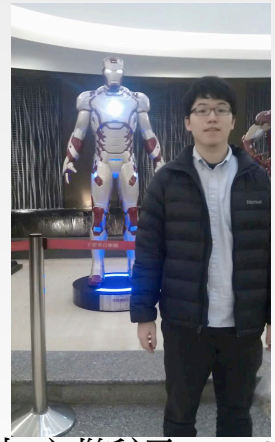
Indexing and Clustering Spamming Botnet using Lucene for Big Data Analysis

應用Lucene對於Spamming Botnet進行快速的索引與分群

Ching-Hao, Eric, Mao Ph. D.,毛敬豪
Institute for Information Industry, 財團法人資訊工業策進會
Cloud Security Alliance Taiwan Chapter, 雲端資安聯盟台灣分會

Contributors

林昶丞 (Spamming Botnet)



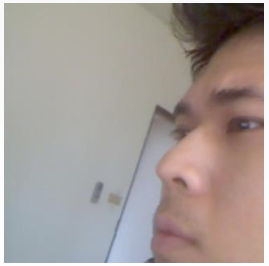
許懷文
(Lucene中文斷詞, Arch)

詹勝宇 Indexing

吳尚遇 同義詞分析

王志群
UI JQuery

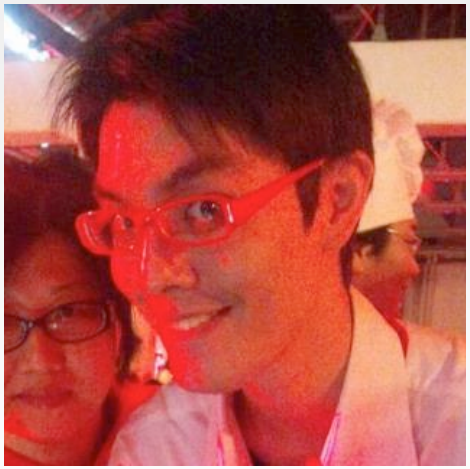
林于安 & 林譽 FB Social Networking analysis



About Speaker

- 財團法人資訊工業策進會，資安科技研究所
- 雲端資安聯盟台灣分會
- 國立台灣科技大學，資訊工程所博士
- 美國卡內基梅隆大學，電腦科學系，訪問學者

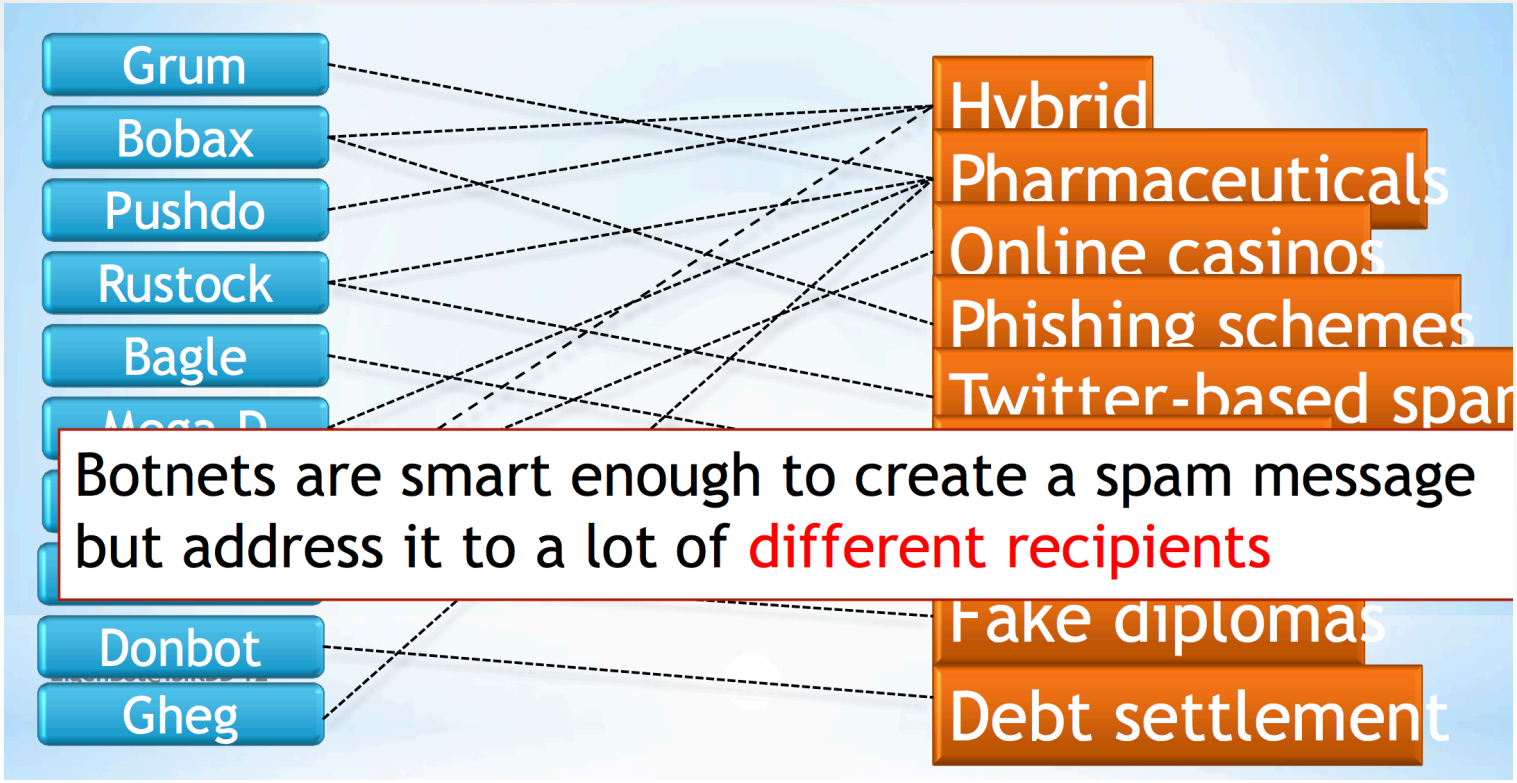
Acknowledge



Outline

- Problem Description
- Concept
- Lucene Speed Up
- Big Data Analysis- Mahout and Pegasus
- Extension: Social Network Analysis
- Conclusions

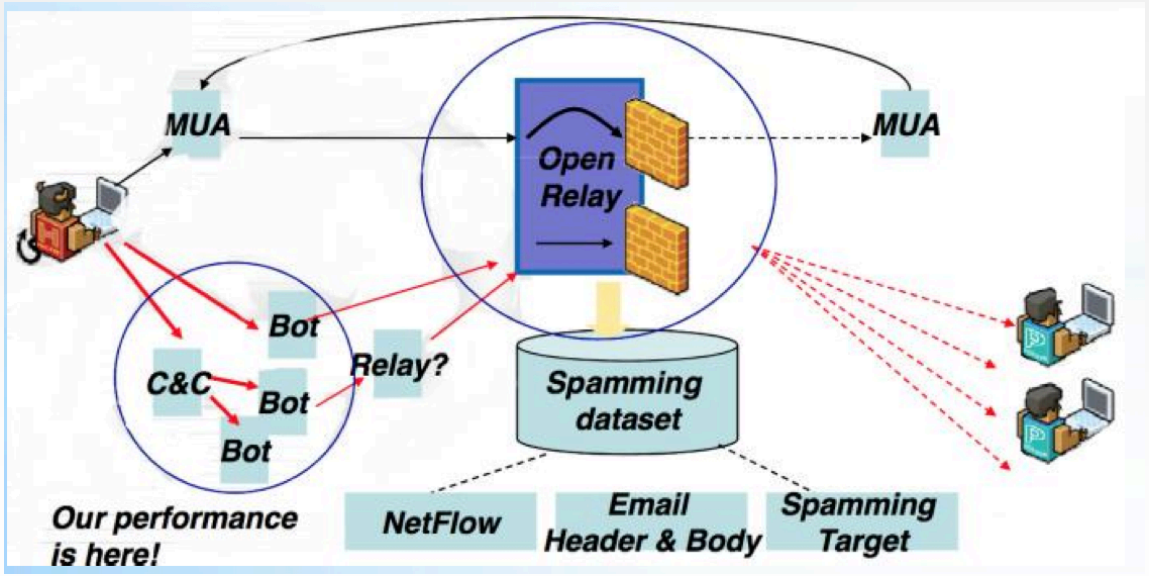
Problem Description



Problem Description- HoneySpam

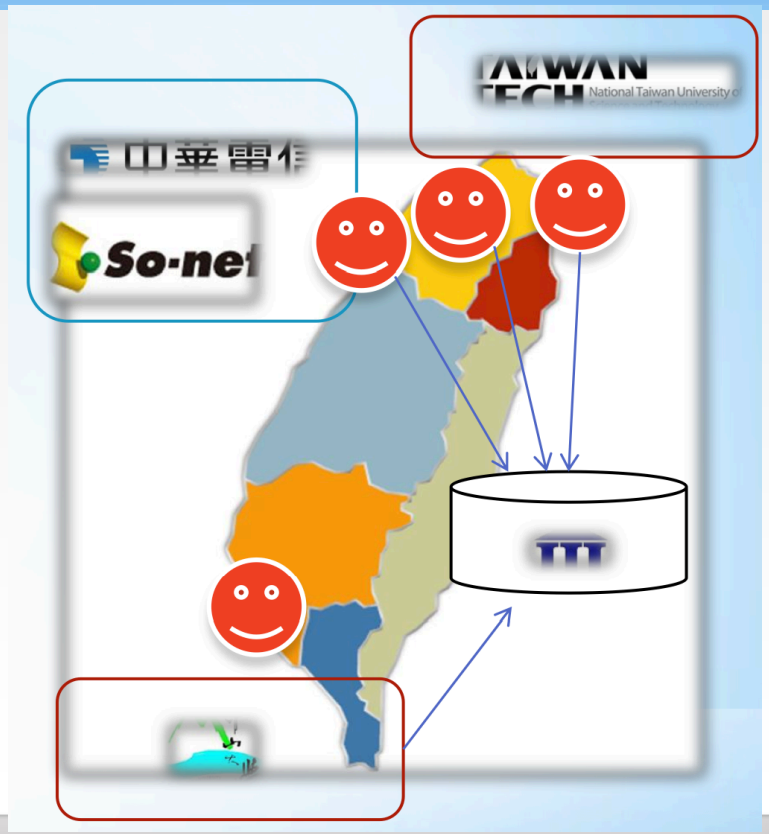
HoneySpam intercept them and does analysis

- Why Spam pass through HoneySpam?
- Heuristic with pattern filtering • E.g., XX_140.ooo.20.31



HoneySpam Deployment

Deployments
*Taiwan Internet Service Provider, Universities...
*Since Aug. 2010
*60 millions + spams
*30,000 + IPs (Bots)
*48/52 (Eng./Chinese)
*1 millions per day

