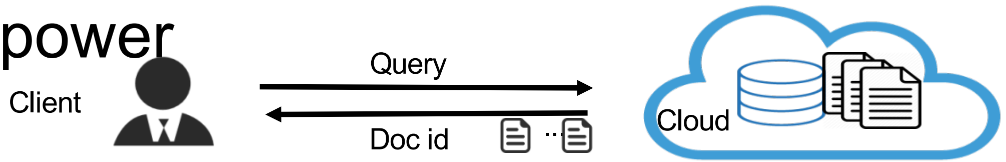
Privacy-aware Ranking with Tree Ensembles on the Cloud

Shiyu Ji, Jinjin Shao, Daniel Agun, Tao Yang
Department of Computer Science
University of California at Santa Barbara

Motivation for Private Search

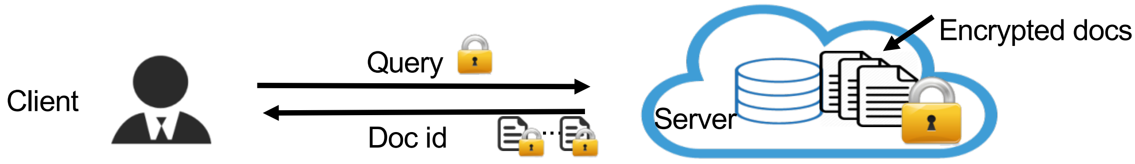
- Client uploads data to the cloud, utilizing its computing power



- Server is honest-but-curious: correctly executes protocols but observes/infers private information
  - Plain text leakage occurs due to various such as accidents, misconfiguration, or employee misuse
    - **"Dropbox Security Bug Made Passwords Optional For Four Hours".** June 2011
  - Even feature leakage such as TFIDF may cause partial document leakage.

Privacy Requirement for Top K Search

- **Given a set of documents feature vectors**
  - Each document d has encrypted feature *i* denoted as E(f_i^d)
- **Indexing and top K search scheme so that**
  - Server can access encrypted document features
  - Rank them within a reasonable response time without knowing underlying feature values
  - E.g. RankScore(E(f_1^{d1}), E(f_2^{d1})) vs RankScore(E(f_1^{d2}), E(f_2^{d2}))



Privacy Challenges in Feature Composition and Rank Computation

- **Ranking requires arithmetic computation and comparison**
  - Feature composition: e.g. TF-IDF, BM25, word distance.
  - Linear/nonlinear rank computation and comparison:
- **Computation and comparability of encrypted features**
  - Compose E($f_1^d + f_2^d$) from E(f_1^d) and E(f_2^d) securely?
  - Compare E($f_1^{d1} + f_2^{d1}$) and E($f_1^{d2} + f_2^{d2}$) securely?
  - Fully Homomorphic encryption [Gentry STOC09]: inefficient
- **No publication on private learning-to-rank tree ensembles**

Previous work on searchable encryption & private search

- **Searchable encryption** [Cash et al. Crypto13, Curtmola et al. Crypto13, Kamara12]– does not address ranking
- **Private decision trees** e.g. [Bost et al. NDSS15]
  - Use computation-heavy cryptographic techniques (e.g. Homomorphic encryption), not scalable.
- **Order Preserving Encryption** [Boldyreva et al. Crypto11] – does not support arithmetic operations
- **Leakage abuse attack of search index, features,** [Cash et al. CCS15, Wang et al. S&P17]
- **Existing research on private additive ranking:**
  - [Cao et al. TPDS14, Xia et. al. TPDS16] works for small database size.
  - [Agun et al. WWW18] relies on client-server collaborative ranking.

Overview of Proposed Private Tree Ranking Scheme (PTR)

1. Restrict computation operators and rely on more raw features
2. Query-length-specific training
3. Hide feature values and tree thresholds with comparison-preserved mapping

- Prove tree ensemble training can be competitive using raw features with restricted feature composition.
- Derive leakage profile and privacy property on what is protected.
- Evaluate relevance competiveness of PTR using TREC Datasets

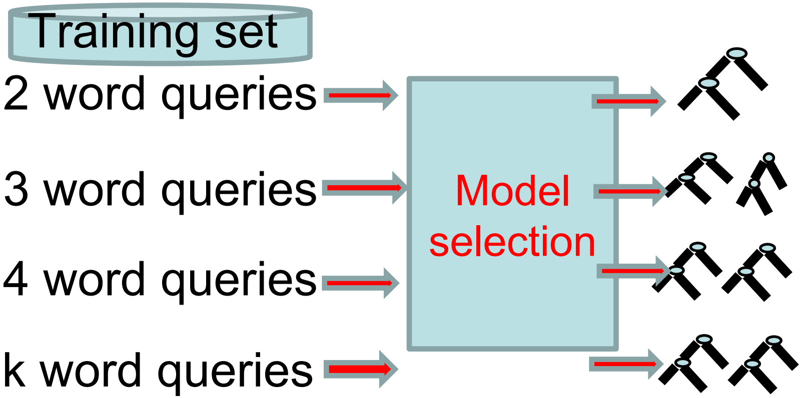
Proposed PTR: Restrict computation operators and rely on more raw features

