Mining the Social Web

社群網站的資料探勘

Matthew A. Russell 著
師蓉・胡為君 譯

O'REILLY®

社群網站的資料探勘

Mining the Social Web

Matthew A. Russell 著

師蓉、胡為君 譯

O'REILLY®

社群網站的資料探勘

譯　　者：師蓉 / 胡為君
企劃編輯：莊吳行世
文字編輯：江雅鈴
設計裝幀：陶相騰
發 行 人：廖文良

發 行 所：碁峰資訊股份有限公司
地　　址：台北市南港區三重路 66 號 7 樓之 6
電　　話：(02)2788-2408
傳　　真：(02)2788-1031
網　　站：www.gotop.com.tw
書　　號：A342
版　　次：2013 年 5 月初版
建議售價：NT$580

國家圖書館出版品預行編目資料

社群網站的資料探勘 / 師蓉, 胡為君譯. -- 初版. -- 臺北市：碁峰資訊, 2013.05
面；　公分
譯自：Mining the Social Web
ISBN 978-986-276-782-5 (平裝)
1.資料探勘　2.網路社群
312.74

前言

與其說網路是一項技術創新，不如說是一項社會創新。我設計的是一種社會效應（幫助大家一起工作），而不是一種高科技玩具。網路的終極目標是支援並改進現實世界中存在的各種各樣的「網」，我們有家庭「網」、協會「網」和公司「網」，我們會親遠疏近。

—Tim Berners-Lee，《*Weaving the* 網頁》(Harper)

是否要閱讀本書

如果你有程式基礎，並且有興趣經由社群網路進行資料探勘與分析，以瞭解周圍事物的機會，你就選對書了。在幾頁簡短介紹之後，我們就會開始動手編寫程式。然而說句實話，你對本書最主要的抱怨，可能就是所有的章節都太短了。遺憾的是，當你試圖抓住每天變化並且有大量機會的空間時，情況總是這樣。也就是說，我非常贊同「80-20 法則」（*http://en.wikipedia.org/wiki/Pareto_principle*），而且我堅信本書在「用 80% 的時間來探討最有趣的 20 % 的內容」這一點，是個很合理的嘗試。

雖然本書篇幅有限，但涵蓋內容包羅萬象。簡而言之，本書訴求廣博而非精深，雖然現在是適合就此主題進行深入探討的時機，但是本書對這些有趣的探勘和分析技術比較沒那麼深入。本書的編排風格，既適合你通讀全書來全面瞭解社群網路資訊方面的入門知識，也適合你根據自己的喜好來挑選感興趣的章節來閱讀。也就是說，每一章的設計都比較簡短且獨立，但是，我在每章內容的編排上還是精心安排了前後順序，以求你在閱讀全書時感到順暢。

在過去的幾年中，Facebook、Twitter 和 LinkedIn 這類社群網站，已經從時尚變為主流，甚至成為全球化現象。在 2010 年第一季，廣受歡迎的社群網站 Facebook 已經超過 Google，成為大家最經常造訪的網站[1]，這也證實了大家網路消費方式的明確轉變。依此事實斷言「網路現在已經成為社會文化現象，而不再是研究和資訊的工具」可能為時尚早；然而，這一資訊的確可以表明，社群網站正在以搜尋引擎所不具備的方式，大規模地滿足了人類的一些基本訴求。社群網路確實正在改變我們的網路生活[2]，它們能夠讓技術給我們呈現出最好的（有時是最壞的）一面。社群網路的爆發只是現實世界和網路世界之間的差距不斷縮小的一種方式。總而言之，本書的每一章都將社群網站與資料探勘、分析和視覺化技術的內容組織在一起，來回答以下問題：

- 誰與誰相識，他們共同的朋友是誰？
- 某人與其他人多久聯繫一次？
- 人與人之間的交流在多少程度上是相對的？
- 網路中最沉默 / 最健談的人是誰？
- 網路中最具影響力／人氣最高的人是誰？
- 大家正在聊什麼（而且它有趣嗎）？