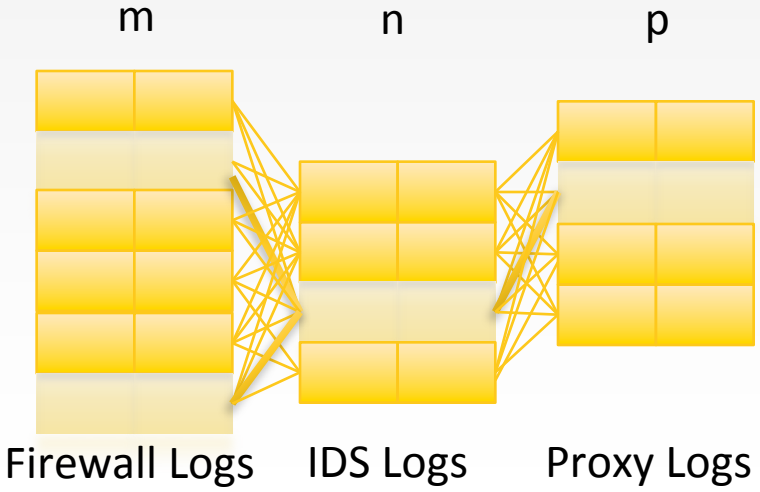


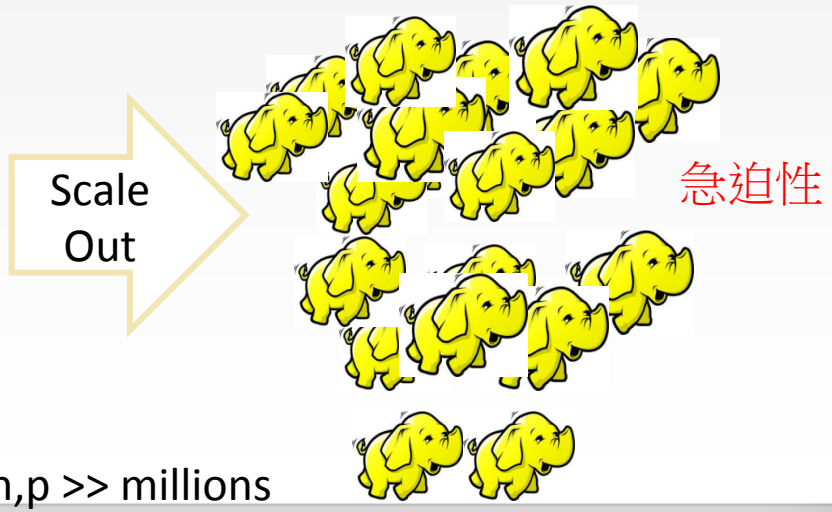
Challenges in Information Security

- **Harsh real time requirements:** Can't analyze the context of all activities (Scale-out can help!)
- **Specificity of detection signatures:** Hard to balance between an overly specific signature and an overly general one (Heterogeneous data analytics can help!)
- **True attacks are rare events**: Looking for a needle in a haystack. So much hay and so little time (Dealing with unbalanced data)
- **Lack of Data Lifecycle Footprint:** Hard to build a footprint among big data (Footprint representation can help!)
- **Difficult in Privacy Leakage Forensics:** (Evidence analytics can help!)

