Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms

Danai Koutra[1], Tai-You Ke[2], U Kang[1], Duen Horng (Polo) Chau[1], Hsing-Kuo Kenneth Pao[2], and Christos Faloutsos[1]

[1] School of Computer Science, Carnegie Mellon University
{danai,ukang,dchau,christos}@cs.cmu.edu
[2] Dept. of Computer Science & Information Engineering
National Taiwan Univ. of Science & Technology
{M9815071,pao}@mail.ntust.edu.tw

Abstract. If several friends of *Smith* have committed petty thefts, what would you say about *Smith*? Most people would not be surprised if *Smith* is a hardened criminal. **Guilt-by-association** methods combine weak signals to derive stronger ones, and have been extensively used for anomaly detection and classification in numerous settings (e.g., accounting fraud, cyber-security, calling-card fraud).
The focus of this paper is to compare and contrast several very successful, *guilt-by-association* methods: *Random Walk with Restarts*, Semi-Supervised Learning, and *Belief Propagation* (BP).
Our main contributions are two-fold: (a) theoretically, we prove that all the methods result in a similar matrix inversion problem; (b) for practical applications, we developed FaBP, a fast algorithm that yields 2× speedup, equal or higher accuracy than BP, and is guaranteed to converge. We demonstrate these benefits using synthetic and real datasets, including YahooWeb, one of the largest graphs ever studied with BP.

Keywords: Belief Propagation, Random Walk with Restart, Semi-Supervised Learning, probabilistic graphical models, inference

1 Introduction

Network effects are very powerful, resulting even in popular proverbs like "birds of a feather flock together". In social networks, obese people tend to have obese friends [5], happy people tend to make their friends happy too [7], and in general, people usually associate with like-minded friends with respect to politics, hobbies, religion etc. Thus, knowing the types of a few nodes in a network, (say, "honest" vs "dishonest"), we would have good chances to guess the types of the rest of the nodes.

Informally, the guilt-by-association problem (or label propagation in graphs) is defined as follows:

Given: a graph with N nodes and M edges; n_+ and n_- nodes labeled as members of the positive and negative class respectively

Find: the class memberships of the rest of the nodes, assuming that neighbors influence each other

The influence can be "homophily", meaning that nearby nodes have similar labels, or "heterophily", meaning the reverse (e.g., talkative people tend to prefer silent friends, and vice-versa). Homophily appears in numerous settings, for example: (a) *Personalized PageRank*: if a user likes some pages, she would probably like other pages that are heavily connected to her favorites. (b) *Recommendation systems*: if a user likes some products (i.e., members of *positive* class), which other products should get *positive* scores? (c) *Accounting and calling-card fraud*: if a user is dishonest, his/her contacts are probably dishonest too.

There are several, closely related methods that address the homophily problem, and some - among which is our proposed FABP method, improved on *Belief Propagation* - that address both homophily *and* heterophily. We focus on three of them: Personalized PageRank (or "Personalized Random Walk with Restarts", or just RWR), Semi-Supervised Learning (SSL), and Belief Propagation (BP). How are these methods related? Are they identical? If not, which method gives the best accuracy? Which method has the best scalability?

These questions are exactly the focus of this work. In a nutshell, we contribute by answering the above questions, and providing a fast algorithm inspired by our theoretical analysis:

- *Theory & Correspondences*: the three methods are closely related, but not identical.
- *Algorithm & Convergence*: we propose FABP, a fast, accurate and scalable algorithm, and provide the conditions under which it converges.
- *Implementation & Experiments*: finally, we propose a HADOOP-based algorithm, that scales to *billion-node* graphs, and we report experiments on one of the largest graphs ever studied in the open literature. Our FABP method achieves about 2× better runtime.

2 Related Work

RWR, SSL and BP are very popular techniques, with numerous papers using or improving them. Here, we survey the related work for each method.

RWR is the method underlying Google's classic PageRank algorithm [2]. RWR's many variations include *Personalized PageRank* [10], *lazy random walks* [20], and more[24, 21]. Related methods for node-to-node distance (but not necessarily *guilt-by-association*) include Pegasus [15], parameterized by *escape probability* and *round-trip probability*.

According to conventional categorization, SSL approaches are classified into four categories [28]: *low-density separation* methods, *graph-based* methods, methods for *changing the representation*, and *co-training* methods. The principle behind SSL is that unlabeled data can help us decide the "metric" between data points and improve the models' performance. A very recent use of SSL for

multi-class settings has been proposed in [12]. In this work, we mainly study the graph-based SSL methods.

BP [23], being an efficient inference algorithm on probabilistic graphical models, has been successfully applied to numerous domains, including error-correcting codes [16], stereo imaging in computer vision [6], fraud detection [19, 22], and malware detection[3]. Extensions of BP include *Generalized Belief Propagation* (GBP), that takes a multi-resolution view point, grouping nodes into regions [27]; however, how to construct good regions is still an open research problem. Thus, we focus on standard BP, which is better understood. Here, we study how the parameter choices for BP helps accelerate the algorithms, and how to implement the method on top of HADOOP [1] (open-source MapReduce implementation). This focus differentiates our work from existing research which speeds up BP by exploiting the graph structure [4, 22] or the order of message propagation [9].

*S*ummary: None of the above papers show the relationships between the three methods, or discuss the parameter choices (e.g., homophily factor). Table 1 qualitatively compares the methods. BP supports heterophily, but there is no guarantee on convergence. Our FABP algorithm improves on it to provide convergence.