要回答這些問題，通常都會涉及兩人或更多的人，並且需要找出他們之間存在關係的上下文。回答這些問題所涉及的工作只是開始，更複雜的分析過程還在後面，但是你必須找個地方下手，因為社群網路 API 和開放原始碼工具套件都很容易掌握。

籠統來說，本書把**社群網站**[3]看成是由人、活動、事件、概念等組成的一幅「圖」。Google 和 Facebook 這些業界領袖，已經開始逐漸推廣以「圖」為中心（graph-centric）的理念，而不再強調以「網路」為中心的說法了，因為它們在同時推廣以「圖」為基礎的 API。事實上，Tim Berners-Lee 建議，也許他應該使用巨大全球圖（Giant Global Graph，GGG）（*http://dig.csail.mit.edu/breadcrumbs/node/215*）來代替全球資訊網（World Wide 網頁，WWW），因為「網」和「圖」可以在定義網際網路的拓樸結構的情境中任意互換。雖然 Tim Berners-Lee 最初的設想能否達成仍有待觀察，但是我們所熟知的網路正在因社群資訊而變得越來越豐富。我們回顧過去的幾年時，最明顯的變化就是：由一個固有的語義網建立的第二級和第三級影響，是達成真正的語義網的必要條件。兩者之間的差距也變得越來越小了。

1 見第 9 章的第一段。

2 Facebook 的創始人馬克·祖伯格，被《時代》雜誌評為 2010 年度人物（*http://www.time.com/time/specials/packages/article/0,28804,2036683_2037183_2037185,00.html*）。

3 參閱 *http://journal.planetwork.net/article.php?lab=reed0704*，換一個角度看待關注數位認證的社群網路。

或許，用不到本書

從零開始構建自己的自然語言處理器、探究視覺化圖庫的典型用法，以及任何與構建相關的最新技術這類活動，都不屬於本書的範圍。如果你想瞭解任何以上內容而購買本書，你必然會失望。然而，切勿因為「在區區幾百頁中進行內文分析或記錄比對，既不實際，也無法表現我們的最佳技術」，就認為本書無法讓你找到疑難問題的合理解決方案，無法將這些方案應用於社群網站，在此過程中並無樂趣可言。當然，這也並不妨礙你在這些誘人的研究領域中培養積極的興趣。本書的篇幅有限，充其量只能培養你在資訊處理方面的興趣，並不能賦予你開天闢地的神奇力量。

也許這樣的提醒顯得多此一舉，但是我還是要強調一點（非常重要的一點）：本書所述內容，通常都假設你已經連接到了網際網路。本書並不適合在度假遠行時隨身攜帶，因為其中包含很多有超連結的參考內容，而且所有的程式碼範例都透過超連結直接與 GitHub 相連，GitHub（*http://github.com*）是一個非常社群化的 Git（*http://git-scm.com*）資料庫，裡頭的範例程式碼持續維護。這種做法希望社群編碼能加強志同道合的讀者之間的協作，例如，有的人想一起擴展範例，也有人想一起探索有趣的問題。但願大家能夠對資源進行分叉、擴展及改進 — 能結識一些志趣相投的新朋友更好。像 API 文件這種現成的資源，透過超連結也可以方便取得，況且我們認為你比較習慣使用線上內容，而不是遲早會失去時效的印刷文字。

搭配本書的程式碼勘誤官方 GitHub 資料庫是 *http://github.com/ptwobrussell/Mining-the-Social-web*。本書的官方 Twitter 帳戶是 *@SocialWebMining*。

