Tweets Analysis -

whether tweets indicate disaster or not

蘇柏庄、吳岱錡、陳威宇、王彥翔、蔡雅欣

資料簡介

分析目的

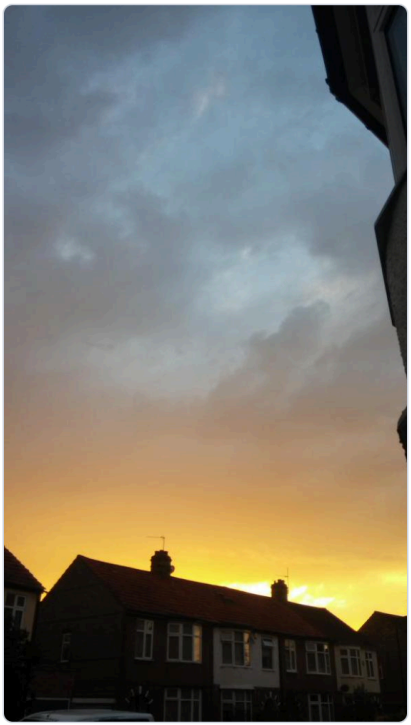
Wildfire Today
@wildfiretoday

#DeltaFire burns into #HirzFire north of Redding, California.
The fire grew by about 12,000 acres on Friday. Total now: 36,970 acres. Interstate 5 is still closed.
#DoNotEnter
Inciweb photo
wildfiretoday.com/2018/09/08/del...

4:54 AM · Sep 9, 2018 · TweetDeck

Anna K
@AnyOtherAnnaK

On plus side LOOK AT THE SKY LAST NIGHT IT WAS ABLAZE



12:43 AM · Aug 6, 2015 · Twitter for Android

BATUHAN
@batuhangcr

Afrin

The death toll rise to 43 after the bomb attack of the seperatist terrorist organization YPG/PKK to vegetable market in Raju. Among the wounded there are women and children. What is YPG/PKK if it is not a terror organization? Arw there any difference between DAESH and YPG?

11:17 PM · Apr 28, 2020 · Twitter for Android

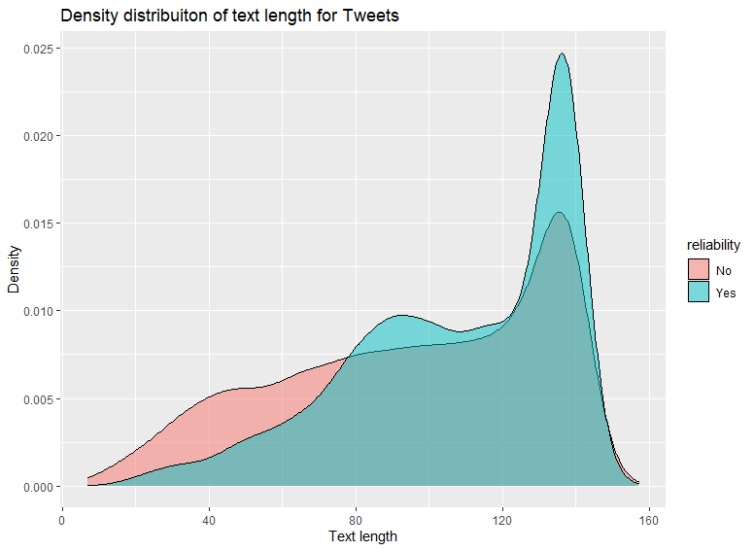
資料說明

變數類別	變數名稱	變數解釋
Covariates	Id	推特貼文識別碼
	Text	推特貼文文字內容
	location	推特的發文地點
	Keyword	貼文中的特定單詞
	TextLength	貼文文字內容長度

- 在 7613 個觀察值中，關於推文發生地點 (Location) 的缺失值占比高達 33%。由於地點資料無法應用其他方式生成，在以下分析中將不予以採用。

探索式資料分析

Analysis on text length



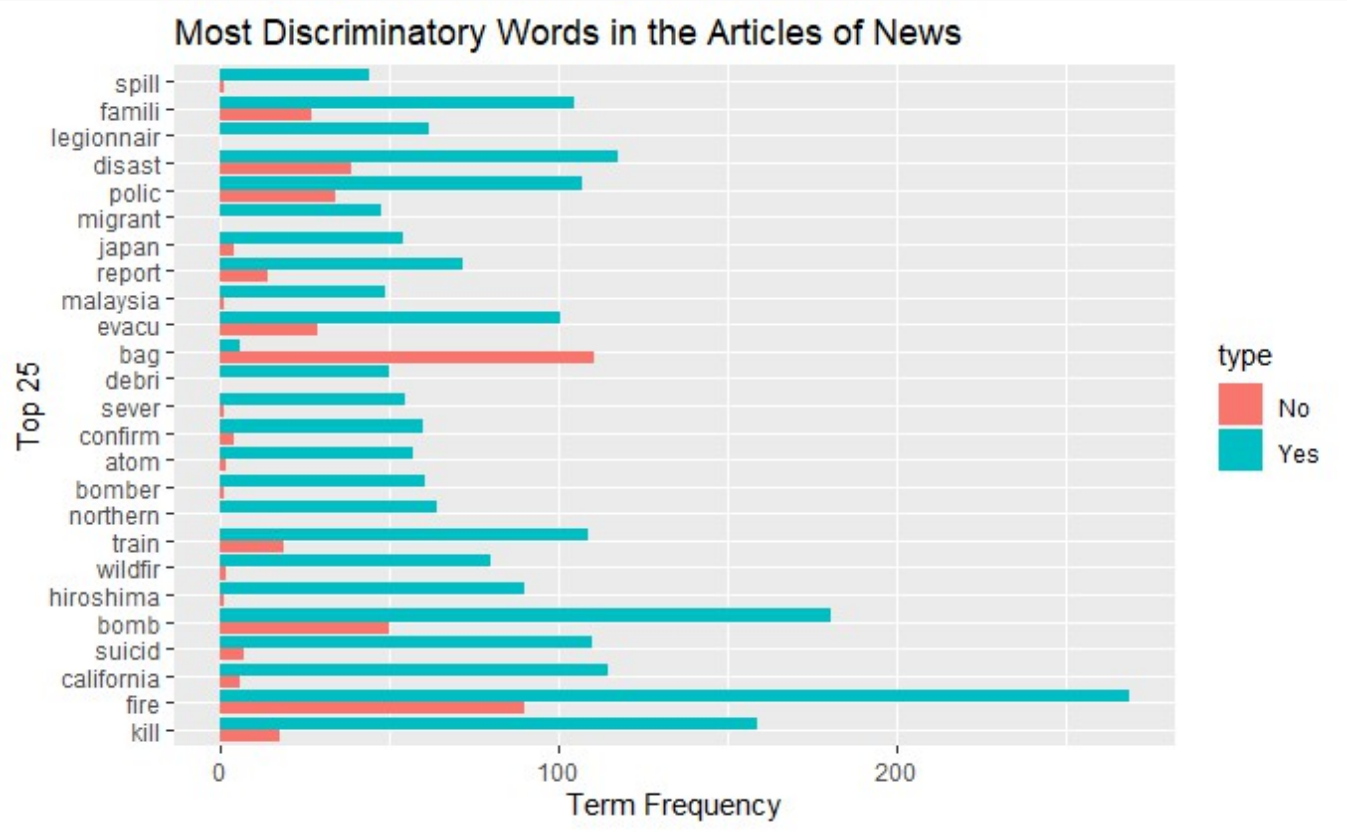
Two Sample T-test	
p-value	$< 2.2e-16$
95% 信賴區間	$(10.94068, 13.87252)$
x 平均值為	108.113
y 平均值為	95.707

- 事故實際未發生時，其分布會較為平均；但倘若實際發生的話，其推文文長會集中在字數較長的區間，而在字數較少的區間其頻率皆低於未發生的情況。

WordCloud

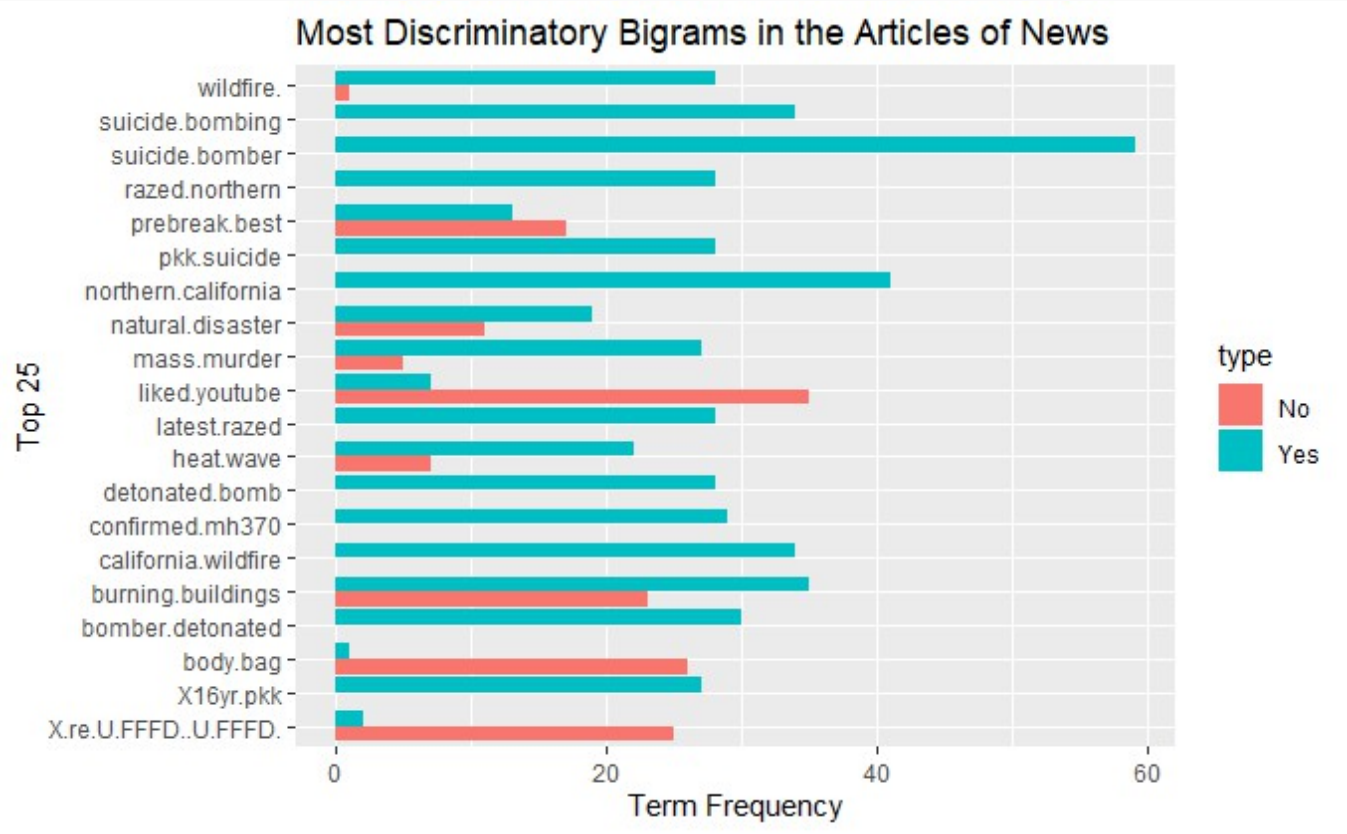
- 真實事故的文字雲出現次數高的詞，其與人員傷亡必會有一定程度的連結
- 錯誤警報的文字雲大多為無關緊要的文字或是使用上較為聳動的用詞。

