The best privacy defense is a good privacy offense: Obfuscating a search engine users profile

Joshua Fenech & Omar Salbrout

University of Jean Monnet

jfenech22@hotmail.com

December 12, 2017

Overview

Companies can't be trusted

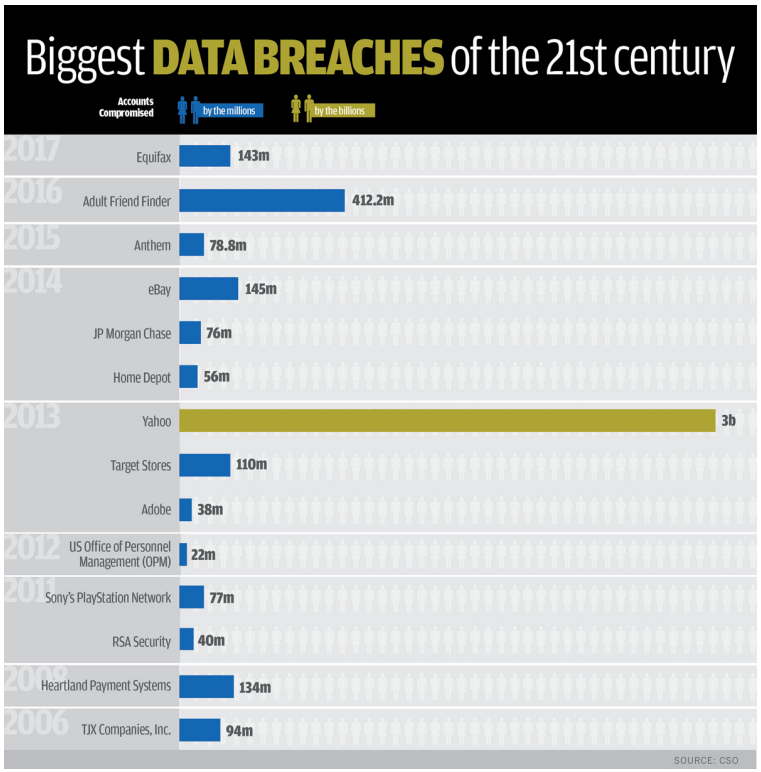


Figure: www.csoonline.com/article/2130877/data-breach/the-16-biggest-data-breaches-of-the-21st-century.html

- We trust that companies will protect our data
- Data breaches are commonplace today
- Unencrypted data is often leaked
- There is currently no or little legal requirement to protect data, and therefore represents an additional cost that some companies try to avoid
- Can we encrypt our own data before it is submitted to such companies?

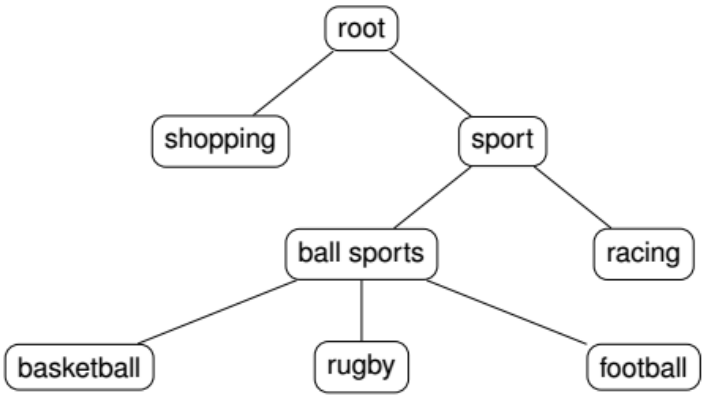
Other methods of obfuscation

- Private browsing - no cookies stored, but... IP still revealed
- Proxy servers to hide IP - web browser fingerprints still revealed
- Ultimately, TOR for maximum anonymity
- Problem - lose benefits that personalisation of websearches provides
- Can an alternative means of securing privacy without more intensive [change intensive here] methods be found?