- **More composition operation types supported → less secure**
- **Strategy:**
  - Restrict type of arithmetic operations in feature and rank computation. Only support min/max based composition from raw features
  - Rely on raw features more with tree branching composition

- For **BM25**, use individual raw features (Avoid addition)
- For **proximity**, use word pair or n-gram scores as basis. Avoid addition, or derivation from word positions

Proposed PTR: Query-length-specific training

- Number of raw features is query-dependent.
- Query-length specific training with hybrid tree ensemble



Allow a different algorithm to be used for a different query length with a different combination of raw/composite features

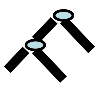
Proposed PTR: Hide feature values and tree thresholds with comparison-preserved mapping

- **Objective: Hide feature values and tree thresholds for better privacy**
- **Option 1: OPM**
  - Order preserved mapping [Boldyreva et al. Cryoto11]
  - $v_1 > v_2 \Leftrightarrow \text{OPM}(v_1) > \text{OPM}(v_2)$
  - $v_1 = v_2 \Leftrightarrow \text{OPM}(v_1) = \text{OPM}(v_2)$

- **Option 2: CPM** (Comparison preserved mapping)

Feature value/threshold mapping only preserves correctness of decision tree branching

Leak less: $v_1 \geq v_2 \cancel{\Leftrightarrow} \text{CPM}(v_1) \geq \text{CPM}(v_2)$

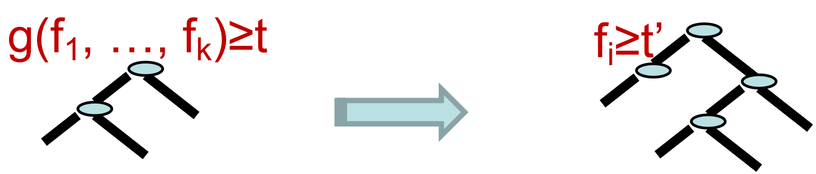


Can tree ensemble training be competitive using raw features with restricted feature composition?

Definition:

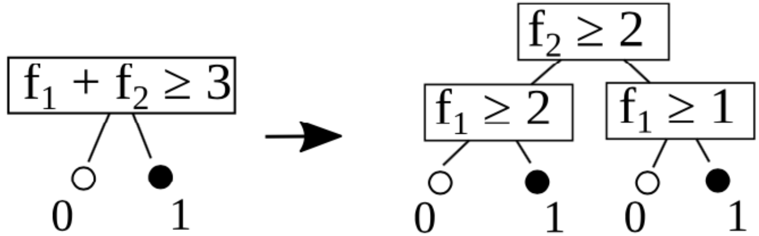
- Composition function $g(f_1, \ldots, f_k)$ is inequality-simplifiable if any inequality $g(f_1, \ldots, f_k) \geq t$ can be transformed as $f_i \geq t'$ given fixed k-1 features except f_i.
- Example: $2f_1 + 3f_2$, $f_1 \log f_2$, $\frac{1}{1+e^{-f_1-f_2}}$

Theorem: A decision tree that uses inequality-simplifiable composite features can be transformed into another tree using raw features only without training loss degradation in terms of squared error or entropy-based information gain



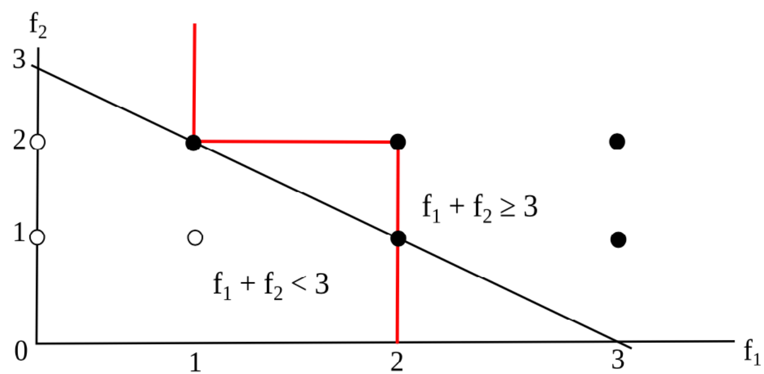
Example of tree transformation by removing sum operators

- Transform a tree with a sum-based composite feature into another tree using raw features only.