Unigram Top Discriminatory Words



- **Bomb**(炸彈)：「有事故」的出現頻率遠高於「實際未發生」。
- **Fire**：「有事故」的出現頻率遠高於「實際未發生」。
- **Kill**：「有事故」的出現頻率遠高於「實際未發生」。

Bigram Top Discriminatory Words



- **suicide** 與 **bomber**、**bombing** 資料集中且多為「真實事故」
- **northern california**、**california wildfire**
- **liked youtube** 同時出現時高機率為「未發生事故」
- **burning buidings** 同時出現時真假「近乎參半」

模型方法

tf-idf: tf and idf

- $tf_{t,d}$ (term frequency，詞頻) 衡量各個詞於各個文件中出現的頻率，並將其標準化。

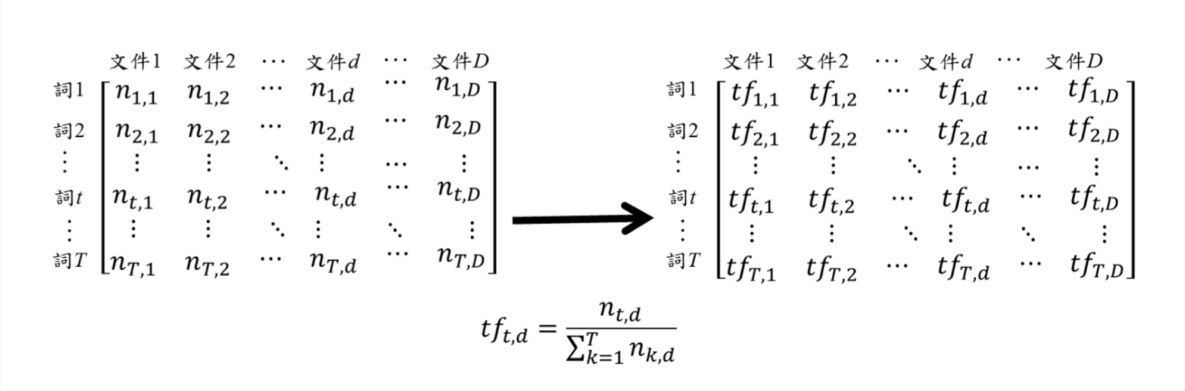


Figure 1: tf - term frequency

- idf_t (inverse document frequency，逆向文件頻率) 則計算每個詞於整份文件中的重要性，若其於每份文件皆曾出現，則較為不重要。

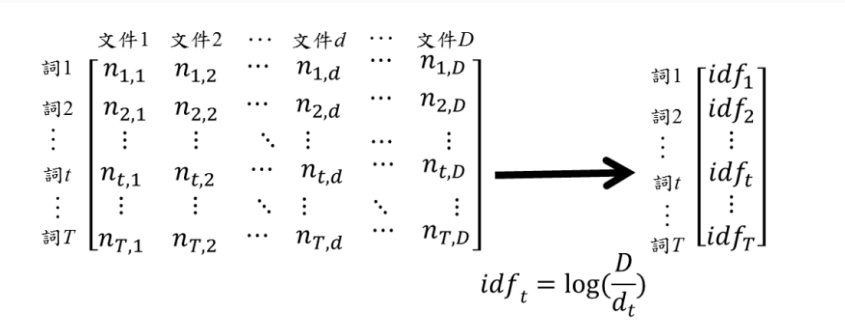


Figure 2: idf - inverse document frequency

- tf-idf 由 $score_{t,d}$ 來計算各個詞於每個文件中的重要性權重，將前兩者所計算的 idf_t 以及 $tf_{t,d}$ 進行矩陣相乘後，可得其值。

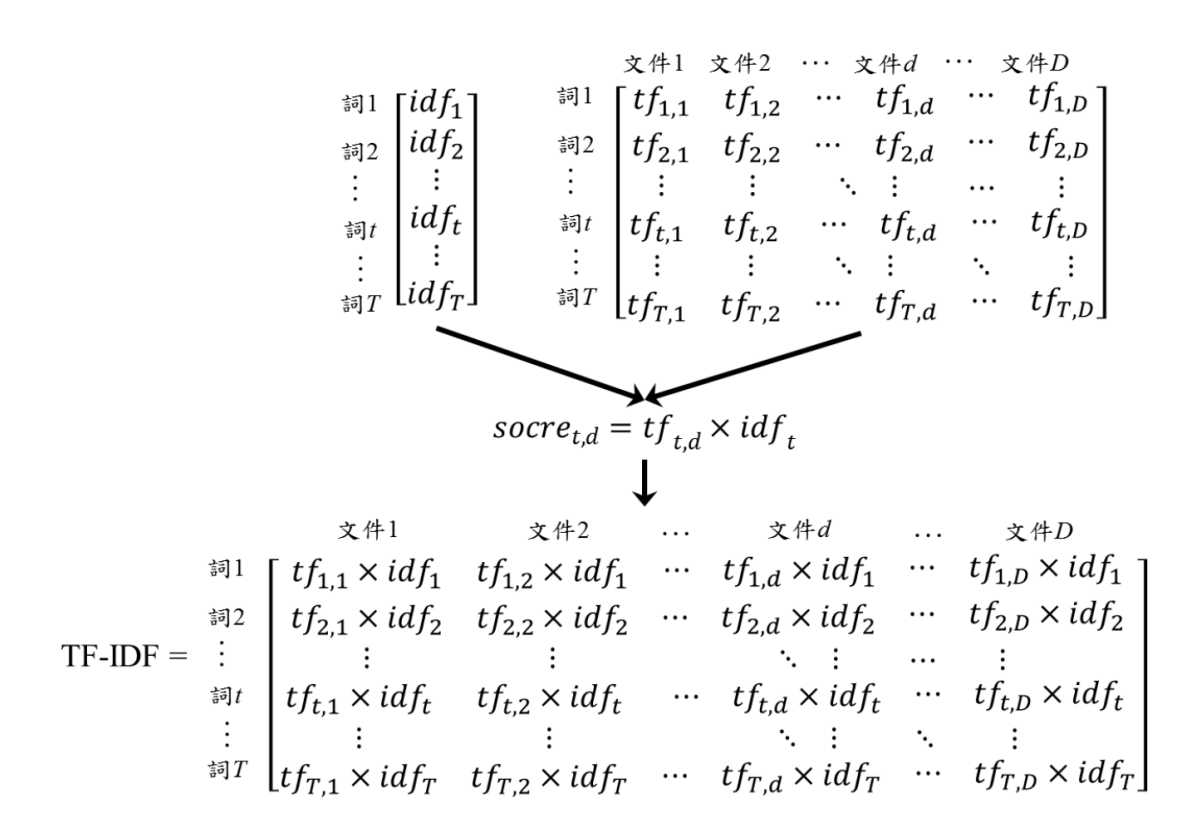


Figure 3: tf-idf

K-Means

K-means 是一個簡單易懂且利於解釋的分類演算法, 其想法為在事先給定 K 個群集下，最小化群內的資料與群心的誤差平方和，公式可以寫爲

$$\underset{\mu}{argmin} \sum_{c=1}^{K} \sum_{i=1}^{n_c} ||x_i - \mu_c||^2 \Big|_{x_i \in S_c}$$

μ_c 就是群心，$||x - y||$ 就是算歐氏距離 (Euclidean distance)，S_c 則代表第 c 個群集 (cluster)。其演算法為：

1. 設定 k 個群心

$$\mu_c^{(0)} \in R^d, \quad c = 1, 2, ..., K$$

2. 將每個樣本分到與其最接近的群心所屬的群集中

$$S_c^{(t)} = \{x_i : ||x_i - \mu_c^{(t)}|| \leq ||x_i - \mu_{c^*}^{(t)}||, \forall\ i = 1, ..., n\}$$

3. 結合新加入的資料計算新的群心 (第 c 群內有 n_c 個資料)

$$\mu_c^{t+1} = \frac{sum(S_c^{(t)})}{n_c} = \sum_{i=1}^{n_c} x_i \Big|_{x_i \in S_c}$$

