。它的起源在第十三章有所提及。Domingos 和 Pazzani(1997)說明了其成功的部分原委。一種改進型的原始貝氏學習贏得了首屆 KDD 杯資料挖掘競賽(Elkan，1997)。Heckerman(1998)提供了一個對貝氏網路學習一般問題的非常出色的介紹。Spiegelhalter 等人(1993)討論了使用 Dirichlet 事前的貝氏網路的貝氏參數學習。BUGS 套裝軟體(Gilks 等人，1994)綜合了許多上述想法，並提供了一個非常強大的對複雜模型進行形式化和學習的工具。第一個用於學習貝氏網路結構的演算法使用了條件獨立性測試的方法(Pearl，1988；Pearl 和 Verma，1991)。Spirtes 等人(1993)開發出一個全面方法，並實作在貝氏網路學習的 TETRAD 此套裝軟體中。從那時起的演算法改進使得貝氏網路學習方法在 2001 年的 KDD 杯資料挖掘競賽中取得了絕對的勝利(Cheng 等人，2002)(這裡的特定任務是一個有 139 351 個特徵的生物資訊學問題！)。Cooper 和 Herskovits(1992)發展出一種基於最大概似的結構學習方法，Heckerman 等人(1994)又改進了它。自從那時的一些演算法的進展，使得完全資料的情況獲得相當可觀的表現(Moore 及 Wong，2003；Teyssier 及 Koller，2005)。其中一個很重要的要素是有效的資料結構—AD-樹，對於變數與值的所有可能的組合作快取計算(Moore 及 Lee，1997)。Friedman 和 Goldszmidt(1996)指出局部條件分佈表示對學習到的結構的影響。

包含隱變數和缺失資料的機率模型學習的一般問題由 Hartley 提出(1958)，其描述的是在後來被稱爲 EM 的一般想法，並且給了幾個例子。進一步將之推廣乃來自 HMM 學習的 Baum-Welch 演算法(Baum 及 Petrie，1966)，其代表的是一個特殊形的 EM。由 Dempster，Laird 與 Rubin(1977)提出的論文，發表了 EM 演算法的一般形式並且分析其收斂性，是電腦科學與統計學兩個領域中最多人引用的論文之一。(Dempster 本人將 EM 視爲一種模式而不是一個演算法，因爲在應用於新的分佈族之前，還需要完成大量的數學工作)。McLachlan 和 Krishnan(1997)爲該演算法及其特性寫了一本完整的書。對於學習混合模型的特定問題，包括混合高斯模型，由 Titterington 等人(1985)加以論述。在人工智慧領域內，第一個針對混合建模問題使用 EM 的成功系統是 AUTOCLASS(Cheeseman 等人，1988；Cheeseman 和 Stutz，1996)。AUTOCLASS 以及被用於許多現實世界的科學分類任務，包括根據光譜資料發現新型恒星(Goebel 等人，1989)以及在 DNA/蛋白序列庫中發現新型蛋白和基因(Hunter 和 States，1992)。

對於含有隱變數的貝氏網路中的最大概似參數學習，EM 和基於梯度的方法在約同時期由 Lauritzen(1995)，Russell 等人(1995)，和 Binder 等人(1997a)發表出來。結構化 EM 演算法是由 Friedman(1998)發展出來並且將之應用到含潛變數的貝氏網路結構的最大概似學習。Friedman 與 Koller(2003)介紹了貝式結構學習

學習貝氏網路結構的能力與從資料中恢復因果資訊的問題關係非常密切。也就是說，是否能夠以這樣一種方式學習貝氏網路：恢復出的網路結構指示出實際的因果影響？多年來，統計學家在迴避這個問題，他們相信觀察資料(而不是那些從實驗中產生的資料)只可能產生相關資訊——畢竟，表現出相關性的任何兩個變數可能事實上受到未知的第三個因果因素的影響，而不是彼此直接相互影響。Pearl(2000)提出了令人信服的相反論據，顯示事實上在許多情況下因果關係是可以得到確認的，並且發展出**因果網路**形式化方法來表達這種因果影響以及普通條件機率。

