Machine Learning

Final Report , Due: 2017, January, 20th 9:00 pm on github

0. Information on our Team

Team Name: NTU_b03901036_新垣結衣我的菜

Team Member: b03901036 陳柏文, b03901024 楊承運, b03901037 鄭宇强,
b03901137 張致綱

Work Division:

ID	Work Description
b03901036	Model Training, Data Processing, Model Design, Final Report
b03901037	Model Training, Data Processing, Final Report
b03901137	Model Training, Data Processing, Model Design
b03901024	Model Training, Data Processing, Model Design

1. Brief Introduction- Outbrain Click Prediction

The internet is a stimulating treasure trove of possibility. Every day we stumble on news stories relevant to our communities or experience the serendipity of finding an article covering our next travel destination. Outbrain, the web's leading content discovery platform, delivers these moments while we surf our favorite sites.

Currently, Outbrain pairs relevant content with curious readers in about 250 billion personalized recommendations every month across many thousands of sites. In this competition, Kagglers are challenged to predict which pieces of content its global base of users are likely to click on. Improving Outbrain's recommendation algorithm will mean more users uncover stories that satisfy their individual tastes.

2. Preprocessing/ Feature Engineering

2.1 documents_categories.csv

檔案內容：document_id 及其對應的 category_id，confidence_level

檔案處理： 將對應的 category_id 轉換為 one hot vector(TF-IDF weighted on confidence_level)

2.2 documents_meta.csv

檔案內容：文章基本資訊，document_id, souce_id, publisher_id, publish_time

檔案處理：source_id, publisher_id, 轉換成 one-hot vector, publish time 轉換成已經出版的時間差，作為 feature

2.3 documents_entities.csv

檔案內容：document_id 及其對應的 entity_id, confidence_level

檔案處理：計算每個 entity_id 讀取的 document_id 數量作為 feature

2.4 promoted_content.csv

檔案內容：廣告基本資訊, ad_id, 出現的 docment_id, compaign_id(宣傳內容), advertised_id

檔案處理：compaign_id, advertised_id 轉換為 one-hot vector

2.5 documents_topics.csv

檔案內容：document_id and its topic_id, confidence level

檔案處理：將 document_id 對應的 topic_id 轉換成 one-hot vector (TFIDF weighed on confidence level))

2.6 page_views_sample.csv

檔案內容：uuid, document_id, timestamp, platform, geo_location, traffic_source

檔案處理：platform，traffic_source 轉換為 one-hot vector 作為 feature，利用 geo_location 轉換時區

2.7 events.csv

檔案內容：display_iduuid, document_id, timestamp, platform, geo_location, traffic_source

檔案處理：timestamp, geo_location(將時間轉換成該地的時間)，weekend or not(可能影響到點擊結果)

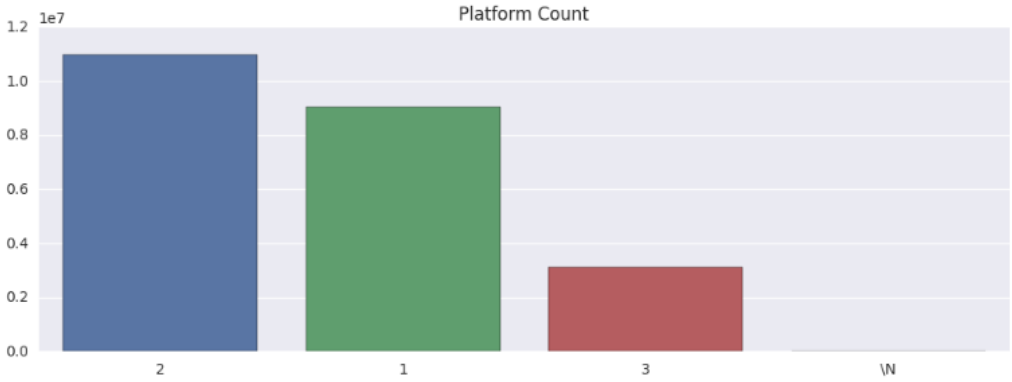
2.8 clicks_train.csv

檔案內容：display_id 及其對應的 ad_id, ad_id 被點及與否(1 for clicked)