4. 重複 2 和 3 直到群心收斂不變

$$S_c^{(t+1)} = S_c^{(t)}, \quad \forall\ c = 1, ..., K$$

Gaussian Mixture Model

高斯混合模型（Gaussian Mixture Model，簡稱 GMM) 是單一高斯機率密度函數的延伸，指的是多個高斯分布函數的線性組合，由於 GMM 能夠平滑地近似任意形狀的密度分佈，因此近來常被用在語音辨識並得到不錯的效果。

- 單一高斯機率密度函數

$$N(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) \right]$$

- 高斯混合模型：

$$p(x|\Theta) = \sum_{k=1}^{K} \alpha_k N(x|\mu_k,\Sigma_k)$$

$$\sum_{k=1}^{K} \alpha_k = 1, \quad 0 \le \alpha_k \le 1$$

$$N(x|\mu_k,\Sigma_k) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_k|}} exp\left[-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) \right]$$

給定樣本集合 $\{x_1, x_2, ..., x_n\}$，在選取 n 個樣本後，其概似函數為：

$$
\begin{aligned}
L(\Theta) &= \prod_{i=1}^{n} p(x_i|\Theta) \\
\Rightarrow lnL(\Theta) &= \sum_{i=1}^{n} lnp(x_i|\Theta) \\
&= \sum_{i=1}^{n} ln\Big\{ \sum_{k=1}^{K} p(x_i|z_k = 1)p(z_k = 1) \Big\} \\
&= \sum_{i=1}^{n} ln\Big\{ \sum_{k=1}^{K} \alpha_k N(x_i|\mu_k, \Sigma_k) \Big\}
\end{aligned}
$$

EM 是一種不斷反覆運算的演算法，所以參數會不斷的更新，此處假設第 t 與 $t+1$ 次估計的參數如下：

$$\Theta^{(t)} = \left\{ \alpha_k^{(t)};\, \mu_k^{(t)},\, \Sigma_k^{(t)} \right\}_{k=1}^{K}$$

$$\Theta^{(t+1)} = \left\{ \alpha_k^{(t+1)};\, \mu_k^{(t+1)},\, \Sigma_k^{(t+1)} \right\}_{k=1}^{K}$$

- 假設給定樣本集合 $x_1, x_2, ..., x_n$ ，
- 初始化參數：設定 K 個數，t(第 t 次計算) 設定為 0

$$\Theta^{(0)} = \left\{ \alpha_k^{(0)};\, \mu_k^{(0)},\, \Sigma_k^{(0)} \right\}_{k=1}^{K}$$

- **E-Step**

  假設所有參數 $\Theta^{(t)}$ 已知，便可估計出位置分類 $p(\mathbf{z}|x)$ 的後驗幾率。

$$w_k^{(t)}(x_i) = p(z_k = 1|x_i) = \frac{\alpha_k^{(t)} N\left(x_i|\mu_k^{(t)}, \Sigma_k^{(t)}\right)}{\sum_{j=1}^{K} N\left(x_i|\mu_j^{(t)}, \Sigma_j^{(t)}\right)},\ \forall i, k$$

- **M-Step**

  再利用 E-step 中計算的 $w_k^{(t)}(x_i)$ 去估計 $Theta^{(t+1)}$：

$$\Theta^{(t+1)} = \left\{ \alpha_k^{(t+1)};\ \mu_k^{(t+1)},\ \Sigma_k^{(t+1)} \right\}_{k=1}^{K}$$

$$\mu_k^{(t+1)} = \frac{1}{n_k^{(t)}} \sum_{i=1}^{n} w_k^{(t)}(x_i) x^{(i)}$$

$$\Sigma_k^{(t+1)} = \frac{1}{n_k^{(t)}} \sum_{i=1}^{n} \sum_{k=1}^{K} w_K^{(t)}(x_i) \left(x^{(i)} - \mu_k^{(t+1)} \right) \left(x^{(i)} - \mu_k^{(t+1)} \right)^T$$

$$\alpha_k^{(t+1)} = \frac{n_k^{(t)}}{n}, \quad n_k^{(t)} = \sum_{(i=1)}^{n} w_k^{(t)}(x_i)$$

- 重複 E-Step 和 M-Step，直到滿足收斂條件 (參數收斂或是概似函數收斂)，便可得到目標參數的最大概似估計值。

結果分析

K-means - Elbow Method

1. 計算在不同個 K 下的不同分群函式
2. 對每個 K，我們計算群組內的組內距離平方和 (Within-cluster Sum of Square)
3. 以 K 為橫軸，WSS 為縱軸作圖
4. 找出上圖中使坡度變化最大的 K 值

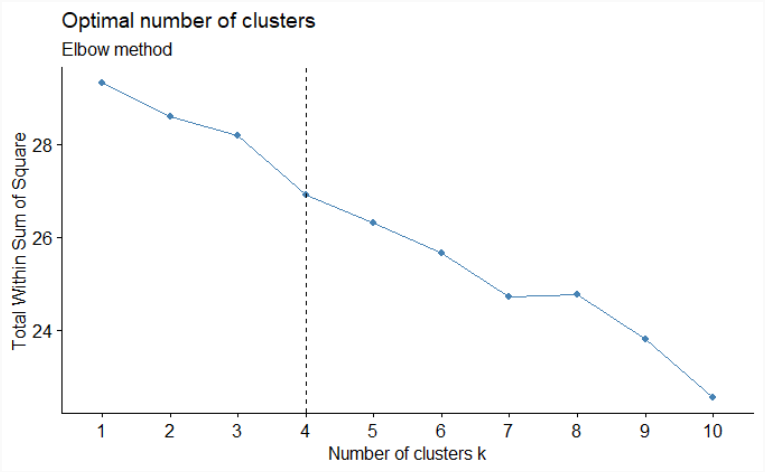


Figure 4: Elbow Method

K-means - Average Silhouette Method

此演算法根據每個資料點 (i) 的內聚力和分散力去衡量分群的效果，首先側影係數 (Silhouette Coefficient) 如下：

$$s(i) = \frac{b(i) - a(i)}{max\{a(i),\ b(i)\}}$$

$a(i)$：資料點 i 與群內其他資料點的平均距離

$b(i)$：資料點 i 與其他群內資料點的平均距離，取最小值

$s(i)$：側影係數，可視為資料點 i 在它所屬的群內是否適當的指標

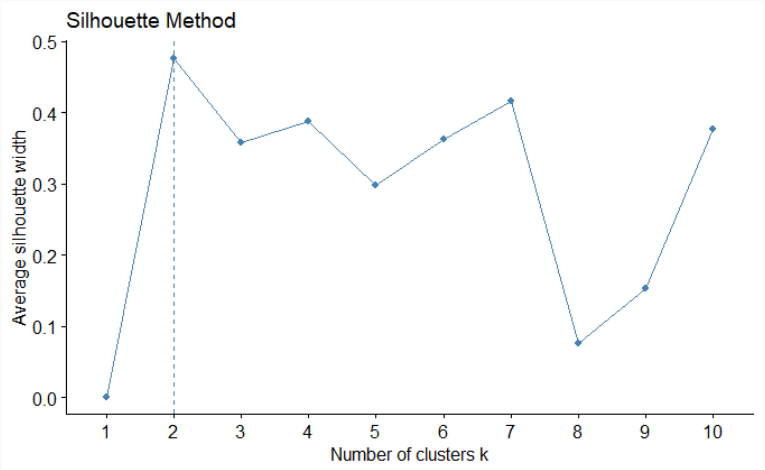


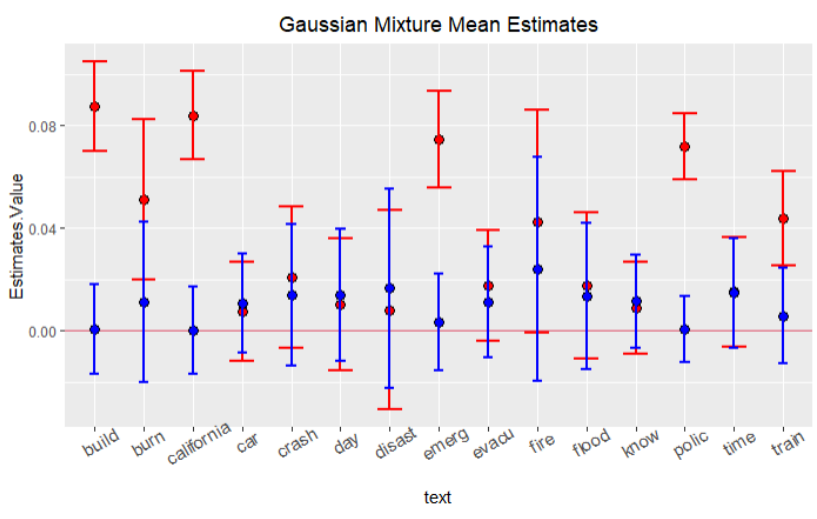
Figure 5: Silhouette Method

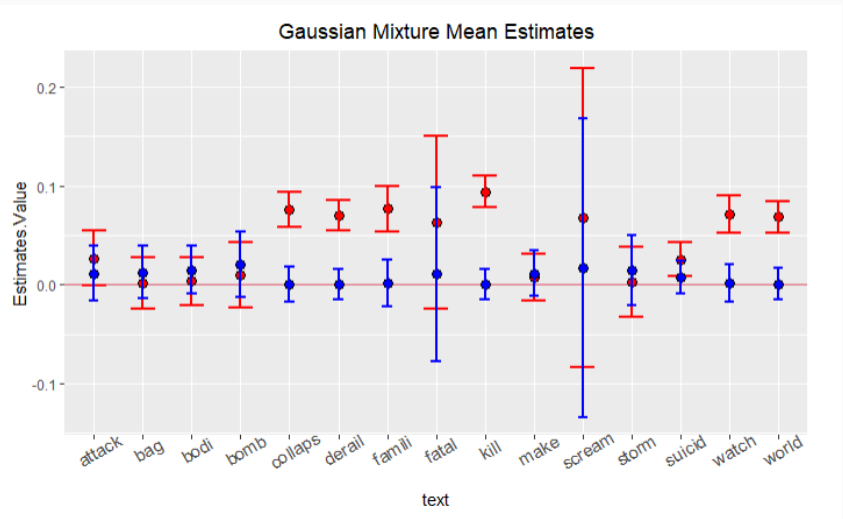
K-means - Cross Table

	Actual		
Predicted	1	2	Row Total
1	84	86	170
	0.494	0.506	0.022
	0.026	0.020	
2	3187	4256	7443
	0.428	0.572	0.978
	0.974	0.980	
Column Total	3271	4342	7613
	0.430	0.570	

