機器學習 - NTUDAC

James Yeh

辛普森悖論(Simpson's paradox)

世界上有三種謊言：謊言，該死的謊言，統計數字。』(There are three kinds of lies: lies, damned lies, and statistics) - 馬克·吐溫

辛普森悖論-實際案例

腎結石治療方案

Kidney stones
Stone passes through ureter and bladder
Often painful when passed

辛普森悖論-實際案例

醫生詢問你要哪種方案？

方案	患者人數	治癒人數	成功率
方案A	350	273	78.00%
方案B	350	289	82.57%

Kidney stones
Stone passes through ureter and bladder
Often painful when passed

辛普森悖論-實際案例

醫生詢問你要哪種方案？

方案	患者人數	治癒人數	成功率
方案A	350	273	78.00%
方案B	350	289	82.57%

勝

Kidney stones
Stone passes through ureter and bladder
Often painful when passed

真的這麼簡單嗎？

辛普森悖論-實際案例

方案	小結石	大結石	患者人數
方案A	(93%)81/87	(73%)192/263	350
方案B	(87%)234/270	(69%)55/80	350

辛普森悖論-實際案例

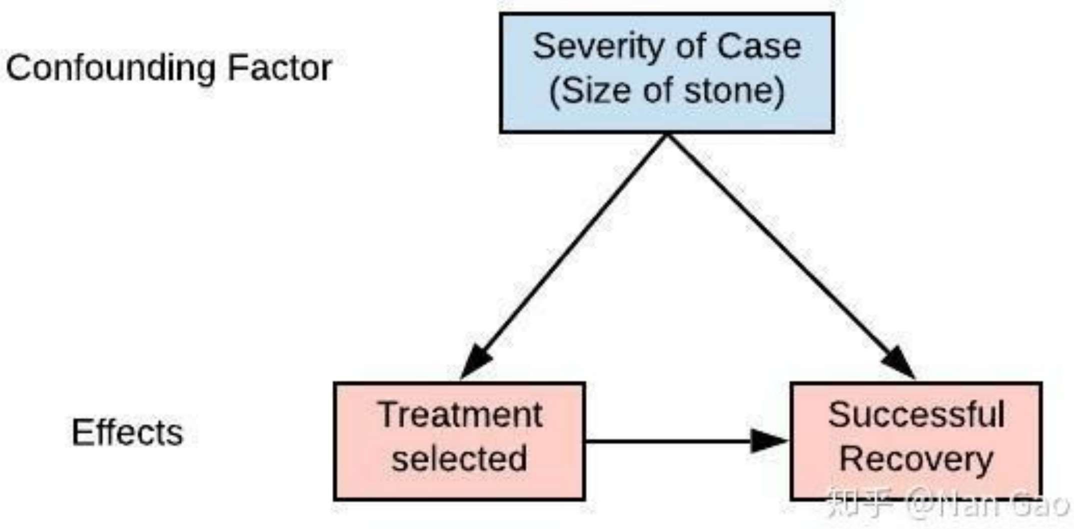
方案	小結石	大結石	患者人數
方案A	勝 (93%)81/87	勝 (73%)192/263	350
方案B	(87%)234/270	(69%)55/80	350

辛普森悖論-實際案例

我少讀書　你不要騙我

I didn’t have much education.　Don’t try to fool me

藏在魔鬼中的細節



Confounding Factor

Severity of Case
(Size of stone)

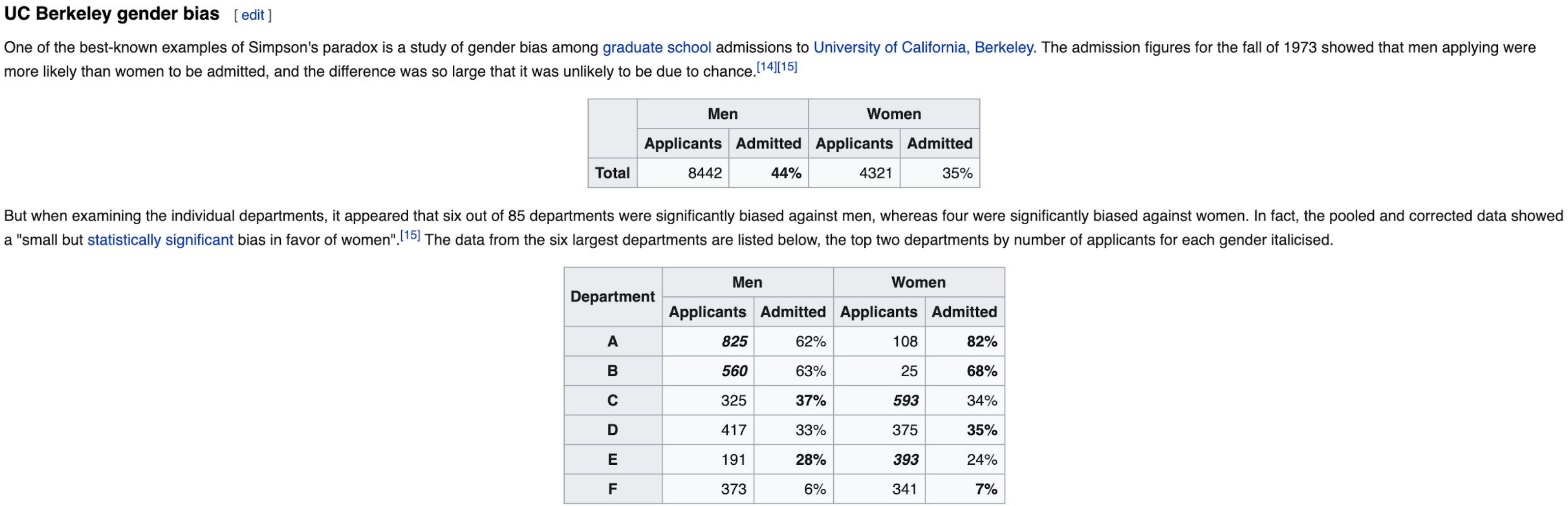
Effects

Treatment selected

Successful Recovery

真實案例

- 柏克萊入學男女比歧視



UC Berkeley gender bias [edit]

One of the best-known examples of Simpson's paradox is a study of gender bias among graduate school admissions to University of California, Berkeley. The admission figures for the fall of 1973 showed that men applying were more likely than women to be admitted, and the difference was so large that it was unlikely to be due to chance.[14][15]

	Men		Women	
	Applicants	Admitted	Applicants	Admitted
Total	8442	**44%**	4321	35%

But when examining the individual departments, it appeared that six out of 85 departments were significantly biased against men, whereas four were significantly biased against women. In fact, the pooled and corrected data showed a "small but statistically significant bias in favor of women".[15] The data from the six largest departments are listed below, the top two departments by number of applicants for each gender italicised.

Department	Men		Women	
	Applicants	Admitted	Applicants	Admitted
A	***825***	62%	108	**82%**
B	***560***	63%	25	**68%**
C	325	**37%**	***593***	34%
D	417	33%	375	**35%**
E	191	**28%**	***393***	24%
F	373	6%	341	**7%**

從辛普森悖論學到？

- ~~騙人的技巧？~~
- 從低維度看到的資訊是偏頗的(合併數據會損失訊息)
- 實際在分析時需結合產業知識, 判斷因果關係
- AB test需注意分群

如何設計滿足辛普森悖論的例子？

https://web.math.sinica.edu.tw/math_media/d412/41205.pdf

参考

1. Wiki Simpson's paradox https://en.wikipedia.org/wiki/Simpson%27s_paradox#
2. 机器学习中的因果关系: 从常见的统计学谬误 —— 辛普森悖论讲起 https://zhuanlan.zhihu.com/p/95461240

辛普森悖論在機器學習的應用

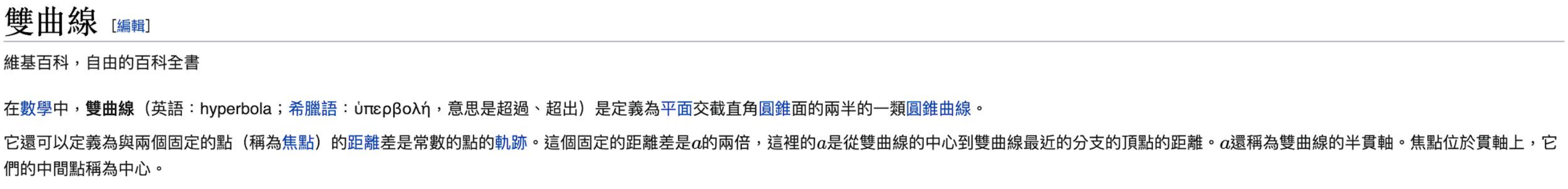
特徵工程添加兩兩特徵關係

$$\phi(x) = w_0 + \sum_{i=1}^{n} w_i x_i$$

$$\Big\downarrow$$

$$\phi(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \boxed{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j}$$

Degree-2 Polynomial



雙曲線 [編輯]

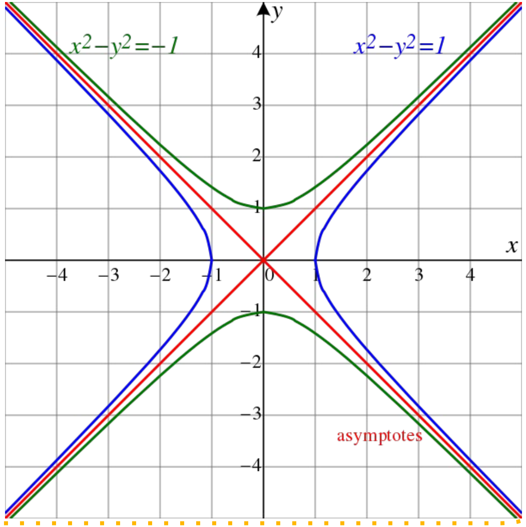
維基百科，自由的百科全書

在數學中，**雙曲線**（英語：hyperbola；希臘語：ὑπερβολή，意思是超過、超出）是定義為平面交截直角圓錐面的兩半的一類圓錐曲線。

它還可以定義為與兩個固定的點（稱為焦點）的距離差是常數的點的軌跡。這個固定的距離差是a的兩倍，這裡的a是從雙曲線的中心到雙曲線最近的分支的頂點的距離。a還稱為雙曲線的半貫軸。焦點位於貫軸上，它們的中間點稱為中心。

從代數上說，雙曲線是在笛卡兒平面上由如下方程式定義的曲線

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$



https://zh.wikipedia.org/wiki/%E5%8F%8C%E6%9B%B2%E7%BA%BF

添加兩兩特徵關係(Degree-2 Polynomial)

$$\phi(x) = w_0 + \sum_{i=1}^{n} w_i x_i$$

↓

複雜度 N平方

$$\phi(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \boxed{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j}$$

Degree-2 Polynomial特徵工程 vs 人工挑選

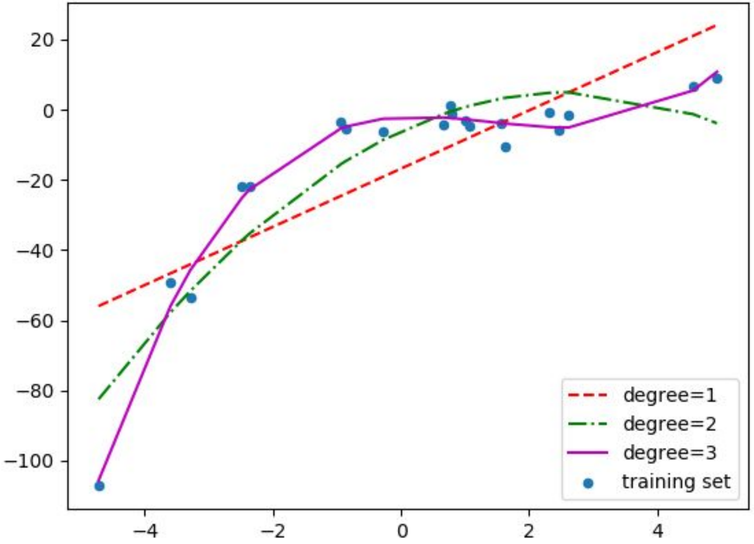
Degree-2 Polynomial選

1. 參數大幅增加，由線性增加至平方級（運算時間以及記憶體需求大幅增加）
2. 樣本非常稀疏(overfit）

人工選

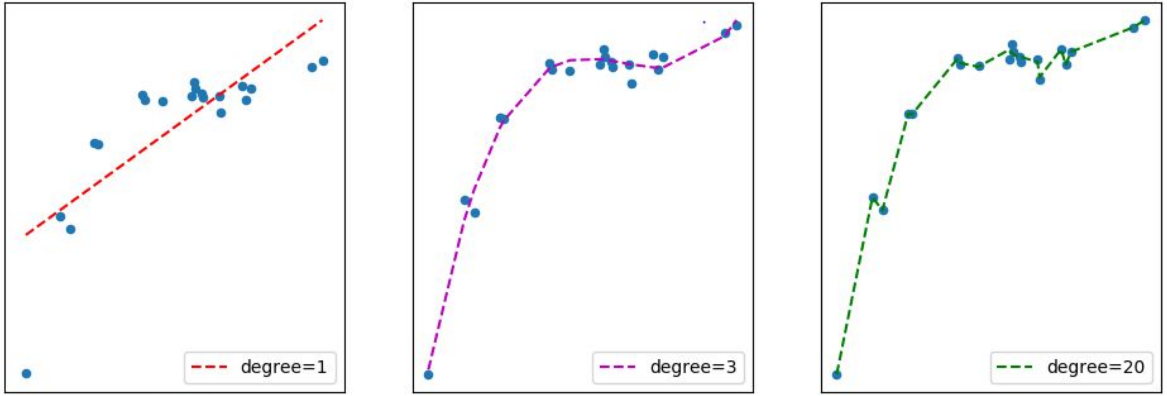
1. 可以大量減少運算量，以及加入產業知識
2. 選擇的方法對於不同的商業場景不通用（產業知識）
3. 人工選非常耗時，還不一定可以選出來

Polynomial Regression



https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

Polynomial Regression



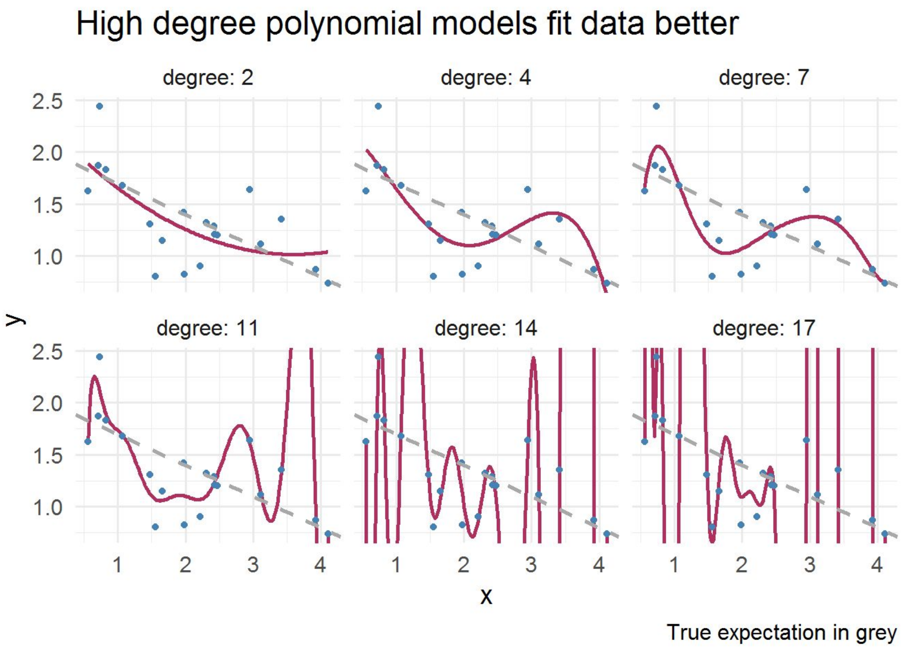
Underfit
High Bias
Low Variance

Correct Fit
Low Bias
Low Variance

Overfit
Low Bias
High Variance

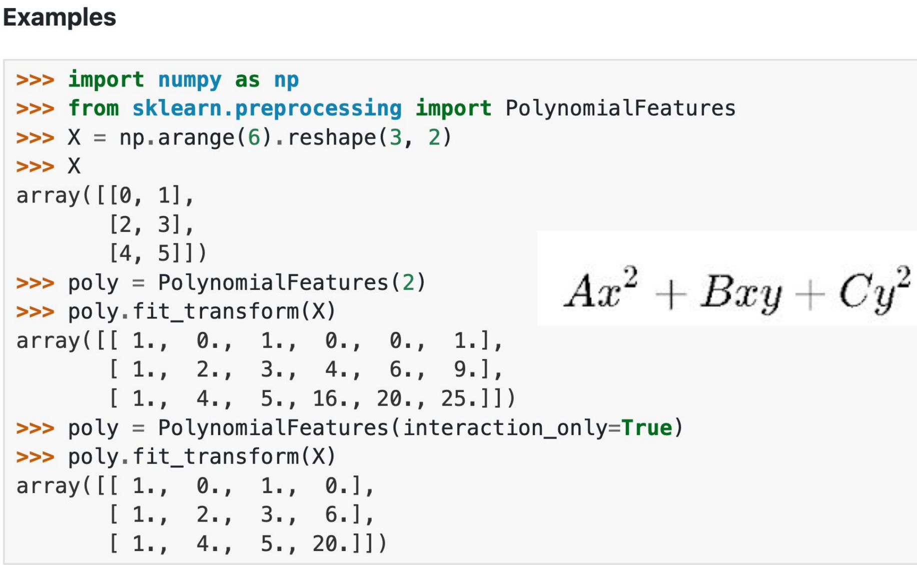
https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