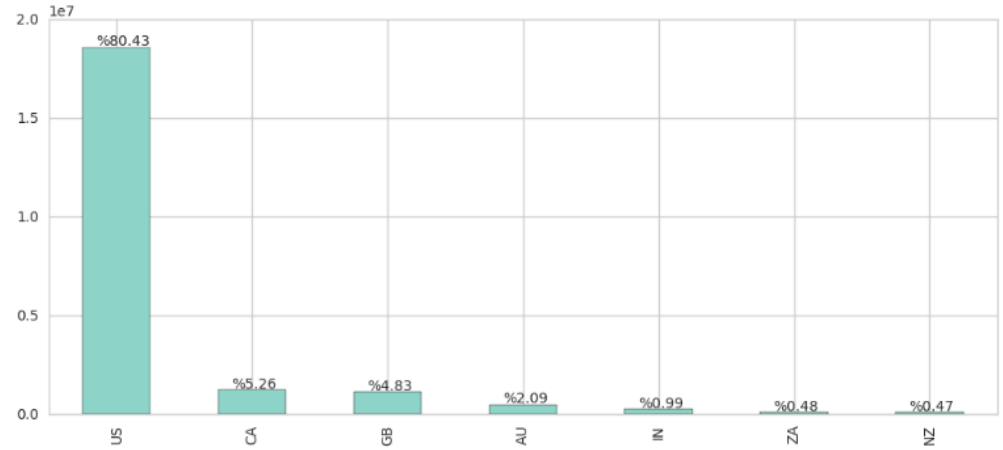
2.9 Analysis in Statistics

以下資料為一些簡單的數據統計(沒有作為 feature 使用)