Table 1: Qualitative comparison of 'guilt-by-association' (GBA) methods.

GBA Method	Heterophily	Scalability	Convergence
RWR	No	Yes	Yes
SSL	No	Yes	Yes
BP	Yes	Yes	?
FABP	Yes	Yes	Yes

3 Theorems and Correspondences

In this section we present the three main formulas that show the similarity of the following methods: binary BP and specifically our proposed approximation, the linearized BP (FABP), Gaussian BP (GAUSSIANBP), Personalized RWR (RWR), and Semi-Supervised learning (SSL).

For the homophily case, all the above methods are similar in spirit, and closely related to diffusion processes: the n_+ nodes that belong to class "+" (say, "green"), act as if they taint their neighbors (diffusion) with green color, and similarly do the negative nodes with, say, red color. Depending on the strength of homophily, or equivalently the speed of diffusion of the color, eventually we have green-ish neighborhoods, red-ish neighborhoods, and bridge-nodes (half-red, half-green).

The solution vectors for each method obey very similar equations: they all involve a matrix inversion, where the matrix consists of a diagonal matrix and a weighted or normalized version of the adjacency matrix. Table 2 has the definitions of the symbols that are used in the following discussion, and Table 3 shows the resulting equations, carefully aligned to highlight the correspondences.

Table 2: Major Symbols and Definitions. (matrices in bold capital, vectors in bold lowercase, and scalars in plain font)

Symbols	Definitions	Explanations
n	number of nodes in the graph	
$\mathbf{A}$	$n \times n$ symmetric adjacency matrix	
$\mathbf{D}$	$n \times n$ diagonal matrix of degrees	$D_{ii} = \sum_j A_{ij}$ and $D_{ij} = 0$ for $i \neq j$
$\mathbf{I}$	$n \times n$ identity matrix	
$\mathbf{b_h}$	"about-half" final beliefs $\boldsymbol{b} - 0.5$	$\boldsymbol{b} = n \times 1$ vector of the BP final beliefs $b(i)\{>0.5, <0.5\}$ means $i \in$ {"+", "-"} class $b(i) = 0$ means i is unclassified (neutral)
$\boldsymbol{\phi_h}$	"about-half" prior beliefs, $\boldsymbol{\phi} - 0.5$	$\boldsymbol{\phi} = n \times 1$ vector of the BP prior beliefs
h_h	"about-half" homophily factor $h - 0.5$	$h = \psi($"+","+"$)$: BP propagation matrix entry $h \to 0$ means strong heterophily $h \to 1$ means strong homophily

Table 3: Main results, to illustrate correspondence: $n \times n$ matrices in bold capital, $n \times 1$ vectors in bold lowercase, and scalars in plain font.

Method	matrix	unknown		known
RWR	$[\mathbf{I} \quad - c\mathbf{A}\mathbf{D}^{-1}]\times$	$\boldsymbol{x}$	$=$	$(1-c)\ \boldsymbol{y}$
SSL	$[\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]\times$	$\boldsymbol{x}$	$=$	$\boldsymbol{y}$
Gaussian BP = SSL	$[\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]\times$	$\boldsymbol{x}$	$=$	$\boldsymbol{y}$
FABP	$[\mathbf{I} + a\mathbf{D} - c'\mathbf{A}]\times$	$\mathbf{b_h}$	$=$	$\boldsymbol{\phi_h}$

Theorem 1 (FABP)**.** *The solution to Belief Propagation can be approximated by the linear system*

$$[\mathbf{I} + a\mathbf{D} - c'\mathbf{A}]\mathbf{b_h} = \boldsymbol{\phi_h} \tag{1}$$

where $a = 4h_h^2/(1-4h_h^2)$*, and* $c' = 2h_h/(1-4h_h^2)$*. The definitions of* h_h*,* $\phi_\mathbf{h}$ *and* $\mathbf{b_h}$ *are given in Table 2. Specifically,* $\phi_\mathbf{h}$ *corresponds to the prior beliefs of the nodes, and node* i*, about which we have no information, has* $\phi_h(i) = 0$*;* $\mathbf{b_h}$ *corresponds to the vector of our final beliefs for each node.*

Proof. The goal behind the "about-half" is the linearization of BP using Maclaurin expansions. The preliminary analysis of FABP, and the necessary lemmas for the linearization of the original BP equations are given in Appendix A. For the detailed proof of this theorem see Appendix B. □

Lemma 1 (Personalized RWR)**.** *The linear system for RWR given an observation* $\boldsymbol{y}$*, is described by the following equation:*

$$[\mathbf{I} - c\mathbf{A}\mathbf{D}^{-1}]\boldsymbol{x} = (1-c)\boldsymbol{y} \tag{2}$$

where $1-c$ *is the restart probability,* $c \in [0,1]$*. Similarly to the BP case above,* $\boldsymbol{y}$ *corresponds to the prior beliefs for each node, with the small difference that* $y_i = 0$ *means that we know nothing about node* i*, while a positive score* $y_i > 0$ *means that the node belongs to the positive class (with the corresponding strength).*

Proof. See [11], [24]. □

Lemma 2 (SSL and Gaussian BP)**.** *Suppose we are given* l *labeled nodes* (x_i, y_i)*,* $i = 1, \ldots, l$*,* $y_i \in \{0, 1\}$*, and* u *unlabeled nodes* $(x_{l+1}, ..., x_{l+u})$*. The solution to a Gaussian BP and SSL problem is given by the linear system:*

$$[\alpha(\mathbf{D} - \mathbf{A}) + \mathbf{I}]\boldsymbol{x} = \boldsymbol{y} \tag{3}$$

where α is related to the coupling strength (homophily) of neighboring nodes, $\boldsymbol{y}$ represents the labels of the labeled nodes and, thus, it is related to the prior beliefs in BP, and $\boldsymbol{x}$ corresponds to the labels of all the nodes or equivalently the final beliefs in BP.