Polynomial Regression(overfit)



https://www.alexpghayes.com/blog/overfitting-a-guided-tour/

Degree-2 Polynomial實際案例



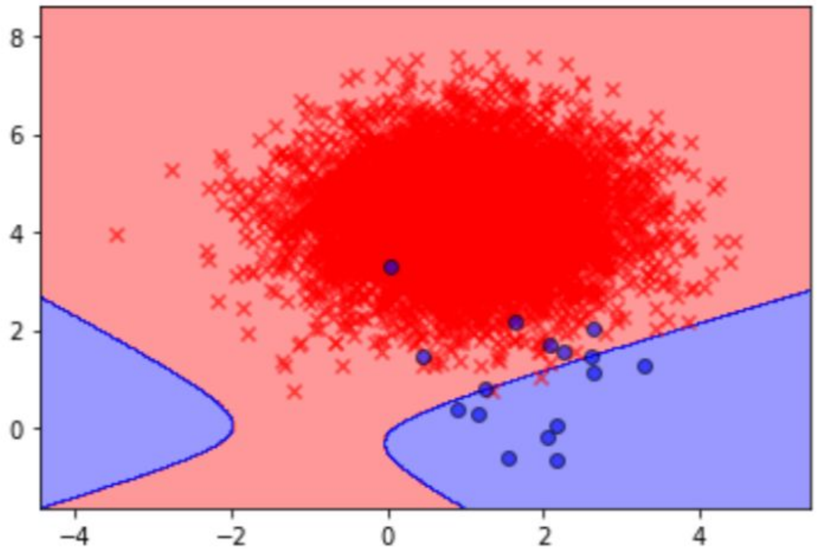
Examples

```
>>> import numpy as np
>>> from sklearn.preprocessing import PolynomialFeatures
>>> X = np.arange(6).reshape(3, 2)
>>> X
array([[0, 1],
       [2, 3],
       [4, 5]])
>>> poly = PolynomialFeatures(2)
>>> poly.fit_transform(X)
array([[ 1.,  0.,  1.,  0.,  0.,  1.],
       [ 1.,  2.,  3.,  4.,  6.,  9.],
       [ 1.,  4.,  5., 16., 20., 25.]])
>>> poly = PolynomialFeatures(interaction_only=True)
>>> poly.fit_transform(X)
array([[ 1.,  0.,  1.,  0.],
       [ 1.,  2.,  3.,  6.],
       [ 1.,  4.,  5., 20.]])
```

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html

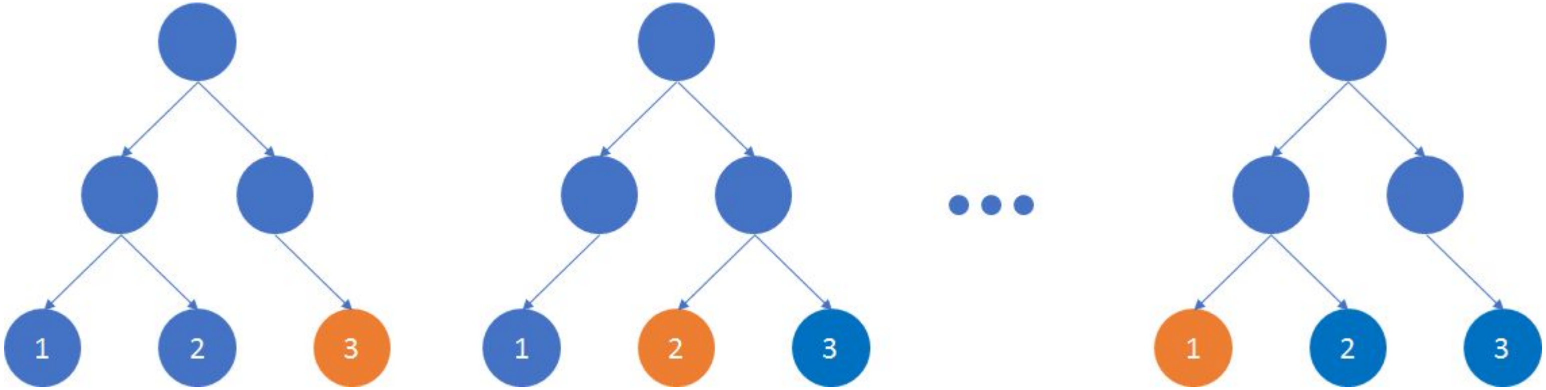
Degree-2 Polynomial實際案例

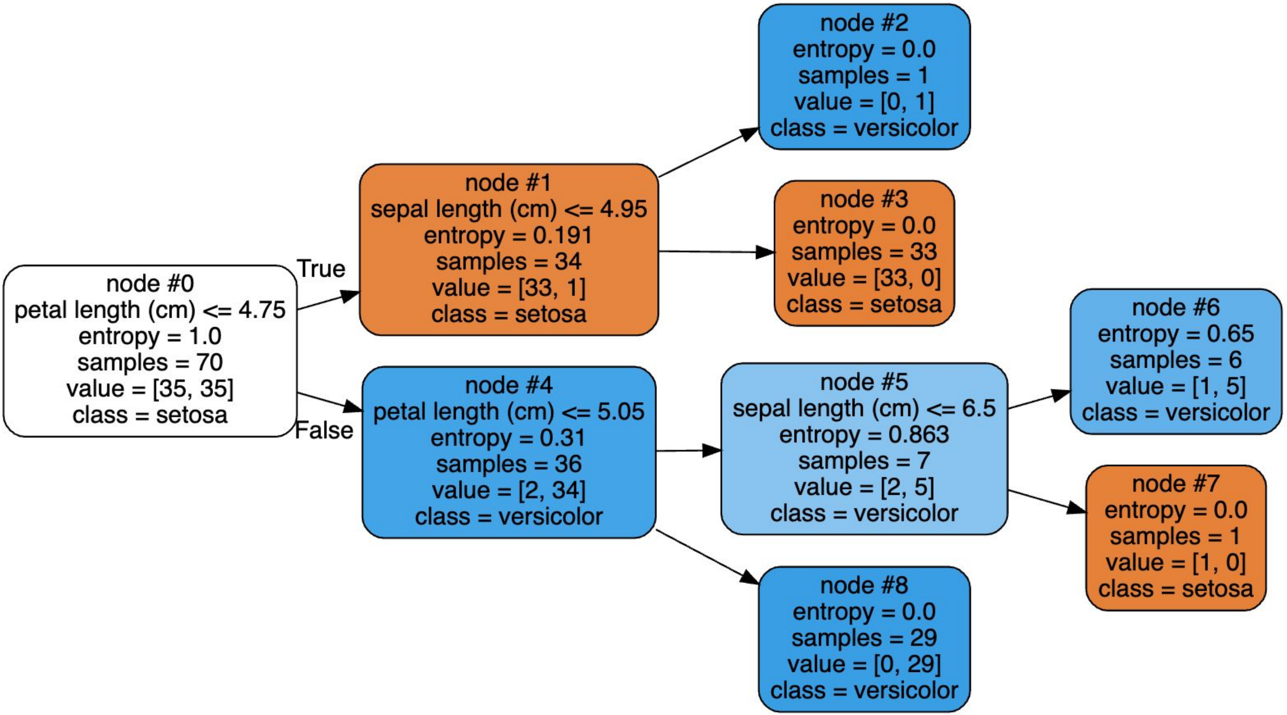


GDBT + LR(利用數模型來做特徵工程)

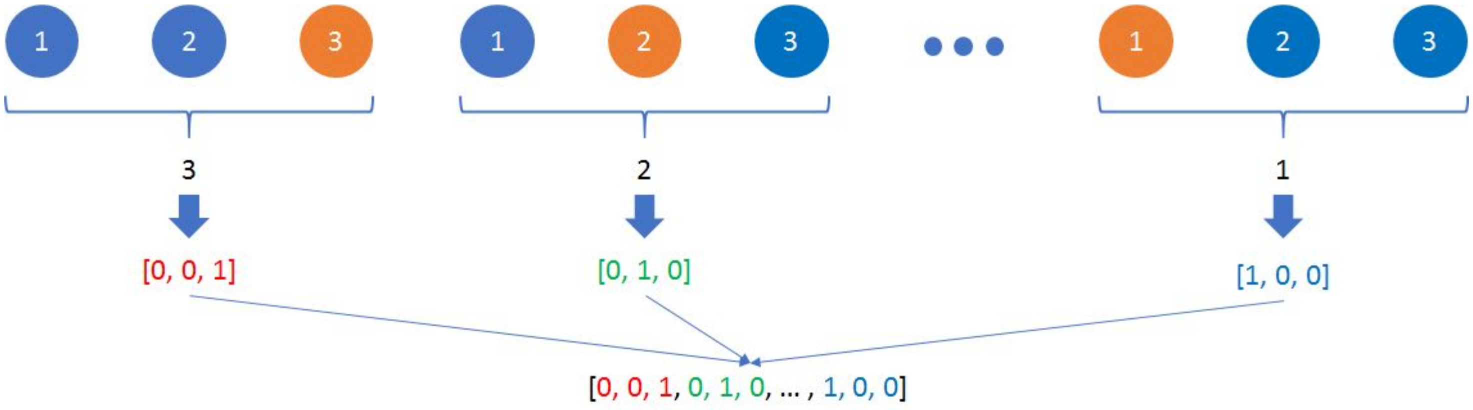
完整Github描述業界應用

GBDT Tree

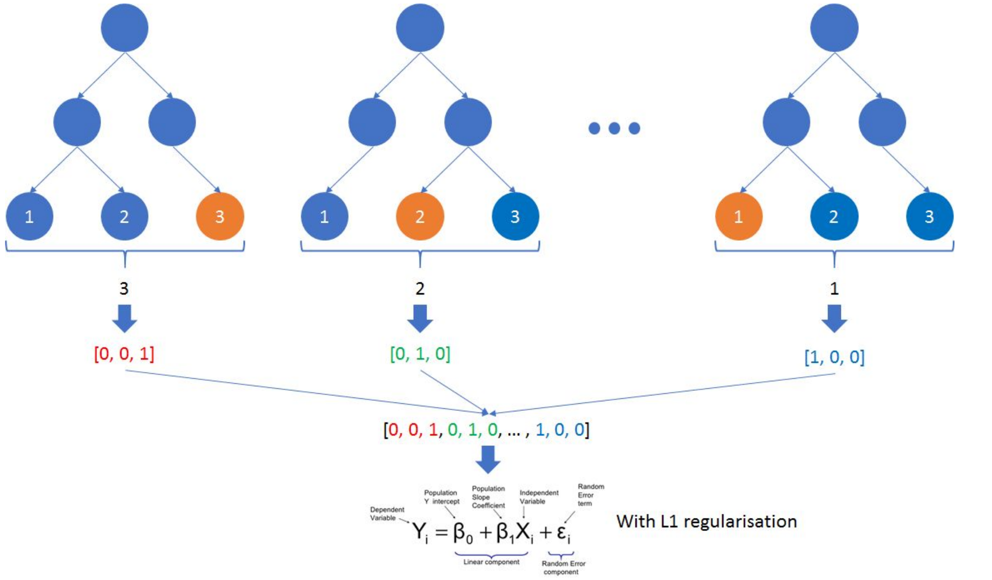


node #0
petal length (cm) <= 4.75
entropy = 1.0
samples = 70
value = [35, 35]
class = setosa
True
False
node #1
sepal length (cm) <= 4.95
entropy = 0.191
samples = 34
value = [33, 1]
class = setosa
node #4
petal length (cm) <= 5.05
entropy = 0.31
samples = 36
value = [2, 34]
class = versicolor
node #2
entropy = 0.0
samples = 1
value = [0, 1]
class = versicolor
node #3
entropy = 0.0
samples = 33
value = [33, 0]
class = setosa
node #5
sepal length (cm) <= 6.5
entropy = 0.863
samples = 7
value = [2, 5]
class = versicolor
node #8
entropy = 0.0
samples = 29
value = [0, 29]
class = versicolor
node #6
entropy = 0.65
samples = 6
value = [1, 5]
class = versicolor
node #7
entropy = 0.0
samples = 1
value = [1, 0]
class = setosa

利用葉子編號做one hot encodding

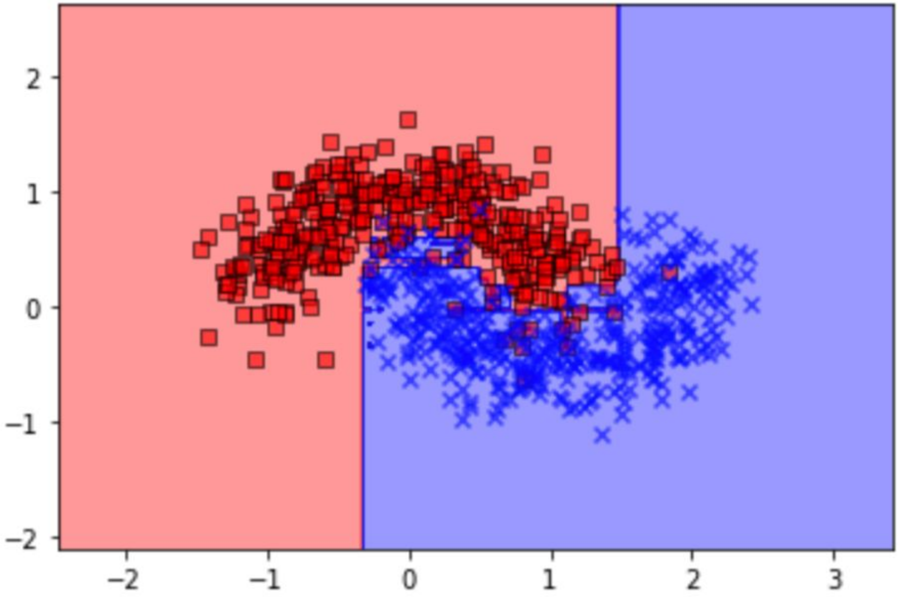


然後提供給logistic regression當作Input



With L1 regularisation

Demo



什麼是embedding？

把文字或類別轉成向量

```
# One Hot Encoding Categoricals
books = ["War and Peace", "Anna Karenina",
          "The Hitchhiker's Guide to the Galaxy"]
books_encoded = [[1, 0, 0],
                 [0, 1, 0],
                  [0, 0, 1]]
```

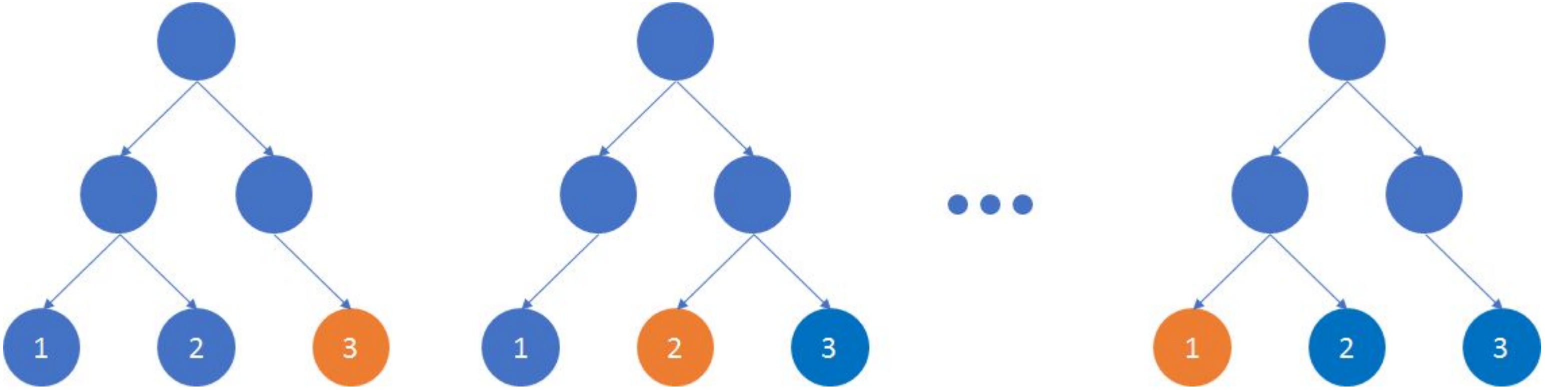
https://zhuanlan.zhihu.com/p/46016518

為什麼可以用tree 的結果做 embedding?