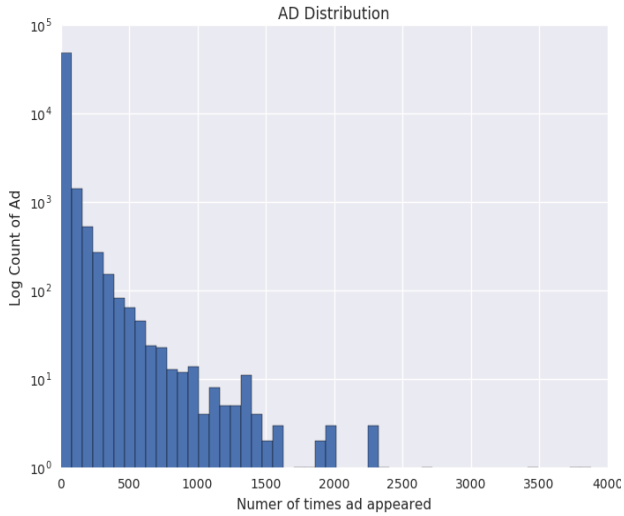
(1) 不同平台上的點擊率比較

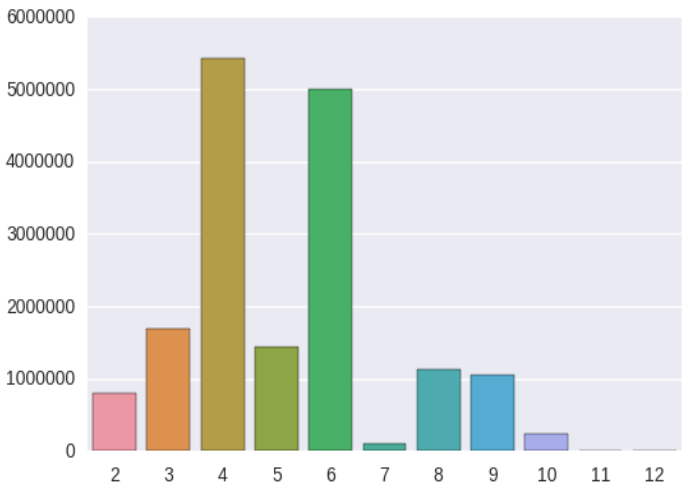


(2) 不同國家對點擊率的貢獻量



(3) 不同廣告出現的總共次數，12 個位置廣告的被點擊次數





3. Model Description

採用典型的 Learning to Rank(L2R)方法，針對不同類型的模型架構以及 Loss function，簡述如下：

3.1 Pointwise approach Model using DNN structure

(1)方法概述：

Let be $X = \{x_1, \dots, x_m\}$ a set of QDPs, and let $Y = \{y_1, \dots, y_m\}$ be a set of rankings for these pairs, where $y_i \in \{1, \dots, s\}$, for some small $s \in \mathbb{N}$ (3,5 in our datasets). For a QDP $x_i = \langle q, u \rangle$ the value of y_i determines the relevance of the document *u* to the query *q*, $y_i = 1$ being irrelevant and $y_i = s$ being most relevant.

Let $F = \{f_1, \dots, f_n\}$ be a set of ranking features such that . For each $x_i \in X$, the learning algorithm receives an *n*-dimensional vector $x = \{f_1(x_i), \dots, f_n(x_i)\}$. Denote . The goal of the learning algorithm, given *X*, *Y*, is to predict a correct ranking of an unseen QDPs set *X'*. The output of such algorithm is a function $\mathrm{H}: \mathrm{X} \mapsto \mathbb{R}$. A loss function $L(X, Y)$ is used in order to measure an algorithm's predictive success.

(2)模型簡述：

4 layers DNN structure with a softmax layer at last

Loss Function: Pointwise Loss (PTL)- the normalized Euclidean distance

$$\mathrm{PTL} = \frac{1}{m}\sum_{i=1}^{m}\big(y_i - H(x_i)\big)^2$$

3.2 Pairwise comparing Model using DNN structure

(1)方法概述：

This approach used *Y* to achieve an order relation '$\preccurlyeq$' over *X*. For $x_i, x_j \in X$ say that $x_i \succcurlyeq x_j$ if $y_i > y_j$. The goal of pairwise algorithms is to output a function *H*, that given x_i, x_j such that $x_i \succcurlyeq x_j$, will satisfy $H(x_i) \geq H(x_j)$. Note that the numerical value of the rank of the QPD is irrelevant

(2)模型簡述：

4 layers DNN structure with a softmax layer at last

Loss Function: Pairwise Weighted Loss (PRWL)

Let $\Phi: X \times \mathrm{X} \mapsto \mathbb{R}$ be a function such that $\Phi(x_i, x_j) = y_j - y_i$. Define $D(x_i, x_j) = max\{0, Z \cdot \Phi(x_i, x_j\}$, where Z is some normalization factor so that *D* will become a distribution over *X*×*X*, namely such that $\sum_{x_i, x_j \in X} D(x_i, x_j) = 1$. Note that this distribution gives higher mass to pairs $x_i, x_j \in X$ such that $y_j - y_i$ is large,

namely, it puts a lot of weight on pairs with disparate ranking. Finally, define

$$\text{PRWL} = \Pr_{(x_i,x_j)\sim D}[H(x_j) \le H(x_i)]$$

3.3 Leakage method

(部分概念參考自 CuteChibiko 於 Kernel 中提出的相關看法)

主辦方提供的 events.csv 中，包含了 train set 和 test set 中每個 display_id 所對應到的點閱紀錄。共有下列敘述的幾個資訊，

