Leveraging network topology for better fake account detection in social networks

Björn Bebensee
bebensee@snu.ac.kr

Nagmat Nazarov
nagmat@snu.ac.kr

Seoul National University

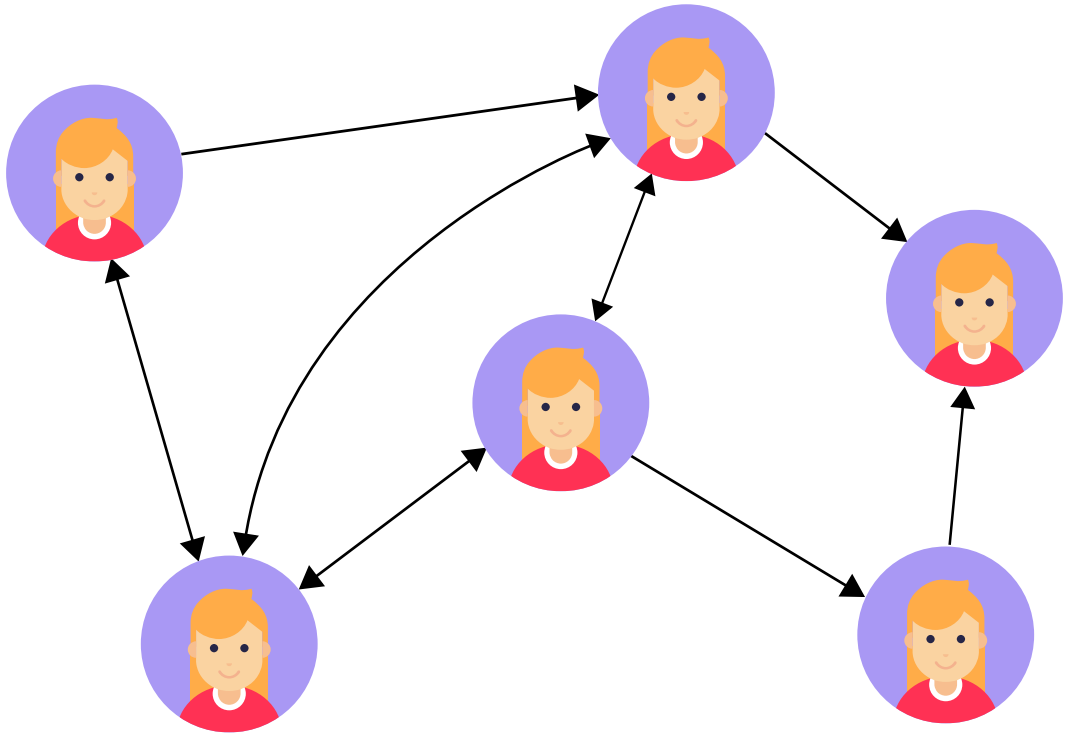
December 2, 2019

Contents

Motivation

Observation

There is a high number of fake accounts in social networks which typically serve malicious purposes.



Problem definition

Given: directed graph $G = (V, E)$ of users and their interactions (i.e. who follows who)

Goal: identify all fake accounts in G

Dataset

We are using labelled data by Cresci, 2018.

This dataset contains users trying to influence stock prices by manipulating public opinion Twitter.

Originally contained: 25,987 accounts, 7,479 human, 18,508 bots.

Dataset

After scraping the still existing Twitter users we end up with a total of 13,091 accounts:

6,082 human accounts
7,009 bot accounts.

Dataset

Retrieved data for each user:

- user_id
- label
- username
- screen_name
- number of followers
- number of following
- location
- url
- description
- number of times user appears in lists
- number of favourites
- number of status
- account age
- default profile
- default profile image
- list of users followed (up to first 5000)
- list of followers (up to first 5000)

Dataset

Extend dataset with neighbors in the social graph.

We scraped profile data of all users followed by and following users in the dataset.