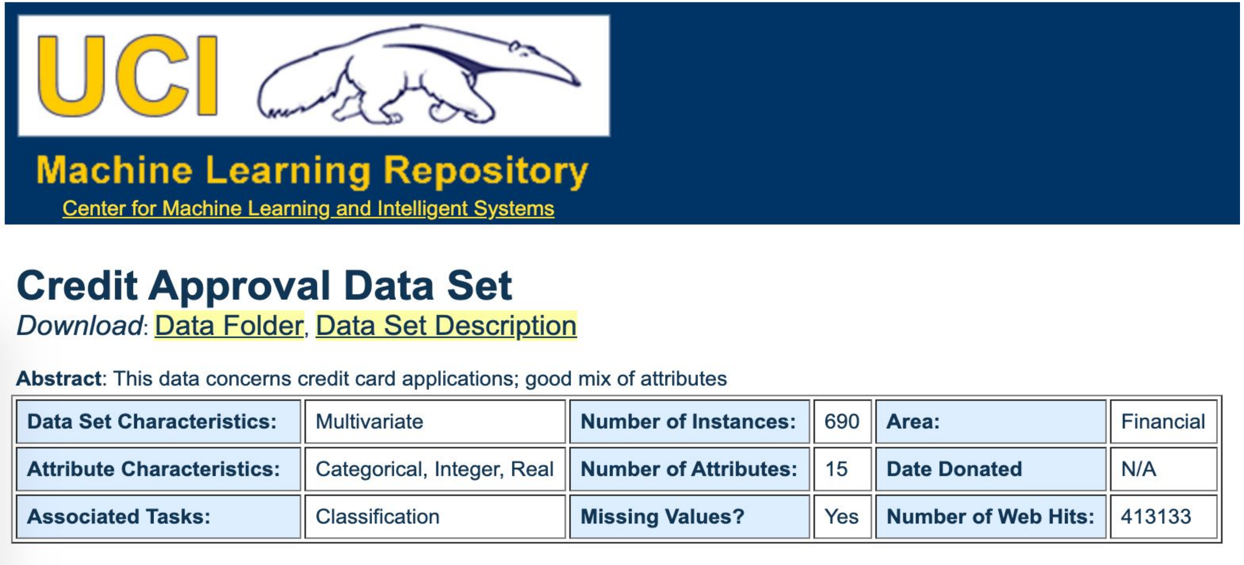
什麼是一個好的embedding？

相似的input，產出來的embedding向量應該要很近

ex:如果兩個input 很近，每棵樹數生出來的結果應該是差不多的



在GBDT+LR信用卡審核資料提升0.03 ROC準度



UCI

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Credit Approval Data Set

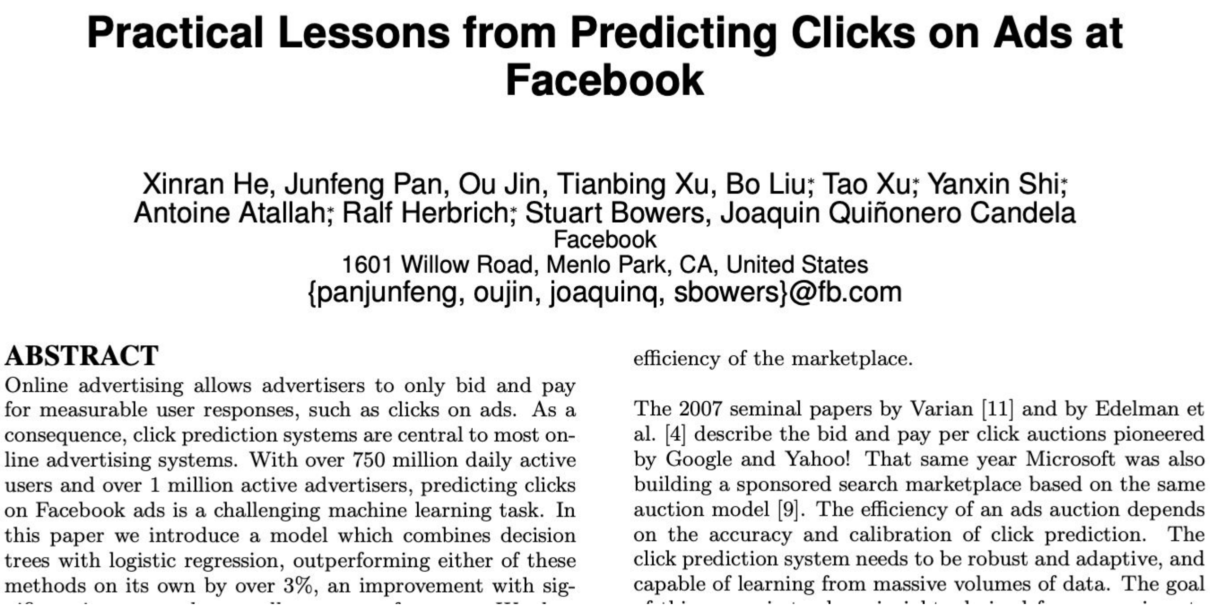
Download: Data Folder, Data Set Description

Abstract: This data concerns credit card applications; good mix of attributes

Data Set Characteristics:	Multivariate	**Number of Instances:**	690	**Area:**	Financial
Attribute Characteristics:	Categorical, Integer, Real	**Number of Attributes:**	15	**Date Donated**	N/A
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	413133

https://towardsdatascience.com/next-better-player-gbdt-lr-for-binary-classification-f8dc6f32628e

提升Facebook 3% ads CTR performance



Practical Lessons from Predicting Clicks on Ads at Facebook

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu*, Tao Xu*, Yanxin Shi*,
Antoine Atallah*, Ralf Herbrich*, Stuart Bowers, Joaquin Quiñonero Candela
Facebook
1601 Willow Road, Menlo Park, CA, United States
{panjunfeng, oujin, joaquinq, sbowers}@fb.com

ABSTRACT

Online advertising allows advertisers to only bid and pay for measurable user responses, such as clicks on ads. As a consequence, click prediction systems are central to most online advertising systems. With over 750 million daily active users and over 1 million active advertisers, predicting clicks on Facebook ads is a challenging machine learning task. In this paper we introduce a model which combines decision trees with logistic regression, outperforming either of these methods on its own by over 3%, an improvement with sig-

efficiency of the marketplace.

The 2007 seminal papers by Varian [11] and by Edelman et al. [4] describe the bid and pay per click auctions pioneered by Google and Yahoo! That same year Microsoft was also building a sponsored search marketplace based on the same auction model [9]. The efficiency of an ads auction depends on the accuracy and calibration of click prediction. The click prediction system needs to be robust and adaptive, and capable of learning from massive volumes of data. The goal

https://research.fb.com/wp-content/uploads/2016/11/practical-lessons-from-predicting-clicks-on-ads-at-facebook.pdf

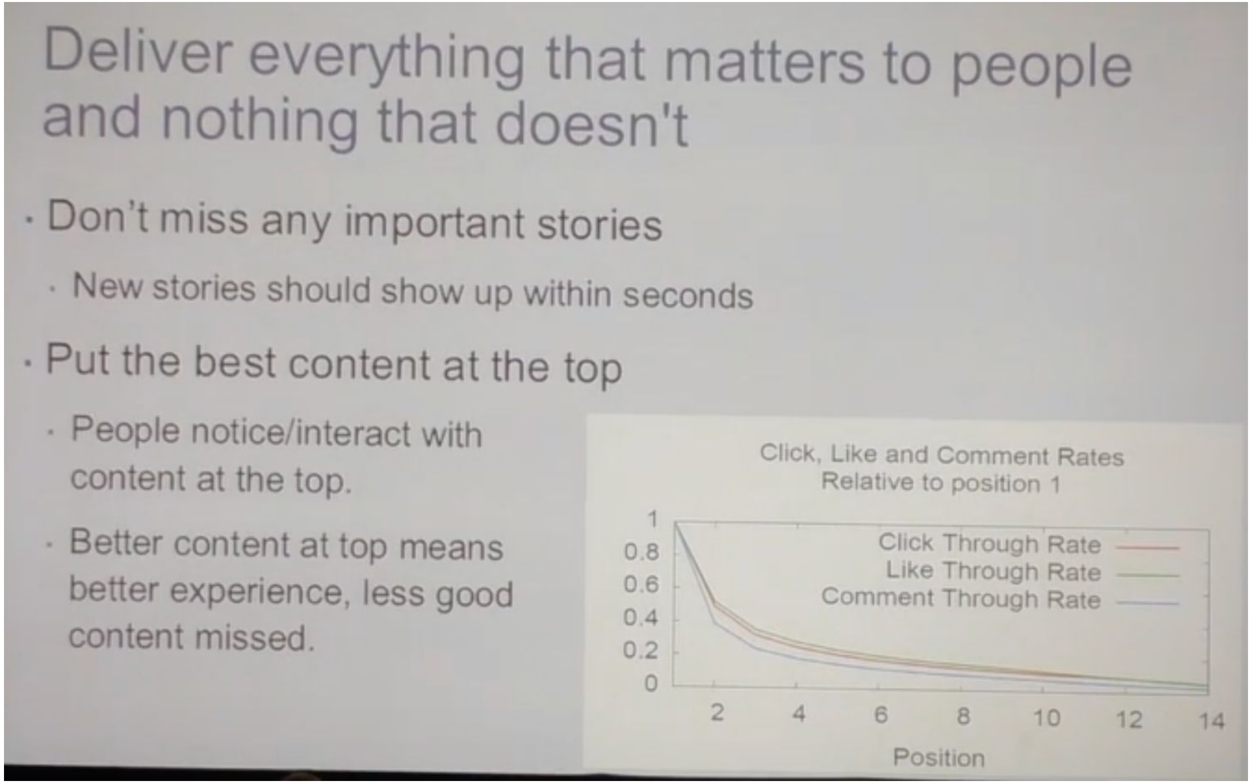
Facebook Newsfeed推薦系統介紹

Lars Backstrom - Serving a billion personalized news feeds

設計一個newsfeed推薦系統

- 如何具體描述什麼是好的newsfeed ranking?
- 實際資料量/QPS有多大？
- Feature會有哪些, model怎麼設計

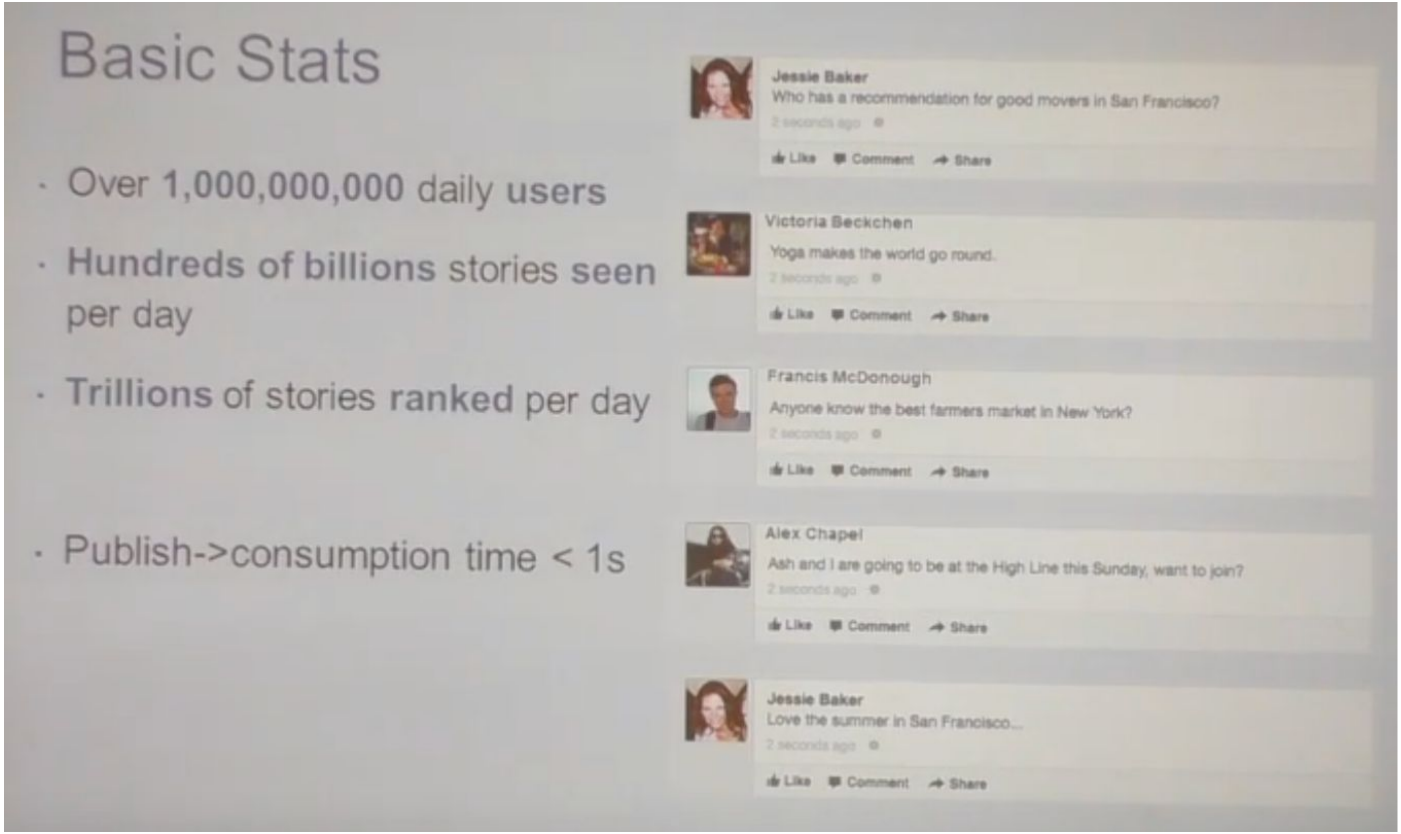
如何具體描述什麼是好的newsfeed ranking?



Deliver everything that matters to people and nothing that doesn't

- Don't miss any important stories
  - New stories should show up within seconds
- Put the best content at the top
  - People notice/interact with content at the top.
  - Better content at top means better experience, less good content missed.

實際資料量/QPS有多大？



Basic Stats

- Over 1,000,000,000 daily users
- Hundreds of billions stories seen per day
- Trillions of stories ranked per day
- Publish->consumption time < 1s

Jessie Baker
Who has a recommendation for good movers in San Francisco?
2 seconds ago

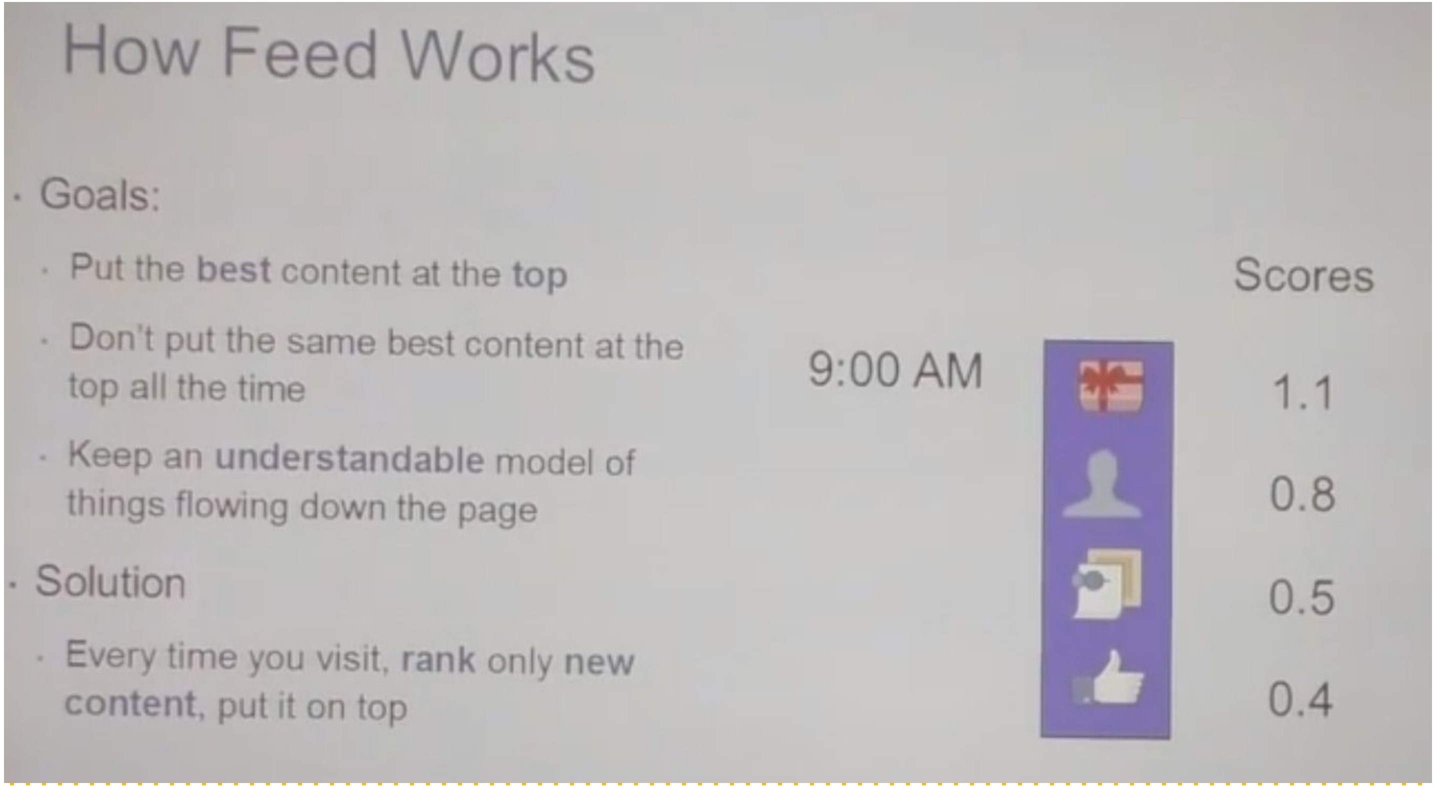
Victoria Beckchen
Yoga makes the world go round.
2 seconds ago

