Deep Entity Classification: Abusive Account Detection for Online Social Networks

Teng Xu\textsuperscript{1}, Gerard Goossen\textsuperscript{1}, Huseyin Kerem Cevahir\textsuperscript{1}, Sara Khodeir\textsuperscript{1}, Yinglezehe Jin\textsuperscript{1}, Frank Li\textsuperscript{1,3}, Shawn Shan\textsuperscript{1,2}, Sagar Patel\textsuperscript{1}, David Freeman\textsuperscript{1}, and Paul Pearce\textsuperscript{1,3}

\textsuperscript{1}Facebook, Inc \quad \textsuperscript{2}University of Chicago \quad \textsuperscript{3}Georgia Institute of Technology
Problem

- Clickbait
- Spam
- Harassment
- Bullying
- Hate Speech
- Nudity

**Abusive account:** an account created for the purpose of abuse (i.e. activity that goes against Facebook's Community Standards).
Abusive Accounts on Facebook

Estimated 5% of monthly active users are abusive accounts.¹

Took down 1.3 Billion Abusive Accounts from 2020 Q4¹, most within minutes of registration, before they could become active users.

¹. Facebook community standards enforcement report: https://transparency.facebook.com/community-standards-enforcement#fake-accounts
Machine Learning Based Detection

- Manual review does not scale
- Heuristic rules are hard to create and maintain
- Adversaries move fast
ML: Traditional Approach

**Account Features**
- Location
- Number of posts
- Number of friends

**Manual Labels**
- Benign
- Abusive

**Model Architecture**
- GBDT
- Deep neural network

...
Solution: deep entity classification

<table>
<thead>
<tr>
<th><strong>Problem</strong></th>
<th><strong>Solution</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Features can be gamed by attackers.</td>
<td>Extract “deep features” of accounts by aggregating properties and behavioral features from direct and indirect neighbors in graph.</td>
</tr>
<tr>
<td>Features are hand written, which only scales to hundreds of features.</td>
<td>Define dozens of features per edge, apply to all edges, and recursively traverse the graph, resulting in tens of thousands of features.</td>
</tr>
<tr>
<td><strong>Obtaining large amounts of ground truth data is difficult.</strong></td>
<td>Use a multi-stage multi-task learning technique using large amounts of low-precision automated labels, and small amounts of high-precision human labels.</td>
</tr>
</tbody>
</table>
Deep Feature Extraction

First order

- Apply aggregation functions to direct features of fanout entities.
- Numeric aggregation functions:
  - max
  - min
  - mean
  - p75
  - p25
  - variance
- Categorical aggregation functions:
  - percentage of the most common category
  - percentage of empty values.
  - entropy of the category values.
  - number of distinct categories.

Avg (# of groups per friend) = 3
Most common percentage (friend country) = 0.67
Deep Feature Extraction

Second order

- Apply aggregation functions to second order fanout entities.
- Aggregate results over first order fan-outs.
- Lots of features, expensive to calculate.

\[
\text{min}_{\text{friends}} \left( \max_{\text{posts}} \left( \# \text{ of likes per post} \right) \right) = 11
\]
How do we avoid overfitting and also obtain benefit of high-quality labels?

**Multi-stage multi-task learning (MS-MTL)**
MS-MTL Model: stage 1

Diagram showing the flow of the MS-MTL model, starting with machine data, followed by a low precision multi-label vector and deep features, leading to a deep neural network with input, hidden layers, and output layers. The output is then embedded.
MS-MTL Model: stage 2
Model Comparisons

1. Only behavioral features + GBDT
2. DEC features + single stage deep neural network (SS)
3. DEC features + MS-MTL
Offline evaluation

**ROC Curve**

- True Positive Rate vs. False Positive Rate

**PR Curve**

- Precision vs. Recall
- Precision at 0.95
Online evaluation

In production: precision over 30 days

In production: recall over 30 days
## Takeaways

1. Extracting graph-based “deep features” of accounts allows us to scale features and resist adversarial adaptation.

2. MS-MTL training leverages both high quantity-low precision, and low quantity-high precision training data to improve model performance.

3. DEC’s two-year deployment has resulted in Facebook taking down hundreds of millions of abusive accounts.

4. Counterintuitively, the deployment of DEC reduced global CPU usage on Facebook despite the high computational load.
Thank you

Contact: xuteng@fb.com for questions and further information