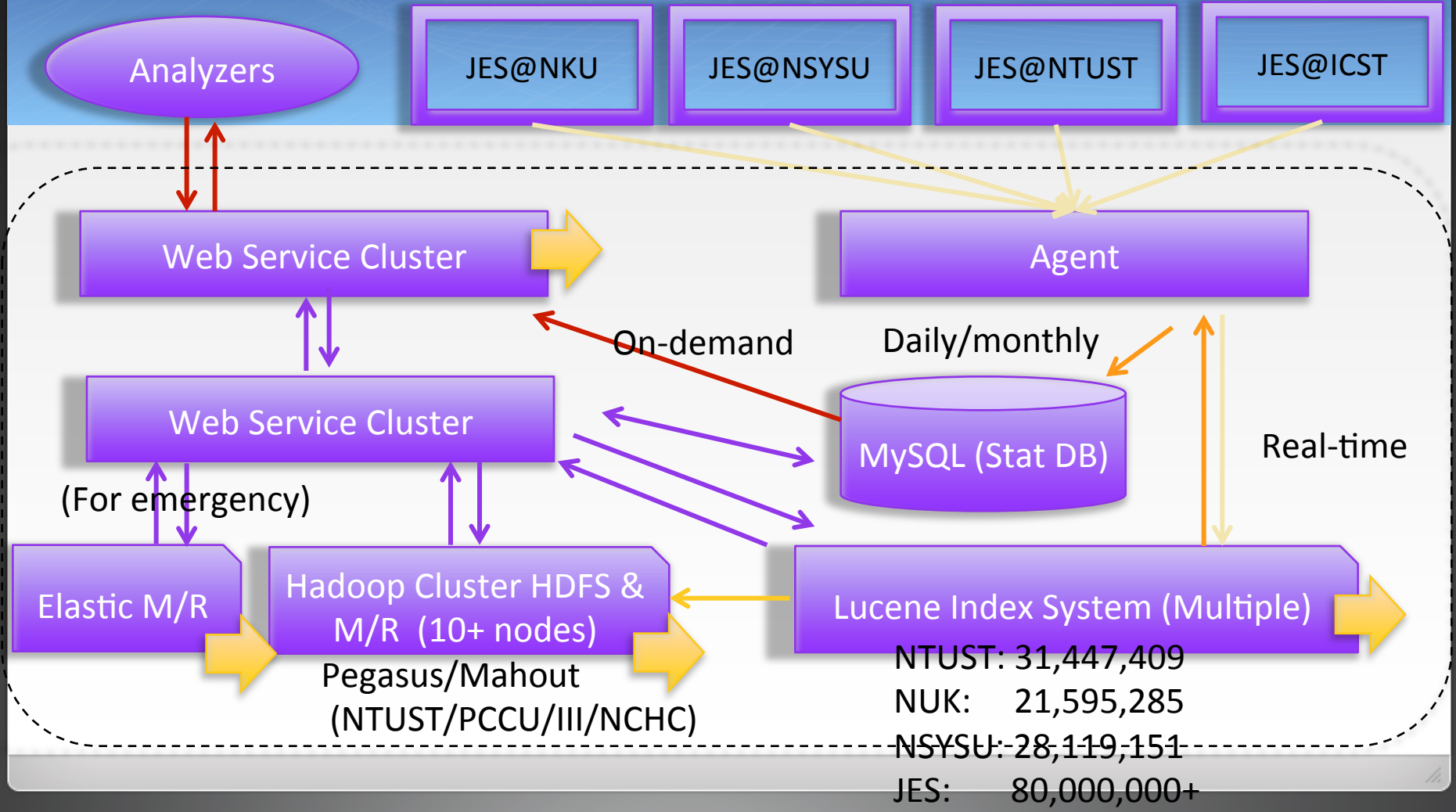
資安監控與巨量資料分析

- 事件追蹤如求解最短路徑，尤其資安遇到的資料多元且大量，時間複雜度極高
- 時效性對於資安事件極為重要，如：情蒐、追蹤、鑑識及事件處理



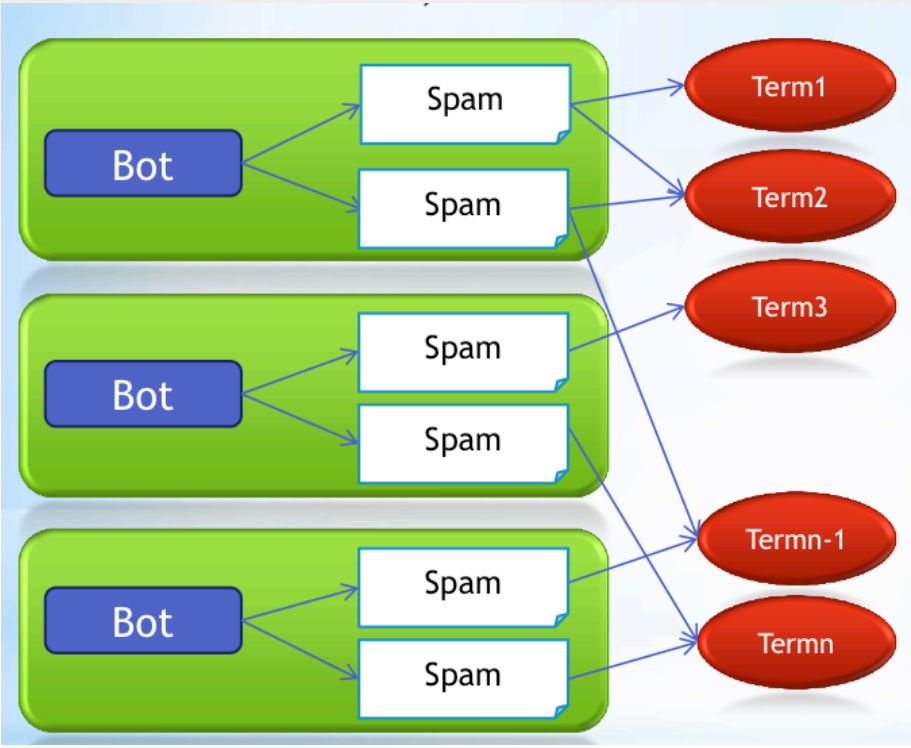


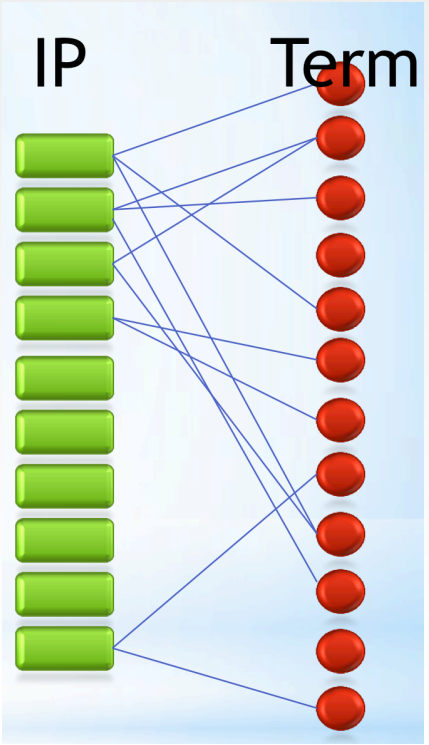
Ecosystem in EigenBot



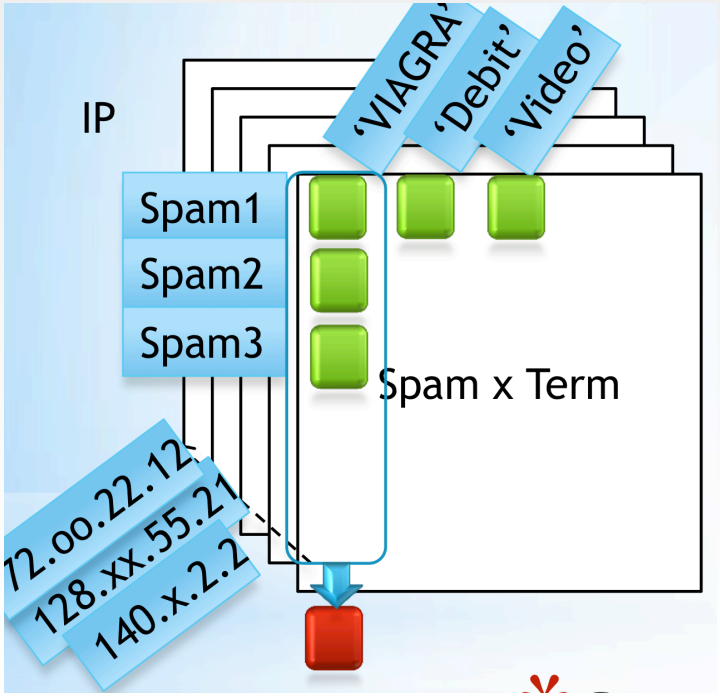
垃圾郵件殭屍電腦語意分群

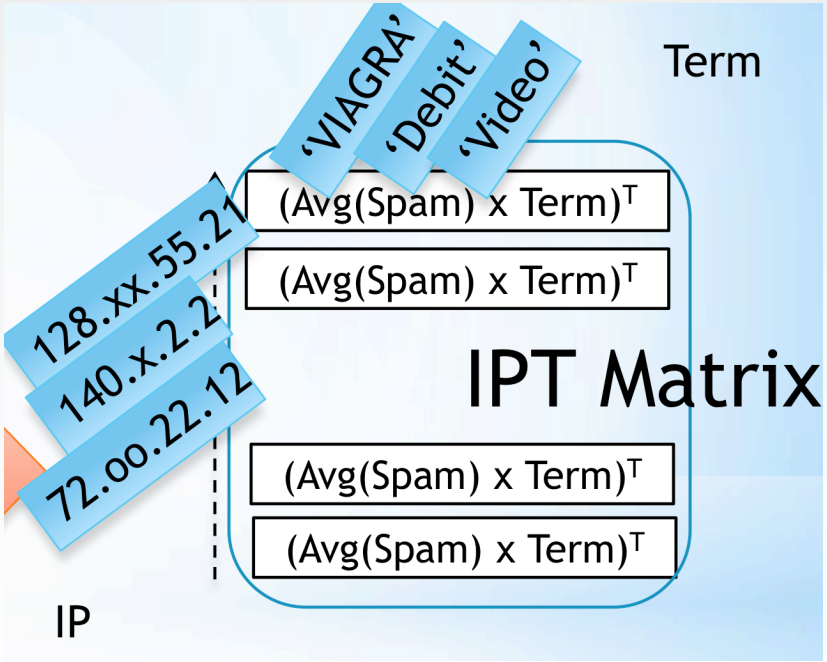
依來源發送之語意進行分群EigenBot





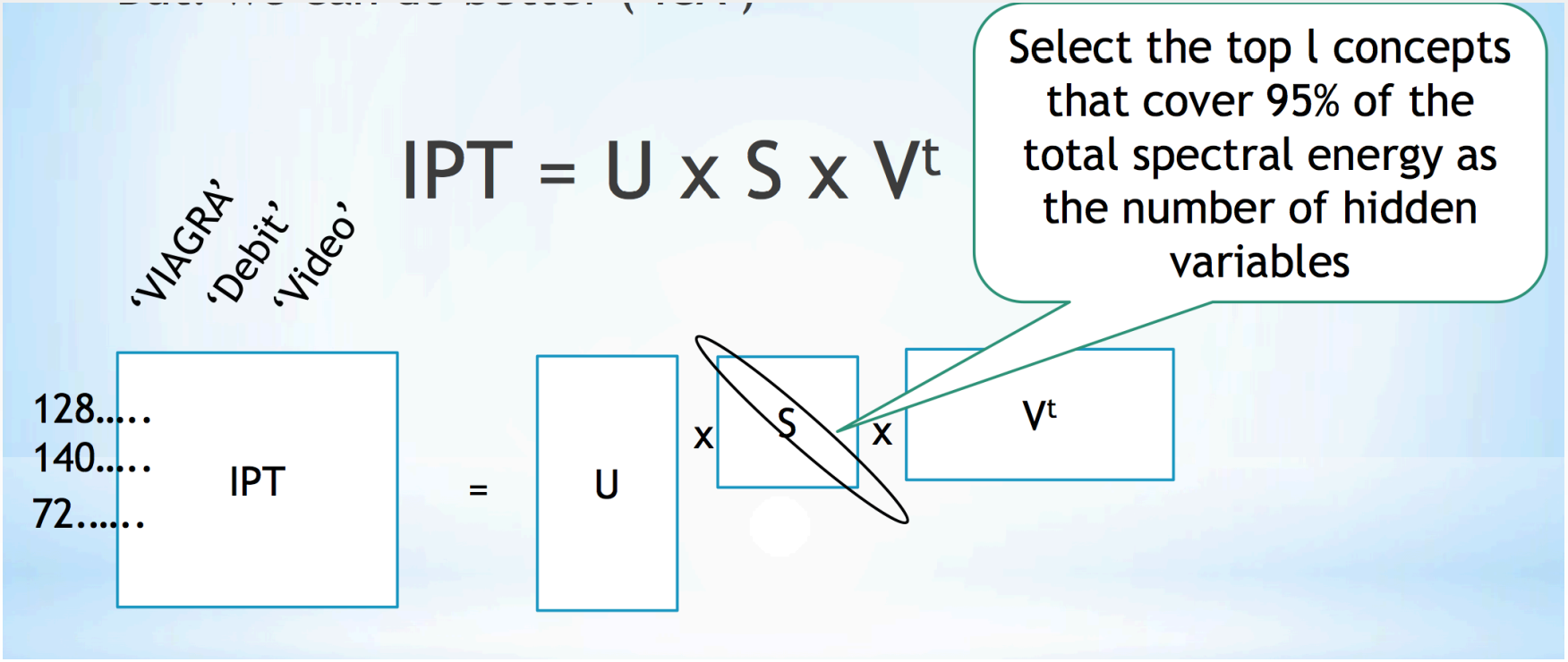
EigenBot Detail





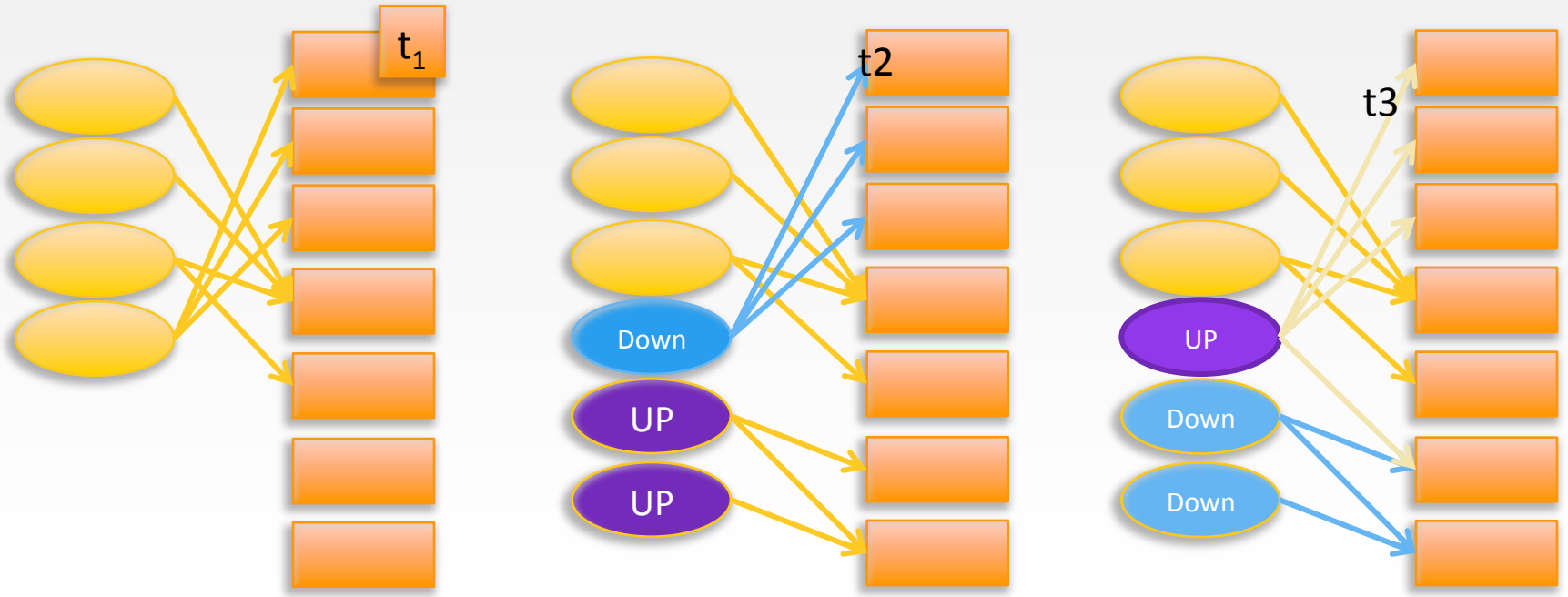
Lucene

Latent Sapmming Intention Finding



Tensor in Spam

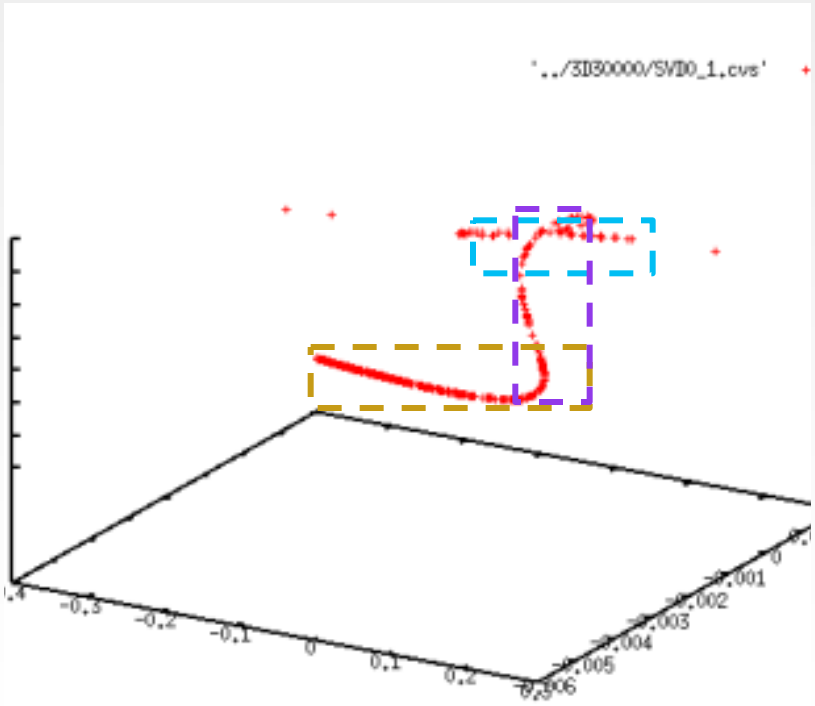
- Maintain ISP anti-spam blocklist



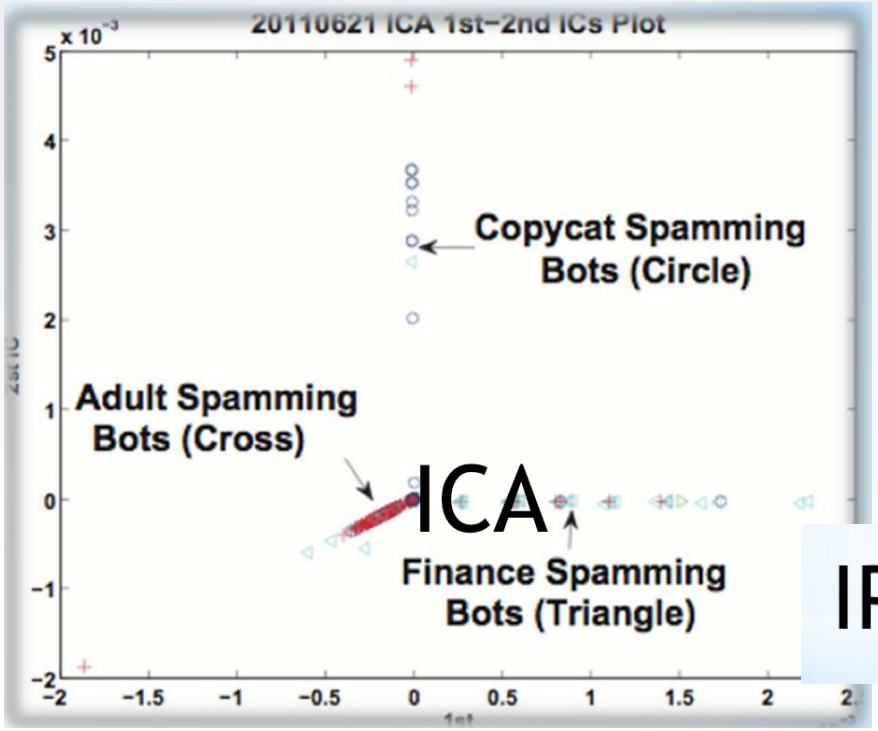
Big Data Visualization in Security

- 可快速將大量攻擊資料透過Big Data投影技術，找出資料關聯
- 群聚資料找出中心，省略逐筆比對之運算
- 實際案例：

1. 30000個攻擊來源IP
2. 藉由Big Data分散式索引技術萃取出特徵
3. 透過Big Data的資料投影演算法投射至3D空間
4. 透過Big Data群聚技術，加速攻擊特徵之比對



使用ICA分解

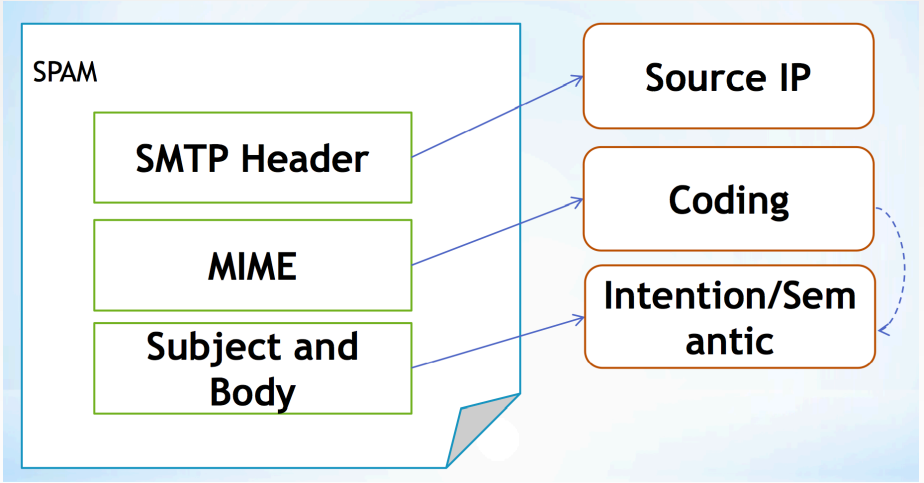


兩天的資料，
已經快算不動了

$$IPT_{[m\times n]} = H_{[m\times l]} \times B_{[l\times n]}$$

Lucene Speed Up

- Decoding
  - SMTP decoding: BASE 64 and quoted-printable
- NLP Segmentation
  - Chinese segmentation: CKIP systems (http://ckipsvr.iis.sinica.edu.tw/)
- Indexing
  - Extracting and weighting the terms via Lucene (http://lucene.apache.org/)



常會遇到的議題

- Lucene是斷詞系統？資料庫？還是一個索引系統？
- 如何擴展Lucene的同義詞庫
- 到底Lucene能否有Distinct? Lucene能否有Group By?速度會比SQL語法快嗎？
- 巨量資料的Lucene Index應該如何處理？