(1) uuid: 點閱者的 id，

(2) document_id: 該 display 出現的網頁的 id，

(3) Timestamp: 該使用者於何時點閱。

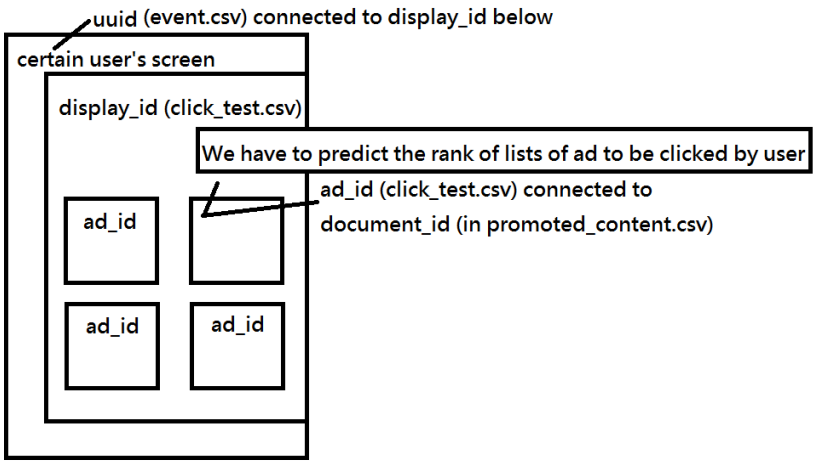
另一個檔案，page_views.csv 中則包含了每個使用者(uuid)的相關點閱紀錄。共有下列敘述的幾個資訊，

(1) document_id: 該使用者所觀看的網頁的 id

(2) timestamp: 該使用者於何時觀看該網頁

因此，我們可以從 page_views.csv 得知某個特定使用者待在某一個網頁的紀錄，累加所有被該特定使用者瀏覽過的網頁後，找出所有被該使用者瀏覽過的網頁中最常被瀏覽的網頁，依瀏覽次數排名。當我們在預測 test data 中某一個 display 所有 ad_id 被點擊的機率大小時，可以參考上面的排名，排序 ad_id 的先後次序。直觀而言，test data 應該與 train data 取自不同的時間區塊，兩個時間區塊的使用者瀏覽紀錄並未重疊，而主辦方提供的 events.csv 和 page_views.csv 應該是取自 train data 採樣的時間區塊。但是事實上，test data 和 train data 採樣於部分重疊的時間區塊中，因此我們稱此為 leakage。就某種程度上而言，我們相當於窺視了部分 test data 不該公開的資訊。

我們的做法大致上如下，先透過 promoted_content.csv 建立連結 (promoted_content.csv 主要提供了 ad_id 和 document_id 之間的對應關係，也就是點擊某個 ad 後會進入哪一個網站的 landing page)。data 有以下關係



因此，我們現在有了 display_id (特定 user uuid 待在某個 display 的紀錄)—ad_id (點擊該 ad 所進入的 document_id)一連串的關係(結合自 click_test.csv、promoted_content.csv、event.sv)，我們可以查找 display_id(uuid)—ad_id(document_id)的對應關係是否存在於 page_views.csv 中。一旦在 page_views.csv 中找到，我們可以直接給予該 ad_id 較大的機率值(EX: 0.95)。當我們在預測時，就能依照某個 ad_id 對應到的機率大小作出 rank。而剩下其無法在 page_views.csv 查找到的對應關係，則依照各個網頁被瀏覽的次數做排序，依序給予相應大小的機率值(BTB 做法 in Kernel)。

另一方面，我們試著去觀察在 leakage 在 train data 中的佔有多少分量。我們估算了 train data 中有多少 document(ad_id)被點擊，有多少在 train data 中 display_id(uuid)—ad_id(document_id)的對應關係能被查找到。以及上述兩集合的交集。以下是我們得到的數據：

在 train data set 上(只包含 page_views_sample.csv)
TP: 4863　FP: 70　FN:1928123　Recall: 0.3%　Precision: 98.6%

對於所有 data (包含完整 page_views.csv)
TP: 724749　FP: 31813　FN: 16149844　Recall: 4.3%　Precision: 95.8%

TP: 被點擊且 display_id(uuid)—ad_id(document_id) 的對應關係在 page_view.csv 中
FP: display_id(uuid)—ad_id(document_id) 的對應關係在 page_view.csv 中卻未被點擊
FN: 被點擊但 display_id(uuid)—ad_id(document_id) 的對應關係未在 page_view.csv 中
Recall: display_id(uuid)—ad_id(document_id) 對應關係在 page_view.csv 中的 data，clicked data 佔的比率
Precision: clicked data 中，display_id(uuid)—ad_id(document_id) 對應關係在 page_view.csv 的 data 佔的比率