Francis McDonough
Anyone know the best farmers market in New York?
2 seconds ago

Alex Chapel
Ash and I are going to be at the High Line this Sunday, want to join?
2 seconds ago

Jessie Baker
Love the summer in San Francisco...
2 seconds ago

News Feed Ranking樣貌（去掉你看過的）

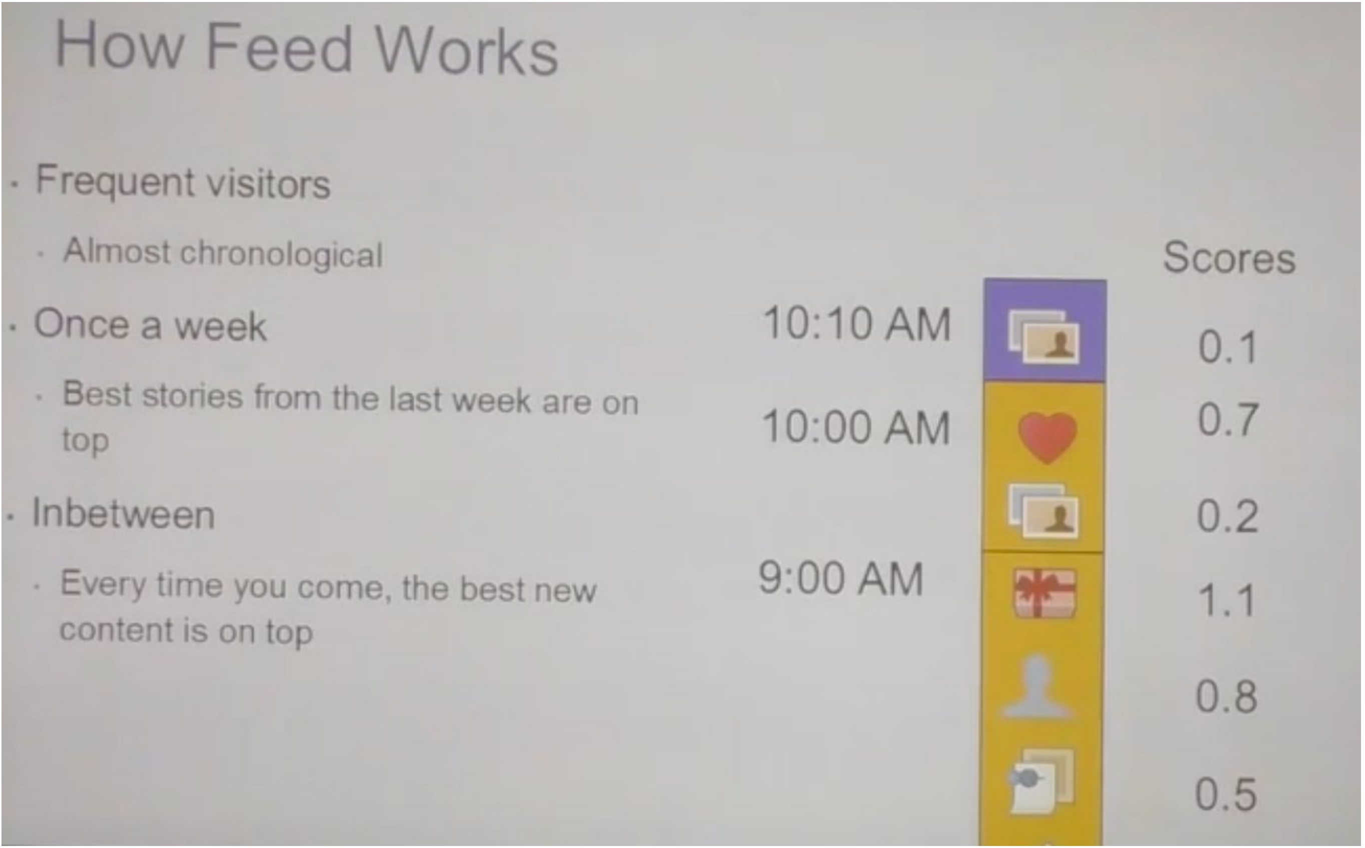


How Feed Works

- Goals:
  - Put the **best** content at the **top**
  - Don't put the same best content at the top all the time
  - Keep an **understandable** model of things flowing down the page
- Solution
  - Every time you visit, **rank** only **new content**, put it on top

Scores
9:00 AM
1.1
0.8
0.5
0.4

News Feed Ranking樣貌（去掉你看過的）

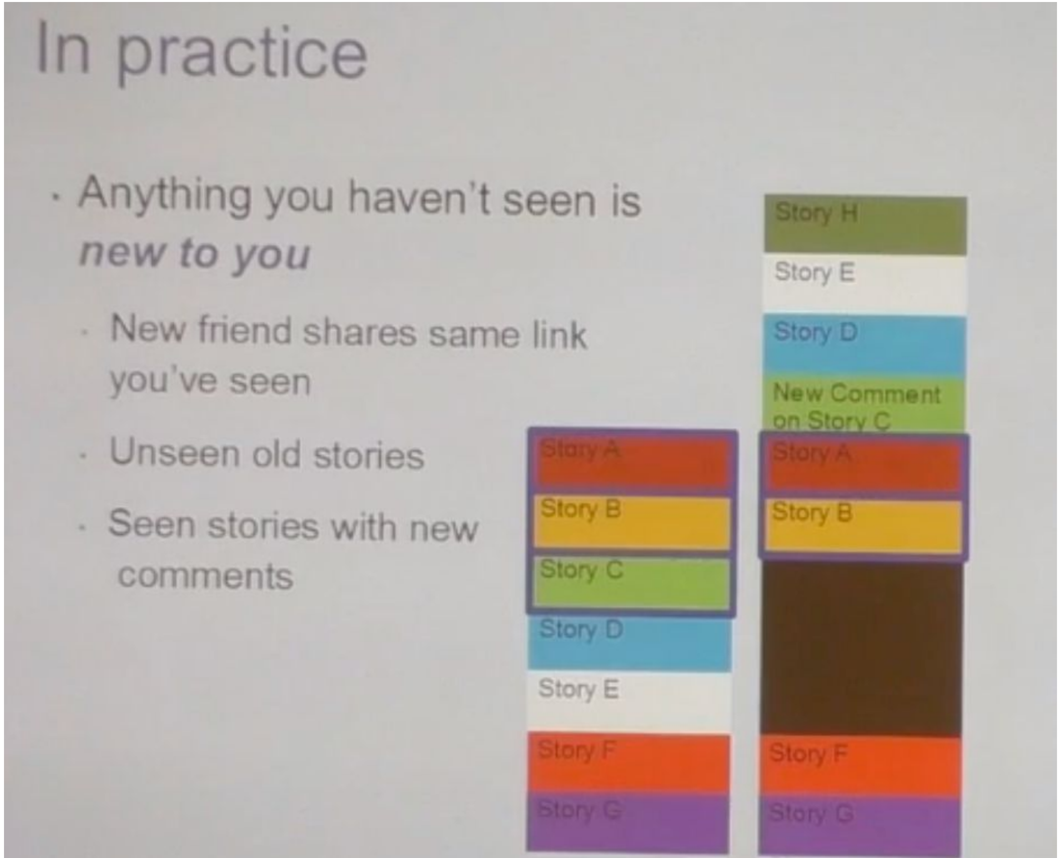


How Feed Works

- Frequent visitors
  - Almost chronological
- Once a week
  - Best stories from the last week are on top
- Inbetween
  - Every time you come, the best new content is on top

	Scores
10:10 AM	0.1
10:00 AM	0.7
	0.2
9:00 AM	1.1
	0.8
	0.5

「你看過的內容」定義比想像中模糊的

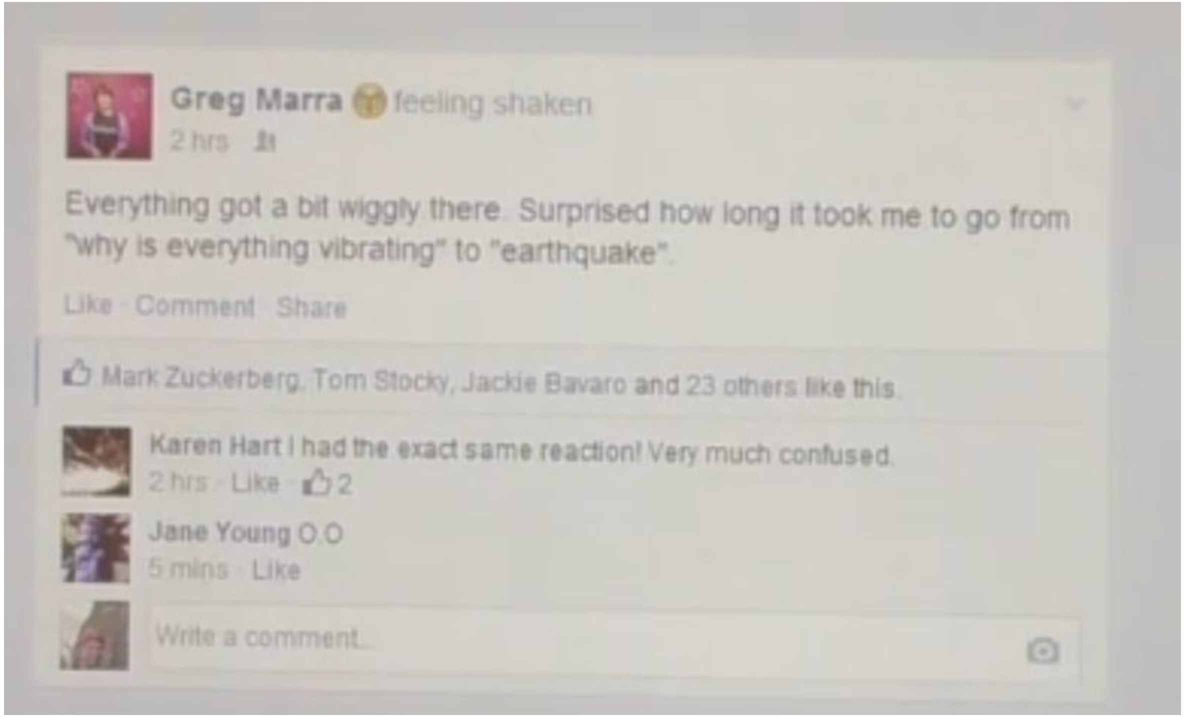


In practice

- Anything you haven't seen is ***new to you***
  - New friend shares same link you've seen
  - Unseen old stories
  - Seen stories with new comments

Story A
Story B
Story C
Story D
Story E
Story F
Story G
Story H
Story E
Story D
New Comment on Story C
Story A
Story B
Story F
Story G

Machine learning的應用



Greg Marra feeling shaken
2 hrs

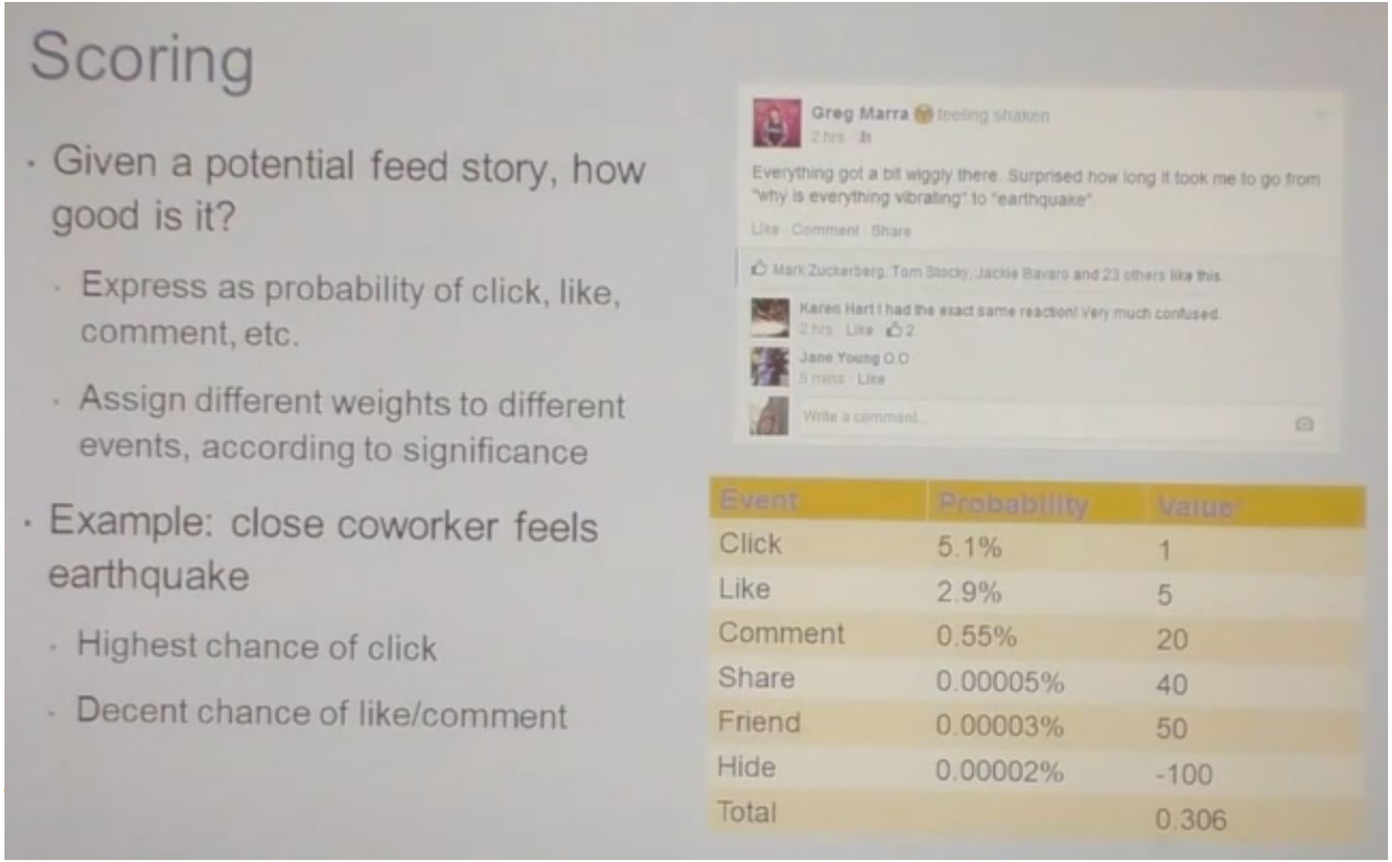
Everything got a bit wiggly there. Surprised how long it took me to go from "why is everything vibrating" to "earthquake".

Mark Zuckerberg, Tom Stocky, Jackie Bavaro and 23 others like this.

Karen Hart I had the exact same reaction! Very much confused.
2 hrs · Like · 2

Jane Young O.O
5 mins · Like

Machine learning的應用



Scoring

- Given a potential feed story, how good is it?
  - Express as probability of click, like, comment, etc.
  - Assign different weights to different events, according to significance
- Example: close coworker feels earthquake
  - Highest chance of click
  - Decent chance of like/comment

Greg Marra feeling shaken
2 hrs

Everything got a bit wiggly there. Surprised how long it took me to go from "why is everything vibrating" to "earthquake".

Like · Comment · Share

Mark Zuckerberg, Tom Stocky, Jackie Bavaro and 23 others like this.