Proof. See Appendix B and [25], [28]. □

Lemma 3 (RWR-SSL correspondence)**.** *On a regular graph (i.e., all nodes have the same degree d), RWR and SSL can produce identical results if*

$$\alpha = \frac{c}{(1-c)d}\,. \tag{4}$$

That is, we need to align carefully the homophily strengths α and c.

Proof. See Appendix B. □

In an arbitrary graph the degrees are different, but we can still make the two methods give the same results if each node has a different α_i instead of α. Specifically, for node i with degree d_i, the quantity α_i should be $\frac{c}{(1-c)d_i}$. The following section illustrates the correspondence between RWR and SSL.

3.1 Arithmetic Examples

Here we illustrate that SSL and RWR result in closely related solutions. We study both the case with variable α_i for each node i, and the case with fixed $\alpha = c/((1-c)\bar{d})$, where $\bar{d}$ is the average degree.

We generated a random graph using the Erdős-Rényi model, $G(n,p) = G(100, 0.3)$. Figure 1 shows the scatter-plot: each node i has a corresponding blue circle (x_{1i}, y_{1i}) for variable α_i, and also a red star (x_{2i}, y_{2i}) for fixed α. The coordinates of the points are the RWR and SSL scores, respectively. Figure 1(b) shows a magnification of the central part of Fig. 1(a). Notice that the red stars (fixed α) are close to the 45-degree line, while the blue circles (variable α_i) are exactly on the 45-degree line. The conclusion is that (a) the SSL and RWR scores are similar, and (b) the rankings are the same: whichever node is labeled as "positive" by SSL, gets a high score by RWR, and conversely.

4 Analysis of Convergence

In this section we provide the sufficient, but not necessary conditions for which our method, FaBP, converges. The implementation details of FaBP are described in the upcoming Section 5. Lemmas 4, 5, and 8 give the convergence conditions. At this point we should mention that work on the convergence of a variant of BP, Gaussian BP, is done in [18] and [25]. The reasons that we focus on BP are that (a) it has a solid, Bayesian foundation, and (b) it is more general than the rest, being able to handle heterophily (as well as multiple-classes, that we don't elaborate here).

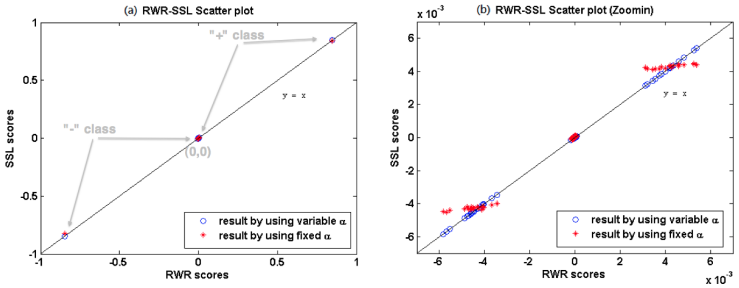


Fig. 1: Scatter plot showing the similarities between SSL and RWR. Scores of SSL and RWR for the nodes of a random graph: blue circles (perfect equality – variable α_i) and red stars (fixed α). The right figure is a zoom-in of the left. Most red stars are on or close to the diagonal: the two methods give similar scores, and identical assignments to positive/negative classes.

All our results are based on the power expansion required to compute the inverse of a matrix of the form $\mathbf{I} - \mathbf{W}$; all the methods undergo this process, as we show in Table 3. Specifically, we need the inverse of the matrix $\mathbf{I}+a\mathbf{D}-c'\mathbf{A} = \mathbf{I} - \mathbf{W}$, which is given by the expansion:

$$(\mathbf{I} - \mathbf{W})^{-1} = \mathbf{I} + \mathbf{W} + \mathbf{W}^2 + \mathbf{W}^3 + ... \tag{5}$$

and the solution of the linear system is given by the formula

$$(\mathbf{I} - \mathbf{W})^{-1}\phi_{\mathbf{h}} = \phi_{\mathbf{h}} + \mathbf{W} \cdot \phi_{\mathbf{h}} + \mathbf{W} \cdot (\mathbf{W} \cdot \phi_{\mathbf{h}}) + ... \tag{6}$$

This method, also referred to as the *Power Method*, is fast since the computation can be done in iterations, each one of which consists of a sparse-matrix/vector multiplication. In this section we examine its convergence conditions.

Lemma 4 (Largest eigenvalue). *The series* $\sum_{k=0}^{\infty} |c'\mathbf{A} - a\mathbf{D}|^k$ *converges iff* $\lambda(\mathbf{W}) < 1$, *where* $\lambda(\mathbf{W})$ *is the magnitude of the largest eigenvalue of* $\mathbf{W}$.