- 為何Lucene是進入巨量資料的一個碁石？

中文分詞器 Lucene

StandardAnalyzer	一元分詞演算	今,天,天,氣,好,棒
CJKAnalyzer	二元分詞 交叉雙字分割法	今天,天天,天氣, 氣好,好棒
ChineseAnalyzer	針對 LOWERCASE_LETTER UPPERCASE_LETTER 處理	今,天,天,氣,好,棒
IK_CAnalyzer	字典,辭典基礎 雙向搜索 文法分析	Lucene project 為主 轉向Java公用套 件

中文分詞器 Third party

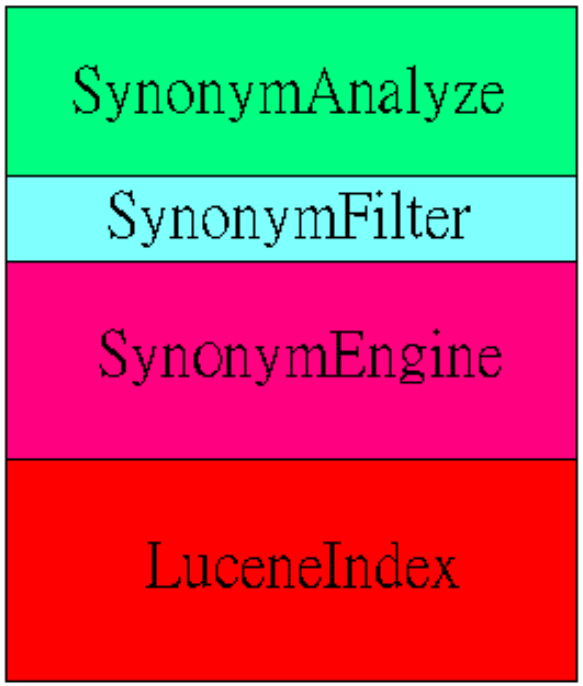
mmseg4j	正向最大匹配 模糊解析規則	UTF8編碼,自訂辭典,高速高準確率,無義詞去除
CKIP	字典 詞頻詞性分類 領域用詞詞庫	廣義知網 語意剖析系統
庖丁解牛 PoadingAnalyzer	辭典中文隱喻詞 全面向文章切分	詞類定義分類 中文語義處理較繁瑣
NLPIR(ICTCLAS)	自然語言處理 辭典自訂義	2013版開放各式編碼(原僅GB) 微博分詞 (社群新詞發現)

http://www.chmao.idv.tw/CKIPChineseAnalyzer.html

同義詞的挑戰-苦工級的任務（SynonymAnalyzer）

- 它的目標首先是檢測具有同義詞的單詞，然後在同一位置插入對應的同義詞。

```
2 : [quick][speedy][fast]
3 : [brown]
4 : [fox][狐狸]
5 : [jumps][hops][leaps]
6 : [over][above]
8 : [lazy][sluggish][apathetic]
9 : [dog][pooch][canine]
10 : [hi][嗨][hello]
```



SynonymEngine

- 包括新增、刪除等許多管理方法都必須在此實作。


```
engine.addWord("幹");
engine.addSynonym("幹", "不爽");
engine.addSynonym("幹", "生氣");

engine.removeWord("雪特");
engine.addSynonym("雪特","shit");

engine.removeSynonym("幹", "生氣");

engine.getPhaseDB();
```

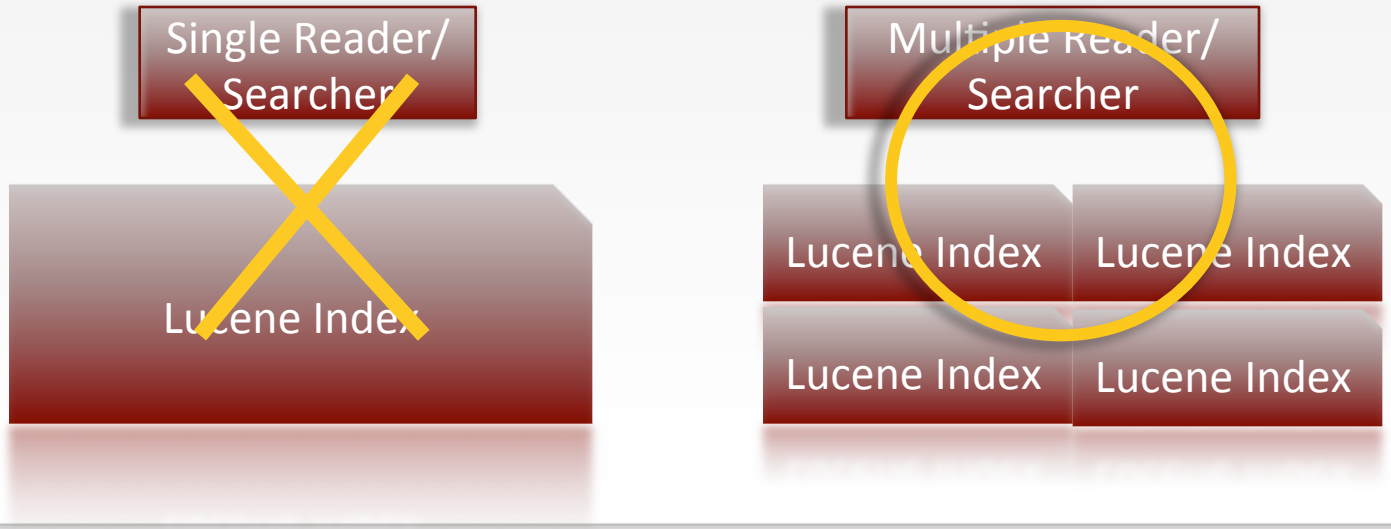
```
successful adds word "幹".

successful adds synonym "不爽".

successful adds synonym "生氣".
word "雪特" is not exist.
successful adds synonym "shit".
successful remove synonym "生氣".
0: word:[jumps] synonym:[跳]
1: word:[雪特] synonym:[shit]
2: word:[幹] synonym:[,不爽]
```

巨量資料的Lucene Index應該如何處理?