如果你需要一本工具書，能讓你在 sharded MySQL 等分散式運算平台或者是 Hadoop、Redis 這樣的 NoSQL（*http://en.wikipedia.org/wiki/NoSQL*）技術上快速進步，我們並不推薦本書。我們確實使用了一些非常規的儲存技術，如 CouchDB（*http://couchdb.apache.org*）和 Redis（*http://code.google.com/p/redis*），但都是在單一主機上執行的，因為這樣便足以解決眼前的問題。然而，如果你興趣強烈且需要維持擴充性，它並不能真正連續不斷地將範例移植到分散式技術中。我強烈建議你首先要掌握好基礎知識，並且要保證程式碼能先在一個相對簡單的環境中執行，然後再將其移植到更複雜的分散式系統中，如此一來，萬一資訊並非來自於本機，你就可以依此調整運算方式，以保持其效能。如果你有意走這條路，Dumbo（*http://github.com/klbostee/dumbo/wiki*）很適合用於研究。請參考本書的 Twitter 帳戶（*@SocialWebMining*），取得有關 Dumbo 的擴充範例。

雖然我們會盡量遵守營運管理相關網站的條款與精神，但是你可能會對從社群網站所獲得的資訊進行加工，本書對於這些做法的法律後果並不提供任何意見。有些無奈的是，許多主流社群網站的授權條款，禁止在它們提供的平台之外使用這些資訊，但目前來說，這種做法是意料中事。大部分社群網站就像是帶圍牆的花園，但是從他們的立場（以及他們的投資者的立場）來說，這些公司的諸多價值目前依賴於控制平台和保護用戶的隱私；這種平衡很難維持，而且短期內不可能有明顯改變。

最後的小提示：本書略微傾向於 *nix 環境[4]，因為有些視覺化內容可能會給 Windows 用戶帶來些小麻煩。然而，一旦遇到這類問題，建議採用合理的替代方案或臨時性方案，例如啟動 VirtualBox（*http://www.virtualbox.org*）在 Linux 環境中執行範例。還好，並不經常發生這種情況，當出現這些問題時，你可以忽略相關的章節繼續閱讀，這並不會影響你閱讀的樂趣。

工具和前提

本書唯一的前提就是需要主動地學習一些 Python 知識，並且做好親自動手處理社群資訊的準備。本書的任何技術或範例都不需要太多資訊分析、高效能運算、分散式系統、機器學習，或者任何其他特別的背景知識。有些範例可能會涉及你以前沒有使用過的結構，如執行緒集區（*http://en.wikipedia.org/wiki/Thread_pool_pattern*），但不必擔心 — 我們使用 Python 程式。Python 直觀的語法、優秀的資訊處理生態系統軟體包和核心資料結構（實際上是 JSON）（*http://www.json.org*），使它成為一種優秀的教學工具，雖然功能強大卻很容易使用。在其他情況下，如處理自然語言時，我們會使用一些處理進階作業的套件，但是我們將會從應用程式設計者的角度來使用這些技術。由於在其他程式語言中也很可能存在非常相似的關聯，因此如果你願意的話，這應該是移植程式碼範例的必備練習方式。（但願這可以在 GitHub 中用得上！）除了上面的介紹之外，本書不會再多談使用 Python 的利弊，因為它是非常適合此作業的工具。如果你是程式新手或者從來沒有見過 Python 語法，那麼你只要保證沒有跳過前幾頁內容即可。如果你正在尋找可靠的介紹資料，網路上有很多優秀的參考文件，Python 的官方教學（*http://docs.python.org/tutorial*）就是很好的起點。

本書試圖從各種視覺化工具和工具套件中選擇性介紹一系列有用的視覺化工具，現有的電子表單類消費類產品之中，也有 Graphviz（*http://www.graphviz.org*）類的業界產品，還有 Protovis（*http://vis.stanford.edu/protovis*）這種尖端的 HTML5（*http://en.wikipedia.org/wiki/HTML5*）技術。每一章都會介紹一些新的視覺化技術，但我們會盡量順其自

4 *nix 是 Linux/Unix 環境的術語，在此處等同於非 Windows。

然，讓它順理成章。從這些工具建立輕量化的原型，你必然會滿意。換言之，本書的大部分視覺化內容只是現成的範例，或者只是對 API 做微幅修改，所以只要願意學，你就能做到。