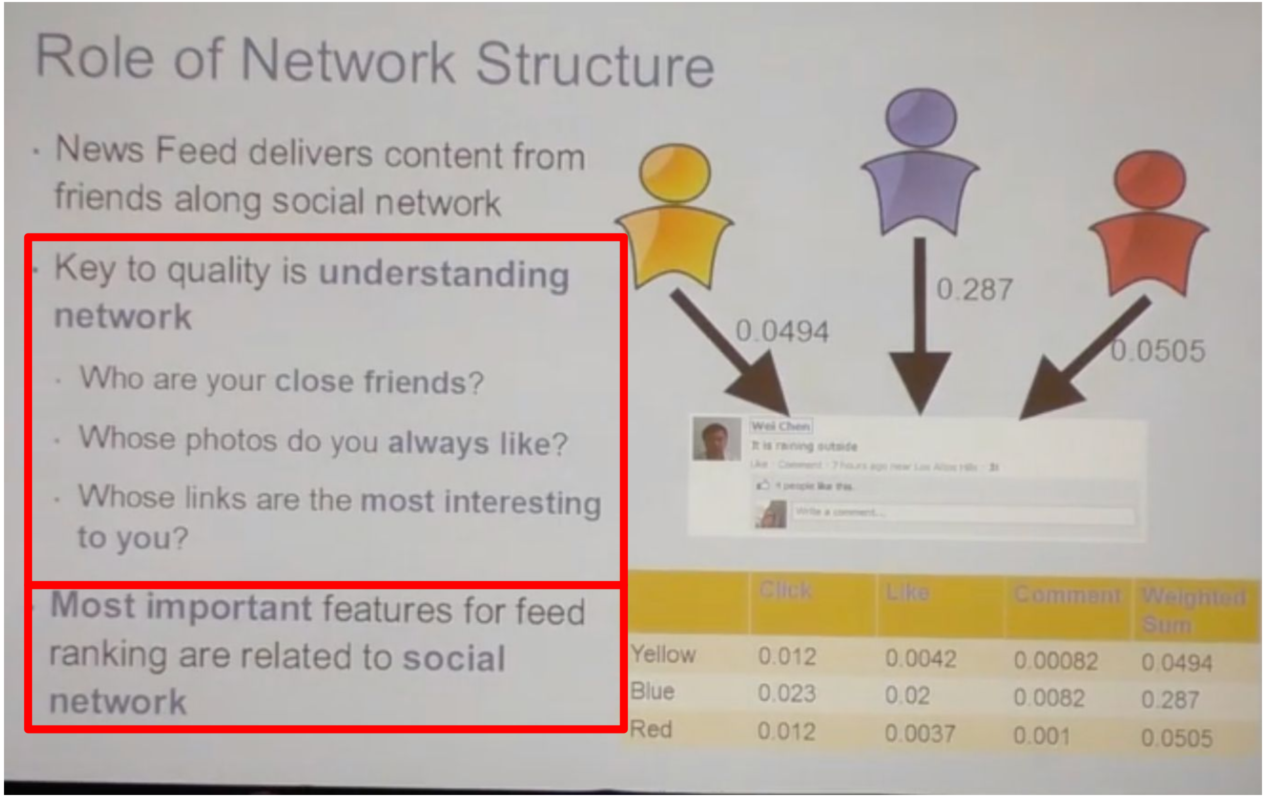
Karen Hart I had the exact same reaction! Very much confused.
2 hrs · Like · 2

Jane Young O.O
5 mins · Like

Write a comment...

Event	Probability	Value
Click	5.1%	1
Like	2.9%	5
Comment	0.55%	20
Share	0.00005%	40
Friend	0.00003%	50
Hide	0.00002%	-100
Total		0.306

Machine learning的應用



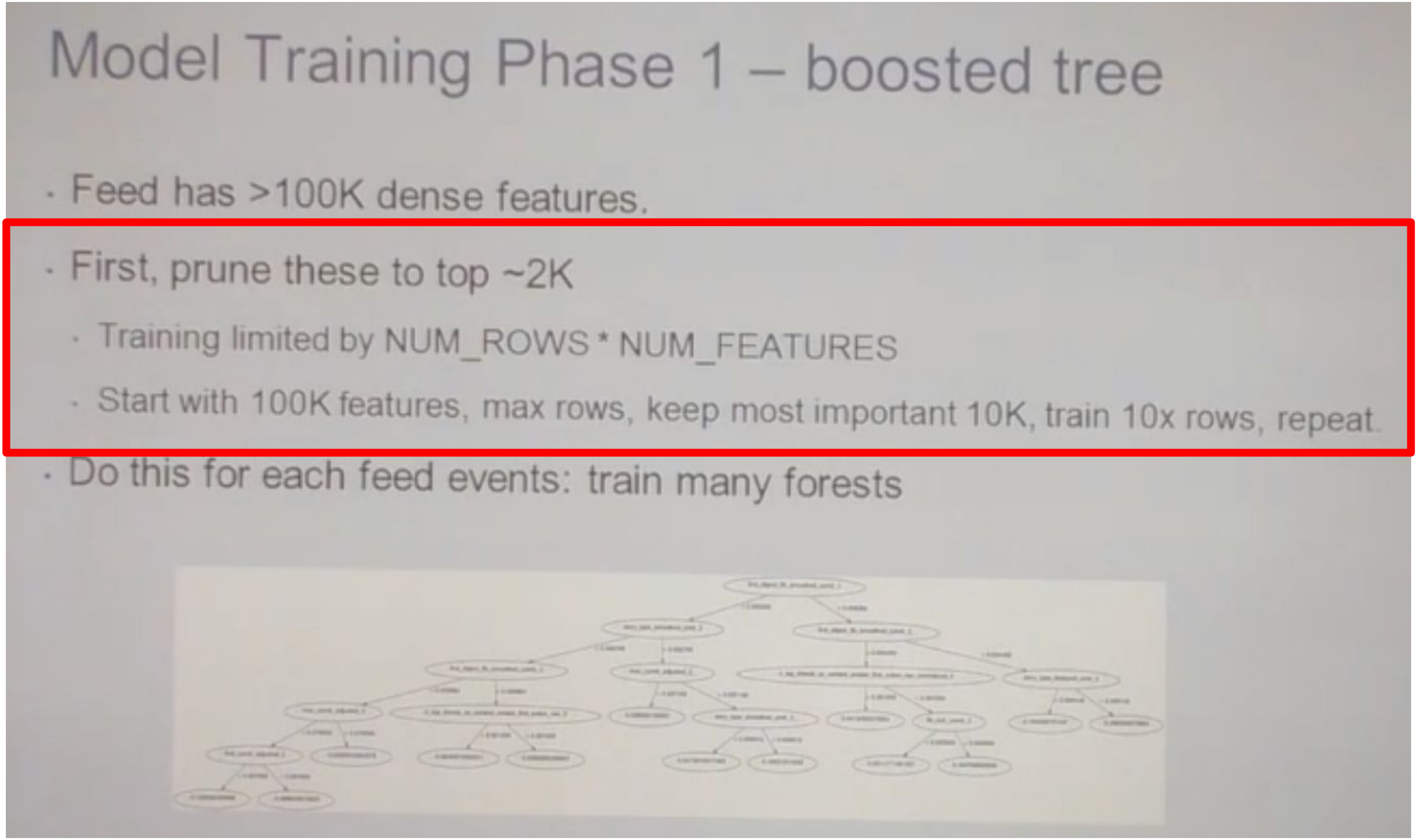
Role of Network Structure

- News Feed delivers content from friends along social network
- Key to quality is **understanding network**
  - Who are your **close friends**?
  - Whose photos do you **always like**?
  - Whose links are the **most interesting** to you?
- **Most important features for feed ranking are related to social network**

0.0494
0.287
0.0505
Wei Chen
It is raining outside
Like · Comment · 7 hours ago near Los Altos Hills ·
4 people like this.
Write a comment...

	Click	Like	Comment	Weighted Sum
Yellow	0.012	0.0042	0.00082	0.0494
Blue	0.023	0.02	0.0082	0.287
Red	0.012	0.0037	0.001	0.0505

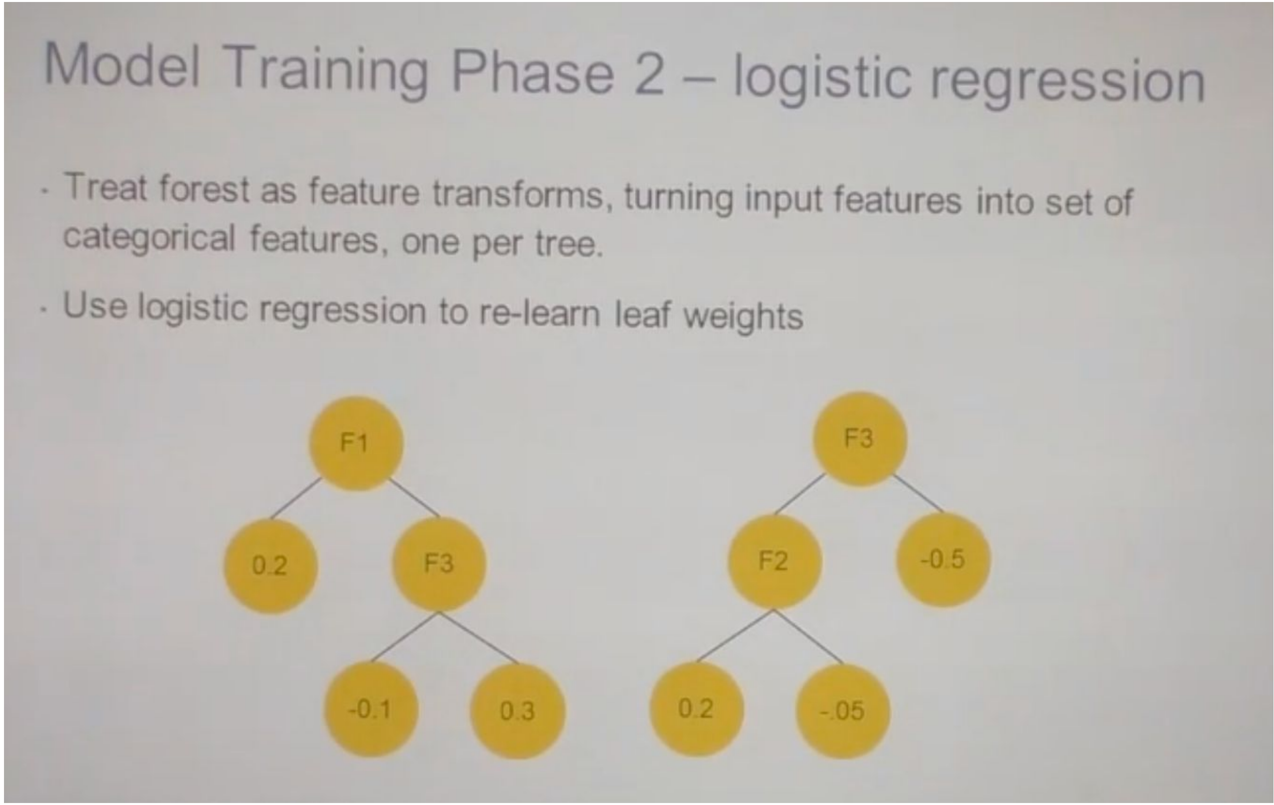
Machine learning的應用



Model Training Phase 1 – boosted tree

- Feed has >100K dense features.
- First, prune these to top ~2K
  - Training limited by NUM_ROWS * NUM_FEATURES
  - Start with 100K features, max rows, keep most important 10K, train 10x rows, repeat.
- Do this for each feed events: train many forests

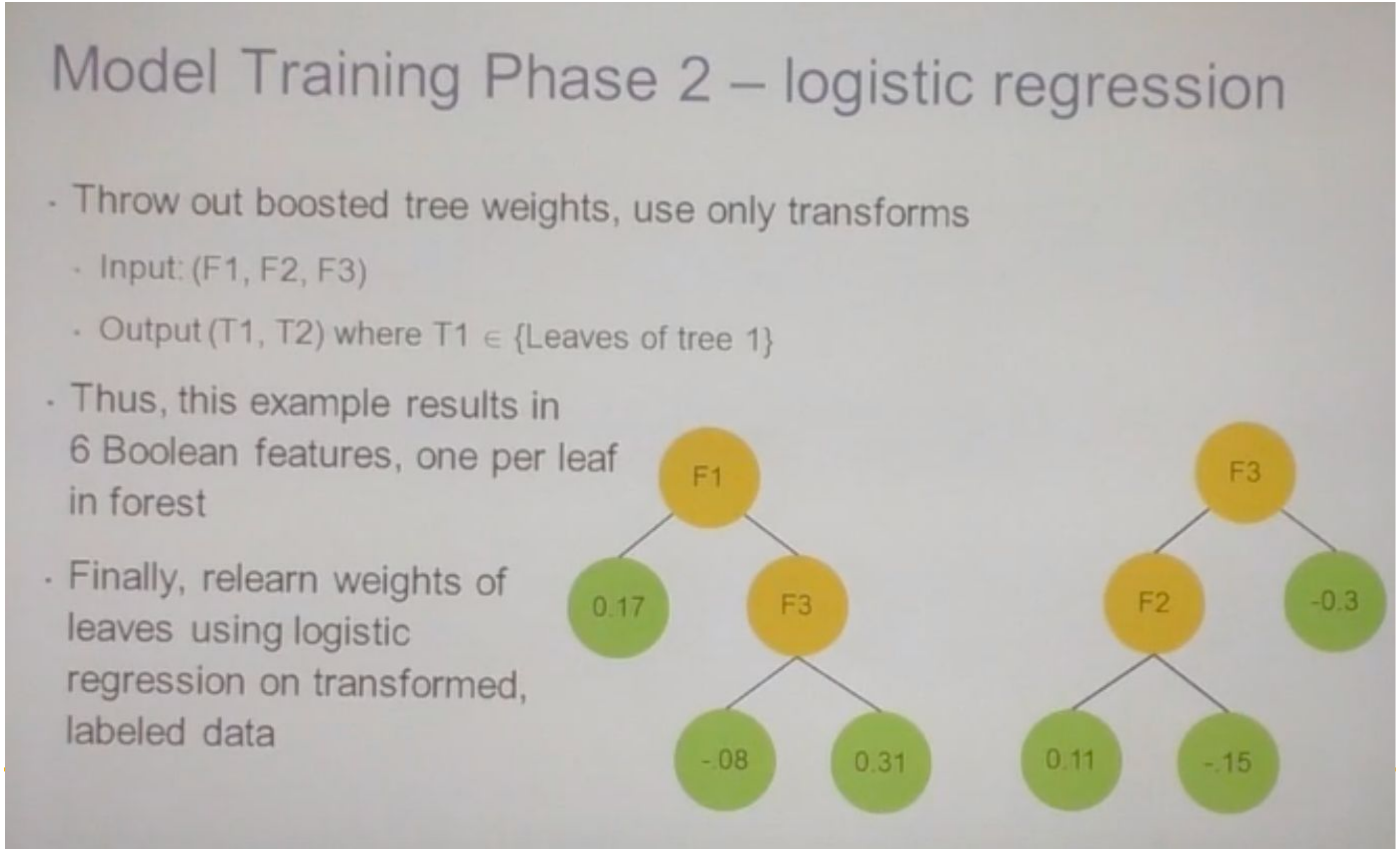
Machine learning的應用



Model Training Phase 2 – logistic regression

- Treat forest as feature transforms, turning input features into set of categorical features, one per tree.
- Use logistic regression to re-learn leaf weights