我們可以發現，在 page_view.csv 中只有 4.3%的 data 有被 train data 記錄其點擊行為。這個大小遠小於我們的想像，這讓我們無法想像其他 95.75 的 landing page 是如何被瀏覽卻未曾被使用者所點擊。可能的原因是某些頁面並非透過點擊 ad 的方式連結，而是以其他方式進入。因此 page_views.csv 有這些頁面的瀏覽紀錄，但是卻未曾透過 ad 被連結。
另一方面，如果採用 leakage information 的話，在 train data 上能夠達到 95.8%的表現。假設 test data 上也能達到如此高的精確度的話，那麼這將是一項極為有用的資訊。

4. Experiments and Discussion

4.1 Evaluation Function

Submissions are evaluated according to the Mean Average Precision @12 (MAP@12)

$$MAP@12 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{min(12,n)} P(k)$$

where |U| is the number of display_ids, P(k) is the precision at cutoff k, n is the number of predicted ad_ids.

4.2 Discussion on Methods adopted

DNN 架構以 one-hot vector 為主要 feature 之形式較難 train，推測原因為資料代表的 vector 過於稀疏(sparse)，原先嘗試將原始 feature 經過 Truncated SVD/ PCA 降維(利用 hw4 的 code，也嘗試了用 Autoencoder, Variational Autoencoder 進行降維)，採用類似 LSA 概念，分別降至 200/ 500/ 1000 維後經過 normalization 或不經過 normalization，效果沒有比沒有降維的 features 好，不過可以減少 training 的時間。

model 訓練部分，Pairwise model 的 performance 無法繼續上升的原因應該是 Feature engineering 準備不足，由於時間關係本次的實作並沒有把 3.3 的 leakage solution 合併到 3.2 pairwise comparison method 作為 feature 進行實作。

4.3 Performance

Kaggle Estimation 結果比較如下：

Method	Performance (MAP@12)
Ad_id statistics grouping	0.63714
Leakage method	0.65249
Pairwise-comparing method	0.63392

由於 pointwise method 的模型不太好 Train(相比於 pairwise)之下，因此用 DNN 架構的 Pointwise model 尚沒有辦法達到良好的誤差值(用 validation set)來看最初的 pairwise-comparing method 可以達到不錯的結果，不過一開始 trainin 時的 feature engineering 沒有做好，加入針對 ad_id 本身的資料統計當作 feature 應該可以達到不錯的效果

此外，除了 3.中提到的方法，亦使用了基本的 ad_id 統計資料－如總點擊數字、標準差，作為排序依據(單看 ad_id)，經過 MAP 評估可以達到 0.63714 的 performance，代表單純的統計資料量可以做為資料變形當作有效的 feature 使用。

4.4 Conclusion－Potential Effort

首次參加 machine learning 的比賽，從一開始問題分類確定 learning to rank 後開始採用不同的方法訓練模型，經過這次比賽了解到數據量大的情況下可以分析不同的資料變形方法加強 training 的效果。

此外，除了 3.中提到的方法之外，下列也是 L2R 問題中常見的解決辦法

(1) Listwise Method

需要較長的 training time

(2) FFM (Field-aware Factorization Machines)

可以用於處理 sparse matrix 的問題，在數據量大以及數據稀疏

參考資料: http://www.csie.ntu.edu.tw/~cjlin/papers/ffm.pdf

(3) Xgboost

boosting 分類器的一種，訓練時間較短並且可以進行有效預測

上述方法中，尤其 FFM 在近年來的 CTR 比賽中表現更是不俗，善於處理資料量大且 sparse 的情況。模型建構之外，對於原始資料的統計數據分析雖然不屬於 machine learning 的課題之內，但卻對整體的訓練過程有相當的幫助，或許在往後參加比賽的過程中需要引入一些 Data mining 的資料分析手法。

5. Reference

1. Introduction to Boosted Tree https://xgboost.readthedocs.io/en/latest/
2. 深入理解 FFM 原理與實踐 http://www.cnblogs.com/zhizhan/p/5238415.html
3. Learning to Rank: From Pairwise Approach to Listwise Approach
   http://www.machinelearning.org/proceedings/icml2007/papers/139.pdf