- 準確度為 0.5701
- 幾乎所有資料皆被預測為第二分群
- 預測結果不佳

Gaussian Mixture Model - 參數估計



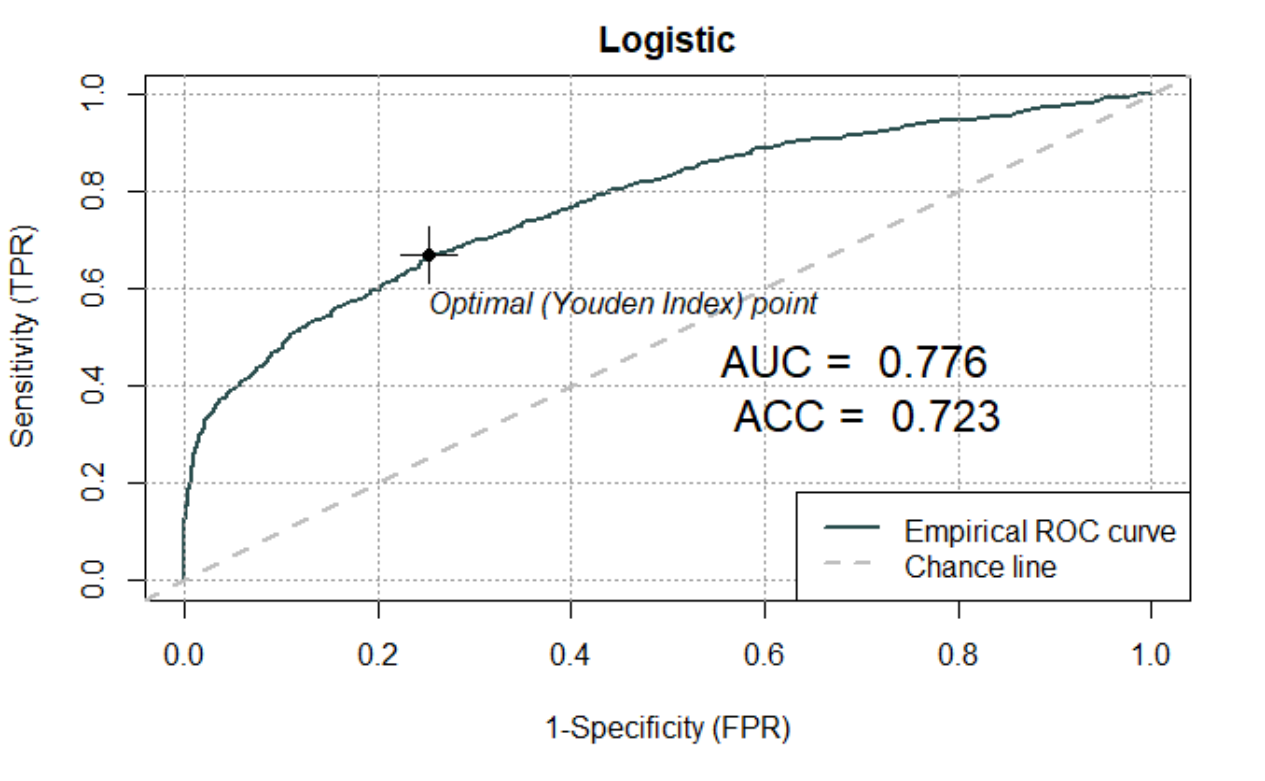


- 單詞「**buildings**」、「**california**」、「**emergency**」、「**police**」、「**train**」的分群效果明顯
- 「**collapse**」、「**derail**」、「**kill**」的分群效果較明顯
- 「**scream**」的信賴區間全距最長，故推測該變數中資料點變異度很大。

Gaussian Mixture Model - Cross Table

	Actual		
Predicted	1	2	Row Total
1	730	353	1083
	0.674	0.326	0.142
	0.223	0.081	
2	2541	3989	6530
	0.389	0.611	0.858
	0.777	0.919	
Column Total	3271	4342	7613
	0.430	0.570	

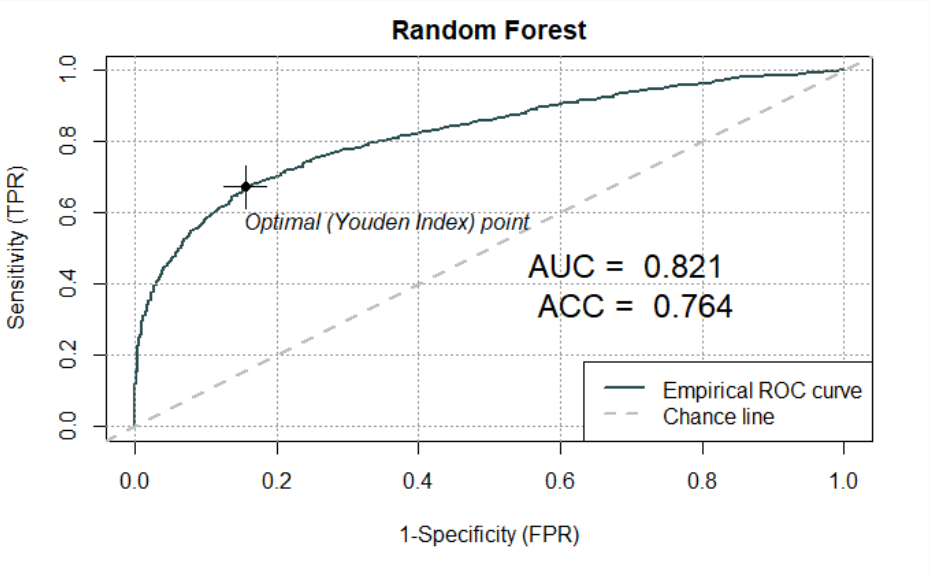
- 準確度為 0.6199，略優於 k-means
- 此群資料的預測分群各佔資料的 15%、85%
- 將許多分類為 1 的錯誤歸類為 2
- 預測結果不佳



在未選定 Cutpoint(預設為 0.5) 的狀況下建立模型取得的準確率為 0.7196671。$0.1 - 0.9$ 中最佳 Cutpoint 為 0.58。

- 精確度為 0.7232。

Random Forest



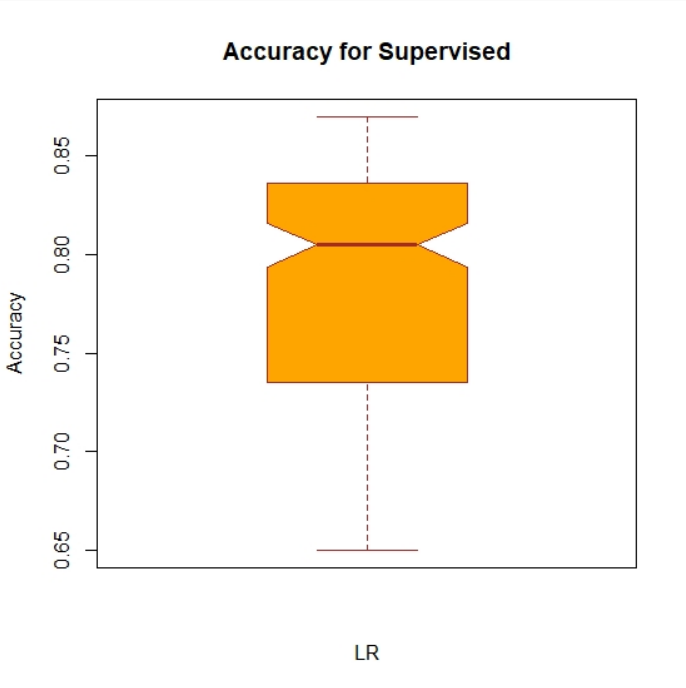
模型參數	參數值
ntree	251
mtry	5
Node_size	17
Sample_size	0.7

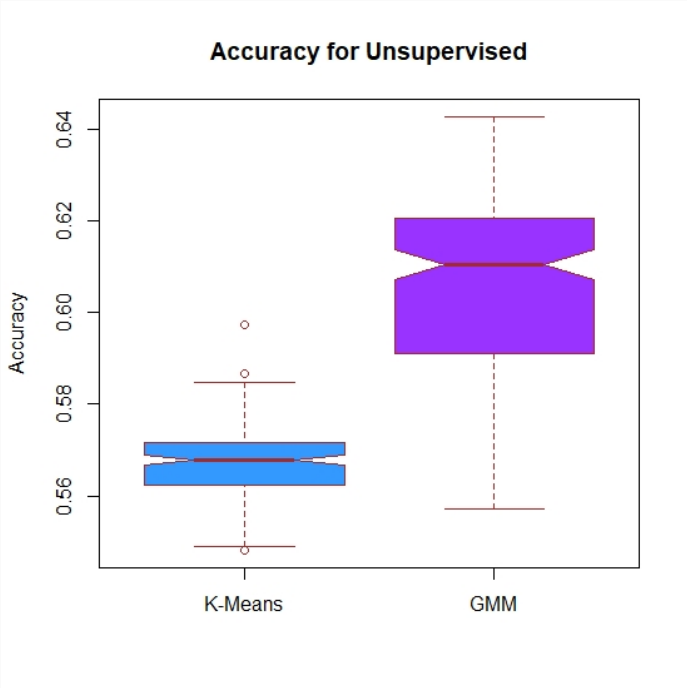
首先以未調整任何參數的情況下建構模型, 判斷最低誤差的最佳 tree size 大約落在 251；而後分別對 3 個參數：mtry, node size, sample size 做 grid search，並以 Out-of-Bag Error 作為標準進行參數調控。