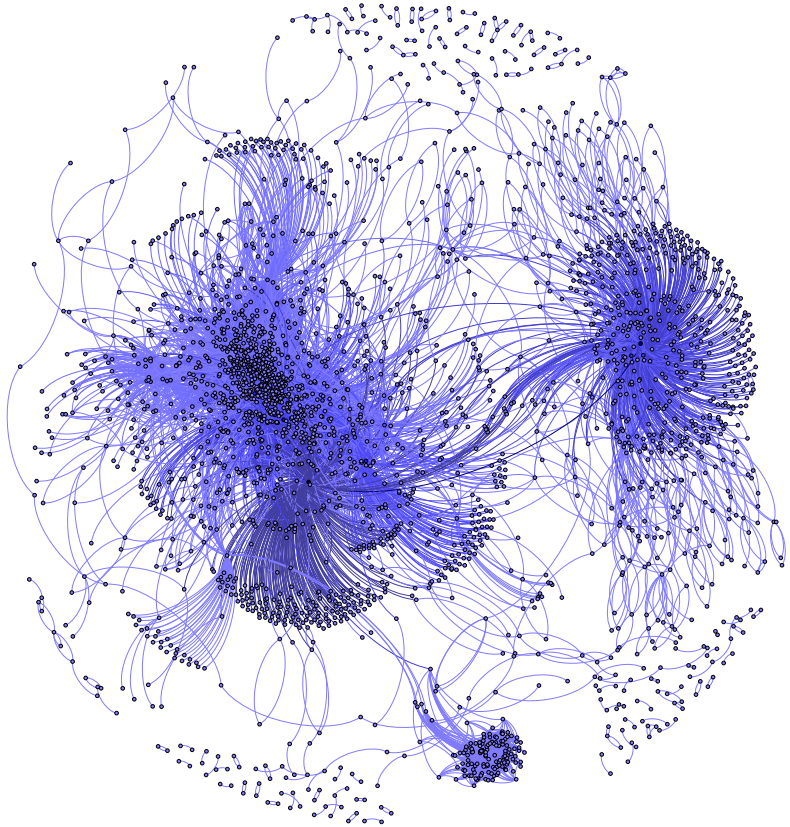
Total: 4.6M unlabelled accounts, 13,091 labelled accounts

Social graph

Vertices: 4,611,170

Edges: 8,514,389

Connected components: 1,244



Social graph

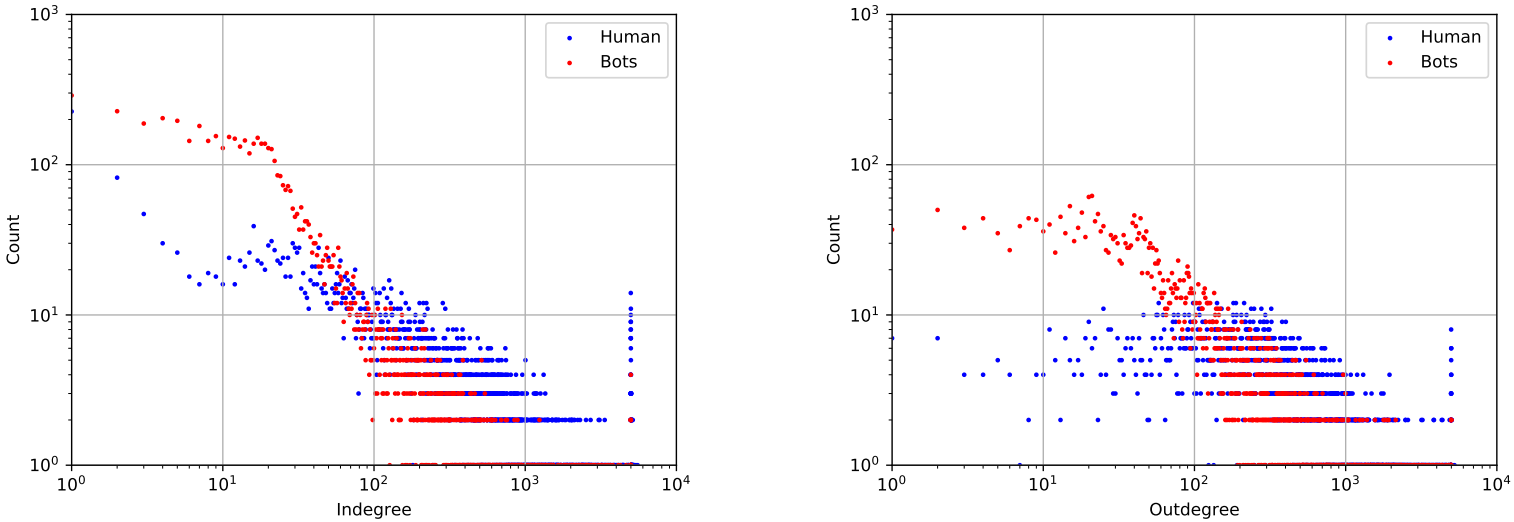
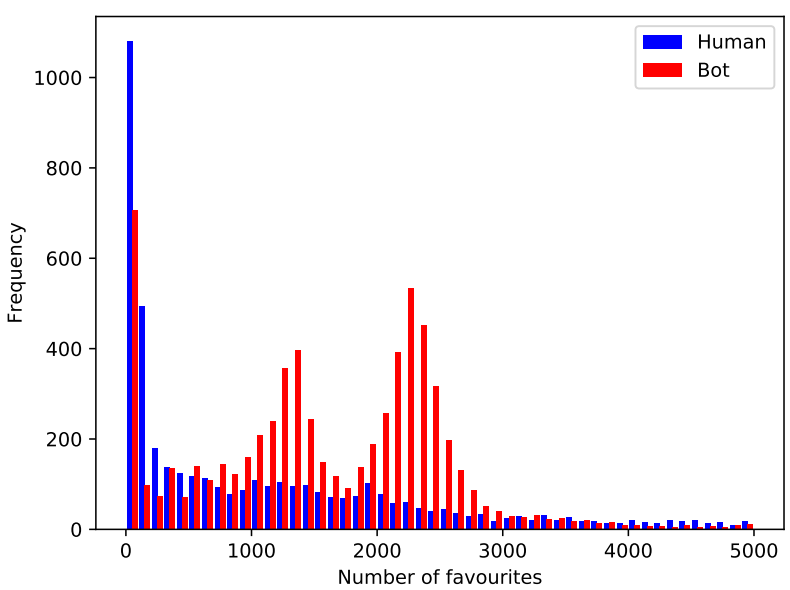