本書編排慣例

以下是本書採用的排版慣例：

楷體字（*Italic*）

用來表示新名詞、URL、電子郵件位址、檔案名稱與副檔名。

定寬字（`Constant width`）

用來表示程式列表，也用在內文段落表示程式元素，例如變數或函式的名稱、資料庫、資料型別、環境變數、敘述，以及關鍵字。

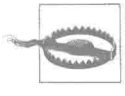
定寬粗體字（**`Constant width bold`**）

用來顯示命令，或其他應該由使用者逐字輸入的文字。

定寬斜體字（*`Constant width italic`*）

用來顯示應由使用者提供的值或由背景環境決定的值換掉的文字。

這個圖示代表提示、建議或一般的註釋。



這個圖示代表警告或注意事項。

使用範例程式

本書的大部分範例都可以從 GitHub（*https://github.com/ptwobrussell/Mining-thesocial-web*）下載。您可以從這裡取得持續更新的程式碼。

本書是為了幫助各位完成工作而存在。一般而言，各位可在自己的程式和文件中使用書裡的程式碼。你不需要聯絡我們以取得授權許可，除非把程式碼的重要部分拿來重製。舉例來說，設計一個程式，其中使用數段來自本書的程式碼，並不需要許可；但是販賣

或散布 O'Reilly 書中的範例，則需要許可。例如引用本書並引述範例碼來回答問題，並不需要許可；但是把本書中的大量程式碼納入你的產品文件，則需要許可。

我們很感激各位註明出處，但並非必要。註明出處時，通常包括書名、作者、出版商、ISBN。例如：「*Mining the Social Web* by Matthew A. Russell. Copyright 2011 Matthew Russell, 978-1-449-38834-8.」。

致謝

撰寫一本技術書籍需要做出很多犧牲。首先，在家裡，我放棄了與妻子 Baseeret 和女兒 Lindsay Belle 相處的時間，這比我樂於承認的時間還要多。雖然我的抱負是有朝一日能在某種程度上征服世界（這只是暫時的，坦白說，我正在盡力擺脫這種狀態），但是我還是要對你們的愛表示感激。

我深信你所做的一切決定，最終都會影響到你的一生（尤其是職業生涯），但是誰都不能孤獨前行，我們要懂得感恩。撰寫本書時，我真的很慶幸能與世界上最聰明的一幫人合作，其中包括像 Mike Loukides 這樣聰明的技術編輯，以及 O'Reilly 這樣極富天賦的製作團隊，還有幫助我完成本書的很多熱心的校稿人。我要特別感謝 Abe Music、Pete Warden、Tantek Celik、J.Chris Anderson、Salvatore Sanfilippo、Robert Newson、DJ Patil、Chimezie Ogbuji、Tim Golden、Brian Curtin、Raffi Krikorian、Jeff Hammerbacher、Nick Ducoff 和 Cameron Marlowe 對本書所用素材的審查或者對本書提出的建設性建議，這些都反映在本書的品質上。在此我也要感謝 Tim O'Reilly 的熱心，他允許我將他的一些 Twitter 和 Google Buzz 資訊在第 4、5、7 章中進行了仔細的研究；這些內容必然使這幾章增色許多。我不可能一一介紹曾經直接或間接地幫助過我或者幫助過本書出版的人，在此一併表示感謝。

最後，要感謝你閱讀這本書，你也許會買一本留作紀念。雖然我已盡了我最大的努力，但你還是會發現本書存在的一些疏漏之處；然而，我堅信，雖然疏漏在所難免，但本書必可讓你覺得值得你花上幾夜甚至幾週的時間來細細研讀，而且你也必將有所收穫。