- 精確度為 0.7643。

重抽様分析

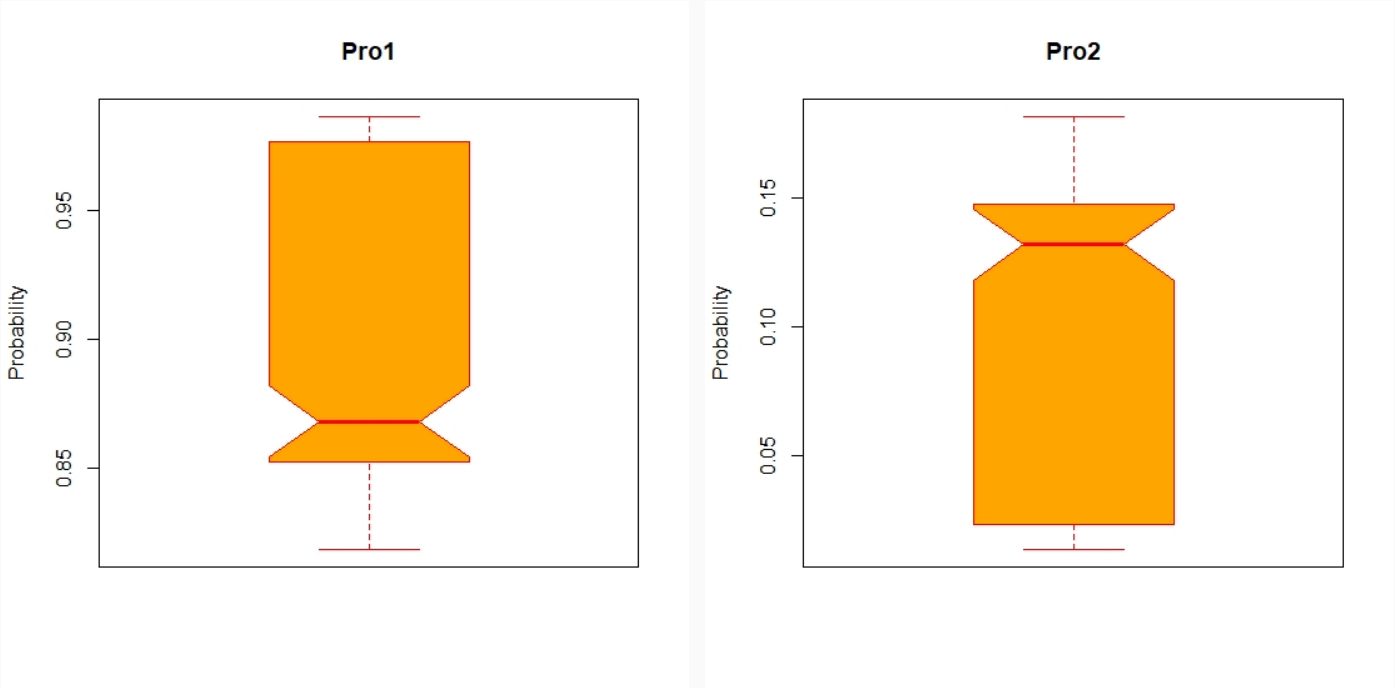
Bootstrap Accuracy





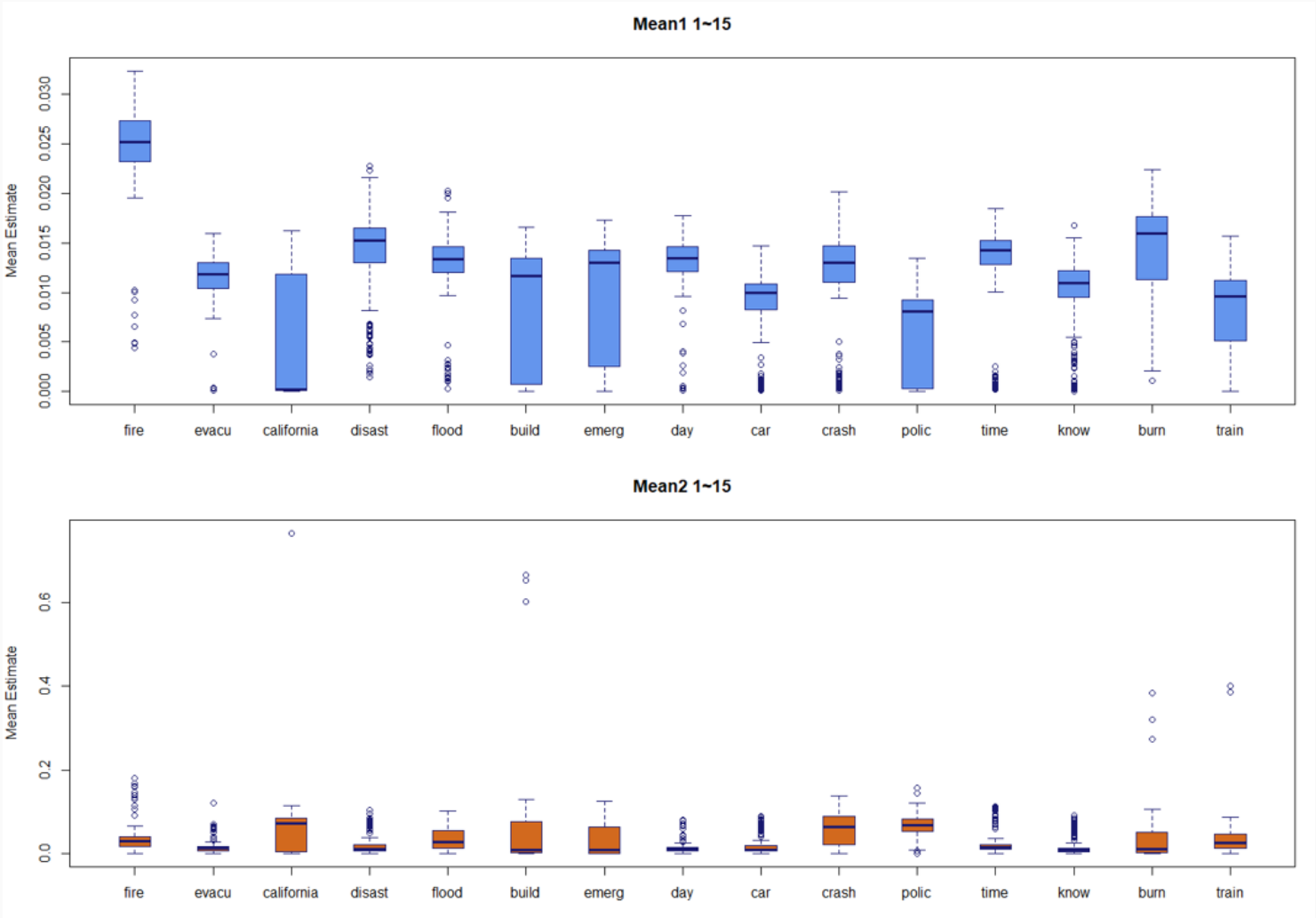
- Gmm 結果明顯優於 k—means
- k-means 的結果分佈更爲集中，受到初始值影響較少
- 所有的 Logistic Regression 結果都比 unsupervised 的優秀