Figure: Degree distributions of human and bot accounts look different

Social graph



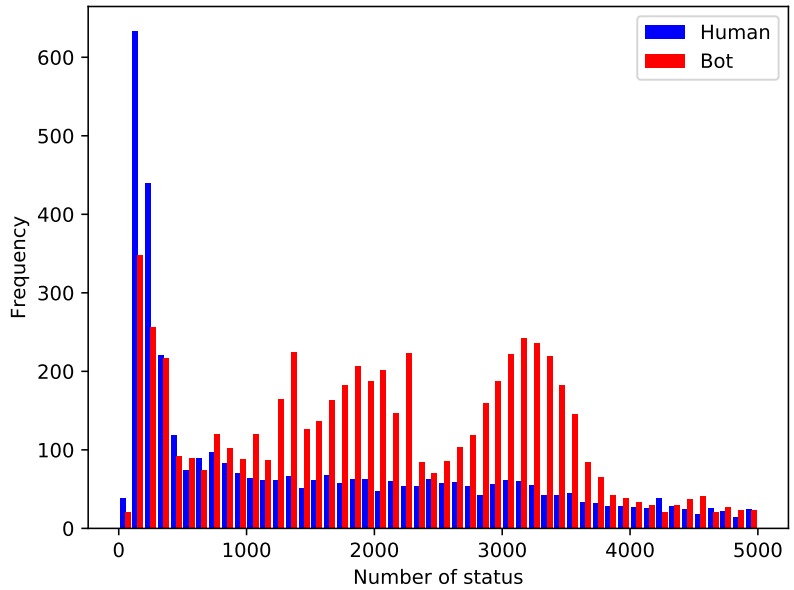


Figure: We can see clear differences in number favorites and status updates as well

Social graph

Some bot accounts have many followers (and some follow many users). But also: bot accounts seem more likely to be followed by other bots.

Observation

Fake accounts and bots do not have the same social structure (and structure in the social graph) as real users.

Idea: Classify user accounts by looking at their neighbors and their egograph!

Proposed method

We use *neighborhood features* and *egograph features* to improve classification.

Using the social graph we created we can extract additional features to aid in classification of bot accounts.

Proposed method

We experimented with many features, but it turns out a lot of them are noisy and do not improve classification for this dataset.

The best features we found:

- median out-degree of predecessors
- median favorites of predecessors
- median status count of predecessors
- median account age of predecessors
- egograph density
- egograph reciprocity

Proposed method

Observation

Features aggregated over predecessors of a user are more helpful and less noisy than those of successors.

Intuitively makes sense: easy to *follow* "normal" users, but it is much less likely for a fake account to *be followed* by "normal" users.

Experiments

Evaluate our model against some baselines. We train the following models as a baseline:

- Gaussian Naïve Bayes (GNB)
- Quadratic Discriminant Analysis (QDA)
- Support Vector Machine (SVM)
- k-Nearest Neighbors (KNN)
- Random Forest (RF) model
- Feedforward Neural Network (NN)

Use basic profile features retrieved using the Twitter API (as done by Kudugunta and Ferrara, 2018)

Experiments

We evaluate using the following measures:

- Accuracy
- True positive rate (TPR)
- False positive rate (FPR)
- F_1 score $\left(2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\right)$
- Area under ROC curve (AUC)