- Lucene Index performs not well when index files growing up to Gigabytes
- Multi-threads and multiple readers are required



巨量資料的Lucene Index應該如何處理?

- MultiReader + IndexSearcher?

或

- IndexReader + MultiSearcher (Deprecated-Lucene 2756)?

If you are using MultiSearcher over IndexSearchers, please use MultiReader instead; this class does not properly handle certain kinds of queries

The fundamental problem is that MultiSearcher first rewrites against individual subs, then uses Query.combine() which simply OR's these sub-clauses.

非(P 且 Q)＝(非 P)或(非 Q)
非(P 或 Q)＝(非 P)且(非 Q)

MUST-NOT 違反迪摩根定理

Lucene如何尋找Finite Set與聚合函數？

- Lucene如何達到Distinct的目的？
  - (Straightforward!!) 透過”for-each” 繞過所有的資料，並存在Set…
    - 但是，當有1億3千萬筆資料時？（找出所有的FiniteSet需花費許多時間…）
    - 為何Luke可以做得這麼快？
  - 透過Reader取出terms…

Subject	IP
I, Love, Hadoop	1.1.1.1
Hadoop, Love, me	2.2.1.1
Mahout, Love, Hadoop	1.1.1.1
…	…

1.1.1.1
2.2.1.1

I, Love, Hadoop, me, Mahout

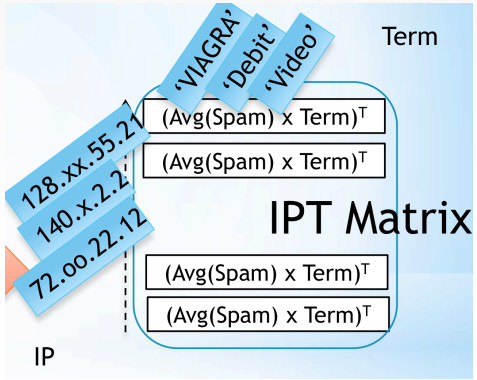
快速的實作Distinct

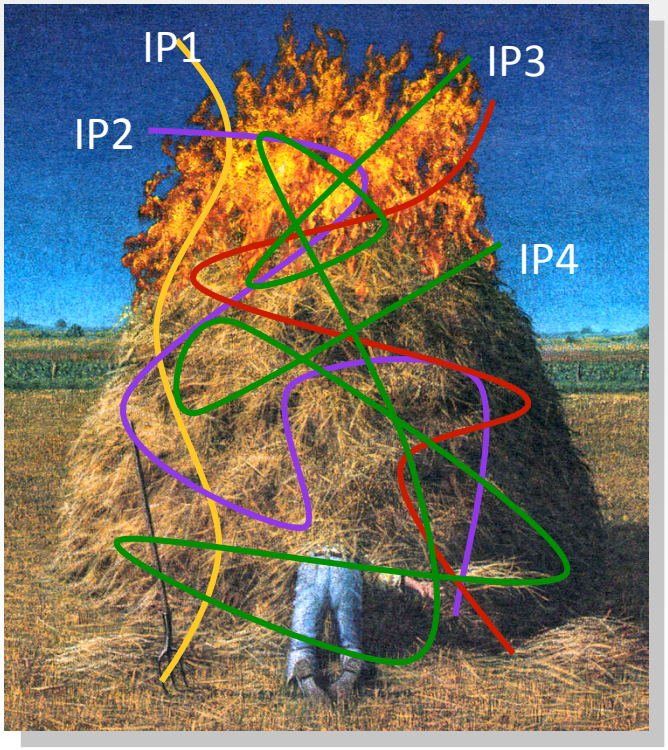
```
TermEnum termEnumSubject = multiReader.terms(new Term("subject", ""));
Term next1 = termEnumSubject.term();
while (next1 != null && next1.field().equals("subject")) {
        next1 = termEnumSubject.next() ? termEnumSubject.term() : null;
        String token = next1.toString().replaceAll("subject:", "”);
}
```

至於Group By… 要用Facet的概念了，超出了範圍了~

為何Lucene對於巨量資料分析是有幫助的？Needle in a Haystack …

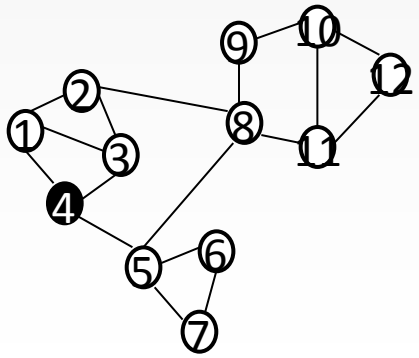
- 快速關聯特定已知特徵的所有行為序列
- 特別對於文字與語意方面的資料特性
- Lucene也是Hadoop Ecosystem的一員
  - 高度與Nutch相關



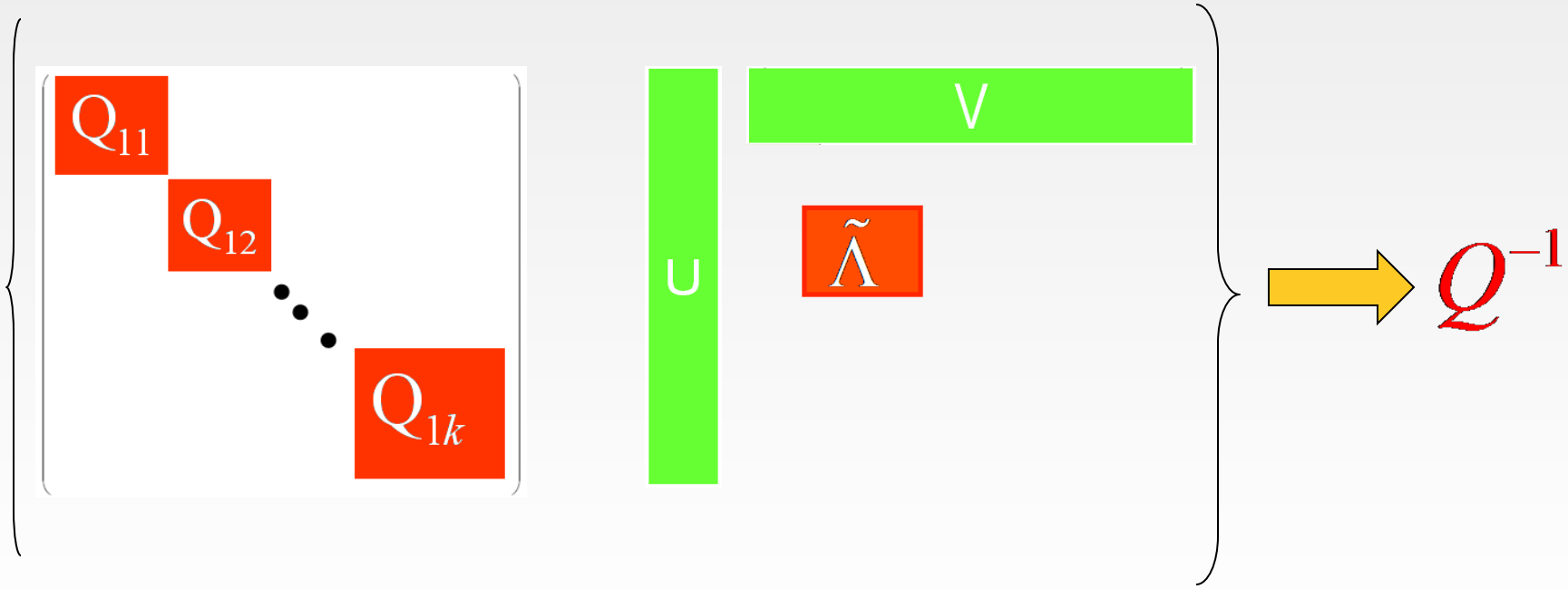


Big Data Analysis- Mahout and Pegasus

- Mahout修改了Data Mining的底層運算子的實作
  - 不只是Density Matrix到Sparse Matrix…（Matlab）就有
  - 更是採用Map-reduce的計算框架
- Pegasus是由卡內基梅隆大學Christos Faloutsos團隊所開發 (http://www.cs.cmu.edu/~pegasus/)
  - 著重於Graph Mining的演算法

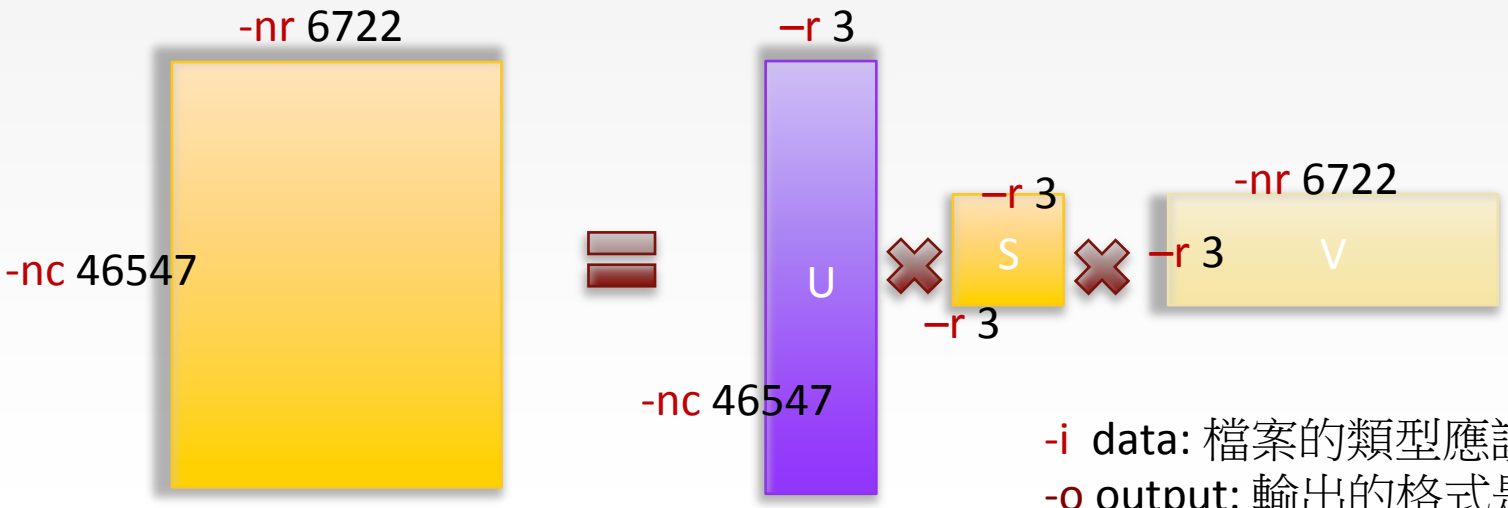


Extension: Social Network Analysis



Singular Value Decomposition (SVD)

