# The many faces of guilt-by-association

**Sofus A. Macskassy** SOFMAC@FETCH.COM
Fetch Technologies, 841 Apollo St., El Segundo, CA 90245

## 1. Introduction

This past decade has seen a remarkable growth in the availability of networked data and interest in analyzing and understanding these networks. This interest spans a wide variety of fields, each having a different perspective and set of analytic tools they bring to bear. Many of the analytical tools available for large networks evolve around global and local metrics to characterize or describe the network. These methods are often generative such as the exponential models from social network analysis (e.g., (Wasserman & Pattison, 1996)) or the Kronecker models that have come out of computer science (e.g., (Leskovec & Faloutsos, 2007)). We have also seen validation of the small worlds phenomena and in particular the six-degrees separation on data such as LinkedIn (Leskovec et al., 2008). More sophisticated tools such as clustering (e.g., (Clauset et al., 2004)) or relational learning (e.g., (Taskar et al., 2001; Neville & Jensen, 2007; Richardson & Domingos, 2006)) are often only tractable on relatively small networks.

We here consider the problem of analyzing the network in the context of *within-network* classification (Macskassy & Provost, 2007). In this setup, one is given a homogeneous weighted network that is partially labeled. A homogeneous network consists of one type of entity (e.g., all people or all companies) and has undirected weighted edges that represent the strength of the relation between the two entities (e.g., how often they call each other, cite each other, co-occur in text, etc.). Further, the network has been partially labeled, meaning that some of the nodes have been categorized or tagged.

This setting lends itself to a variety of interesting problems that can be solved or questions that can be asked. For example, some of the problems that can be addressed in this setting include:

- What is the influence of a given node? We could compute the information centrality of the node (computationally very expensive), or we could use simpler guilt-by-association models to *approximate* the influence of the node much faster.
- Given a set of "interesting" nodes, what are some other interesting nodes? We could build a relational model which took the *profile* of the entities and their relations into account to build a predictive model (computationally expensive) or we could run a fast approximate inference algorithm such as guilt-by-association.
- Given a set of "interesting" nodes which also have profile information, what are the underlying similarities between the nodes of interest that set them apart from the other nodes (and how can we find more like them?). We could use sophisticated learning algorithms to build predictive models or we could leverage the assumption of guilt-by-association and identify underlying implicit relations between these nodes that would make them "closer" with respect to guilt-by-association. These relations can be used twofold: (1) they are by themselves interesting because they "explain" how the nodes are similar in an intuitive way and (2) we can use these relations to augment the network and find other "interesting" nodes using guilt-by-association.
- Given examples of different categories of nodes (e.g., companies in different sectors, people of different demographics), compute the likely categories of the remainding nodes in the network. We can here use graph-partition methods (computationally expensive) or faster approximate methods such as guilt-by-association.
- Getting labels for nodes in a network can be expensive. In this case, if we wanted to categorize all the nodes in the network, which minimal set of nodes should we get labeled? There is a rich literature known as active learning which considers this problem. State-of-the-art methods on networks are expensive to use to identify the nodes which ought to be labeled. By leveraging guilt-by-association and standard social network analysis metrics, we can speed the process up by orders of magnitude.

The core theme in the above problem formulations is that many existing sophisticated methods are computationally expensive and not tractable on large networks. However, fast approximate methods can be often be used to address all of the above problems with little to no loss in performance of the final analysis over that of more sophisticated methods (see (Macskassy & Provost, 2007)).

We next describe our approach for fast inference and then provide some details of how this can be used to answer the above questions.

## 2. Guilt-by-Association and Relaxation Labeling (wvRN-RL)

The approach that we take is to use a guilt-by-association model which we call the weighted-vote relational neighbor with relaxation labeling (wvRN-RL) (Macskassy & Provost, 2007). This particular model leverages an underlying assumption of *homophily* (the principle that a contact between similar people occurs at a higher rate than among dissimilar people (McPherson et al., 2001)). In particular, we can estimate the "interestingness" or "category" of a node by aggregating the "interestingness" or "category" of the neighboring nodes. Simply put, the likelihood that a node belongs to a category is the (weighted) ratio of neighbors that belong to that category. We can put this in mathematical terms as:

$$P(x_i = c|\mathcal{N}_i) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_i} w_{i,j} \cdot P(x_j = c|\mathcal{N}_j), \quad (1)$$

where $\mathcal{N}_i$ is the set of neighboring nodes to node $i$, $w_{i,j}$ is the weight of the relation between node $i$ and node $j$, and $P(x_i = c|\mathcal{N}_i)$ is the probability that node $i$ belongs to category $c$ given its neighborhood. This is clearly a circular definition as neighbors are dependent on each other. To address this, we use a technique known as *collective classification* or *joint inference* (see, e.g., (Jensen et al., 2004)), which finds the most likely joint categorization of all nodes simultaneously. Doing exact joint inference is often intractable to do on graphs of even medim size (a few hundred nodes), so we instead use a fast *approximate inference* method. There are numerous possible such approximate methods including Gibbs Sampling, Iteratice Collective Classification, (Loopy) Belief Propagation and Relaxation Labeling. All these inference method, with the exception of Iterative Classification and Relaxation Labeling, are still quite computationally expensive and generally not tractable on large graphs. We therefore use a simulated annealing variant of *relaxation labeling*, which "freezes" current estimates of all nodes and use those to compute the "new" estimates in the following manner:

$$\mathbf{c}_i^{(t+1)} = \beta^{(t+1)} \cdot \text{wvRN}(\mathbf{C}^{(t)}) + (1-\beta^{(t+1)}) \cdot \mathbf{c}_i^{(t)}, \quad (2)$$

where $\mathbf{c}_i^{(t)}$ is a vector of probabilities (probability distribution) which represents an estimate of $P(\mathbf{x}_i|\mathcal{N}_i)$ at time step $t$ and wvRN($\mathbf{C}^{(t)}$) denotes applying wvRN using all the estimates from time step $t$. We define the simulated annealing constants as:

$$\begin{aligned} \beta^0 &= k \\ \beta^{(t+1)} &= \beta^{(t)} \cdot \alpha, \end{aligned} \quad (3)$$

where $k$ is a constant between 0 and 1, which for the case study we set to 1.0, and $\alpha$ is a decay constant, which we set to 0.99. We note that wvRN-RL is quite robust to a large range of values for $\alpha < 1$.

## 3. Using Guilt-by-Association

We have applied wvRN-RL on a variety of networks, synthetic and real, ranging from a few hundred to over 100K nodes (see, e.g., (Macskassy & Provost, 2007; Macskassy & Provost, 2005b)), comparing it to other more sophisticated methods when appropriate. It has consistently performed as well as, or better than, other methods with orders of magnitude speedup in providing answers–often providing answers where more sophisticated methods fail due to the size of the network. In fact, many methods become intractable as soon as the network size reaches a few thousand nodes.

We here describe how we can use our method to answer the five questions above, providing enough detail on our results.

### 3.1. Computing the influence of a node

There are well-defined social network analytic metrics for computing the "importance" of a node in a social network. These metrics are generally known as "centrality" metrics and they vary depending on the importance that is being measured (see, e.g., (Wasserman & Faust, 1994). The most often-used centrality metrics include degree, betweenness, closeness, and information centrality.

The last centrality metric, information centrality, computes the amount of information a node can disseminate in the network. We can view this as an "influence" metric, which we can directly compute with our guilt-by-association method.

We can compute the "influence" of a node by setting the category of a single node to "interesting" and then use wvRN-RL to compute the probability that all other nodes are "interesting". The final average sum of probabilities is then the influence the node has in the network. These probabilities can also be visualized to provide a *node situational awareness* which is difficult to get from just looking at a graph with no visual markup (see, e.g., (Macskassy & Nanjo, 2008)). Figure 1 shows an example graph where a node was picked and we can see that it is quite strongly connected to nodes throughout the graph–something which would have been difficult to see otherwise.

### 3.2. Finding other nodes of interest

In many cases such as fraud (Fawcett & Provost, 1997; McGlohon et al., 2009) or counter-terrorism (Macskassy

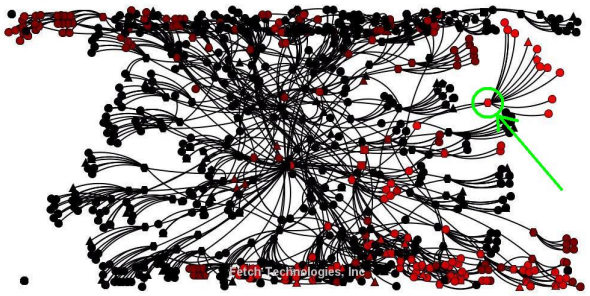


*Figure 1.* Example of visualizing influence of a node (circled above).