Structure of Presentation

- How do search engines 'know' what we want?
- Present a new method of obfuscation related to adversarial data mining
- Approach is explored in common setting of Internet search engines
- A learning method is presented for environments where a user can get feedback from her or his counterpart

Personalised Advertising



- Which ad is displayed depends on The submitted query The user profile
- Ads are assigned to categories
- Users are assigned to categories

Privacy Offense

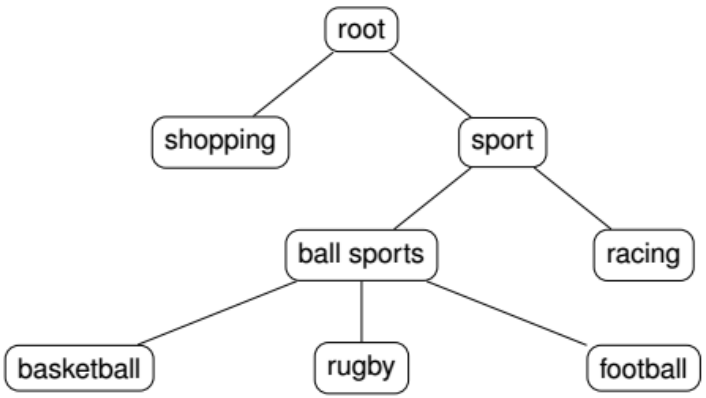
- To implement a method to defend privacy, we need:
- A way to measure the privacy, i.e. objective function
- $\sigma(\kappa_i, P) = \sum(p_j d_T(\kappa_i, \kappa_j))$
- User interest category κ, distribution of probabilities P, category tree T , tree distance d_T
- Score σ is the weighted distance between user interest category and current category the user is assigned to

Privacy Offense

- To implement a method to defend privacy, we need:
- A way to measure the privacy, i.e. objective function
- $\sigma(\kappa_i, P) = \sum(p_j d_T(\kappa_i, \kappa_j))$
- User interest category κ, distribution of probabilities P, category tree T , tree distance d_T
- Score σ is the weighted distance between user interest category and current category the user is assigned to

- To implement a method to defend privacy, we need:
- A way to measure the privacy, i.e. objective function
- Method to use feedback (ads)

Category Prediction of an Ad



- Search engines provide example queries for each category
- Use sample queries of category tree as input and train independent classifiers one for each category
- Classifiers can be applied to queries, as well as any other text
- Predictions on ads work very well due to similar structure of the text

Privacy Offense

- To implement a method to defend privacy, we need:
- A way to measure the privacy, i.e. objective function
- Method to use feedback (ads)
- A set of actions

Actions

Definition of actions to choose one category κ in the set of categories K using category tree T based on reference category $_{ref}$

- Random: $a_{random}(T, \kappa_{ref}) = randomselect(\kappa_r \in K)$

Actions

Definition of actions to choose one category κ in the set of categories K using category tree T based on reference category $_{ref}$

- Random: $a_{random}(T, \kappa_{ref}) = randomselect(\kappa_r \in K)$
- Same: $a_{same}(T, \kappa_{ref})$

Actions

Definition of actions to choose one category κ in the set of categories K using category tree T based on reference category $_{ref}$

- Random: $a_{random}(T, \kappa_{ref}) = randomselect(\kappa_r \in K)$
- Same: $a_{same}(T, \kappa_{ref}) = \kappa_{ref}$
- Sibling: $a_{same}(T, \kappa_{ref}) = sibling(\kappa_{ref} \in K)$

Actions

Definition of actions to choose one category κ in the set of categories K using category tree T based on reference category $_{ref}$

- Random: $a_{random}(T, \kappa_{ref}) = randomselect(\kappa_r \in K)$
- Same: $a_{same}(T, \kappa_{ref}) = \kappa_{ref}$
- Sibling: $a_{same}(T, \kappa_{ref}) = sibling(\kappa_{ref} \in K)$
- Most general: $a_{general}(T, \kappa_{ref}) = max\_parent(\kappa_{ref} \in K)$

Actions

Definition of actions to choose one category κ in the set of categories K using category tree T based on reference category $_{ref}$