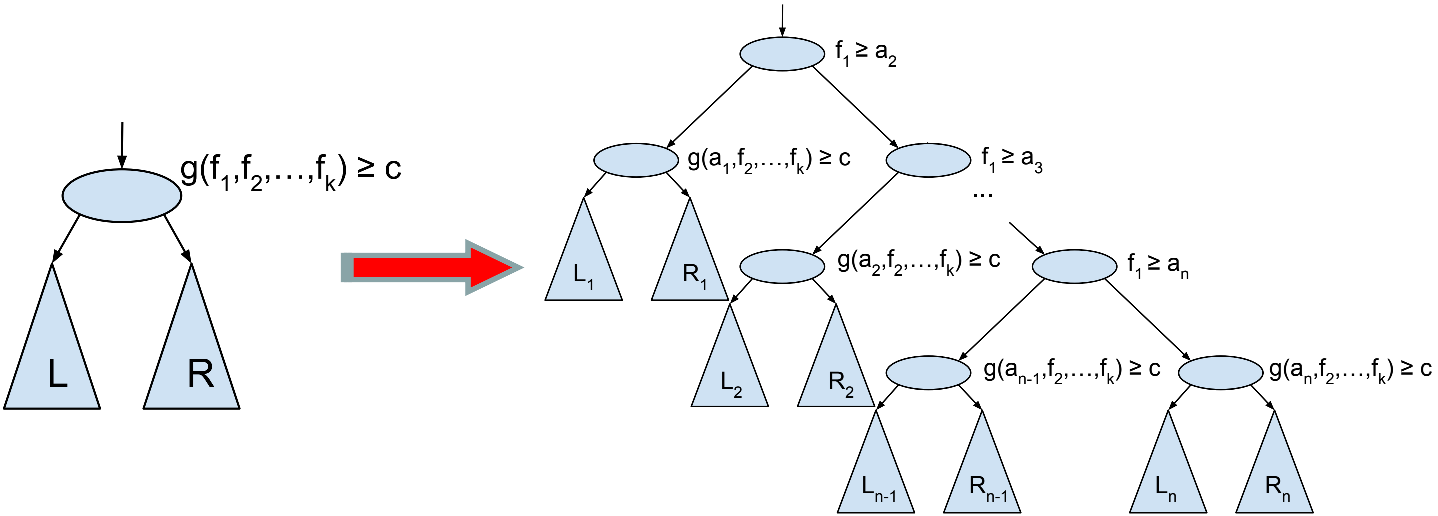
Sum is inequality-simplifiable

- The new tree can separate white and black circles as accurate as the old tree

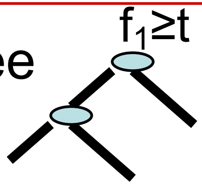


Tree ransformation with inequality-simplifiable composite features

Inequality-simplifiable composition using k raw features can be transformed, using at most k-1 raw features without loss in terms of squared error or information gain.



CPM: Comparison Preservering Mapping



Objective: Index data hides feature values and tree thresholds

Step 1: Partition document feature values and tree thresholds into disjoint comparable groups

- Each group contains all raw feature values and min/max composite features and associated tree thresholds comparable in decision trees.

Step 2: Apply CPM to each group.

Let sorted distinct thresholds be $[t_1, t_2, \ldots, t_r]$. Then $CPM(t_i) = i$.

For any feature value f, if $f < t_1$, $CPM(f) = 0$.

If f is in $[t_{i-1}, t_i]$, $CPM(f) = i-1$.

Example of CPM

Comparable group

with 3 thresholds

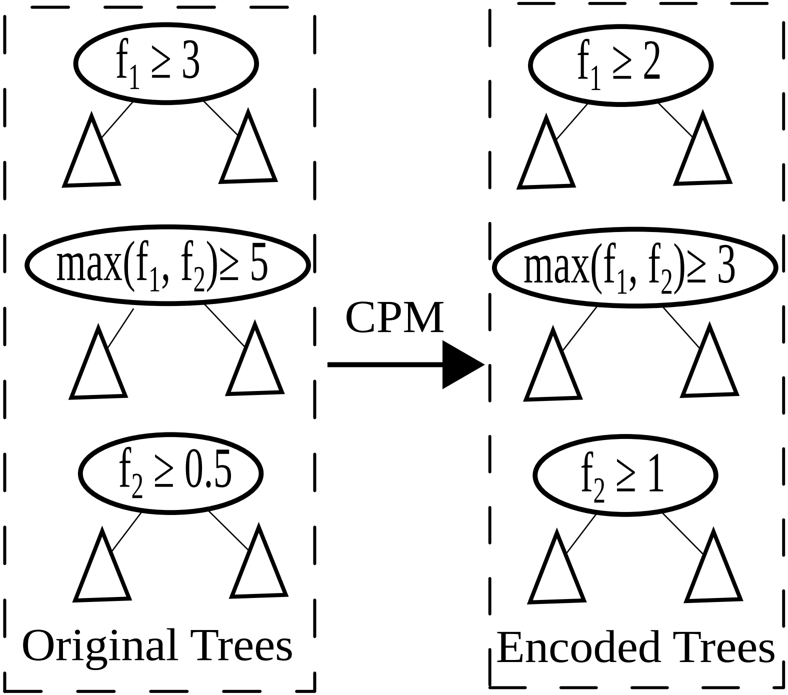
[0.5, 3, 5]