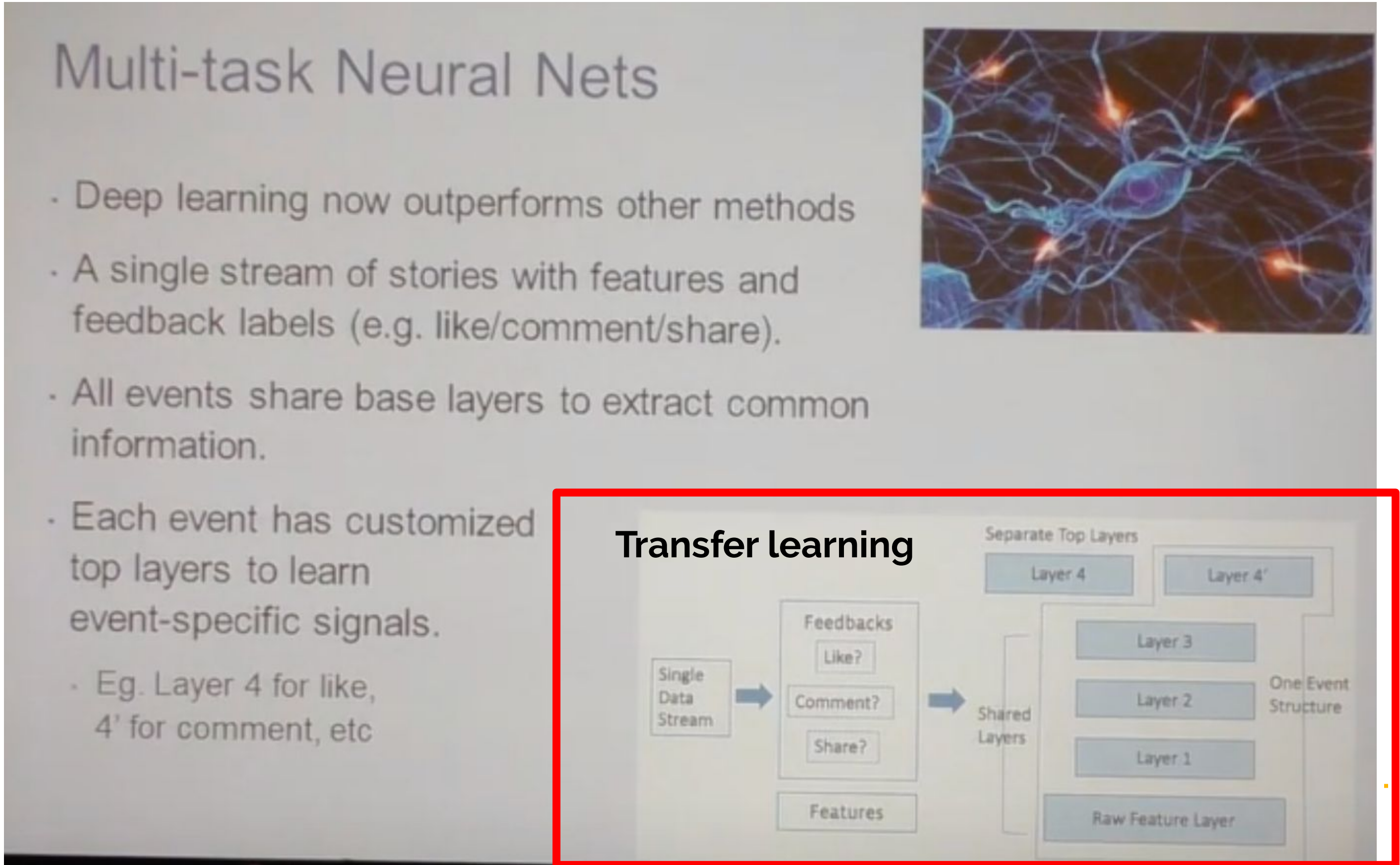
Machine learning的應用



Model Training Phase 2 – logistic regression

- Throw out boosted tree weights, use only transforms
  - Input: (F1, F2, F3)
  - Output (T1, T2) where T1 ∈ {Leaves of tree 1}
- Thus, this example results in 6 Boolean features, one per leaf in forest
- Finally, relearn weights of leaves using logistic regression on transformed, labeled data

Machine learning的應用



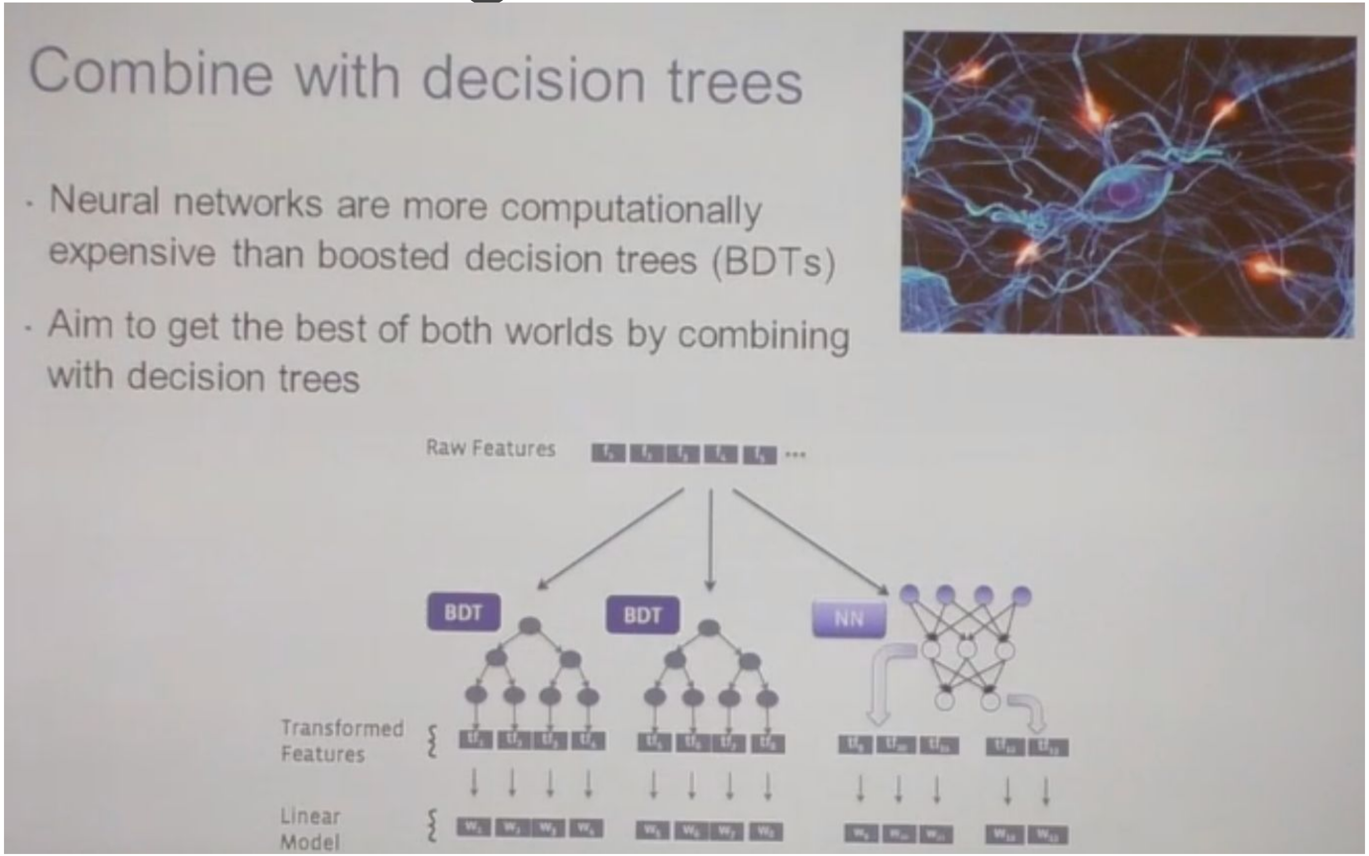
Multi-task Neural Nets

- Deep learning now outperforms other methods
- A single stream of stories with features and feedback labels (e.g. like/comment/share).
- All events share base layers to extract common information.
- Each event has customized top layers to learn event-specific signals.
  - Eg. Layer 4 for like, 4' for comment, etc

Transfer learning

Separate Top Layers
Layer 4
Layer 4'
Feedbacks
Like?
Comment?
Share?
Single Data Stream
Shared Layers
Layer 3
Layer 2
Layer 1
One Event Structure
Features
Raw Feature Layer

Machine learning的應用

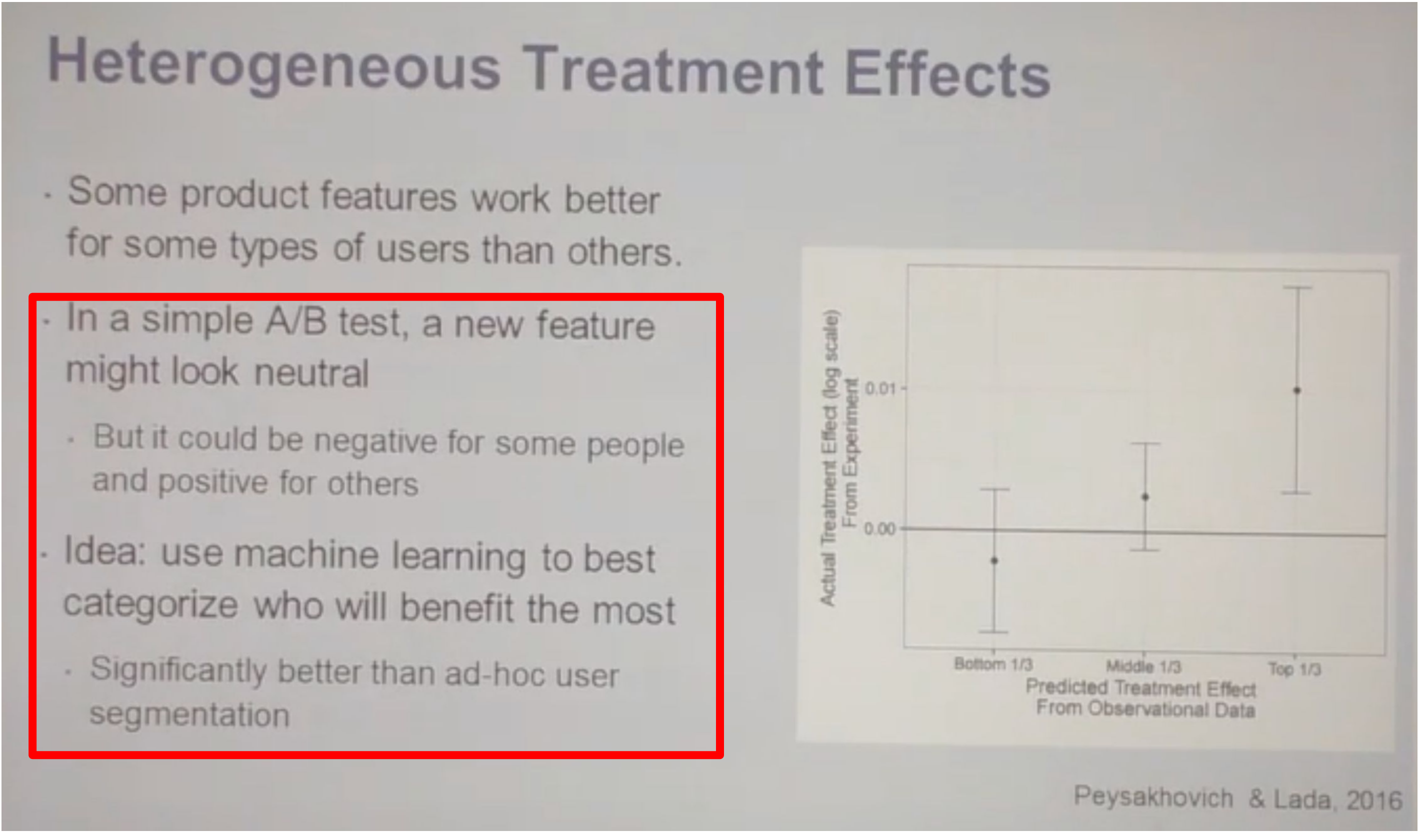


Combine with decision trees

- Neural networks are more computationally expensive than boosted decision trees (BDTs)
- Aim to get the best of both worlds by combining with decision trees

Raw Features
f_1 f_2 f_3 f_4 f_5 ...
BDT
BDT
NN
Transformed Features
tf_1 tf_2 tf_3 tf_4 tf_5 tf_6 tf_7 tf_8 tf_9 tf_{10} tf_{11} tf_{12} tf_{13}
Linear Model
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10} w_{11} w_{12} w_{13}

Machine learning的應用(Grouping)



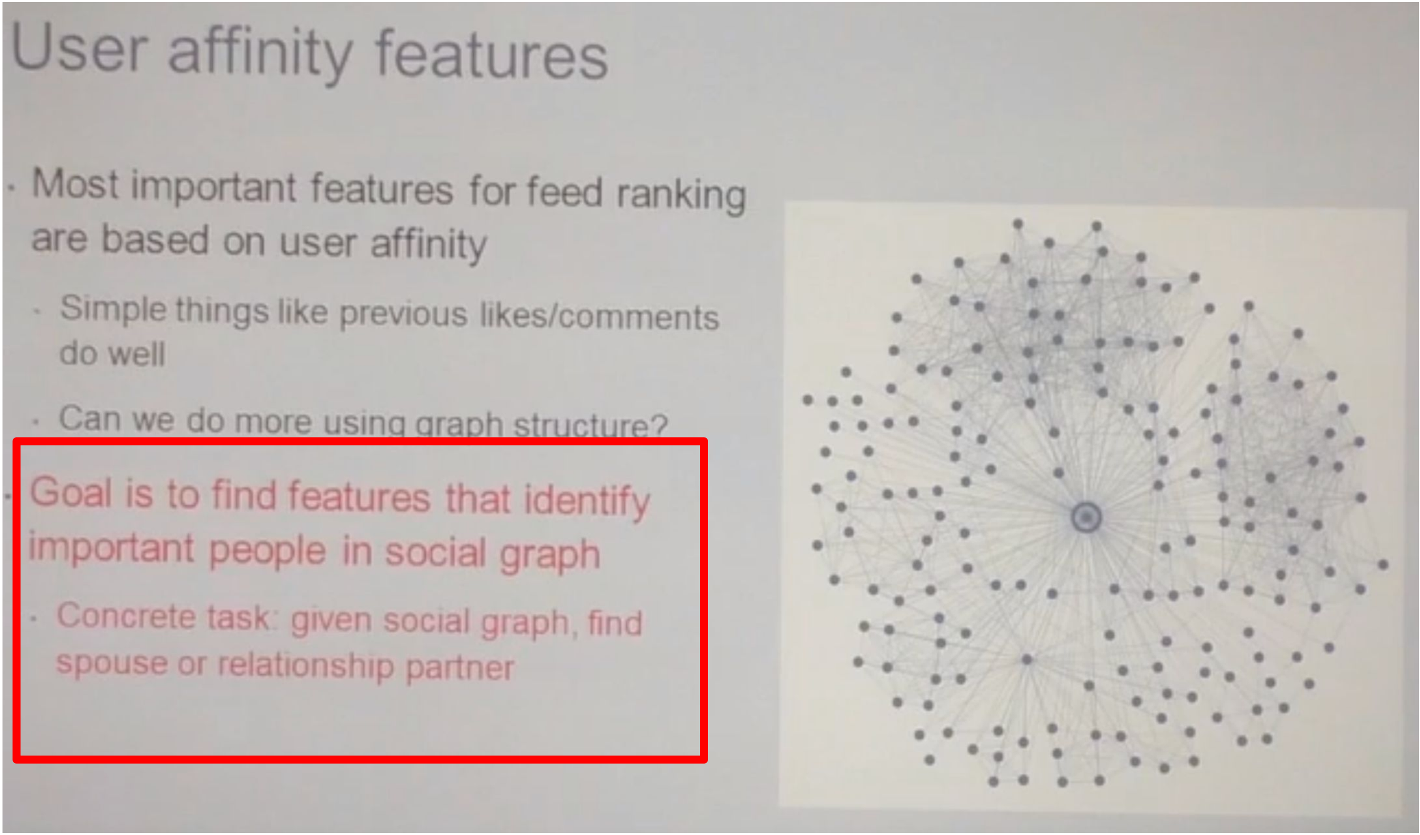
Heterogeneous Treatment Effects

- Some product features work better for some types of users than others.
- In a simple A/B test, a new feature might look neutral
  - But it could be negative for some people and positive for others
- Idea: use machine learning to best categorize who will benefit the most
  - Significantly better than ad-hoc user segmentation

Actual Treatment Effect (log scale)
From Experiment
0.01
0.00
Bottom 1/3
Middle 1/3
Top 1/3
Predicted Treatment Effect
From Observational Data