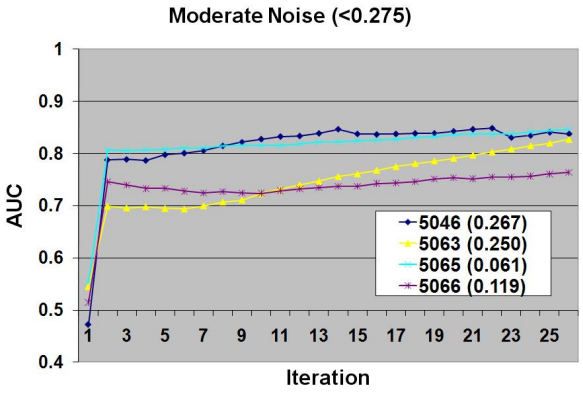


*Figure 2.* Identifying "bad" nodes in a large noisy networks.

& Provost, 2005a), one can label the few known "bad" nodes in the network and find the most likely other "bad" nodes in the network. Standard predictive models which predict yes or no are not good because they often are black box tools which return a set of predicted "bad" nodes–these sets are often too small or too large to be useful. Guilt-by-association can be used to *rank* the nodes in the network based on the likelihood that they are "bad" (Macskassy & Provost, 2005a), making this a useful and pragmatic aid in decision-making and the analytic workflow.

This setting is the dual of computing the influence of a node, where we instead look at how much nodes are influenced by the "bad" nodes in the network. The nodes which are the most influenced are the ones that are "most similar" to the nodes that are bad. Leveraging guilt-by-association in this manner has been shown to be quite robust to noise. For example, Figure 2 shows how well one can separate the true "bad" nodes from the rest as a function of acquiring more edges in a network. The graph shows results on four datasets, where the noise is between $0.061$ and $0.267$, where noise is defined as the number of nodes that have been labeled as "bad" which really were not "bad". The performance measure, AUC, is the area under the ROC curve and reflects how well the "bad' are separated from the "good". A value of $0.5$ is performing no better than random and a value of $1.0$ is perfect separation.

### 3.3. Leveraging profile information

If nodes in a network contain profile information, then we can leverage this information in various ways. In particular, if some nodes are labeled as "interesting", we might want to know what it is about them that makes them interesting or find other nodes that are also of interest. We can leverage the notion of homophily by using a metric known as *node-based assortativity* (Macskassy & Provost, 2007) (cf. (Newman, 2003)), which computes the amount of homophily present in a network (based on the nodes whose labels are known). This metric is computed from a network's node-based *assortativity matrix*—a CxC matrix, where cell $e_{ij}$ represents, for (all) nodes of class $c_i$, the average weighted fraction of their weighted links that link them to nodes of class $c_j$, such that $\sum_{ij} e_{ij} = 1$. The node-based assortativity coefficient, $A_E$, is then calculated as follows:

$$A_E = \frac{\sum_i e_{ii} - \sum_i a_i \cdot b_i}{1 - \sum_i a_i \cdot b_i},$$

where $a_i = \sum_j e_{ij}$ and $b_j = \sum_i e_{ij}$.

The way to use this measure is to have the overall influence of a type of edge be tied to its observed assortativity score, $A_E$ in such a way that types of edges which have high assortativity count for more than types of edges which have low assortativity scores. This can be done in two steps for each of edge type:

1. Normalize the edge-weights.
2. Rescale the edges by multiplying them with their respective $A_E$ score. If the $A_E$ score is negative, then set the edge-weight to zero as the behavior of negative edge weights are undefined in the wvRN model.

The advantage of this approach is that it is very general and can easily be used with an arbitrary number of edge types, each having their own semantics of edge-weights and edge statistics.

We can use this metric to "search" for relations such as whether nodes share a particular profile attribute or to "weight" a particular relation by the amount of assortativity is present in the network defined by that particular. This latter process was used to leverage textual information of documents and news articles to get better classification performance (Macskassy, 2007). While there are other ways of augmenting a network by adding links (see, e.g., (Gallagher et al., 2008)), there are still very few research efforts into augmenting a network based on profile information.