Given that the computation of the largest eigenvalue is non-trivial, we suggest using one of the following lemmas, which give a closed form for computing the "about-half" homophily factor, h_h.

Lemma 5 (1-norm). *The series* $\sum_{k=0}^{\infty} |c'\mathbf{A} - a\mathbf{D}|^k$ *converges if*

$$h_h < \frac{1}{2 + 2\max_j d_{jj}} \tag{7}$$

where d_{jj} *are the elements of the diagonal matrix D.*

Proof. The proof is based on the fact that the power series converges if the 1-norm, or equivalently the ∞-norm, of the symmetric matrix $\mathbf{W}$ is smaller than 1. The detailed proof is shown in Appendix C. □

Lemma 6 (Frobenius norm)**.** *The series* $\sum_{k=0}^{\infty} |c'\mathbf{A} - a\mathbf{D}|^k$ *converges if*

$$h_h < \sqrt{\frac{-c_1 + \sqrt{c_1^2 + 4c_2}}{8c_2}} \tag{8}$$

where $c_1 = 2 + \sum_i d_{ii}$ *and* $c_2 = \sum_i d_{ii}^2 - 1$.

Proof. This upper bound for h_h is obtained when we consider the Frobenius norm of matrix $\mathbf{W}$ and we solve the inequality $\| \mathbf{W} \|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |\mathbf{W}_{ij}|^2} < 1$ with respect to h_h. We omit the detailed proof. □

Equation (8) is preferable over (7) when the degrees of the graph's nodes demonstrate considerable standard deviation. The 1-norm yields small h_h for very big highest degree, while the Frobenius norm gives a higher upper bound for h_h. Nevertheless, we should bear in mind that h_h should be a sufficiently small number in order for the "about-half" approximations to hold, because of the "about-half" approximations done in the analysis of FaBP.

5 Proposed Algorithm: FaBP

Based on the analysis in Sections 3 and 4, we propose the FaBP algorithm:

- **Step 1**: Pick h_h to achieve convergence: $h_h = \max\{(7), (8)\}$ and compute the parameters a and c' as described in Theorem 1.
- **Step 2**: Solve the linear system (1). Notice that all the quantities involved in this equation are close to zero.
- **Step 3** (optional): If the achieved accuracy is not sufficient, run a few iterations of BP using the values computed in Step 2 as the prior node beliefs.

In the datasets we studied, the optional step was not required, as FaBP achieved equal or higher accuracy than BP, while running in less time.

6 Experiments

We present experimental results to answer the following questions:

Q1: How accurate is FaBP?
Q2: Under what conditions does FaBP converge?
Q3: How sensitive is FaBP to the values of h and ϕ?
Q4: How does FaBP scale on very large graphs with billions of nodes and edges?

The graphs we used in our experiments are summarized in Table 4. To answer the first three questions, we used the DBLP dataset [8], which consists of 14,376 papers, 14,475 authors, 20 conferences, and 8,920 terms. Each paper is connected to its authors, the conference in which it appeared and the terms in its title. Only a small portion of the nodes are labeled: 4,057 authors, 100 papers, and all the conferences. We adapted the labels of the nodes to two classes: AI (Artificial Intelligence) and *not* AI (= Databases, Data Mining and Information Retrieval). In each trial, we ran FaBP on the DBLP network where $(1-p)\%$ of the labels of the papers and the authors had been discarded, with $p \in \{0.1\%, 0.2\%, 0.3\%, 0.4\%, 0.5\%, 5\%\}$. Then, we tested the classification accuracy on the nodes whose labels were discarded. To avoid combinatorial explosion in the presentation of the results, we consider $\{h_h, priors\} = \{0.002, \pm 0.001\}$ as the anchor values, and then, we vary one parameter at a time. When the results are the same for different values of $p\%$, due to lack of space, we randomly pick the plots to present.

To answer the last question, we used the YahooWeb dataset, as well as Kronecker graphs – synthetic graphs generated by the Kronecker generator [17]. YahooWeb is a Web graph containing 1.4 billion web pages and 6.6 billion edges; we automatically labeled 11 million educational and 11 million adult web pages. We used 90% of these labeled data to set the node priors, and the remaining 10% to evaluate the accuracy. For parameters, we set h_h to 0.001 using Lemma 6 (Frobenius norm), and the magnitude of the prior beliefs to ± 0.001.

Table 4: Order and size of graphs.

Dataset	**YahooWeb**	**Kronecker 1**	**Kronecker 2**	**Kronecker 3**	**Kronecker 4**	**DBLP**
# nodes	1,413,511,390	177,147	120,552	59,049	19,683	37,791
# edges	6,636,600,779	1,977,149,596	1,145,744,786	282,416,200	40,333,924	170,794

6.1 Q1: Accuracy

Figure 2 shows the scatter plots of beliefs (FaBP vs BP) for each node of the DBLP data. We observe that FaBP and BP result in practically the same beliefs for all the nodes in the graph, when ran with the same parameters, and thus, they yield the same accuracy. Conclusions are identical for any labeled set-size we tried (0.1% and 0.3% shown in Fig. 2).