Bootstrap Parameters - $\vec{\pi}$

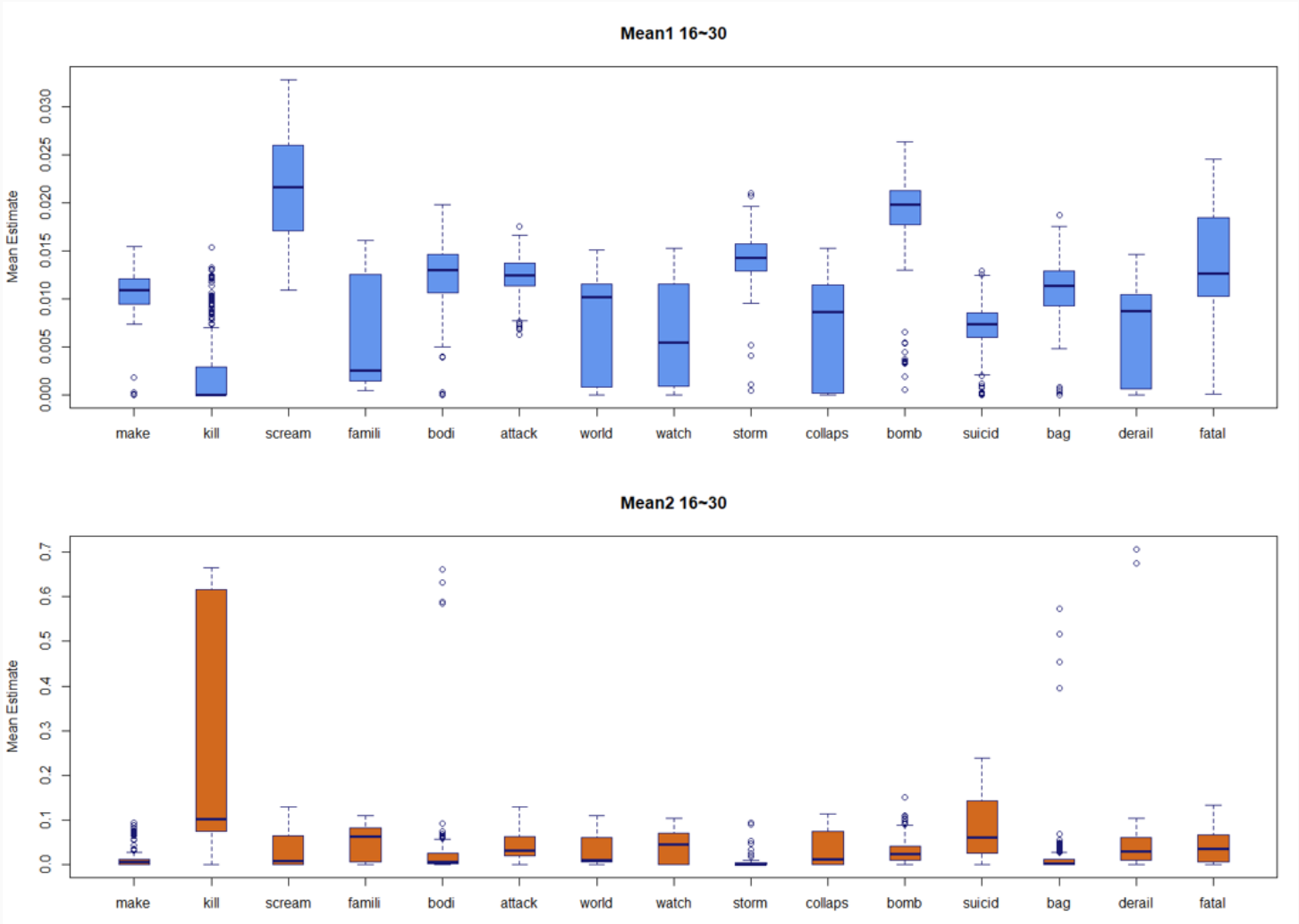


- 兩群組對應幾率差異顯著
- 代表事件實際發生的群組的幾率明顯高於事件未發生

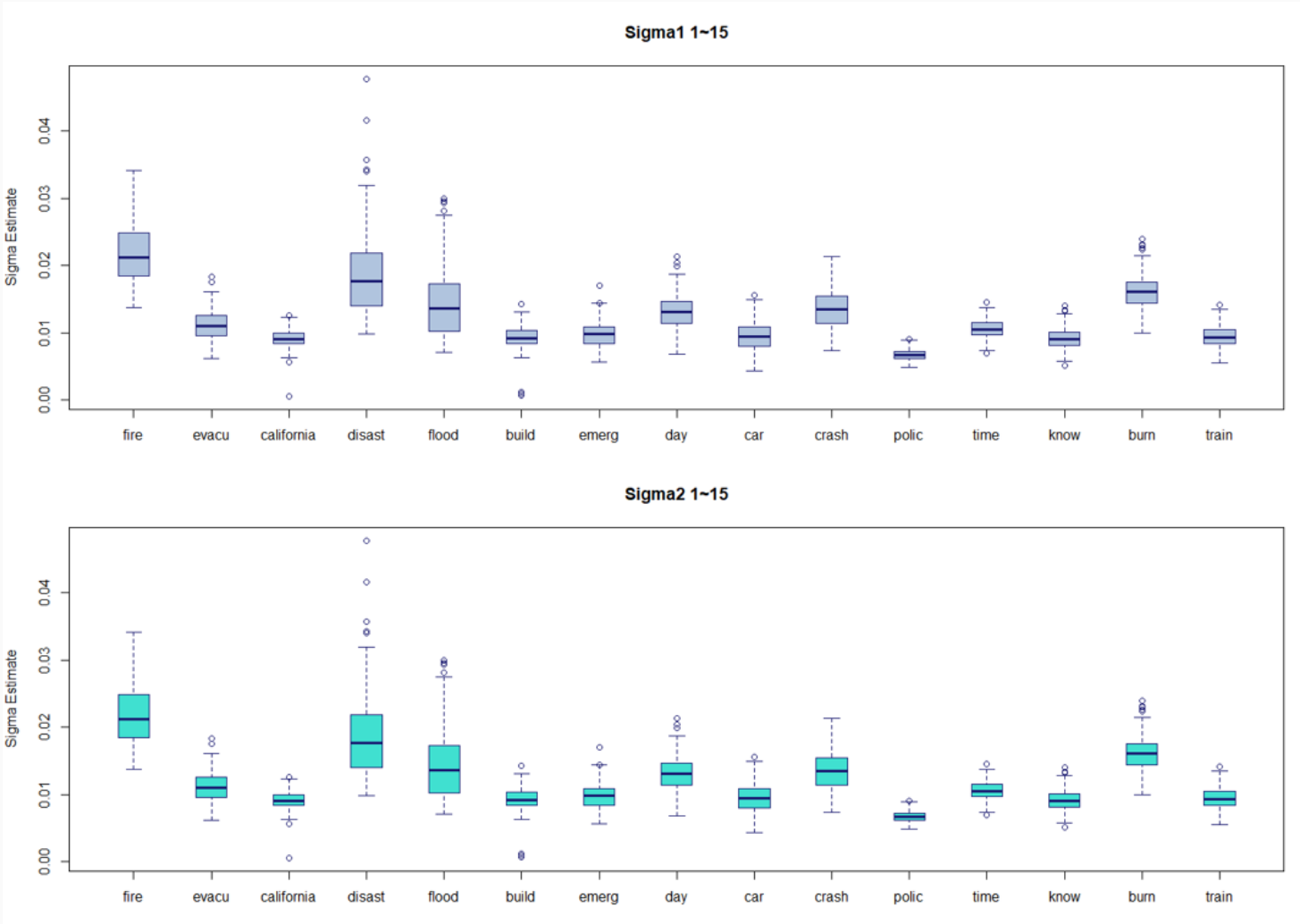
Bootstrap Parameters - $\vec{\mu}_1$



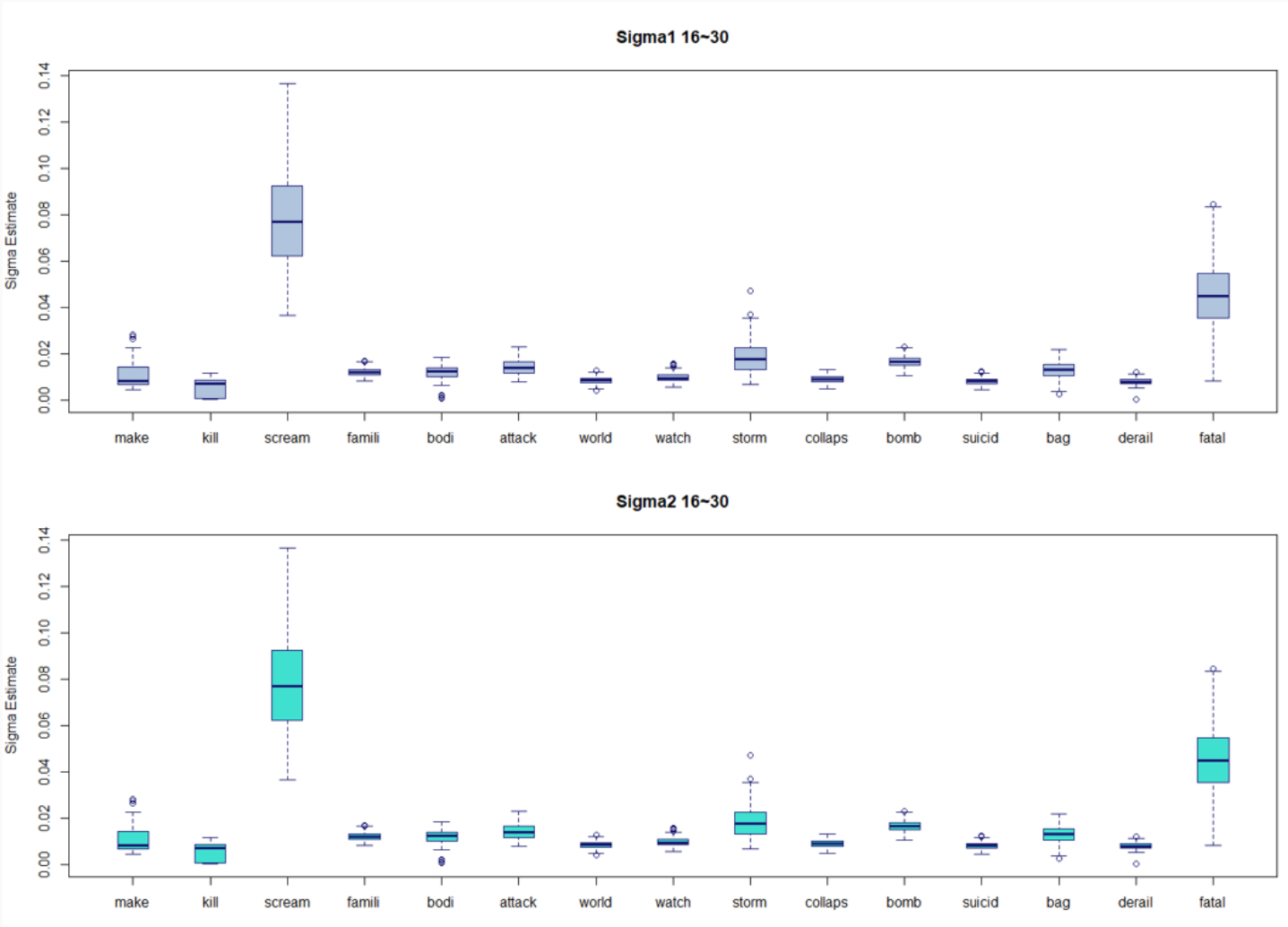
Bootstrap Parameters - $\vec{\mu}_2$



Bootstrap Parameters - $\vec{\sigma}_1$



Bootstrap Parameters - $\vec{\sigma}_2$



補充 - Latent Dirichlet Allocation

生成文檔架構

1. 先按照先驗機率從 $P(d_m)$ 抽取一篇文檔 d_m
2. 從 $Dirichlet\ \alpha$ 中抽樣生成文檔 d_m 的主題分布 θ_m
3. 從主題的 Multinomialθ_m 中抽樣生成文檔 d_m 第 n 個詞的主題 $z_{m,n}$
4. 從 $Dirichlet\ \beta$ 中抽樣生成主題 $z_{m,n}$ 對應的詞語分布 $\phi_{z_{m,n}}$
5. 從詞語的多項式分布 $\phi_{z_{m,n}}$ 中採樣最終生成詞語 $w_{m,n}$