目錄

第一章

緒論：Twitter 資料的處理

我們本應從具體的社群網站 API（Application Programming Interface，應用程式介面）、無模式設計（schemaless design）或者其他主題展開討論，但還是讓我們先透過範例來看看，社群網站的資訊有多容易收集和分析。本章相當於「駕駛指南」，旨在激發你開始對後繼章節所談論的話題展開深入思考，隨後的各章將會逐一對這些話題進行詳細的討論。我們首先會準備好開發環境，然後馬上進入對 Twitter 資訊的收集和分析。

Python 開發工具的安裝

本書的範例程式碼是用 Python 語言編寫，因此，如果你已經在系統中安裝了最新版本的 Python 和 `easy_install`，顯然你已經對 Python 有所瞭解了，你就可以跳過這一部分內容。如果你還沒有安裝 Python，說明目前你還是個 Python 菜鳥。但不必擔心，很快你就會成為一名 Python 高手了，因為 Python 很容易自學。不論是什麼操作系統平台，用戶都可以從 *http://www.python.org/download/* 中找到相應的下載和安裝說明，但我強烈推薦 Windows 用戶安裝 ActivePython（*http://www.activestate.com/activepython*），它會在 Windows 命令提示符號（後文稱為「終端機」）中自動將 Python 添加到你的路徑中，而且它還內建了 `easy_install` 工具套件，我們稍後會討論。書中的範例都是用最新的 Python 2.7 編寫、測試，但在更新的版本上應該也能正確地執行。編寫本書時，Python 2 仍然是主流版本（*http://wiki.python.org/moin/Python2orPython3*），建議你也使用此版本；如果你認為 Python 3 已經完全移植了你需要的全部所需功能，而且你也樂意除錯版本升級引起的任何問題，那就另當別論。

安裝了 Python，就可以在終端機中輸入 `python` 來啟動一個直譯器了。請試試範例 1-1。

範例 *1-1* 第一個真正的 *Python* 直譯器會話

```
>>> print "Hello World"
Hello World
>>> #this is a comment
...
>>> for i in range(0,10): # a loop
...     print i, # the comma suppresses line breaks
...
0 1 2 3 4 5 6 7 8 9
>>> numbers = [ i for i in range(0,10) ] # a list comprehension
>>> print numbers
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> if 10 in numbers: # conditional logic
...     print True
... else:
...     print False
...
False
```

另一個你會用到的工具是 easy_install[1]，類似於 Linux 系統中的套件管理器；有了它，就可以輕鬆地安裝 Python 套件，而不必自己下載程式碼，然後再構建、安裝。你可以從 *http://pypi.python.org/pypi/setuptools* 中下載最新版的 easy_install，針對不同的操作系統平台，網站上都有詳細的安裝說明。一般來說，*nix 系統的用戶會用 sudo easy_install 命令將對應模組寫入 Python 的完整安裝目錄中。假設 Windows 用戶已經按照我的建議使用了 ActivePython，安裝程式已經內建了 easy_install。

Windows 用戶也可以參考一篇名為 "Installing easy_install...could be easier"（參見 *http://blog.sadphaeton.com/2009/01/20/python-development-windows-patr-2-installing-easyinstallcould-be-easier.html*）的文章。文中討論了當你執行 easy_install 時，可能會常遇到的一些有關 C 程式碼編譯的常見問題。

只要正確設定了 easy_install，就可以執行下列命令來安裝 NetworkX（本書通篇都會用它來構建和分析圖），並且會得到類似的輸出結果：

```
$ easy_install networkx
Searching for networkx
```

1 雖然本書範例使用的是大家所熟知的 easy_install，但在 Python 社群內，pip（*http://pip.openplans.org/*）工具逐漸普及，能用 easy_install 處理的套件，也都可以用 pip 處理。如果你已經安裝了 git 工具，就能直接用 pip 從 GitHub 安裝第 9 章「探討 Graph API」一節中需要用到的 PyPi（*http://pypi.python.org/pypi*）。

```
...truncated output...

Finished processing dependencies for networkx
```