Rosenblatt(1956)和 Parzen(1962)最早研究了核密度估計方法，也稱爲 **Parzen 視窗**密度估計。從那時起，大量的文獻對各種不同的估計運算元進行了研究。Devroye(1987)對此提供了一個詳盡的介紹。同時在無參數的貝氏方法的文獻上也有快速的成長，Ferguso(1973)在 **Dirichlet 程序**有原創性且對後來影響深遠的研究，可以被認爲是做爲 Dirichlet 分布上的一個分支。這些方法對於未知數目元素的混合格外的有幫助。Ghahramani(2005)與 Jordan(2005)提供有用的教學在統計學習方法的很多應用上。Rasmussen 與 Williams(2006)的論文提到了**高斯程序**，給出一種定義在連續函數空間上事先分佈的方法。

本章中的材料彙集了來自統計學、模式識別以及類神經網路等領域的研究工作，所以同一個故事被以多種方式講述了很多遍。貝氏統計的較好的課本有 DeGroot(1970)的、Berger(1985)的、以及 Gelman 等人(1995)的。Bishop(2007)及 Hastie 等人(2009)提供了一個關於統計機器學習的極佳介紹。對於模式分類，使用了很多年的經典課本是 Duda 和 Hart(1973)編寫的，現在有了更新版本(Duda 等人，2001)。一年一度的「NIPS 會議」(類神經資訊處理會議)，其會議論文集被作爲《類神經資訊處理系統進展》(*Advances in Neural Information Processing Conference*)系列出版，現在都以貝氏相關論文爲最大宗。貝氏網路學習方面的論文也出現在「人工智慧與機器學習中的不確定性」會議(Uncertainty in AI and Machine Learning conference)上以及一些統計學的會議上。類神經網路的專門期刊包括《神經計算》(*Neural Computation*)、《類神經網路》(*Neural Networks*)，以及《IEEE 類神經網路學報》(*IEEE Transactions on Neural Networks*)。特定的貝氏集會包括了貝式統計與貝氏分析期刊在 Valencia 國際會議(Valencia International Meetings)中。

❖ 習題 EXERCISES

20.1 圖 20.1 中使用的資料可以視爲由 h_5 產生的。對其他 4 個假設的每一個，分別產生一個長度爲 100 的資料集，並爲 $P(h_i \mid d_1, \ldots, d_m)$和 $P(D_{m+1} = lime \mid d_1, \ldots, d_m)$畫出相對應的圖。評論你的結果。

20.2 設 Ann 對櫻桃和酸橙糖果的效用值分別是 c_A 和 l_A，而 Bob 的則是 c_B 和 l_B。(但是只要 Ann 打開了一顆糖果的包裝，Bob 就不會買這顆了)。可以想到，如果 Bob 比 Ann 更喜歡酸橙糖，那麼對於 Ann 來說聰明的做法是只要她足夠確信一包糖果是酸橙味的，就把它賣給 Bob。另一方面，如果 Ann 在這個過程中打開太多的糖果，這包糖的價值就損失了。討論如何確定賣這包糖果的最優點的問題。給定第 20.1 節中的事前分佈，確定最佳過程的期望效用。

20.3 兩個統計學家去看病，得到了相同的診斷：40%的機率是一種致命疾病 A，60%的機率是可能致命疾病 B。幸運的是，抗 A 和抗 B 的藥都不貴，100%有效，並且沒有副作用。統計學家可以選擇吃其中一種藥、兩種都吃或者兩種藥都不吃。第一個統計學家(狂熱的貝氏論者)會怎麼做？總是用最大概似假設的第二個統計學家呢？

醫生研究發現疾病 B 實際上有兩種類型：右旋 B 和左旋 B，有同樣的可能性並且都可以同樣地用抗 B 藥治療。現在有了 3 個假設，兩個統計學家會怎麼做呢？

20.4 解釋第 18 章的 boosting 方法如何用於原始貝氏學習。用餐館學習問題測試得到的演算法性能。

20.5 考慮有 m 個資料點(x_j, y_j)，其中全部 y_j 是根據公式(20.5)中的線性高斯模型從 x_j 集合產生的。找出當資料達到最大條件對數概似時 θ_1，θ_2 和 σ 的值。