[1, 2, 3]

and 6 feature values

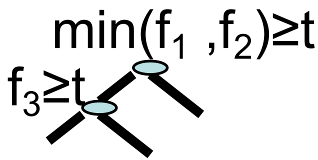
[0.3,0.8,1.5,2.5,3.8,5.1]

[0, 1, 1, 1, 2, 3]



Correctness and Space Efficiency of CPM

- Encoding of feature values and thresholds does not affect the correctness of comparison in decision tree computation
- For any feature value f and tree threshold t,
  - $f \geq t \Leftrightarrow CPM(f) \geq CPM(t)$
  - $min(f_1, \ldots, f_k) \geq t \Leftrightarrow min(CPM(f_1), \ldots, f_k) \geq CPM(t)$
  - $max(f_1, \ldots, f_k) \geq t \Leftrightarrow max(CPM(f_1), \ldots, f_k) \geq CPM(t)$



- **Storage space requirement**: each encoded value requires log N bits where N is the number of distinct tree thresholds.
  - 2-3 bytes in practice

Leakage Profile: What is leaked to the server?

- **Partial order leakage of feature values within each comparable group**
  - $CPM(v_1) > CPM(v_2) \Rightarrow v_1 > v_2$
  - $CPM(v_1) = CPM(v_2) \not\Rightarrow v_1 = v_2$
- **Partial distribution information:** The number of distinct thresholds, the number of encoded feature values between two consecutive thresholds in each group.
- **Tree ensemble structure information:** 1) the number of trees, 2) the topology of each tree, 3) the membership of comparable group, 4) score value difference between every two leaves in a tree.

Privacy Properties: What information is protected

- Server cannot compare feature values and thresholds associated with different comparable groups.
  - Within the same group, $CPM(v_1) = CPM(v_2)$, the server cannot figure out the order of v_1 and v_2
- A server cannot well approximate the actual values of feature values, their difference, and their ratios.
  - If it can do within an error bound, then it has to distinguish the original data from an infinite number of other possible datasets beyond the error bound, which is unlikely.
  - Cannot well approximate actual values and their difference of thresholds

Evaluation

- **Privacy-aware indexing and runtime support**
  - Key-value store scheme to fetch feature values for private search [Agun et. al. WWW 2018]
- **Evaluation objective:** Can PTR with hybrid tree ensembles using raw and min/max compositions perform competitively?

Query length	1	2	3	4	5
Robust04, 0.5M	11	70	140	25	4
Robust05, 1M	1	19	24	5	1
ClubeWeb09-12, 50M	64	70	52	14	0
ClubeWeb, MQ09, 50M	98	294	232	53	9

Relevance of PTR with Restricted Features

Compared to Existing Methods with no Restriction
5-fold validation NDCG@20 results

Collections	λ-MART	GBRT	Random Forest	PTR
Robust04	0.3936	0.4025	**0.4114**	0.3975 (-3.3%)
Robust05	0.2765	0.2778	**0.2945**	0.2928 (-0.6%)
ClueWeb09-12	**0.2235**	0.1906	0.2100	0.2160 (-3.4%)
ClueWeb09, MQ09	**0.2603**	0.2419	0.2395	0.2573 (-1.2%)

PTR is close to the best constantly with small degradation

Relevance with different query lengths

NDCG@20 of ClueWeb09, MQ09. Features include raw indivdual BM25 for title/body, word-pair distance with min/max composition, PageRank, and Wikipedia indicator

Q-length	λ-MART	GBRT	Random Forest	PTR
2	**0.2712**	0.2457	0.2612	**0.2712** (0%)
3	0.2683	0.2185	0.2284	**0.2767** (+3.1%)
4	0.2280	0.2296	**0.2369**	0.2296 (-3%)
5	**0.0913**	0.0843	0.0388	**0.0913** (0%)

PTR gives the more stable results than others by selecting the best configuration with query-length specific optimization.

Contributions and Conclusions

- Addressed an open problem for server-side privacy aware ranking using tree ensembles.
- Three techniques are proposed in private tree ranking (PTR) scheme
  - Restricting decision trees using raw features and min-max composition is a sound tradeoff for privacy with competitive relevance.
  - Query-length specific training
  - Comparison-preserving mapping scales well for large datasets with sound privacy properties.
- Future work is to consider other nonlinear ranking including neural nets.