NetworkX 安裝好後，你可能認為只要從直譯器中導入即可正確執行，但有時，某些套件會出點意外。例如，可能會發生以下情況：

```
>>> import networkx
Traceback (most recent call last):

... truncated output ...

ImportError: No module named numpy
```

只要出現 ImportError 錯誤就表示有一個套件缺失。在本例中，我們安裝的 networkx 模組有一個未做處理的附屬項 —numpy，它是一個高度優化的科學計算工具集。通常，再次呼叫 easy_install 就可以解決這個問題，處理 numpy 缺失的方法與安裝 NetworkX 的方法一樣。只要關閉直譯器，並且在終端機輸入 easy_install numpy 即可安裝該附屬項：

```
$ easy_install numpy
Searching for numpy

...truncated output...

Finished processing dependencies for numpy
```

既然 numpy 已經安裝好了，就應該可以打開一個新的直譯器，執行 import networkx 命令，並用它來構建圖表，如範例 1-2 所示。

範例 *1-2* 使用 *NetworkX* 建立一個由節點和邊構成的圖

```
>>> import networkx
>>> g=networkx.Graph()
>>> g.add_edge(1,2)
>>> g.add_node("spam")
>>> print g.nodes()
[1, 2, 'spam']
>>> print g.edges()
[(1, 2)]
```

現在，你已經安裝好了 Python 核心開發工具，並且已經準備好做一些更有趣的任務。如果你已經掌握了這一節的主要內容，那麼在深入學習之前最好查閱一下官方 Python 教學（*http://docs.python.org/tutorial*）。

Twitter 資訊的收集和處理

你不可能是連 Twitter 都沒聽過的火星人吧！ Twitter 是一個高度社群化的即時微部落格服務，它允許發布不超過 140 個字元的短文；這些短文稱為 tweet。與 Facebook、LinkedIn 這些關係對等的社群網路不同，Twitter 中的非對等關係有「好友」和「跟隨者」之分。假設你有一個 Twitter 帳戶，那麼你的好友就是你跟隨的用戶，而你的跟隨者則是跟隨你的那些用戶。雖然你可以選擇跟隨所有關注你的用戶，但通常你不會這樣做，因為你希望在自己的首頁 Timeline[2] 更新中顯示那些自己覺得有趣的 tweet。無論是從其龐大的用戶數量，還是其作為一種行銷手段，以及近期作為協力廠商訊息服務的傳輸層來說，Twitter 都是一個重要的現象。它提供了大量的 API，即使不註冊你也可以使用其中的大多數 API，但如果你建立自己的網路並對其進行探勘的話會更有趣。在你做大規模開發之前，先花點時間看一下 Twitter 自由的服務條款（*http://twitter.com/tos*）、API 文件（*http://apiwiki.twitter.com*）和 API 規則（*http:/ /twitter.com/apirules*），以確保你有效利用 Twitter 資訊來做你想做的任何事情。本書後續的章節假設你已經有一個可以用於探勘資訊的 Twitter 帳戶和足夠多的好友 / 跟隨者。

本書的官方 Twitter 帳戶是 *@SocialWebMining*。

處理 Twitter 的 API

如需和 Twitter 網路 API 相關的最精簡封裝，可以利用一個叫做 `twitter` 的軟體套件（*http://github.com/sixohsix/twitter*），按照 `easy_install` 的如下規範安裝：

```
$ easy_install twitter
Searching for twitter

...truncated output...

Finished processing dependencies for twitter
```

本套件還包含簡單易用的命令列實用工具和 IRC 機器人程式，因此在安裝完這個模組之後，就可以在 shell 中直接輸入 `twitter` 來瞭解如何使用這個命令列實用工具。然而，我們著重的是互動式 Python 直譯器的使用。我們會透過一些範例來進行瞭解，但要注意，你隨時可以在終端機執行 `pydoc` 來瀏覽這些文件。*nix 用戶可以簡單地輸入 `pydoc`

2 *http://support.twitter.com/entries/164083-what-is-a-timeline*