20.6 考慮第 14.3 節描述的發燒的「雜訊或」模型。解釋如何應用最大概似學習，找到能適配一個完全資料集的模型參數。(**提示**：使用偏微分的鏈式法則)。

20.7 本習題考察公式(20.6)中定義的 β 分佈的特性。

a. 透過在區間[0, 1]上進行積分，證明 $\beta\,[a, b]$分佈的正規化常數可由 $\alpha = \Gamma(a + b)/\Gamma(a)\Gamma(b)$得到，其中 $\Gamma(x)$是 γ 函數(**Gamma** 函數)，其定義爲 $\Gamma(x + 1) = x\Gamma(x)$及 $\Gamma(1) = 1$。(對於整數 x，$\Gamma(x + 1) = x!$)。

b. 證明平均值爲 $a / (a + b)$。

c. 找到 mode(s)(θ 的最可能值)。

d. 描述當 ε 取很小值時的分佈 $\beta\,[\varepsilon, \varepsilon]$。當更新這樣的分佈時會發生什麼情況？

20.8 考慮一個任意的貝氏網路、該網路的完全資料集，以及資料集與網路相關的概似率。給出一個簡單的證明，當我們在網路中增加一個新連接時，資料的概似率不會減小，並重新計算參數值的最大概似。

20.9 考慮單一布林隨機變數 Y (指「分類」)。令事前機率 $P(Y = true)$爲 π。讓我們試著找出 π，且給定訓練集 $D = (y_1,\ldots,y_N)$，其中 N 爲 Y 的獨立取樣數。再者，假設在 N 中 p 爲正且 p 爲負。

a. 寫出由 π，p 與 n 這些項所組成 D 概似度的運算式(也就是說這個實例的特定序列的機率，給出 π 的固定值)。

b. 對概似度 $\log L$ 做微分，找出 π 的值使得資料的概似度最大化。

c. 現在再假設我們增加 k 個布林隨機變數 $X_1, X_2, \ldots, X_k$(指「屬性」)來描述每個取樣，且假設我們設屬性爲各自條件獨立得到目標 Y。畫出對應此假設的貝示網路。

d. 寫出包含屬性資料的概似度，利用下列額外的符號：

- α_i 爲 $P(X_i = true \mid Y = true)$。
- β_i 爲 $P(X_i = true \mid Y = false)$。
- p_i^+爲 $X_i = true$ 與 $Y = true$ 的取樣數。
- n_i^+爲 $X_i = false$ 與 $Y = true$ 的取樣數。
- p_i^-爲 $X_i = true$ 與 $Y = false$ 的取樣數。
- n_i^-爲 $X_i = false$ 與 $Y = false$ 的取樣數。

(**提示**：先考慮看到具有 $X_1, X_2, \ldots, X_k$ 和 Y 之特定值的單一實例的機率。

e. 對概似度 $\log L$ 做微分，找出 α_i 與 β_i 的值(依據不同計數)使得概似度最大化並且敘述這些值代表的意義。

f. 令 $k = 2$，並考慮一個含所有 4 個可能的 XOR 函數實例的資料集合。計算出 $\pi, \alpha_1, \alpha_2, \beta_1$ 和 β_2 的最大的概似度估計值。

g. 利用得到的 $\pi, \alpha_1, \alpha_2, \beta_1$ 和 β_2 估計值，每個例子的事後機率 $P(Y = true \mid x_1, x_2)$各是多少？

20.10 給定公式(20.7)中的眞實參數，考慮用 EM 學習圖 20.13(a)中的網路參數。

a. 解釋爲什麼如果模型中只有 2 個而不是 3 個屬性時，EM 演算法就無法工作？

b. 說明從公式(20.8)開始 EM 進行第一次疊代的計算。

c. 如果我們開始的時候把所有參數都設爲相同的值 p，將會發生什麼？(**提示**：在得到一般性結果之前進行一些實驗調查可能會很有幫助)。

d. 以參數寫出 20.3.2 的糖果表格資料的 log 概似表示式，並計算對每一參數的偏導數，以及探究在(c)中到達的不動點其性質。

本 章 註 腳

[1] 熟悉統計學的讀者會意識到，這種模式是**盒子與球**(urn-and-ball)問題的一個變形。我們發現盒子與球不如糖果吸引人；而且，糖果可以引申到其他任務，例如決定是否和朋友交易糖果包的問題——參見習題 20.2。

[2] 前面我們說過糖果包要很大；否則，獨立同分佈的假定會不成立。技術上說，把每塊糖果在檢視之後再重新包好放回包中才更準確(但會不太衛生)。

[3] 之所以稱為超參數是因為它們對 θ 的分佈進行參數化，而 θ 本身就是一個參數。

[4] 其他共軛事前包括針對離散多值分佈參數的 **Dirichlet** 族和針對高斯分佈參數的**正規-Wishart** 族。參考 Bernardo 及 Smith(1994)。

[5] 在實際情況下最好隨機進行選取，以避免由於對稱而產生的局部極大。