Peysakhovich & Lada, 2016

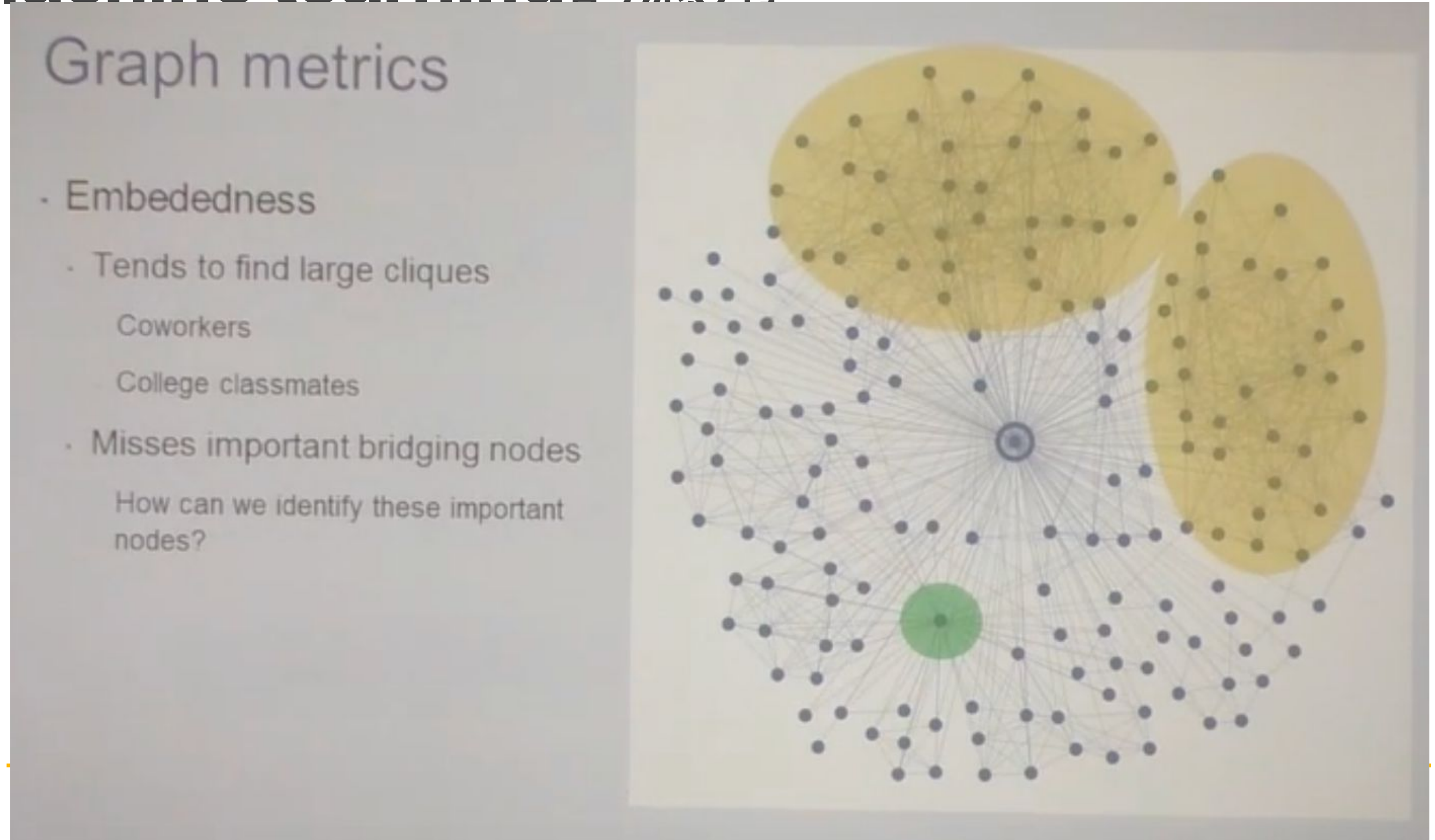
Machine learning的應用



User affinity features

- Most important features for feed ranking are based on user affinity
  - Simple things like previous likes/comments do well
  - Can we do more using graph structure?
- Goal is to find features that identify important people in social graph
  - Concrete task: given social graph, find spouse or relationship partner

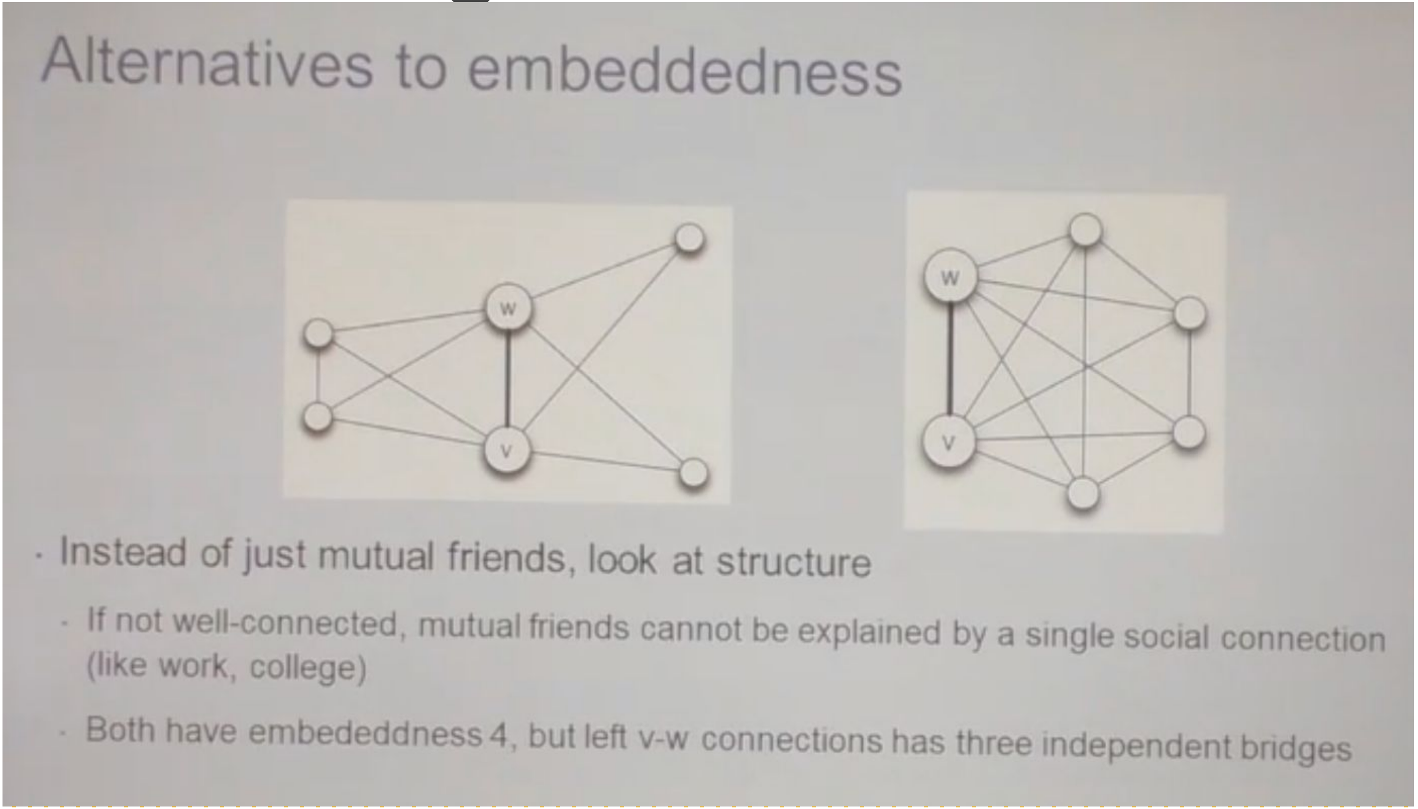
Machine learning的應用



Graph metrics

- Embededness
  - Tends to find large cliques
    - Coworkers
    - College classmates
  - Misses important bridging nodes
    - How can we identify these important nodes?

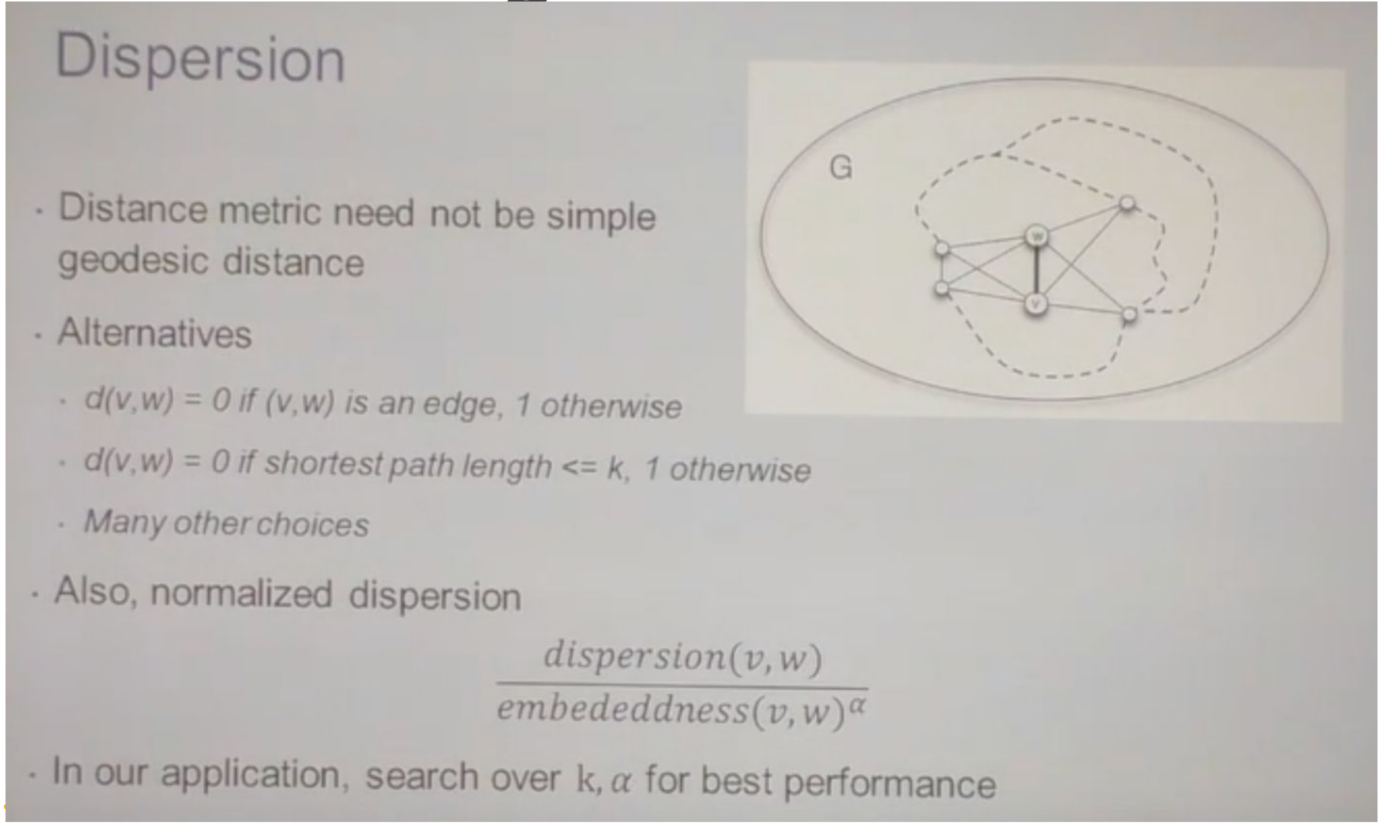
Machine learning的應用



Alternatives to embeddedness

- Instead of just mutual friends, look at structure
  - If not well-connected, mutual friends cannot be explained by a single social connection (like work, college)
  - Both have embededdness 4, but left v-w connections has three independent bridges

Machine learning的應用



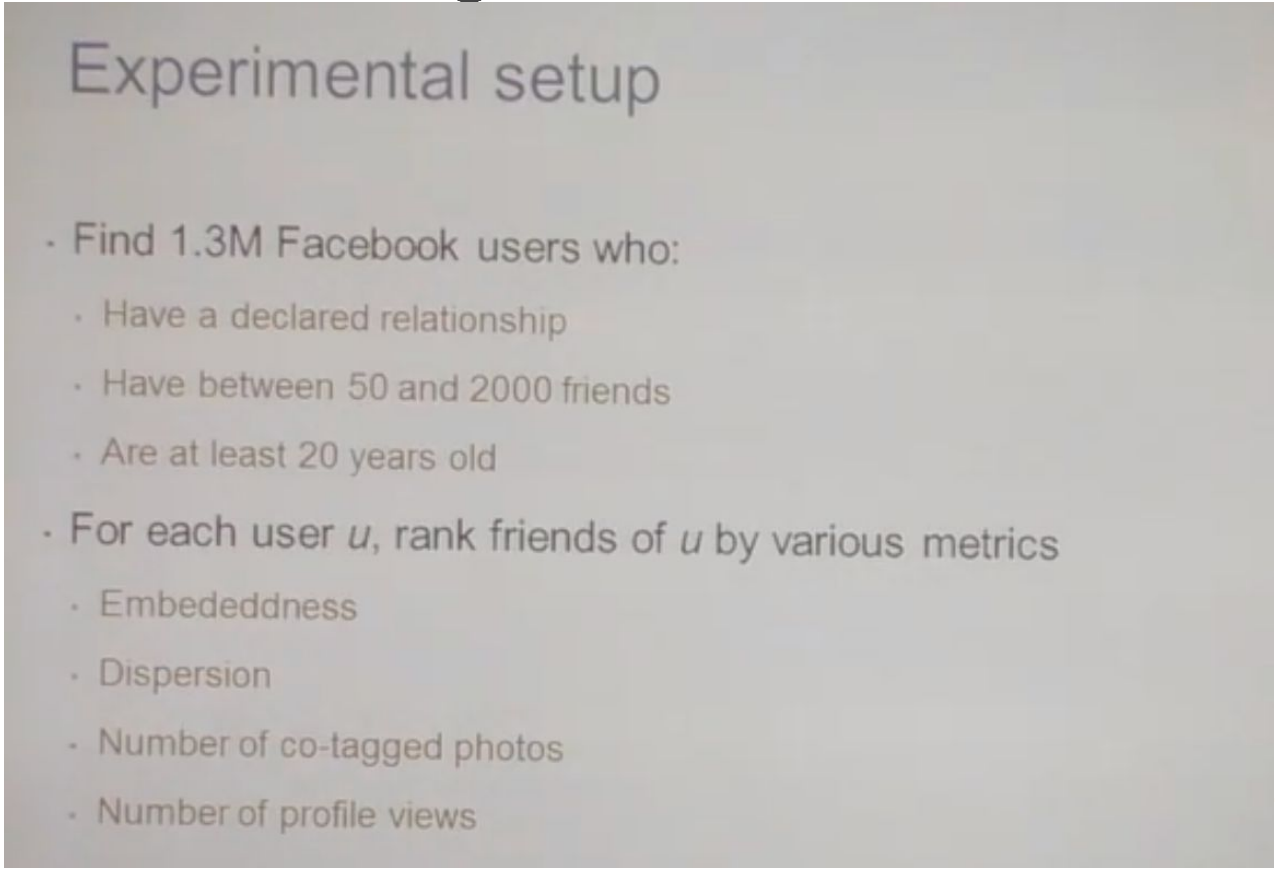
Dispersion

- Distance metric need not be simple geodesic distance
- Alternatives
  - *d(v,w) = 0 if (v,w) is an edge, 1 otherwise*
  - *d(v,w) = 0 if shortest path length <= k, 1 otherwise*
  - *Many other choices*
- Also, normalized dispersion

$$\frac{dispersion(v,w)}{embededdness(v,w)^{\alpha}}$$

- In our application, search over k, α for best performance

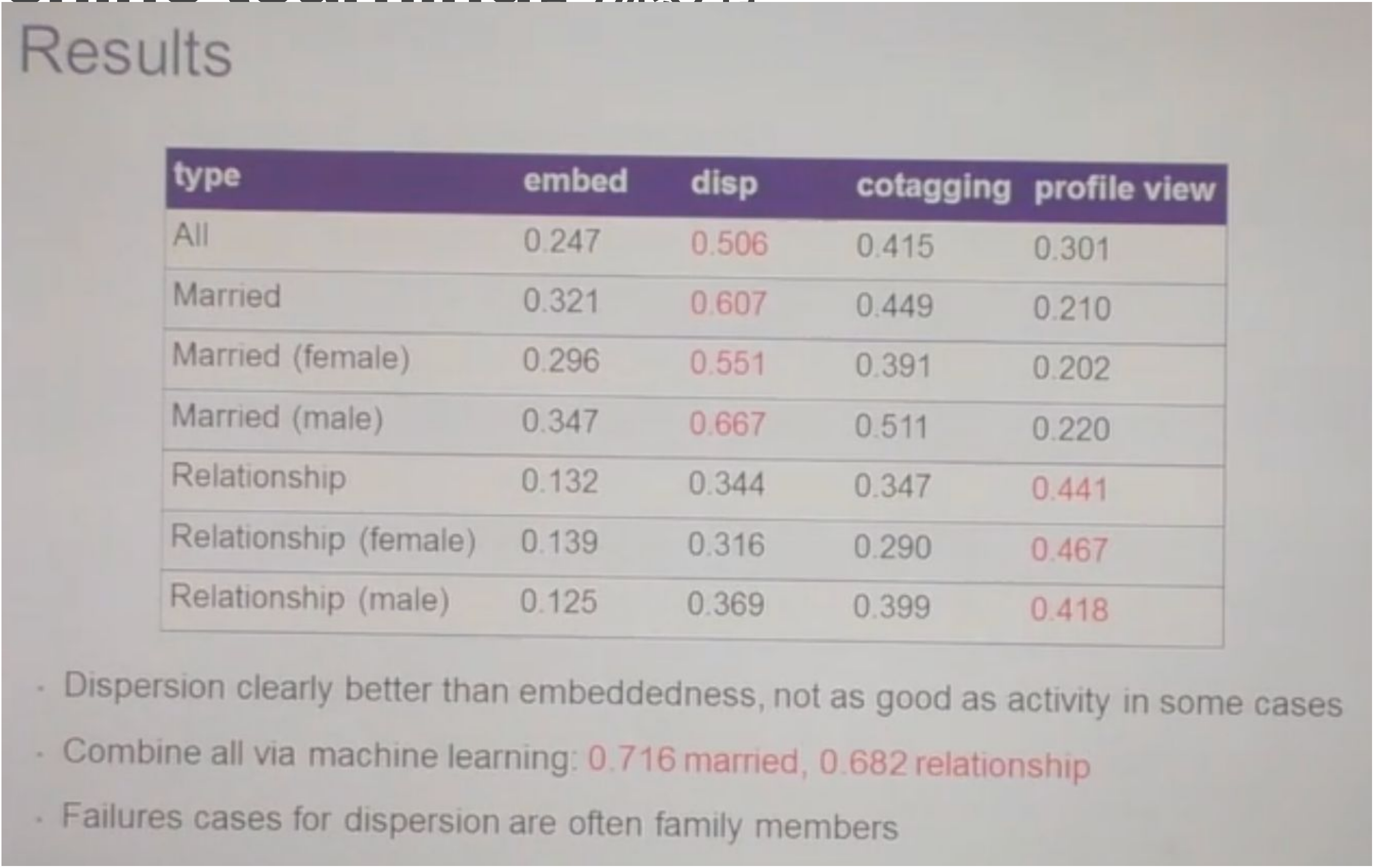
Machine learning的應用



Experimental setup

- Find 1.3M Facebook users who:
  - Have a declared relationship
  - Have between 50 and 2000 friends
  - Are at least 20 years old
- For each user *u*, rank friends of *u* by various metrics
  - Embededdness
  - Dispersion
  - Number of co-tagged photos
  - Number of profile views