Observation 1. FaBP *and* BP *agree on the classification of the nodes when ran with the same parameters.*

6.2 Q2: Convergence

We examine how the value of the "about-half" homophily factor affects the convergence of FaBP. In Fig. 3 the red line annotated with "max $|eval| = 1$" splits the plots into two regions; (a) on the left, the Power Method converges and FaBP is accurate, (b) on the right, the Power Method diverges resulting

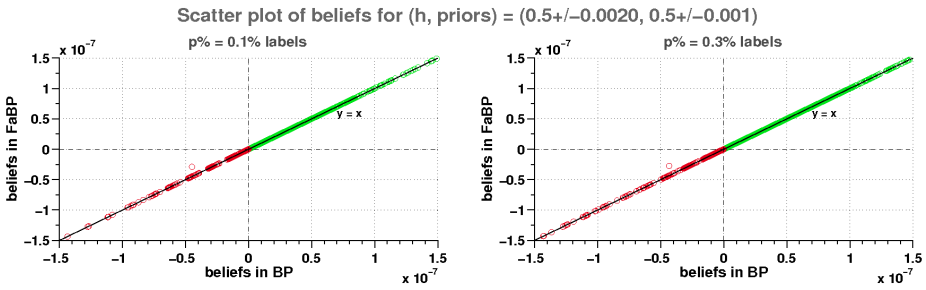


Fig. 2: The quality of scores of FaBP is near-identical to BP, i.e. all the points are on the 45-degree line in the scatter plot of the DBLP sub-network node beliefs (FaBP vs BP); red/green points correspond to nodes classified as "AI/not-AI" respectively.

in significant drop in the classification accuracy. We annotate the number of classified nodes for the values of h_h that leave some nodes unclassified due to numerical representation issues. The low accuracy scores for the smallest values of h_h are due to the unclassified nodes, which are counted as misclassifications. The Frobenius norm-based method yields greater upper bound for h_h than the 1-norm based method, preventing any numerical representation problems.

Observation 2. *Our convergence bounds consistently coincide with high-accuracy regions. Thus, we recommend choosing the homophily factor based on the Frobenius norm using* (8).

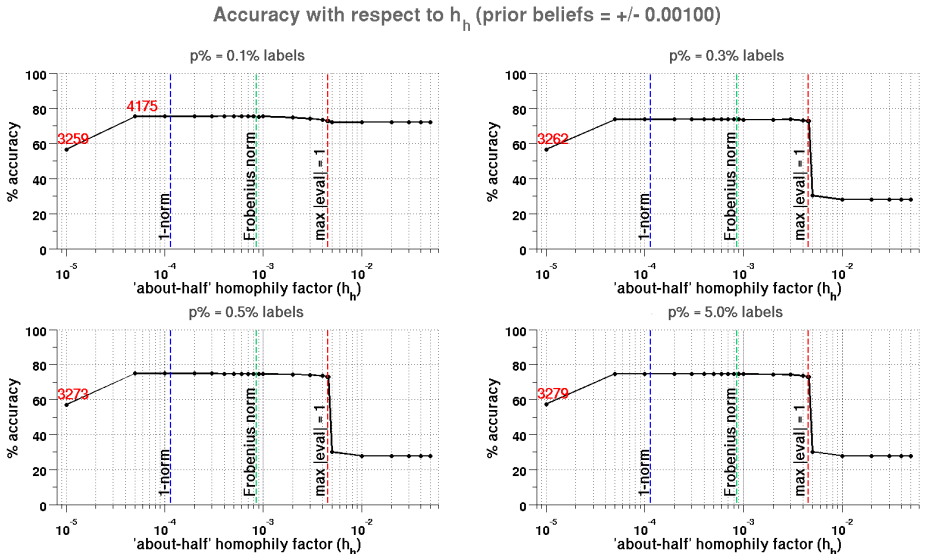


Fig. 3: FaBP achieves maximum accuracy within the convergence bounds. The annotated red numbers correspond to the classified nodes when not all nodes were classified by FaBP.

6.3 Q3: Sensitivity to parameters

Figure 3 shows that FABP is insensitive to the "about-half" homophily factor, h_h, as long as the latter is within the convergence bounds. Moreover, in Fig. 4 we observe that the accuracy score is insensitive to the *magnitude* of the prior beliefs. For brevity, we show only the cases $p \in \{0.1\%, 0.3\%, 0.5\%\}$, as for all values except for $p = 5.0\%$, the accuracy is practically identical. Similar results were found for different "about-half" homophily factors, but the plots are omitted due to lack of space.

Observation 3. *The accuracy results are insensitive to the* magnitude *of the prior beliefs and the homophily factor - as far as the latter is within the convergence bounds we gave in Section 4.*

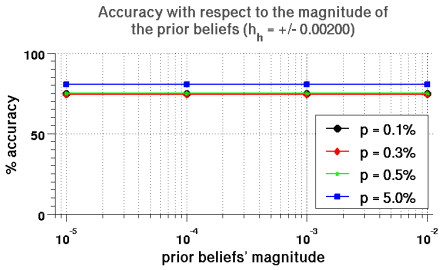


Fig. 4: Insensitivity of FABP to the *magnitude* of the prior beliefs.