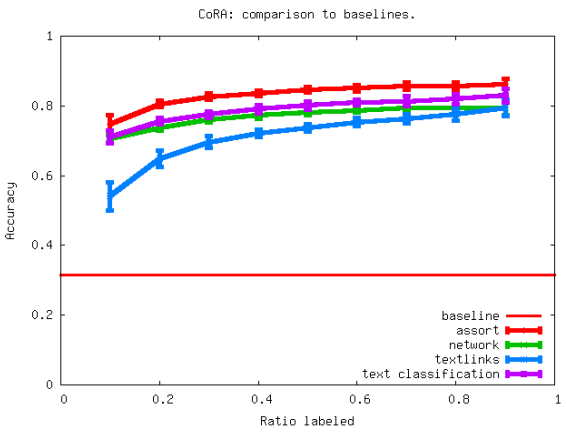


*Figure 3.* Comparing different ways of classifying papers by either combining links, using a link-type by itself or using plan text classification.

Figure 3 shows the results in the academic paper domain of CoRA (McCallum et al., 2000), which is a data set of computer science research papers categorized into 7 subcategories of machine learning. In this particular case, we have explicit links between papers in the forms of coauthors and citations and we can use profile information of the papers represented by their abstract and title. We generate an edge between papers if they have high text similarity (Macskassy, 2007). We compare in the figure how to combine edges using assortativity (assort), using only the explicit network (network) or only the text-mined links (textlinks) or just text classification. We can see that combining links using assortativity, a guilt-by-association metric, outperforms other ways of using the information available.

### 3.4. Categorization of nodes

Guilt-by-association can also be used in a standard machine learning context where some nodes have been labeled with a category and we can then use guilt-by-association to propagate these labels to the rest of the network (Macskassy & Provost, 2007). The final probabilities are then used as the predictions for the nodes that were not initially labeled. This is the most explored setting of our approach and it has consistently performed comparably to other methods.

For example, this particular setting has been used in areas such as fraud detection (Fawcett & Provost, 1997; McGlohon et al., 2009), academic papers (Macskassy & Provost, 2007), email classification (Gallagher et al., 2008), social network marketing (Hill et al., 2006), just to name a few.

### 3.5. Which nodes should be labeled next?

Recently guilt-by-association has been applied in the setting of active learning: given that only a few nodes in the network can be labeled, which should be labeled to get best classification performance. One interesting but related question is which nodes are key nodes in the current network for "splitting" the network into sub-communities? In particular, the answer can either be the nodes which are bordering the communities (i.e., the nodes where two communities meet) or the central nodes in each community, which are nodes that "define" the community. We have shown that we can indeed identify these nodes consistently by combining guilt-by-association and standard social network analysis metrics (Macskassy, 2009).

For example, Figure 4 shows two graphs of how one can use SNA and community-finding with guilt-by-association to perform comparably to state-of-the-art active learning but with an order of magnitude speedup in the running time (not shown) (Macskassy, 2009). The figure first compares using SNA metrics to chose nodes to label against empirical risk minimization (ERM). In this case, ERM consistently outperformed using these metrics by themselves. The second graph in the figure shows the performance when one uses a hybrid approach of community finding, SNA metrics and uncertainty to identify the next node to label. This hybrid approach is an order of magnitude faster in identifying nodes than what can be done with ERM.

## 4. Discussion

We have shown the wide range of uses of guilt-by-association models on large networks. There are still plenty of research directions where this type of model can be used. Its particular strengths lends themselves to learge networks: (1) it is very simple and very fast, (2) it consistently performs as well as more sophisticated methods, and (3) its underlying assumption of homophily has been observed widely in human as well as human-created networks.

## Acknowledgments

This work was sponsored in part by the Office of Naval Research under award number N00014-07-C-0923. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of Naval Research or the U.S. Government.