Machine learning的應用

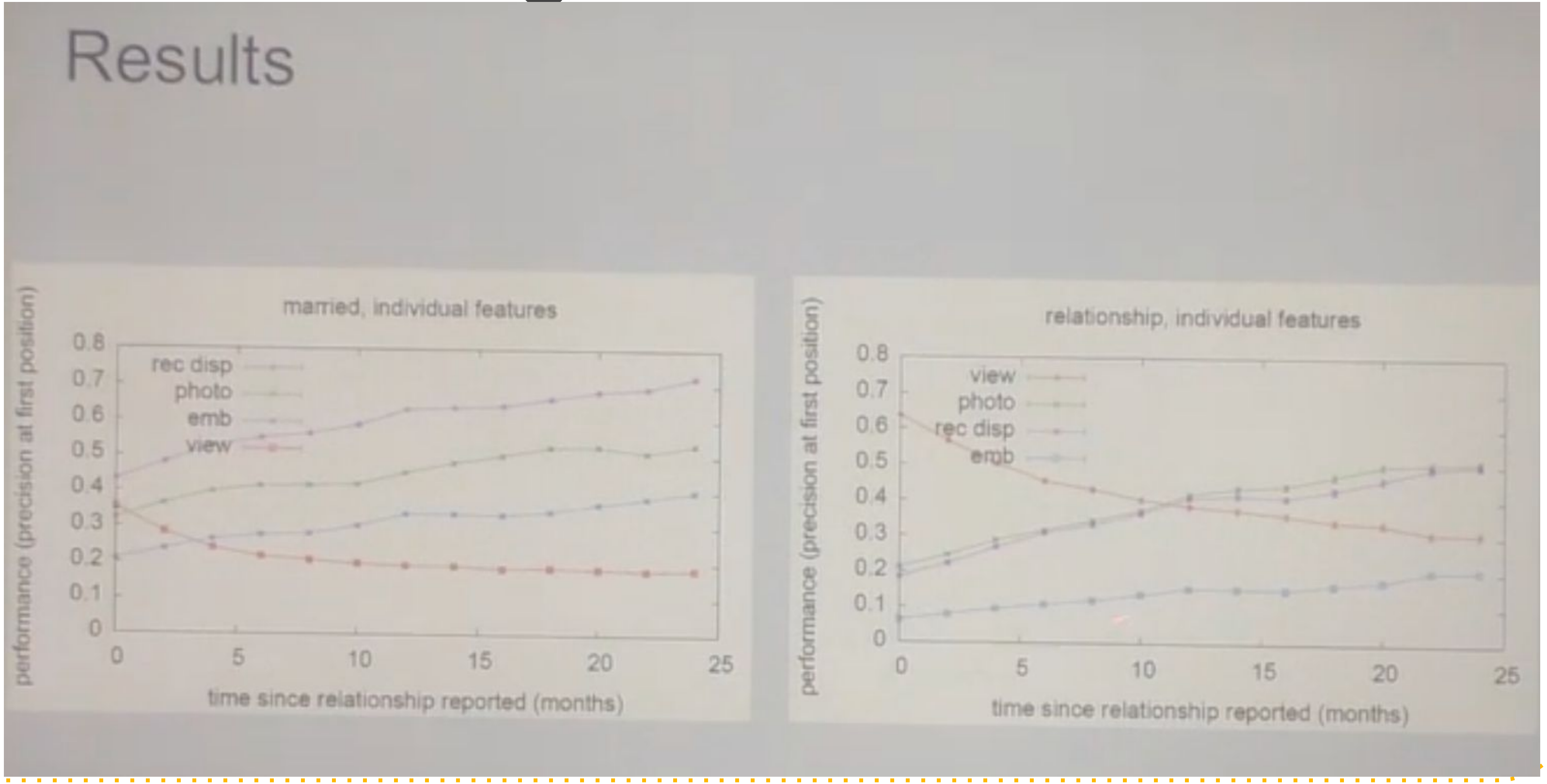


Results

type	embed	disp	cotagging	profile view
All	0.247	0.506	0.415	0.301
Married	0.321	0.607	0.449	0.210
Married (female)	0.296	0.551	0.391	0.202
Married (male)	0.347	0.667	0.511	0.220
Relationship	0.132	0.344	0.347	0.441
Relationship (female)	0.139	0.316	0.290	0.467
Relationship (male)	0.125	0.369	0.399	0.418

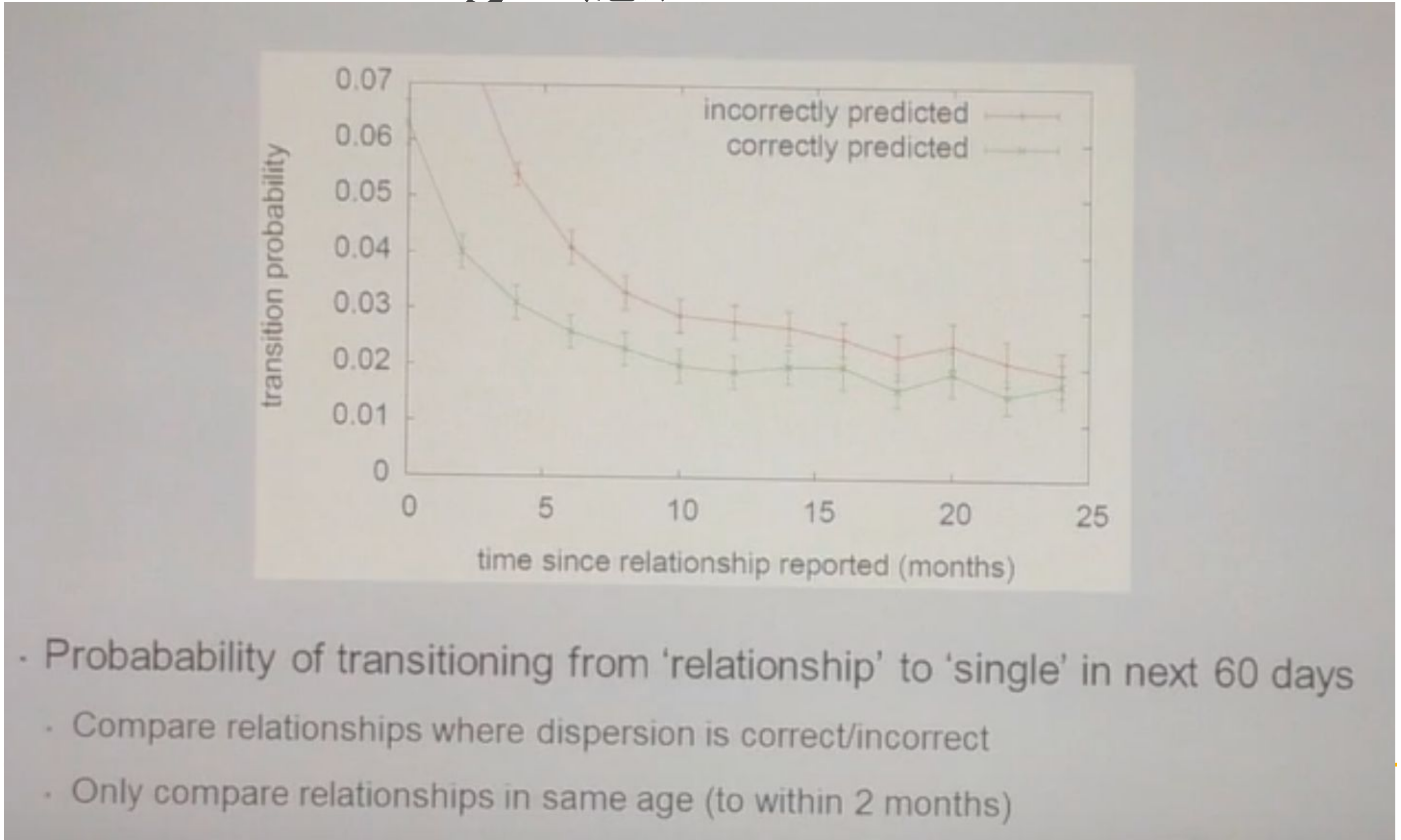
- Dispersion clearly better than embeddedness, not as good as activity in some cases
- Combine all via machine learning: 0.716 married, 0.682 relationship
- Failures cases for dispersion are often family members

Machine learning的應用



Results

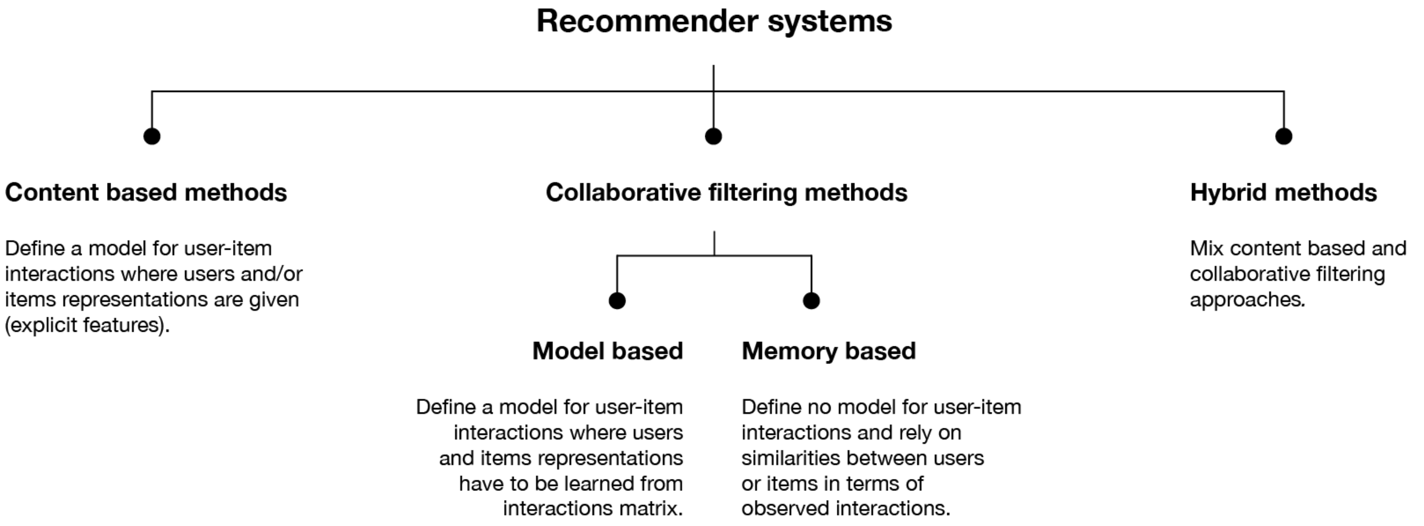
Machine learning的應用



- Probabability of transitioning from ‘relationship’ to ‘single’ in next 60 days
  - Compare relationships where dispersion is correct/incorrect
  - Only compare relationships in same age (to within 2 months)

推薦系統整體介紹

推薦系統概要



Recommender systems

Content based methods

Define a model for user-item interactions where users and/or items representations are given (explicit features).

Collaborative filtering methods

Model based

Define a model for user-item interactions where users and items representations have to be learned from interactions matrix.

Memory based

Define no model for user-item interactions and rely on similarities between users or items in terms of observed interactions.

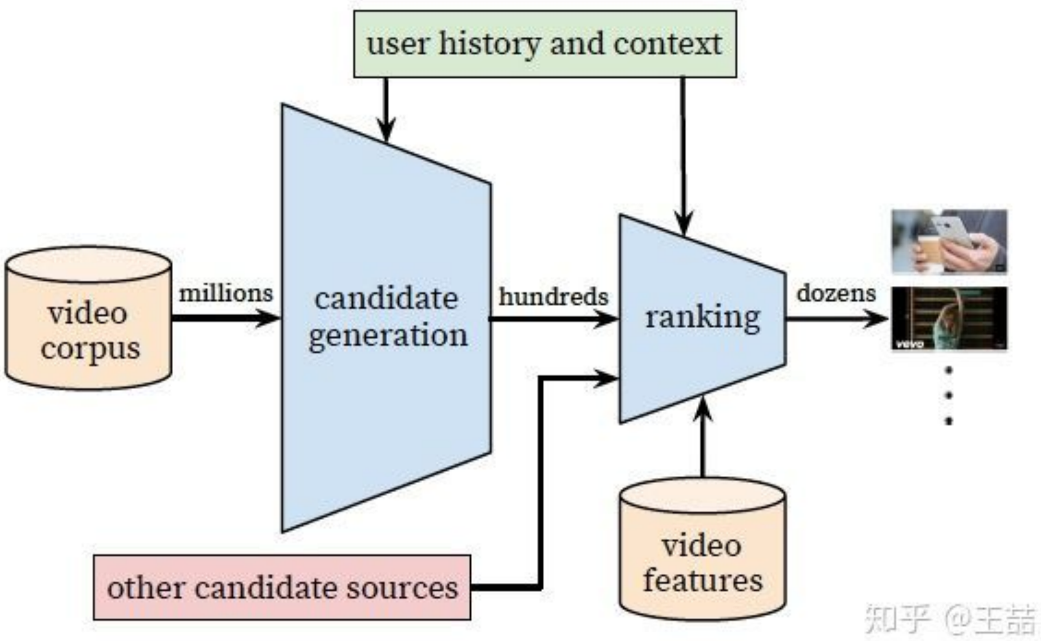
Hybrid methods

Mix content based and collaborative filtering approaches.

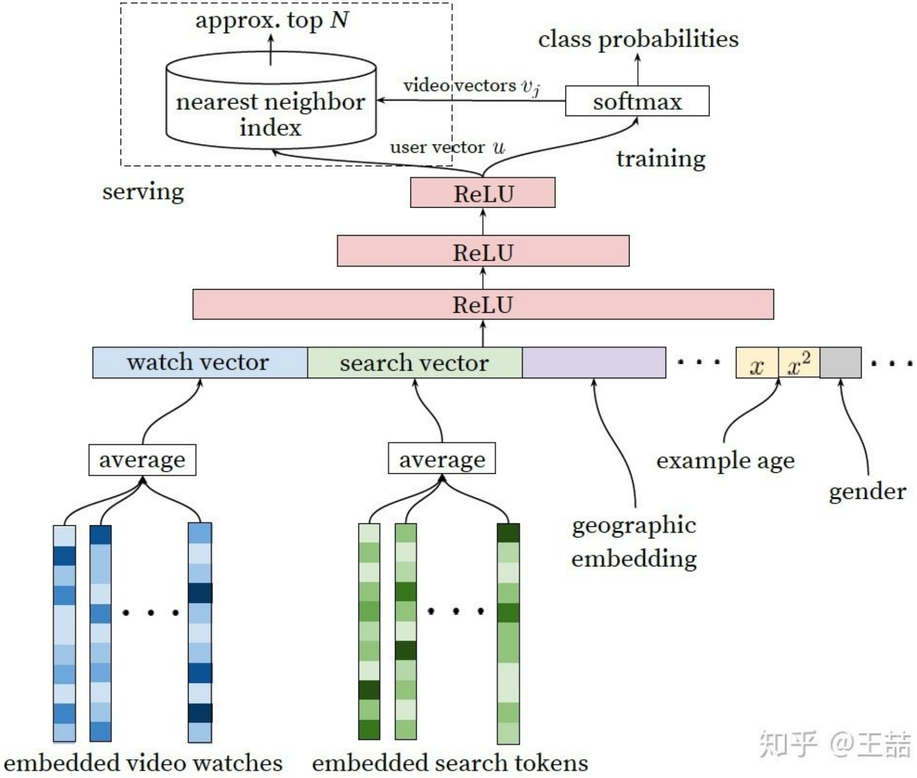
Youtube 推薦系統

https://zhuanlan.zhihu.com/p/52169807

推薦系統業界標準架構


user history and context
video corpus
millions
candidate generation
hundreds
ranking
dozens
other candidate sources
video features
vevo

candidate generation


approx. top N
nearest neighbor index
video vectors v_j
user vector u
class probabilities
softmax
training
serving
ReLU
ReLU
ReLU
watch vector
search vector
x
x^2
average
average
geographic embedding
example age
gender
embedded video watches
embedded search tokens