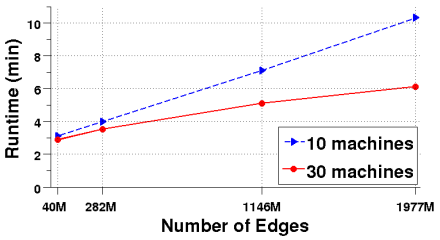
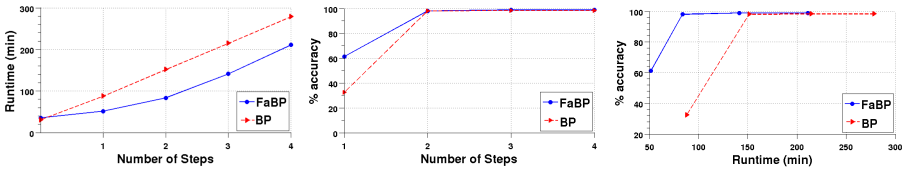


Fig. 5: FABP runtime vs # edges of Kronecker graphs for 10 and 30 machines on HADOOP.

6.4 Q4: Scalability

To show the scalability of FABP we implemented FABP on HADOOP, an open source MAPREDUCE framework, which has been successfully used for large scale graph analysis [14]. We first show the scalability of FABP on the number of edges of Kronecker graphs. As seen in Fig. 5, FABP scales linearly on the number of edges. Next, we compare HADOOP implementation of FABP and BP [13] in terms of running time and accuracy on YahooWeb graph. Figures 6(a-c) show that FABP achieves the maximum accuracy level after two iterations of the Power Method and is $\sim 2\times$ faster than BP. This is explained as follows: BP needs to store the updated messages for 2 states on disks for large graphs, and thus, it stores $2|E|$ records in total, where $|E|$ is the number of edges. In contrast, FABP stores n records per iteration, where n is the number of nodes. Given that $n < 2|E|$, FABP is faster than BP.

Observation 4. FABP *is linear on the number of edges, with* $\sim 2\times$ *faster running time than* BP *on* HADOOP.



(a) Runtime vs # of iterations (b) Accuracy vs # iterations (c) Accuracy vs runtime

Fig. 6: Performance on the YahooWeb graph (best viewed in color): FABP wins on speed and wins/ties on accuracy. In (c), each of the method has 4 points that correspond to one step from 1 to 4. FABP achieves the maximum accuracy after 84 minutes, while BP achieves the same accuracy after 151 minutes.

7 Conclusions

Which of the many *guilt-by-association* methods one should use? We answered this question, and we developed FABP, a new, fast algorithm to do such computations. The contributions of our work are the following:

- *Theory & Correspondences*: We showed that successful, major *guilt-by-association* approaches (RWR, SSL, and BP variants) are closely related, and we proved that some are even equivalent under certain conditions (Theorem 1, Lemmas 1, 2, and 3).
- *Algorithms & Convergence*: Thanks to our analysis, we designed FABP, a fast and accurate approximation to the standard belief propagation (BP), which has convergence guarantee (Lemmas 5 and 6).
- *Implementation & Experiments*: We showed that FABP is significantly faster, about 2×, and has the same or better accuracy (AUC) than BP. Moreover, we showed how to parallelize it with MAPREDUCE (HADOOP), operating on *billion-node* graphs.

Thanks to our analysis, our guide to practitioners is the following: among all 3 *guilt-by-association* methods, we recommend belief propagation, for two reasons: (1) it has solid, Bayesian underpinnings and (2) it can naturally handle heterophily, as well as multiple class-labels. With respect to parameter setting, we recommend to choose homophily score, h_h, according to the Frobenius bound in (8).

Future work could focus on time-evolving graphs, and label-tracking over time. For instance, in a call-graph, we would like to spot nodes that change behavior, e.g. from "telemarketer" type to "normal user" type.

Acknowledgments This work is partially supported by an IBM Faculty Award, by the National Science Foundation under Grants No. CNS-0721736 IIS0970179, under the project No. NSC 98-2221-E-011-105, NSC 99-2218-E-011-019, under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. DE-AC52-07NA27344, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as

representing the official policies, either expressed or implied, of the Army Research Laboratory, the U.S. Government, NSF, or any other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] Hadoop information. http://hadoop.apache.org/.
[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7), 1998.
[3] D. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos. Polonium: Tera-scale graph mining and inference for malware detection. *SDM*, 2011.
[4] A. Chechetka and C. Guestrin. Focused belief propagation for query-specific inference. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, May 2010.
[5] N. A. Christakis and J. H. Fowler. The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 357(4):370–379, 2007.
[6] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006.
[7] J. H. Fowler and N. A. Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study. *BMJ*, 2008.
[8] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based Consensus Maximization among Multiple Supervised and Unsupervised Models. In *NIPS*, 2009.
[9] J. Gonzalez, Y. Low, and C. Guestrin. Residual splash for optimally parallelizing belief propagation. In *AISTAT*, 2009.
[10] T. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, pages 784–796, 2003.
[11] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University, 2003.
[12] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao. Graph regularized transductive classification on heterogeneous information networks. ECML PKDD, 2010.
[13] U. Kang, D. H. Chau, and C. Faloutsos. Mining large graphs: Algorithms, inference, and discoveries. In *ICDE*, pages 243–254, 2011.
[14] U. Kang, C. Tsourakakis, and C. Faloutsos. Pegasus: A peta-scale graph mining system - implementation and observations. *IEEE International Conference on Data Mining*, 2009.
[15] Y. Koren, S. C. North, and C. Volinsky. Measuring and extracting proximity in networks. In *KDD*, pages 245–255. ACM, 2006.
[16] F. Kschischang, B. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.
[17] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. *PKDD*, 2005.
[18] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 7:2031–2064, 2006.