- Mahout實作SVD，非常容易使用
  - mahout svd –r 3 -nr 6722 -nc 46547 –i data -o result



-i data: 檔案的類型應該是？
-o output: 輸出的格式是？

Singular Value Decomposition (SVD)

- -i data: 檔案的類型應該是？
  1. 必須要是Sparse Matrix: row_id, col_id, value
  2. 必須要轉換成HDFS sequential file

  (http://bickson.blogspot.tw/2011/02/mahout-svd-matrix-factorization.html)

0,0,1
0,1,2
1,0,3
1,1,4

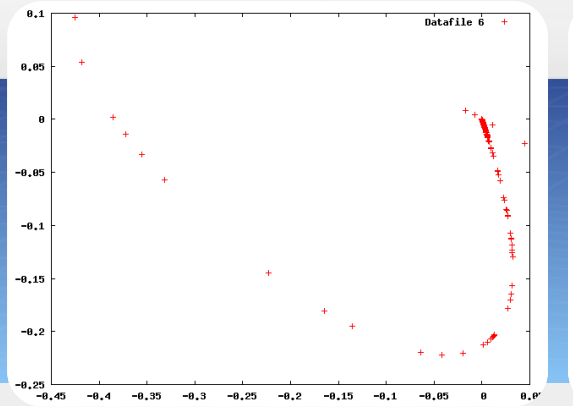
- -o output: 輸出的格式是？
  1. Output亦是HDFS sequential file
  2. 必須轉換為plain text，透過mahout

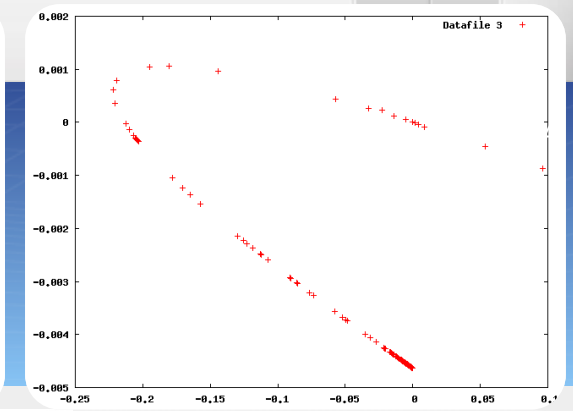
  mahout seqdumper --seqFile hdfs://Cluster01:9000/… --output …

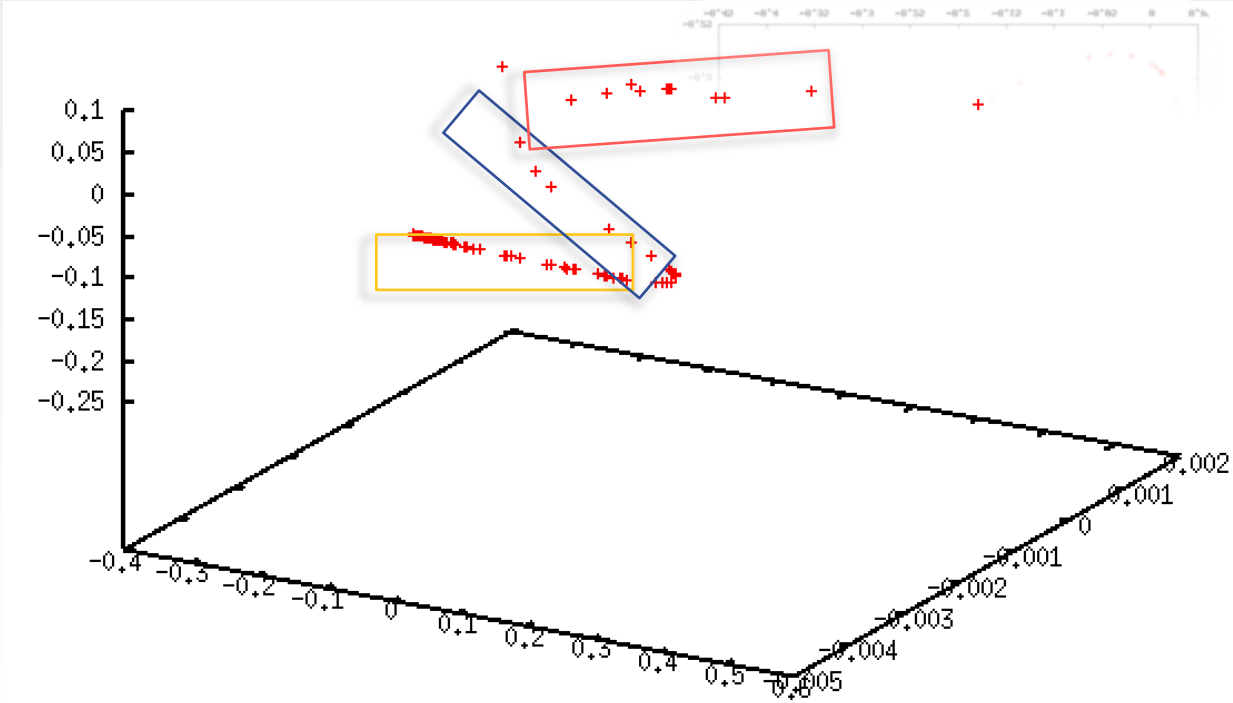
SVD可以告訴我們什麼？

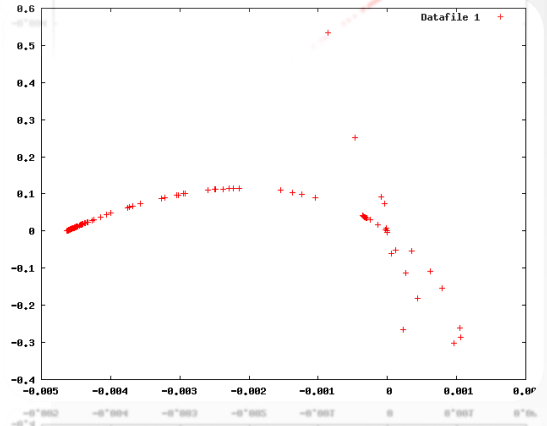
- Visualization: 它可以降到可瞭解的維度，了解各實體間的關聯
- Clustering：配合Vector-based演算法，進行分群
- Fast Indexing：可不需逐筆索引，加快索引速度