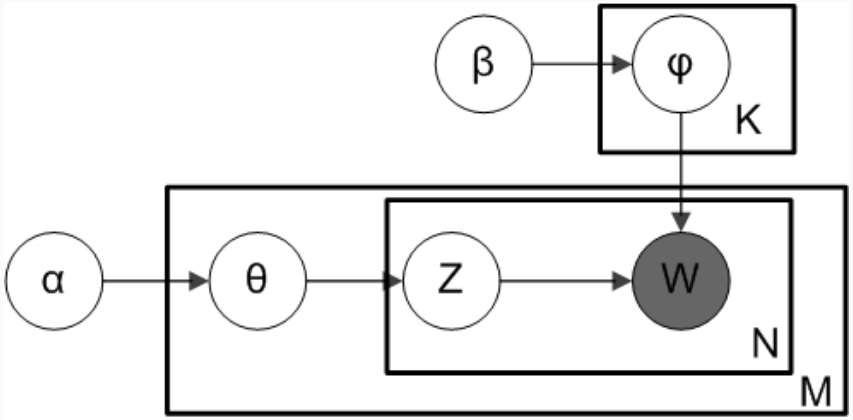


Figure 6: Bayesian Network of Latent Dirichlet Allocation

- $Dirichlet - Multinomial$ 共軛結構

$$\vec{\theta}_m \sim Dir(\vec{\theta}_m|\vec{\alpha}) \qquad z_{m,n} \sim Multi(z_{m,n}|\vec{\theta}_m)$$
$$\vec{\phi}_k \sim Dir(\vec{\phi}_k|\vec{\beta}) \qquad w_{m,n} \sim Multi(w_{m,n}|\vec{\phi}_k, z_{m,n})$$

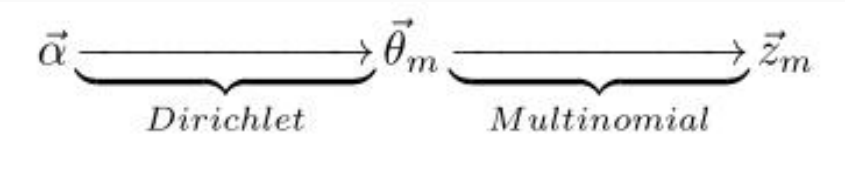


Figure 7: 生成文檔中的所有詞語對應的主題

Figure 8: 生成文檔中的每個主題所對應的詞語

聯合分布

$$p(\vec{z}, \vec{w}|\vec{\alpha}, \vec{\beta}) \quad = \quad p(\vec{w}|\vec{z}, \vec{\beta})p(\vec{z}|\vec{\alpha})$$

$p(\vec{w}|\vec{z}, \vec{\beta})$：根據確定主題 $\vec{z}$ 與詞語分布的先驗分布參數 β 採樣詞語

$$p(\vec{w}|\vec{z}, \vec{\beta}) = \prod_{k=1}^{K} \frac{\Delta(\vec{n}_k + \vec{\beta})}{\Delta(\vec{\beta})}, \quad \vec{n}_k = \{n_k^{(t)}\}_{t=1}^{V}$$

$p(\vec{z}|\vec{\alpha})$：根據主題分布的先驗分布參數 α 採樣主題

$$p(\vec{z}|\vec{\alpha}) = \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}, \quad \vec{n}_m = \{n_m^{(k)}\}_{k=1}^{K}$$

其中兩個因子當中的 $\Delta(\vec{\alpha})$ 為 Dirichlet 分布的歸因化係數；

$\vec{n}_k$ 為構成主題 k 的詞語項數向量，$\vec{n}_m$ 為構成文檔 m 的主題數向量。

Latent Dirichlet Allocation - Latent Variable z

有了聯合分布 $p(\vec{w}, \vec{z})$ 後，便可以通過其計算在給定觀測變量 $\vec{w}$ 下的隱變量 (latent variable)z 的後驗分布 $p(\vec{z}|\vec{w})$ 來進行貝斯分析 (Bayesian Analysis)。

$$p(z_i = k|\vec{z}_{-i}, \vec{w}) \propto \int \vartheta_{m,k} Dir(\vec{\vartheta}_m|\vec{n}_{m,-i} + \vec{\alpha}) \mathrm{d}\vec{\vartheta}_m \cdot \int \varphi_{k,t} Dir(\vec{\varphi}_k|\vec{n}_{k,-i} + \vec{\beta}) \mathrm{d}\vec{\varphi}_k$$
$$= E(\vartheta_{m,k}) \cdot E(\varphi_{k,t})$$

- Posterior of $\vec{\theta}_m, \vec{\varphi}_k$

$$p(\vec{\vartheta}_m|\vec{z}_m, \vec{\alpha}) = Dir(\vec{\vartheta}_m|\vec{n}_m + \vec{\alpha}) \qquad p(\vec{\varphi}_k|\vec{z}, \vec{w}, \vec{\beta}) = Dir(\vec{\varphi}_k|\vec{n}_k + \vec{\beta})$$

- Exepectation of θ and ϕ

$$\vartheta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_k (n_m^{(k)} + \alpha_k)} \qquad \varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_t (n_k^{(t)} + \beta_t)}$$

Latent Dirichlet Allocation - Latent Variable z

最後將 $\vartheta_{m,k}, \varphi_{k,t}$ 代入 $p(z_i = k|\vec{z}_{-i}, \vec{w})$，可得：

$$p(z_i = k|\vec{z}_{-i}, \vec{w}) \propto \frac{n_{m,-i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{m,-i}^{(k)} + \alpha_k)} \cdot \frac{n_{k,-i}^{(t)} + \beta_t}{\sum_{t=1}^{V}(n_{k,-i}^{(t)} + \beta_t)}$$

- Gibbs Sampling

$$doc \overset{\vartheta_{m,k}}{\longrightarrow} topic \overset{\varphi_{k,t}}{\longrightarrow} word$$

Gibbs Sampling 將會在初始設定好的 topic 數量 k 上對上圖路徑 (k 條) 進行採樣，至 $\vec{z}$ 實現收斂，而每篇文檔對應的主題分布 $\vec{\vartheta}_m$ 及每個主題下對應的詞語分布 $\vec{\varphi}_k$ 也就達到收斂。

Latent Dirichlet Allocation - Analysis on $Topic_1$

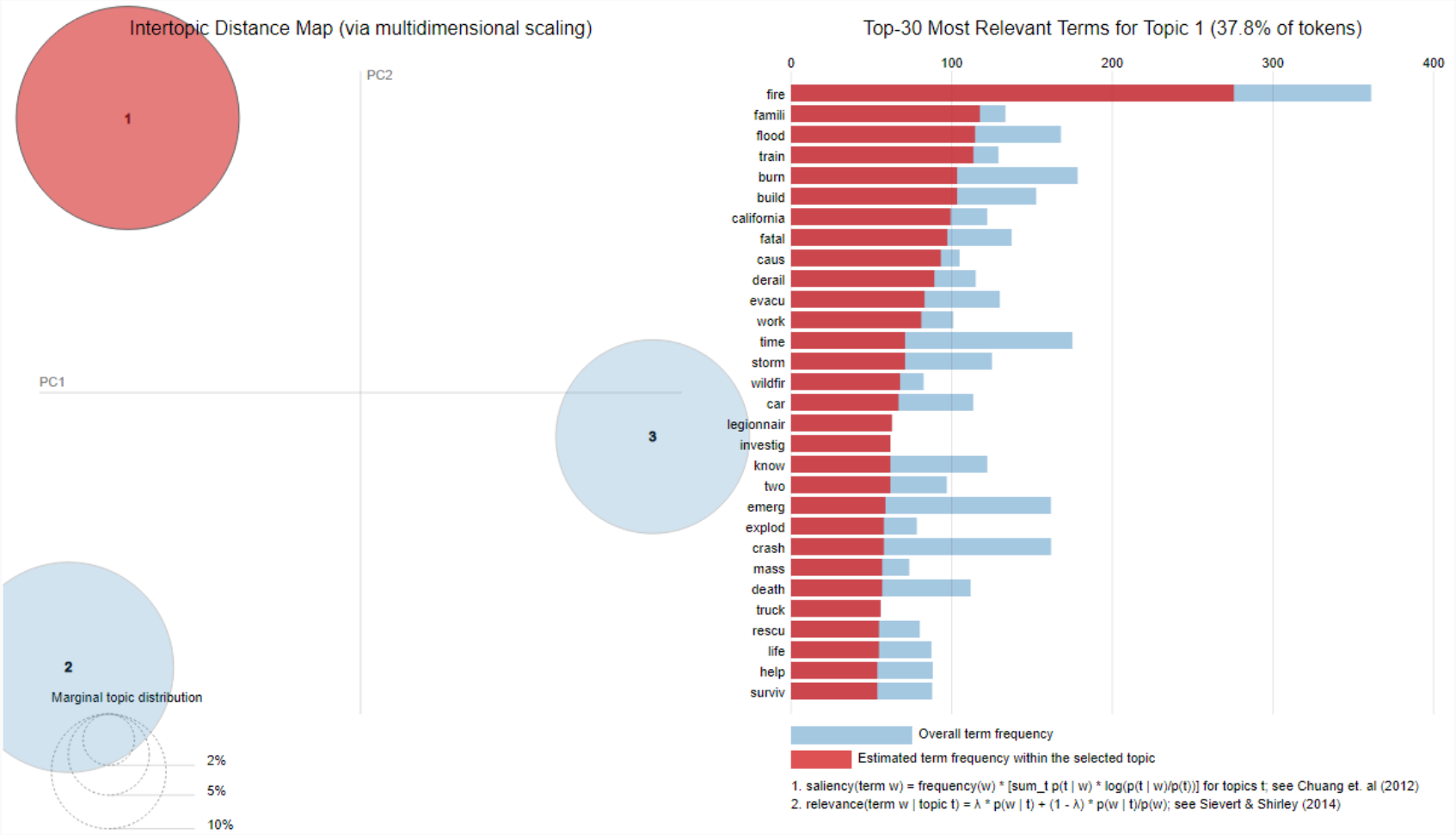


Figure 9:
$Topic_1 : fire,\ flood,\ burn,\ fatal,\ derail,\ evacu(ation),\ storm,\ wildfir(e)$