[19] M. McGlohon, S. Bay, M. G. Anderle, D. M. Steier, and C. Faloutsos. Snare: a link analytic system for graph labeling and risk detection. KDD, 2009.
[20] E. Minkov and W. Cohen. Learning to rank typed graph walks: Local and global approaches. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 1–8. ACM, 2007.
[21] J. Pan, H. Yang, C. Faloutsos, and P. Duygulu. Gcap: Graph-based automatic image captioning. In *MDDE*, 2004.
[22] S. Pandit, D. Chau, S. Wang, and C. Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, 2007.
[23] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the AAAI National Conference on AI*, pages 133–136, 1982.
[24] H. Tong, C. Faloutsos, and J. Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.
[25] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural computation*, 12(1):1–41, 2000.
[26] J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.
[27] J. Yedidia, W. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.
[28] X. Zhu. Semi-supervised learning literature survey, 2006.

Appendix A: Preliminaries - Analysis of FaBP

In this appendix we present the lemmas that are needed to prove Theorem 1 (FaBP), which gives the linearized version of BP. We start with the original BP equations, and we derive the proof by:

- using the *odds ratio* $p_r = p/(1-p)$, instead of probabilities. The advantage is that we have only one value for each node, $p_r(i)$, instead of two, $p_+(i)$ and $p_-(i)$; also, the normalization factor is not needed. Moreover, working with the *odds ratios* results in the substitution of the propagation matrix entries by a scalar homophily factor.
- assuming that all the parameters are close to 1/2, using Maclaurin expansions to linearize the equations, and keeping only the first order terms. By doing so we avoid the sigmoid/non-linear equations of BP.

Traditional BP equations: In [26], Yedidia derives the following update formulas for the messages sent from node i to node j and the belief of each node that it is in state x_i

$$m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i) \cdot \psi_{ij}(x_i, x_j) \cdot \prod_{n \in N(i) \setminus j} m_{ni}(x_i) \tag{9}$$

$$b_i(x_i) = \eta \cdot \phi_i(x_i) \cdot \prod_{j \in N(i)} m_{ij}(x_i) \tag{10}$$

where the message from node i to node j is computed based on all the messages sent by all its neighbors in the previous step except for the previous message sent from node j to node i. $N(i)$ denotes the neighbors of i and η is a normalization constant that guarantees that the beliefs sum to 1.

Table 5: Additional Symbols and Definitions

Symbols	Definitions
p	P(node in positive class) = P("+")
m	message
$< var >_r$	odds ratio = $\frac{<var>}{1-<var>}$, where $< var >= b$, ϕ, m, h
$B(a, b)$	blending function of the variables a and $b = \frac{a \cdot b+1}{a+b}$.

Lemma 7. *Expressed as ratios, the BP equations become:*

$$m_r(i, j) \leftarrow B[h_r, b_{r,adjusted}(i, j)] \tag{11}$$

$$b_r(i) \leftarrow \phi_r(i) \cdot \prod_{j \in N(i)} m_r(j, i) \tag{12}$$

where $b_{r,adjusted}(i, j)$ is defined as $b_{r,adjusted}(i, j) = b_r(i)/m_r(j, i)$. The division by $m_r(j, i)$ subtracts the influence of node j when preparing the message $m_r(i, j)$.

Proof. The proof is straightforward. Notice that $1 - v_+(i) = v_-(i)$ for $v \in \{b, \phi, m\}$, eg., $b_-(i) = 1 - b_+(i) = \eta \cdot (1 - \phi_+(i)) \cdot \prod_{j \in N(i)}(1 - m_+(i, j))$. □

Lemma 8 (Approximations)**.** *Fundamental approximations for all the variables of interest, $\{m, b, \phi, h\}$, are:*

$$v_r = \frac{v}{1 - v} = \frac{1/2 + v_h}{1/2 - v_h} \approx 1 + 4v_h \tag{13}$$

$$B(a_r, b_r) \approx 1 + 8a_h b_h \tag{14}$$

where $B(a_r, b_r)$ is the blending function for any variables a_r, b_r.

Sketch of proof. Use the definition of "about-half" approximations, apply the appropriate Maclaurin series expansions and keep only the first order terms. □

Lemmas 9-11 are useful in order to derive the linear equation of FaBP. Note that we apply several approximations, but omit the "≈" symbol to make the proofs more readable.

Lemma 9. *The "about-half" version of the belief equation becomes, for small deviations from the half-point:*

$$b_h(i) \approx \phi_h(i) + \sum_{j \in N(i)} m_h(j, i)\,. \tag{15}$$

Proof. We use (12) and (13) and apply the appropriate Maclaurin series expansions:

$$b_r(i) = \phi_r(i) \prod_{j \in N(i)} m_r(j,i) \Rightarrow$$
$$log\,(1 + 4b_h(i)) = log\,(1 + 4\phi_h(i)) + \sum_{j \epsilon N(i)} log\,(1 + 4m_h(j,i)) \Rightarrow$$
$$b_h(i) = \phi_h(i) + \sum_{j \epsilon N(i)} m_h(j,i)\,. \qquad \square$$

Lemma 10. *The "about-half" version of the message equation becomes:*

$$m_h(i,j) \approx 2h_h[b_h(i) - m_h(j,i)]\,. \tag{16}$$

Proof. We combine (11), (13) and (14) to deduce

$$m_r(i,j) = B[h_r, b_{r,adjusted}(i,j)] \Rightarrow m_h(i,j) = 2h_h b_{h,adjusted}(i,j)\,. \tag{17}$$