Visualization



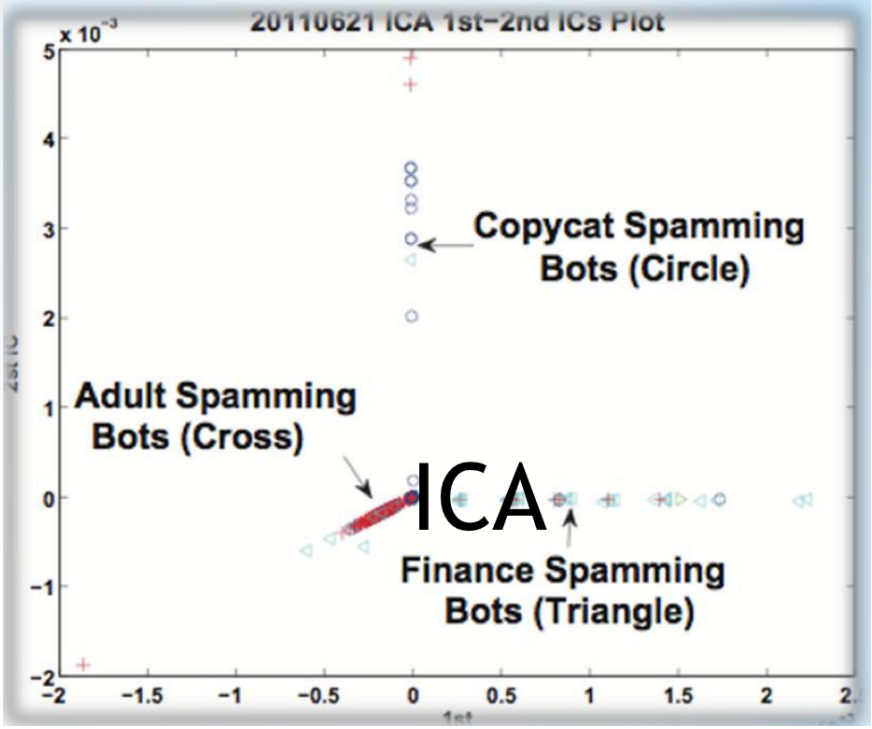






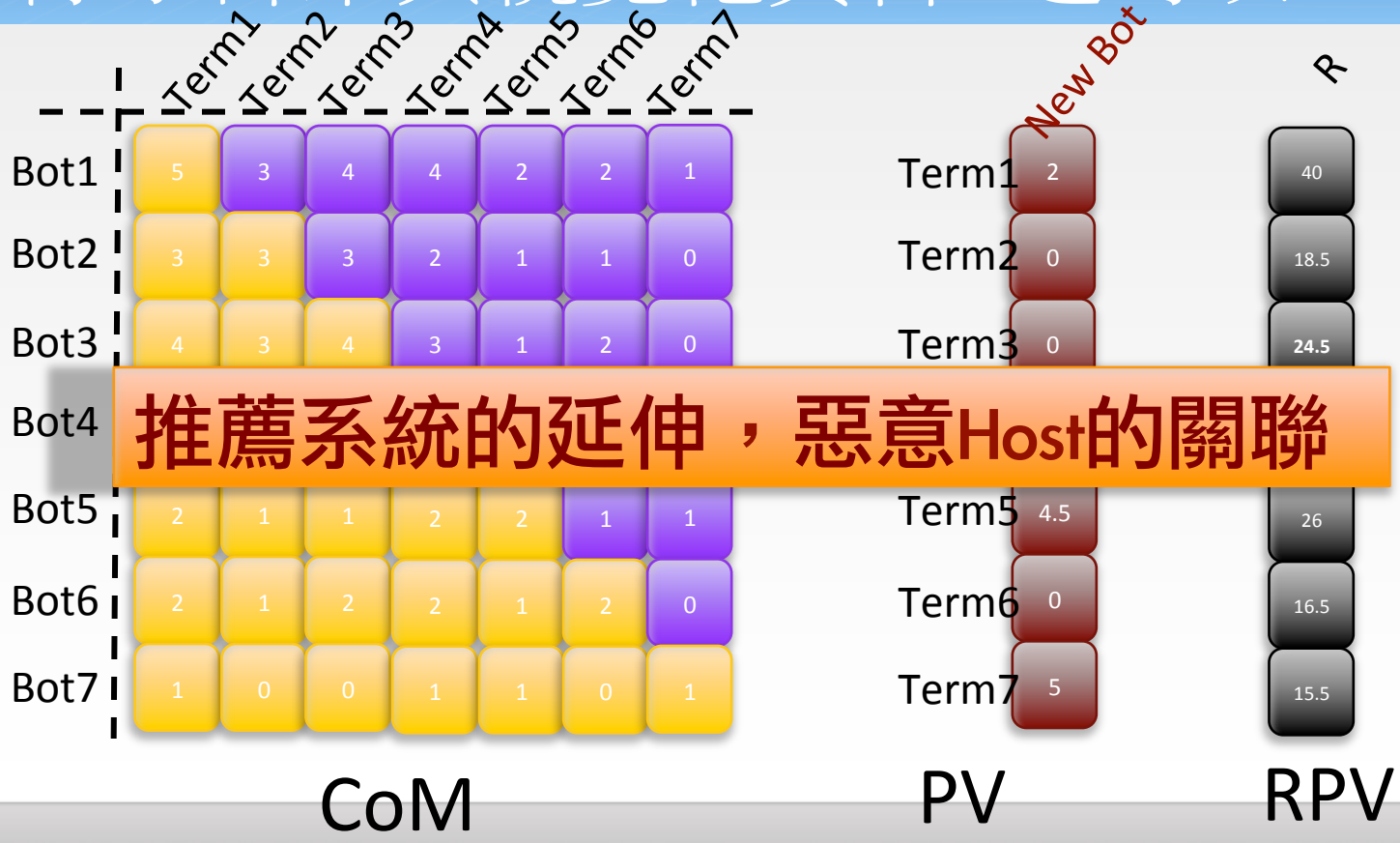
每一個紅點代表發送垃圾郵件的IP位置

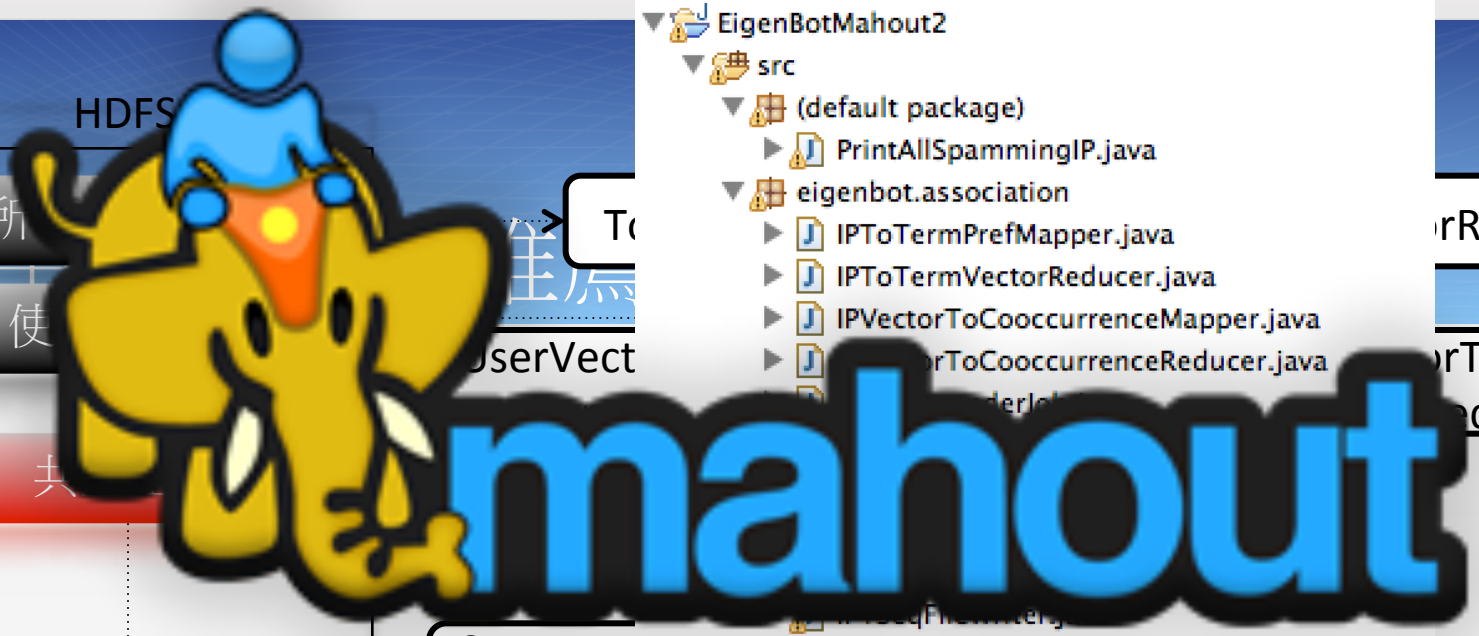
Visualization



不同群的Spamming Botnet被區分開來

SVD除了群聚與視覺化資料...還可以





HDFS

所

使

共

預先部分乘積向量

部分乘積向量

推薦清單

To

rReducer

rToCoocc

ducer

Cooccurr

nWrapp

PartialMu

p

ndPrefRe

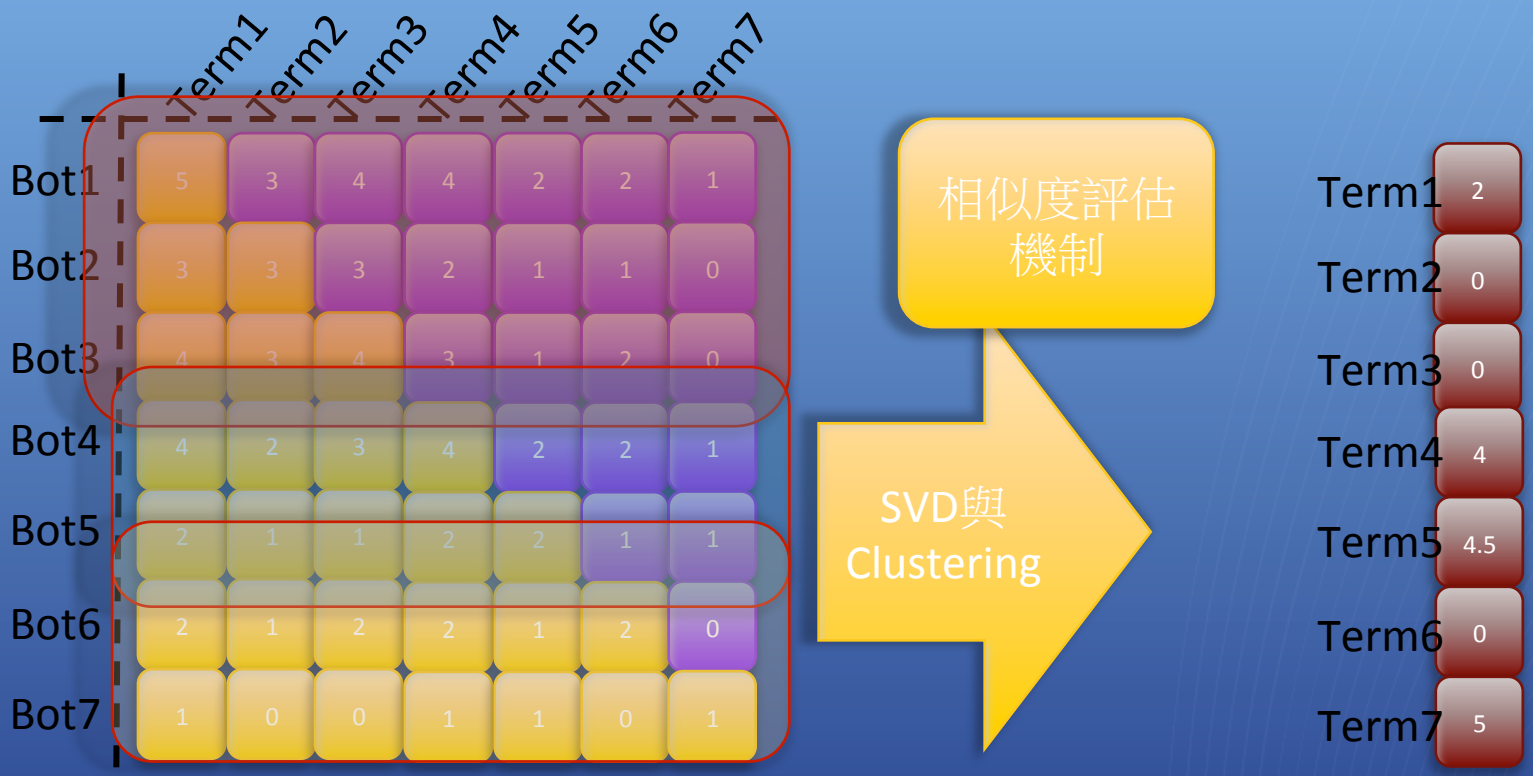
cer

eAndReco

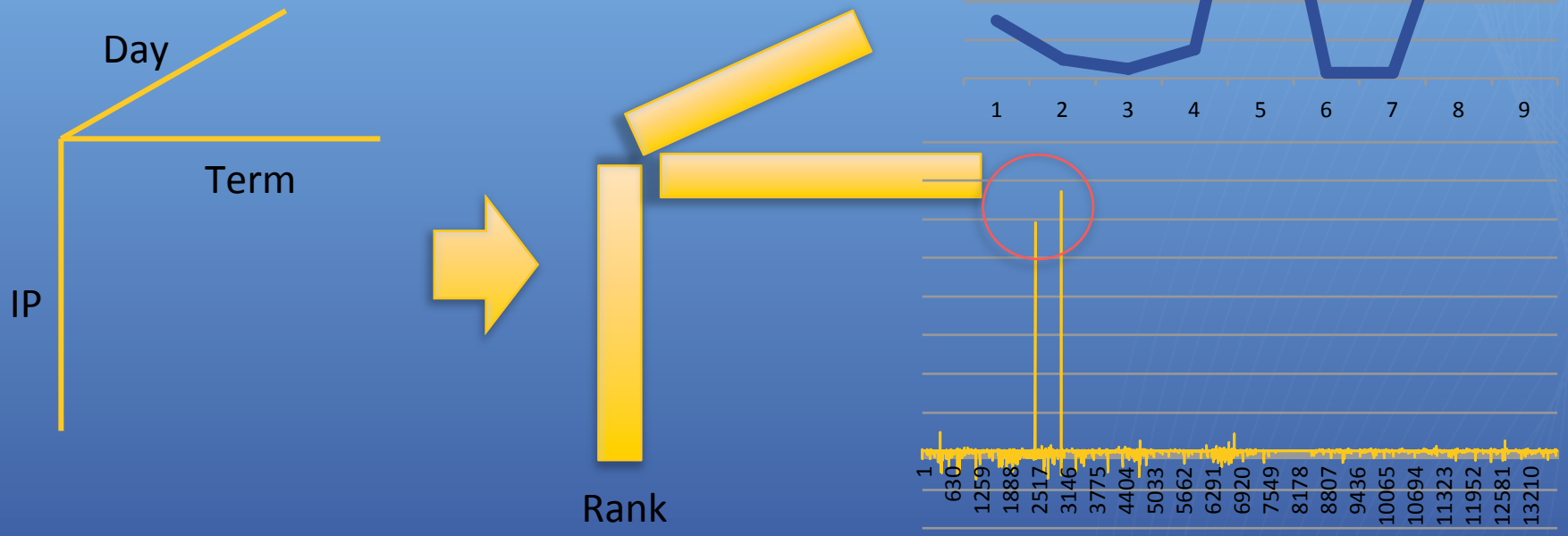
Reducer

EigenBotMahout2
- src
  - (default package)
    - PrintAllSpammingIP.java
  - eigenbot.association
    - IPToTermPrefMapper.java
    - IPToTermVectorReducer.java
    - IPVectorToCooccurrenceMapper.java
    - rToCooccurrenceReducer.java
    - PrintAllSpammingIP.java
- JRE System Library [Java SE 7 (MacOS X Default)]
- xmlenc-0.52.jar – /Users/ericmao/Downloads/had
- slf4j-log4j12-1.4.3.jar – /Users/ericmao/Downloa
- slf4j-api-1.4.3.jar – /Users/ericmao/Downloads/h
- servlet-api-2.5-20081211.jar – /Users/ericmao/D
- oro-2.0.8.jar – /Users/ericmao/Downloads/hadoop
- mockito-all-1.8.5.jar – /Users/ericmao/Downloads
- log4j-1.2.15.jar – /Users/ericmao/Downloads/had
- kfs-0.2.2.jar – /Users/ericmao/Downloads/hadoop
- junit-4.5.jar – /Users/ericmao/Downloads/hadoop
- jsch-0.1.42.jar – /Users/ericmao/Downloads/hado
- jetty-util-6.1.26.jar – /Users/ericmao/Downloads/

