# Anomaly Detection and Featurization on Social Media Platform

Joel Christiansen

## OVERVIEW

Detecting anomalous accounts in social media is a problem which requires the ability to both retrieve and understand massive amounts of data. This capability can be achieved using a combination of various techniques, all of which have many alternatives with various advantages and disadvantages specific to the situation they are being applied to. In this paper we covered the various methods by which it is possible to gather, quantify, and categorize the data needed to correctly identify a social media account as anomalous.

## MY CONTRIBUTIONS

The primary focus of my portion of the paper is on featurization, the quantification and selection of numerical features that distinguish a social media account as "anomalous" or "not anomalous". This selection is critical to the successful categorization, as a feature set that is too large, too small, or the wrong set of features can easily make the difference between a 60% accuracy and a 95%+ accuracy[1].

The easiest, most obvious, and optimal solution to this problem is to simply search through the space of all possible feature sets and select the best one. Unfortunately, this has one major drawback; it is computationally infeasible to do so, with a time complexity of $O(2^n)$. As such, the majority of research into featurization involves approximating this approach or otherwise devising algorithms that, while not optimal, are good enough to return a viable feature set. This is the area my contributions to the paper focused.

There are a few commonly used approaches to selecting which features to include in the final set to be used for training the classifier. The most common is to use a greedy algorithm, whereby you start with either the full set of features or the empty one, and iteratively add or remove (respectively) one feature at a time until no more improvement is gained[2]. It is also possible, though less common, to use a genetic algorithm to find a usable feature set. There are tradeoffs to all three of these approaches, and the best approach possible will heavily depend on the specific goals you hope to accomplish.

The advantage to a greedy algorithm that starts with an empty feature set is speed. The method of testing a feature set is usually to simply train the network with that set of features and see what accuracy you can get with that set. Because it is much faster to train a network with a small number of features than with a large one, it is faster to start with an empty set than with a full one and you will receive your final set that much sooner. The downside to this method is that it is possible that while neither one feature nor another provide an advantage on their own, the two together do. A greedy algorithm that starts with an empty set will miss these interactions, reducing the accuracy of the final classifier.

A greedy algorithm that starts with the full set of features does not suffer from the same downside. The tradeoff in this case is again speed, for exactly the same reason. Every feature that is used to train a network increases the time it takes to do so. While this does not impact the time complexity per se, it *does* impact the real-world time taken to test features significantly. This approach works best when you know the final feature set will have a high degree of interconnectivity, or when the initial set is small.

The last set of algorithms I found in my research was genetic algorithms. These algorithms do not suffer from the long testing time of a full set greedy algorithm, nor do they suffer from the missed connections found in the empty set greedy algorithms. The tradeoff for genetic algorithms are instead in unreliability and complexity. With either of the greedy approaches, you know that given a set of possible features, you will always receive the same output set for the same input set. Genetic algorithms do not have that guarantee, as there is some randomness inherent to the functionality of the algorithm that makes it less reliable. It is also far more difficult to implement these algorithms, as greedy algorithms are quite simple and genetic algorithms are very much not. However, what genetic algorithms give up in terms of simplicity and reliability, they make up for in results. A genetic algorithm consistently outperforms both types of greedy algorithms in the final feature set returned.

In addition to general feature selection algorithms that are applicable to any classification problem, I also looked at deriving features that are specific to social media accounts. These features generally involved different ways of looking at how different accounts may be connected to one another in a quantifiable way.

One such set of features looks at what could be categorized as "following", tracking which accounts are watching which other accounts[3]. For example, the measure of "Account A and account B are both following account C" would be a CoFollowing relationship on account C. The measure of "Account A is followed by account B and account C" would be a CoFollowed relationship on account A. These kinds of relationships allow us to categorize an account not only on how the account itself acts, but on which other accounts it choses to relate to and how *they* act.

Another approach to choosing features is to look at the relevancy of a feature. In broad strokes, the intuition to this is that if you have two features X and Y and the set of features S, if given feature X feature Y does not have an impact on the classification then feature Y is irrelevant with respect to feature X.[4] If this is only true for some subset of S, then it is a weakly relevant feature with respect to X.

The last kind of feature I looked at was the primacy of a feature. That is, given feature X, the probability of an instance being classified in a certain way given X is not the same as it being classified without X. If this is true, then X is considered primary. Otherwise, it is considered contextual. This kind of feature ties into the relevancy of a feature. If a feature is primary it must be strongly relevant, otherwise it may be either weakly relevant or irrelevant.

## CONCLUSION

The featurization of an instance is critical to the successful categorization of it regardless of the type of instance being looked at, be it a text document or a social media account or anything else. In the context of social media, we have several additional features and classes of features that can be applied to the categorization problem that may increase the accuracy and depth of the categorization and allow us to better detect anomalies and bad actors. This increased depth is achieved by looking at social media accounts not just as discrete things with no relation to each other as is normal with other kinds of categorization, but also as interconnected instances with several kinds of relations between one another.

## REFERENCES

[1] Dy, J. G., & Brodley, C. E. (2000). Feature selection for unsupervised learning

[2] Vafaie, H., & De Jong, K. (1993). Robust feature selection algorithms. Paper presented at the 356-363. doi:10.1109/TAI.1993.633981

[3] Tang, J., & Liu, H. (2014). Feature selection for social media data. *ACM Transactions on Knowledge Discovery from Data (TKDD), 8*(4), 1-27. doi:10.1145/2629587

[4] Turney, P. D. (2002). The identification of context-sensitive features: A formal definition of context for concept learning.