In order to derive $b_{h,adjusted}(i,j)$ we use (13) and the approximation of the Maclaurin expansion $\frac{1}{1+\epsilon} \approx 1 - \epsilon$ for a small quantity ϵ:

$$b_{r,adjusted}(i,j) = b_r(i)/m_r(j,i) \Rightarrow$$
$$1 + b_{h,adjusted}(i,j) = (1 + 4b_h(i))(1 - 4m_h(j,i)) \Rightarrow$$
$$b_{h,adjusted}(i,j) = b_h(i) - m_h(j,i) - 4b_h(i)m_h(j,i)\,. \tag{18}$$

Substituting (18) to (17) and ignoring the terms of second order, leads to the about-half version of the message equation. $\square$

Lemma 11. *At steady state, the messages can be expressed in terms of the beliefs:*

$$m_h(i,j) \approx \frac{2h_h}{(1 - 4h_h^2)}\,[b_h(i) - 2h_h b_h(j)]\,. \tag{19}$$

Proof. We apply Lemma 10 both for $m_h(i,j)$ and $m_h(j,i)$ and we solve for $m_h(i,j)$. $\square$

Appendix B: Proofs of Section 3 (Theorems)

Here we give the proofs of the theorems and lemmas presented in Section 3.

Proof of Theorem 1. We substitute (16) to (15) and we obtain:

$$b_h(i) - \sum_{j \in N(i)} m_h(j,i) = \phi_h(i) \Rightarrow$$
$$b_h(i) + \sum_{j \in N(i)} \frac{4h_h^2 b_h(j)}{1 - 4h_h^2} - \sum_{j \in N(i)} \frac{2h_h}{1 - 4h_h^2} b_h(i) = \phi_h(i) \Rightarrow$$
$$(\mathbf{I} + a\mathbf{D} - c'\mathbf{A})\mathbf{b_h} = \boldsymbol{\phi_h}\,. \qquad \square$$

Proof of Lemma 2. Given l labeled points (x_i, y_i), $i = 1, \ldots, l$, and u unlabeled points $x_{l+1}, \ldots, x_{l+u}$ for a semi-supervised learning problem, based on an energy minimization formulation, we find the labels x_i by minimizing the following function E

$$E(\boldsymbol{x}) = \alpha \sum_{j \in N(i)} a_{ij}(x_i - x_j)^2 + \sum_{1 \le i \le l} (y_i - x_i)^2 \,, \tag{20}$$

where α is related to the coupling strength (homophily) of neighboring nodes, and $N(i)$ denotes the neighbors of i. If *all* points are labeled, (20) becomes, in matrix form,

$$\begin{aligned} E(\boldsymbol{x}) &= \boldsymbol{x}^T[\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]\boldsymbol{x} - 2\boldsymbol{x} \cdot \boldsymbol{y} + K(\boldsymbol{y}) \\ &= (\boldsymbol{x} - \boldsymbol{x}^*)^T[\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})](\boldsymbol{x} - \boldsymbol{x}^*) + K'(\boldsymbol{y}) \,, \end{aligned}$$

where $\boldsymbol{x}^* = [\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]^{-1}\boldsymbol{y}$, and K, K' are some constant terms which depend only on $\boldsymbol{y}$. Clearly, E achieves the minimum when

$$\boldsymbol{x} = \boldsymbol{x}^* = [\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]^{-1}\boldsymbol{y} \,.$$

The equivalence of SSL and Gaussian BP can be found in [25]. □

Proof of Lemma 3. Based on (2) and (3), the two methods will give identical results if

$$\begin{gathered} (1-c)[\mathbf{I} - c\mathbf{D}^{-1}\mathbf{A}]^{-1} = [\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]^{-1} \Leftrightarrow \\ \left(\frac{c}{1-c}\right)[\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}] = \alpha(\mathbf{D} - \mathbf{A}) \Leftrightarrow \\ \left(\frac{c}{1-c}\right)\mathbf{D}^{-1} = \alpha\mathbf{I} \,. \end{gathered}$$

This cannot hold in general, unless the graph is "regular": $d_i = d$ $(i = 1, \ldots, n)$, or $\mathbf{D} = d \cdot \mathbf{I}$, in which case the condition becomes

$$\alpha = \frac{c}{(1-c)d} \Rightarrow c = \frac{\alpha d}{1 + \alpha d} \tag{21}$$

where d is the common degree of all the nodes. □

Appendix C: Proofs of Section 4 (Convergence)

Proof of Lemma 5. In order for the power series to converge, a sub-multiplicative norm of matrix $\mathbf{W} = c\mathbf{A} - a\mathbf{D}$ should be smaller than 1. In this analysis we use the 1-norm (or equivalently the ∞-norm). The elements of matrix $\mathbf{W}$ are either $c = \frac{2h_h}{1-4h_h^2}$ or $-ad_{ii} = \frac{-4h_h^2 d_{ii}}{1-4h_h^2}$. Thus, we require

$$\begin{gathered} \max_j (\sum_{i=1}^{n} |\mathbf{W}_{ij}|) < 1 \Rightarrow (c + a) \cdot \max_j d_{jj} < 1 \Rightarrow \\ \frac{2h}{1-2h} \max_j d_{jj} < 1 \Rightarrow h_h < \frac{1}{2(1 + \max_j d_{jj})}. \end{gathered}$$

□