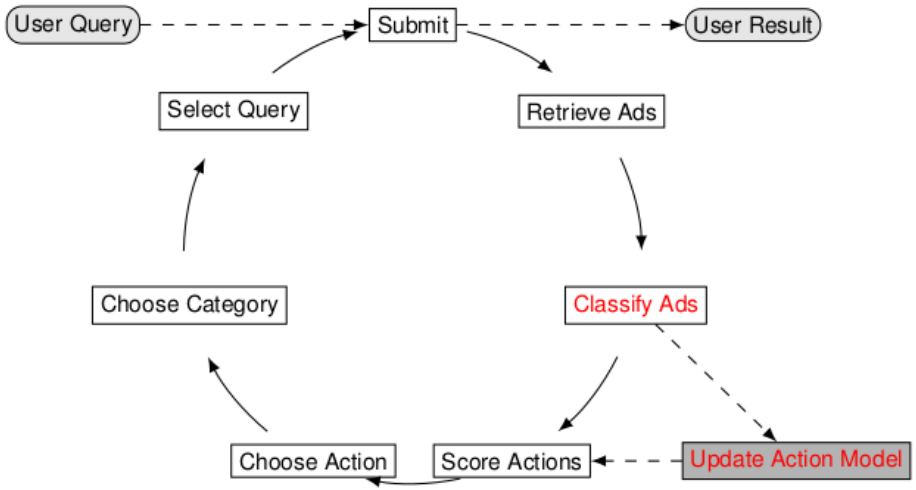
- Random: $a_{random}(T, \kappa_{ref}) = randomselect(\kappa_r \in K)$
- Same: $a_{same}(T, \kappa_{ref}) = \kappa_{ref}$
- Sibling: $a_{same}(T, \kappa_{ref}) = sibling(\kappa_{ref} \in K)$
- Most general: $a_{general}(T, \kappa_{ref}) = max\_parent(\kappa_{ref} \in K)$
- Most specialized of sibling:
  $a_{specialized}(T, \kappa_{ref} = lowest\_child(all_siblings(\kappa_{ref} \in K))$

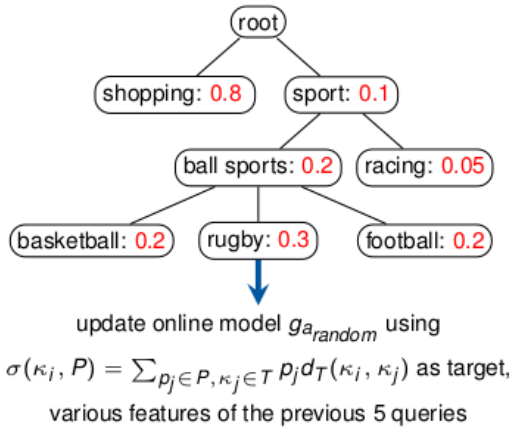
Actions

Definition of actions to choose one category κ in the set of categories K using category tree T based on reference category $_{ref}$

- Random: $a_{random}(T, \kappa_{ref}) = randomselect(\kappa_r \in K)$
- Same: $a_{same}(T, \kappa_{ref}) = \kappa_{ref}$
- Sibling: $a_{same}(T, \kappa_{ref}) = sibling(\kappa_{ref} \in K)$
- Most general: $a_{general}(T, \kappa_{ref}) = max\_parent(\kappa_{ref} \in K)$
- Most specialized of sibling:
  $a_{specialized}(T, \kappa_{ref} = lowest\_child(all\_siblings(\kappa_{ref} \in K))$
- Distance-based:
  $a_{dist}(T, \kappa_{ref}) = \kappa_r : \forall \kappa_t \in K, d(\kappa_r, \kappa_{ref}) \geq d(\kappa t, \kappa_{ref})$

Figure


User Query
Submit
User Result
Select Query
Retrieve Ads
Choose Category
Classify Ads
Choose Action
Score Actions
Update Action Model