然而，當資料幾百萬筆時...

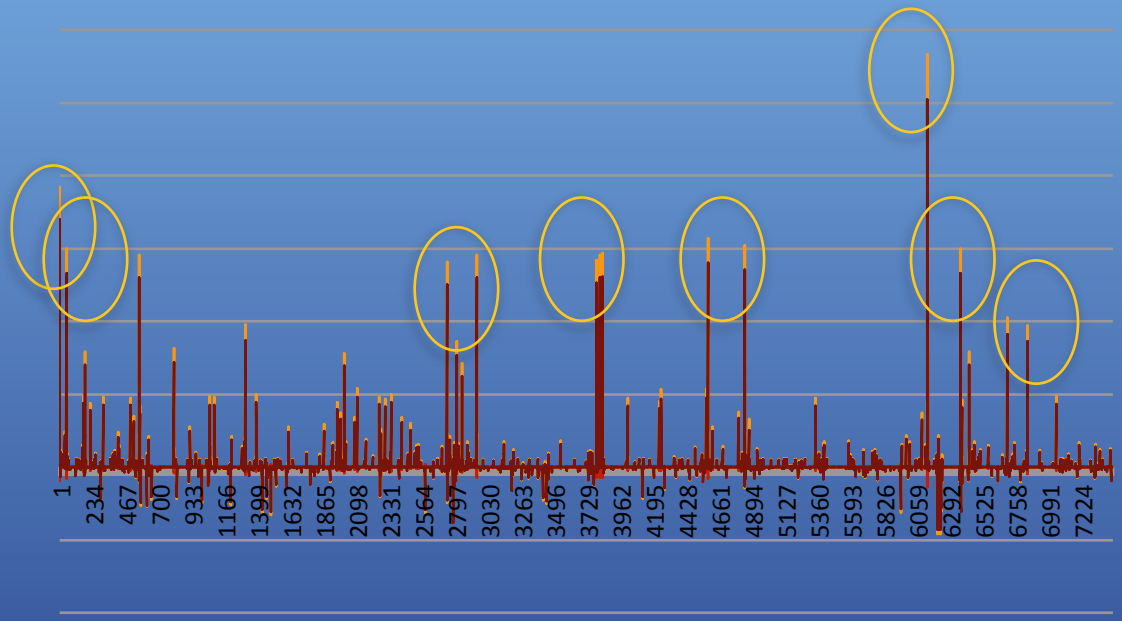

Term1
Term2
Term3
Term4
Term5
Term6
Term7
Bot1
Bot2
Bot3
Bot4
Bot5
Bot6
Bot7
5 3 4 4 2 2 1
3 3 3 2 1 1 0
4 3 4 3 1 2 0
4 2 3 4 2 2 1
2 1 1 2 2 1 1
2 1 2 2 1 2 0
1 0 0 1 1 0 1
相似度評估機制
SVD與Clustering
Term1 2
Term2 0
Term3 0
Term4 4
Term5 4.5
Term6 0
Term7 5

GigaTensor in Pegasus



U. Kang, Evangelos Papalexakis, Abhay Harpale, Christos Faloutsos, GigaTensor: scaling tensor analysis up by 100 times - algorithms and discoveries, ACM SIGKDD 2012.

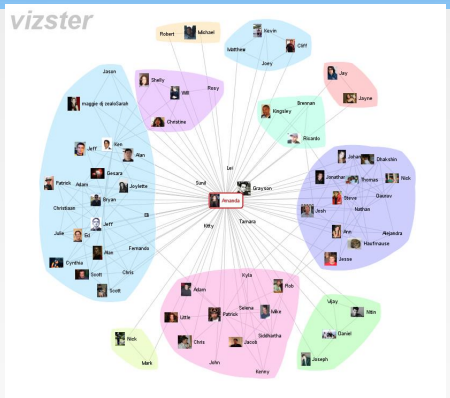
GigaTensor- 考慮時間與IP在字詞的特徵



使用步驟

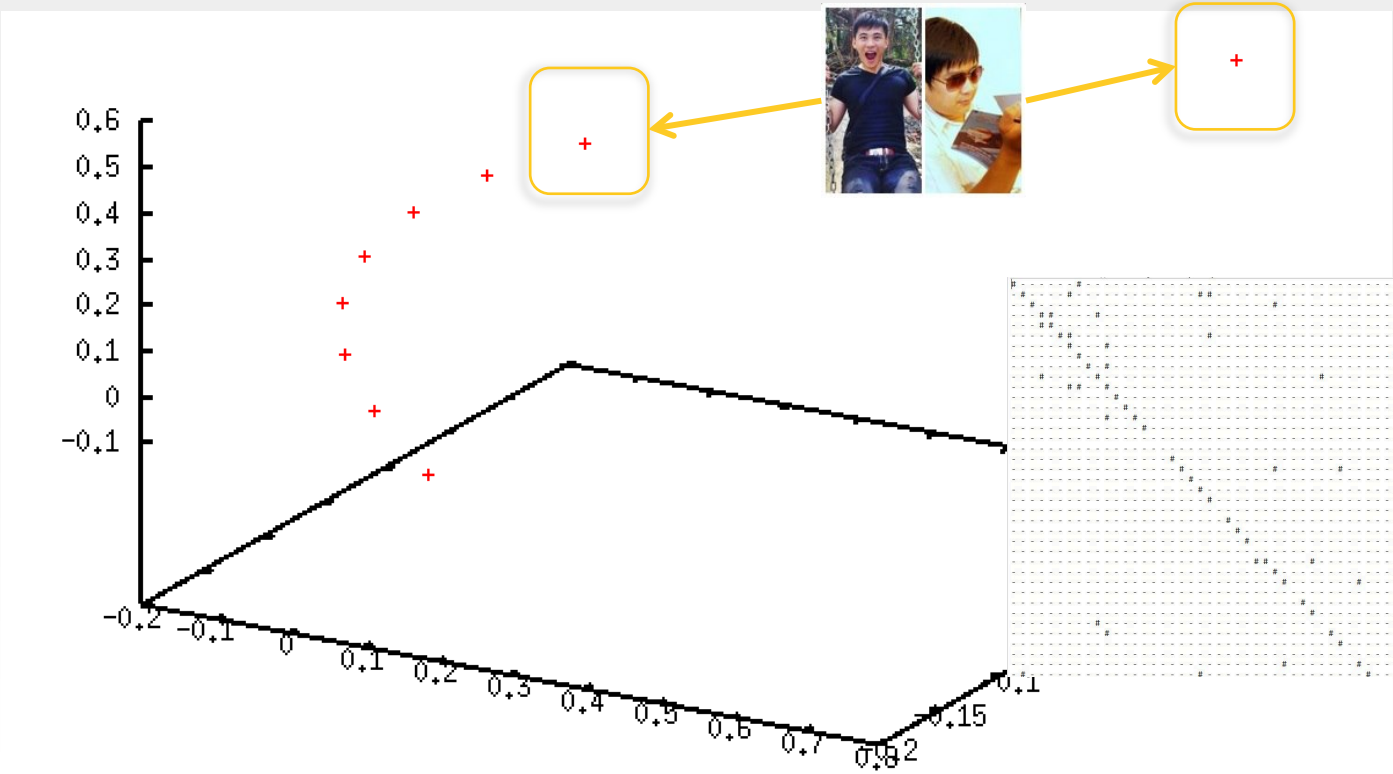
1. 取得GigaTensor的code
2. 參考do_parafac.sh

Extension: Social Network Analysis



- 抓朋友的朋友
  - 無法透過FB的API”直接”取得朋友與朋友的關係
- 解決方法
  - 找到朋友的ID
  - 再用朋友ID看到Po文（包括：讚與回覆)
  - 透過朋友塗鴉牆上的朋友互動，建立朋友與朋友的關係 (朋友數x朋友數)
  - 該關係稱為朋友關係矩陣（0代表沒有互動過，1代表有互動過)
- 透過Mahout SVD進行視覺化與群據分析

同樣分析方法所產生的結果



Summary

- Big Data最花時間的是在問題的前處理與分析
  - 去除雜訊及前處理
  - 判斷Pattern是否存在，需要高度domain expert的投入
  - 視覺化投射有助於找出pattern
- Mining的工具現已Graph較為常用於現實環境
  - 監督式學習，訓練資料的不完整性，會造成塑模的困難
  - 非監督式學習仍無法完全解決問題
  - 半監督式學習與主動學習是較貼切現實環境地學習框架
    - Belief Propagation…

Conclusions

- Lucene提供了及效率的索引平台
- Mahout提供基本的Big Data Analysis的Algorithms
- Pegasus可是用於大規模Graph Analysis
- SVD與Tensor適合協助找尋Patterns

Thanks & Questions

chmao2008@gmail.com 毛敬豪