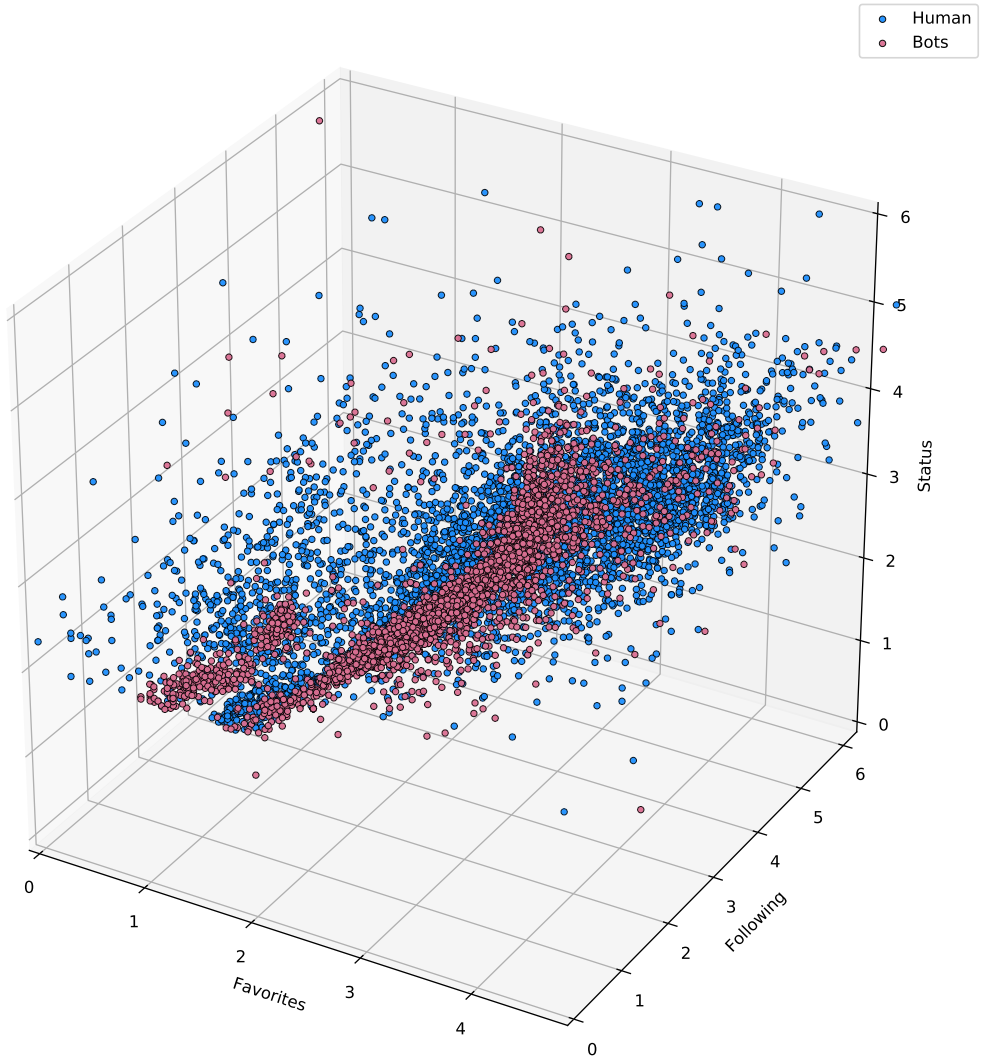
Experiments

Model	Accuracy	TPR	FPR	F_1 score	AUC
GNB	0.6399	**0.9622**	0.7313	0.741	0.7652
QDA	0.6751	0.8937	0.5768	0.7465	0.6405
SVM	0.7797	0.7746	0.2145	0.7901	0.7801
KNN	0.8496	0.8752	0.18	0.8617	0.9074
RF	0.8675	0.8745	0.1405	0.876	0.9426
NN	0.8648	0.8673	0.138	0.8729	0.9154
NN + NF (ours)	0.8702	0.8623	**0.1208**	0.8767	0.928
NN + NF + GF (ours)	0.8751	0.8894	0.1413	0.8841	0.9367
RF + NF + GF (ours)	**0.8774**	0.888	0.1348	**0.8858**	**0.9468**

Table: Results on the CRESCI-2018 dataset for various baseline models (top) and our model using neighborhood features (bottom).

Experiments

But: few certain features sufficiently separate this dataset in multi-dimensional space.



Experiments

RF classifier trained only on three (!) features

$$X = \{\texttt{favourites\_count}, \texttt{following}, \texttt{status\_count}\}$$

can achieve F_1 score of 0.8738 and AUC of 0.9313.

Conclusion

In hindsight: dataset may not be ideal.

Data in Cresci-2018 is algorithmically annotated based on suspicious behaviour.

But: Genuine users may exhibit suspicious behaviour too! Even human accounts might try to influence stock price (e.g. if they hold the stock).

It would be interesting to see how our approach performs on a higher quality dataset.

Conclusion

We proposed a novel method to improve bot classification in graphs using data from adjacent, unlabelled nodes.

Our approach provides a "free" improvement upon baseline models.

Future work: evaluate on a different dataset.

Thank you for your attention.