## References

Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding community structure in very large networks. *Physi-*

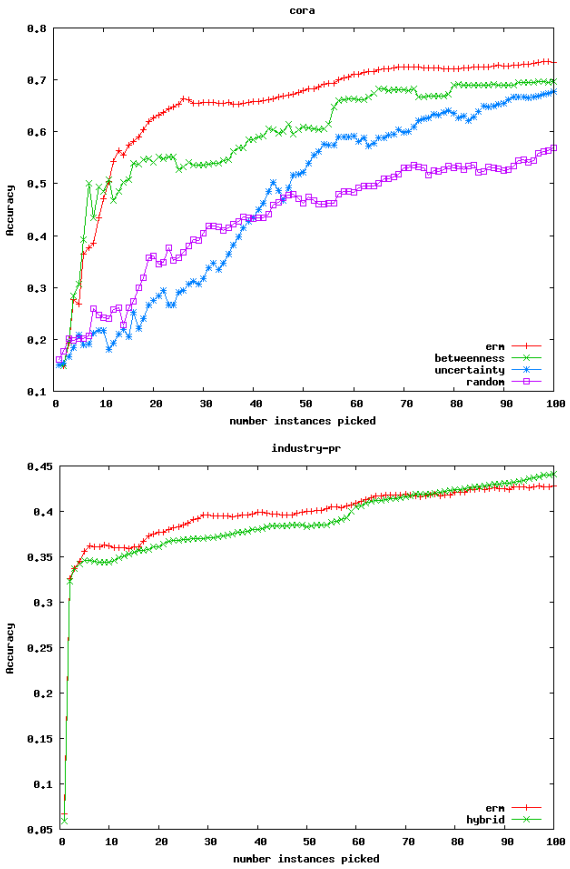


*Figure 4.* Comparing state-of-the-art active learning (ERM=empirical risk minimization) to that of a hybrid approach in the context of guilt-by-association.

*cal Review E*, *70*. 066111.

Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, *3*, 291–316.

Gallagher, B., Tong, H., Eliassi-Rad, T., & Faloutsos, C. (2008). Using ghost edges for classification in sparsely labeled networks. *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, NV.

Hill, S., Provost, F., & Volinsky, C. (2006). Network-based Marketing: Identifying Likely Adopters via Consumer Networks. *Statistical Science*, *22*.

Jensen, D., Neville, J., & Gallagher, B. (2004). Why Collective Inference Improves Relational Classification. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Leskovec, J., Backstrom, L., Kumar, R., & Tomkins, A. (2008). Microscopic evolution of social networks. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Leskovec, J., & Faloutsos, C. (2007). Scalable modeling of real graphs using kronecker multiplication. *International Conference on Machine Learning*.

Macskassy, S. A. (2007). Improving learning in networked data by combining explicit and mined links. *Proceedings of the Twenty-Second Conference on Artificial Intelligence*. Vancouver, Canada.

Macskassy, S. A. (2009). Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. *Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Paris, France.

Macskassy, S. A., & Nanjo, C. C. (2008). Graph mining using graph pattern profiles. *Proceedings of the 2008 International Conference on Artificial Intelligence*.

Macskassy, S. A., & Provost, F. (2005a). Suspicion scoring based on guilt-by-association, collective inference, and focused data access. *International Conference on Intelligence Analysis*.

Macskassy, S. A., & Provost, F. (2005b). Suspicion scoring of entities based on guilt-by-association, collective inference, and focused data access. *Annual Conference of the North American Association for Computational Social and Organizational Science (NAACSOS)*.

Macskassy, S. A., & Provost, F. (2007). Classification in Networked Data: A toolkit and a univariate case study. *Journal of Machine Learning Research (JMLR)*, *8*, 935–983.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval*, *3*, 127–163.

McGlohon, M., Bay, S., Anderle, M., Steier, D., & Faloutsos, C. (2009). SNARE: A link analytic system for graph labeling and risk detection. *Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Paris, France.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, *27*, 415–444.

Neville, J., & Jensen, D. (2007). Relational dependency networks. *Journal of Machine Learning Research*.

Newman, M. E. J. (2003). Mixing patterns in networks. *Physical Review E*, *67*. 026126.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.

Taskar, B., Segal, E., & Koller, D. (2001). Probabilistic Classification and Clustering in Relational Data. *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 870–878).

Wasserman, S., & Faust, K. (1994). *Social network analysis*. Cambridge: Cambridge University Press.

Wasserman, S., & Pattison, P. E. (1996). Logit models and logistic regressions for social networks. *Psychometrika*, *61*, 401–425.