Latent Dirichlet Allocation - Analysis on $Topic_2$

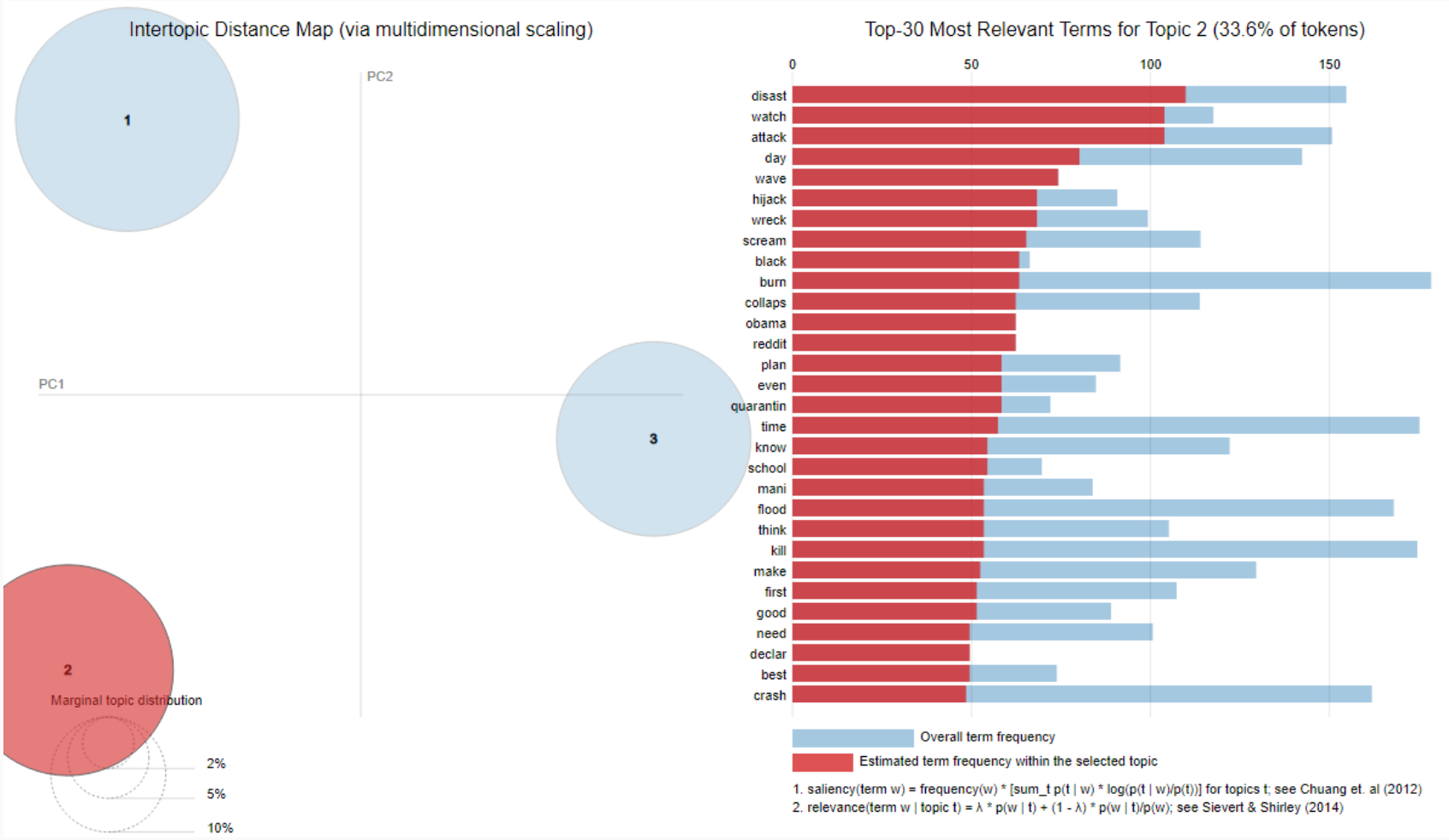


Figure 10:
$Topic_2$: *disaster*, *attack*, *hijack*, *wreck*, *burn*, *collaps(e)*, *Obama*, *reddit*

Latent Dirichlet Allocation - Analysis on $Topic_3$

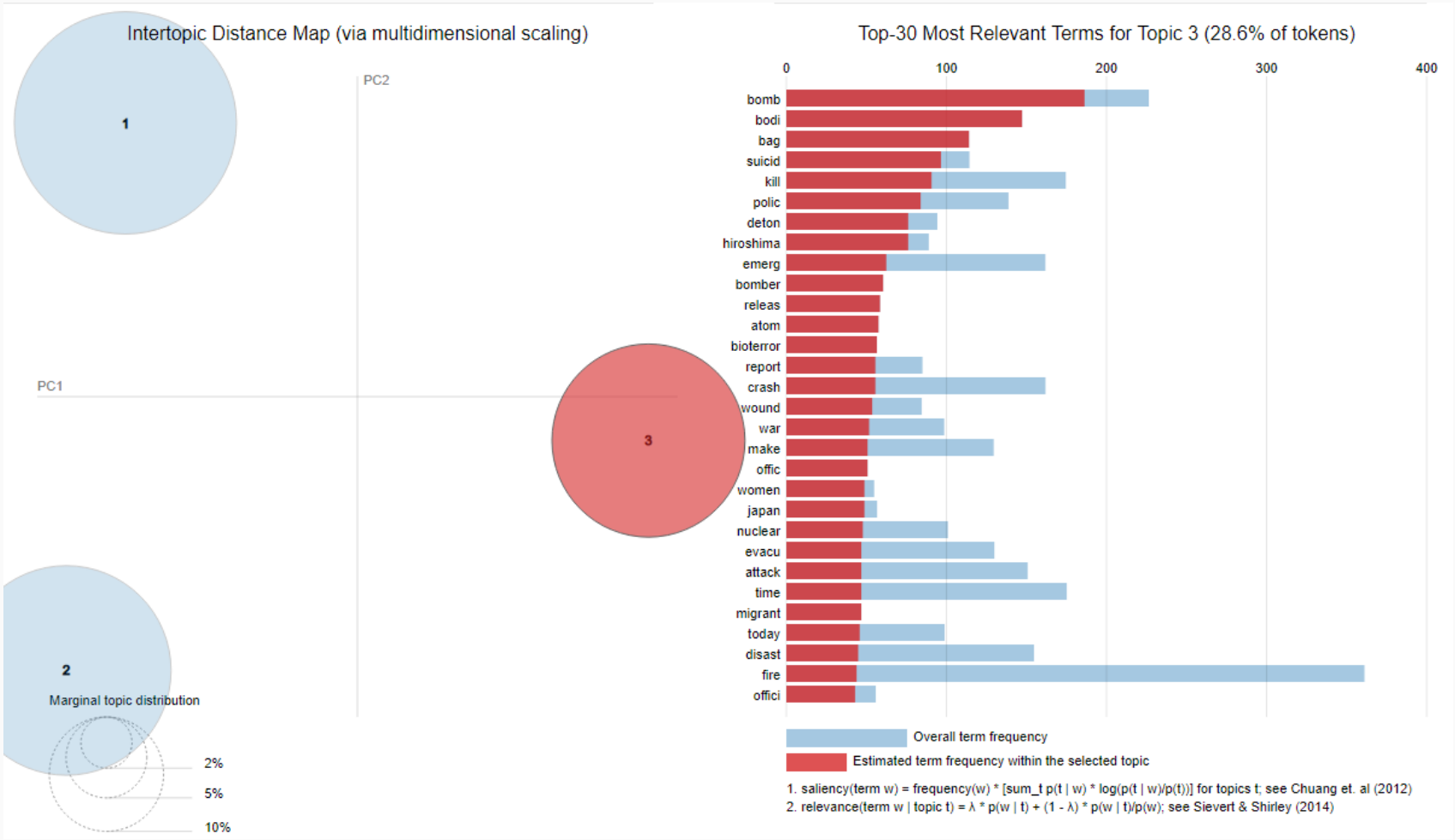


Figure 11:
$Topic_3$: $bomb$, $body$, $suicid(e)$, $kill$, $polic(e)$, $emerg(ency)$, $bioterror$

Conclusion

Conclusion

分類效果

TF-IDF 轉化影響 - 文檔 d_m 在沒有文字 $(w_{m,n})$ 的 $tf - idf = 0$

Supervised method 分群結果優於 Unsupervised method

包含 Labeled 資料，對於模型本身樣本信息更多

資料配適問題

文字資料的出現頻率、tf-idf 值，並不能對 GMM 的假設分布達到配適;

Latent Dirichlet Allocation 生成模型的方式是由文檔文字生成的方式建立

Reference

- Ramos, J.(2003). Using $tf - idf$ to determine word relevance in document queries. Proceedings of the First Instructional Conference on Machine Learning, 242, 133–142. Piscataway, NJ.
- Trstenjak, B., Mikac, S., & Donko, D. (2014). KNN with TF-IDF based framework for text categorization. Procedia Engineering, 69, 1356–1364.
- Blei, David M.; Ng, Andrew Y.; Jordan, Michael I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research. 3 (4–5): pp. 993–1022.