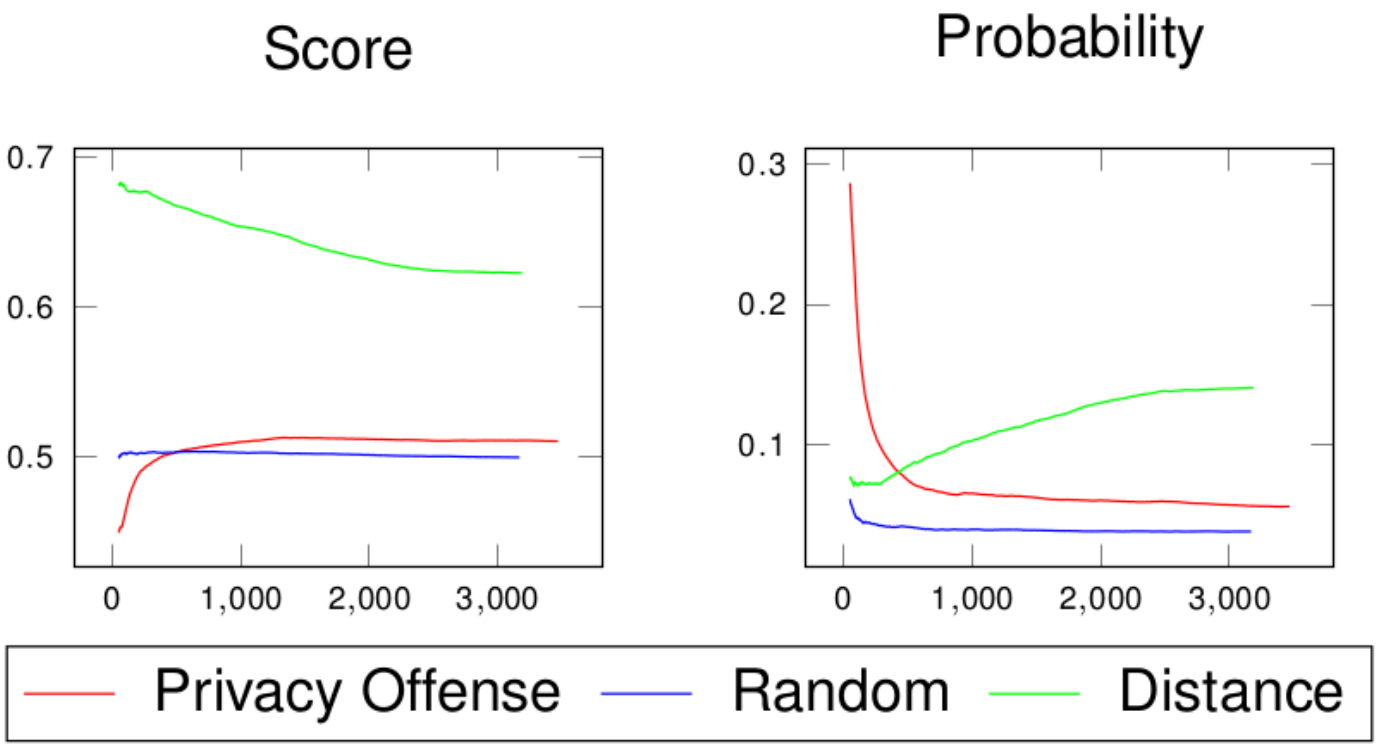

root
shopping: 0.8
sport: 0.1
ball sports: 0.2
racing: 0.05
basketball: 0.2
rugby: 0.3
football: 0.2
update online model $g_{a_{random}}$ using
$\sigma(\kappa_i, P) = \sum_{p_j \in P, \kappa_j \in T} p_j d_T(\kappa_i, \kappa_j)$ as target,
various features of the previous 5 queries

Experiments

Users are given one interest category and either:

- Use the proposed method, or
- Submit queries from random categories, or
- Submit queries from the category that is the furthest away from their interest category
- All users submit in 10% of the cases random queries from their interest category

Results


Score
Probability
0.7
0.6
0.5
0.3
0.2
0.1
0
1,000
2,000
3,000
Privacy Offense
Random
Distance

Results

- To implement a method to defend privacy, we need:
- A way to measure the privacy, i.e. objective function
- $\sigma(\kappa_i, P) = \sum(p_j d_T(\kappa_i, \kappa_j))$
- User interest category κ, distribution of probabilities P, category tree T , tree distance d_T
- Score σ is the weighted distance between user interest category and current category the user is assigned to

Conclusions

- Does it work?
  - Maybe
- Simplified Model
  - Only one interest category
  - Discard more aspects of the search engines model, e.g., time and date
- Future Work
  - More sophisticated model
  - Use more feedback than just the ads
  - Extend the use beyond search engines

Questions