Collective Opinion Spam Detection using Active Inference

Shebuti Rayana
Stony Brook University
srayana@cs.stonybrook.edu

Leman Akoglu
Stony Brook University
leman@cs.stonybrook.edu

Abstract

Opinion spam has become a widespread problem in the online review world, where paid or biased reviewers write fake reviews to elevate or relegate a product (or business) to mislead the consumers for profit or fame. In recent years, opinion spam detection has attracted a lot of attention from both the business and research communities. However, the problem still remains challenging as human labeling is expensive and hence labeled data is scarce, which is needed for supervised learning and evaluation. There exist recent works (e.g., FraudEagle [2], SpEagle [19]) which address the spam detection problem as an unsupervised network inference task on the review network. These methods are also able to incorporate labels (if available), and have been shown to achieve improved performance under the semi-supervised inference setting, in which the labels of a random sample of nodes are consumed.

In this work, we address the problem of active inference for opinion spam detection. Active inference is the process of carefully selecting a subset of instances (nodes) whose labels are obtained from an oracle to be used during the (network) inference. Our goal is to employ a label acquisition strategy that selects a given number of nodes (a.k.a. the budget) wisely, as opposed to randomly, so as to improve detection performance significantly over the random selection. Our key insight is to select nodes that (i) exhibit high uncertainty, (ii) reside in a dense region, and (iii) are close-by to other uncertain nodes in the network. Based on this insight, we design a utility measure, called Expected Un-Certainty Reach (EUCR), and pick the node with the highest EUCR score at every step iteratively. Experiments on two large real-world datasets from Yelp.com show that our method significantly outperforms random sampling as well as other state-of-the-art active inference approaches.

Keywords: opinion spam, active inference, expected uncertainty reach

1 Introduction

In the modern era of e-commerce, online reviews are gaining increasing importance in the decision making of consumers for buying products or services. Unlike advertisements, online reviews are endorsements of real consumers about products. A study by Luca [11] shows that $+1$ star rating increase of a product or business increases the revenue by 5–9%. Due to the financial gain associated with online reviews, paid or biased reviewers write fake reviews to promote or demote a product (or business) to mislead the consumers. This very act of false endorsement about products by fraudulent reviewers is known as opinion spam [7].

Opinion spam has become a widespread problem in recent years. However, it remains challenging, as unlabeled data is abundant, but labels are difficult or expensive to obtain where human judges are only slightly better than random [17]. Scarcity of labeled data makes the supervised learning and evaluation hard [7, 16, 17]. As such, most existing works on opinion spam detection are unsupervised [2, 5, 6, 15, 22, 23, 24]. A useful trade-off is semi-supervised learning, which uses only a small set of labeled data to improve detection performance over the unsupervised setting. Most recently, Rayana et al. proposed a flexible approach called SpEagle [19] which addresses the spam detection problem as a network inference task on the review network, with the potential of seamless label incorporation. Importantly, they showed that a small fraction of labeled data improves the detection performance significantly.

In SpEagle, the labels of a *random* sample of nodes are consumed to improve the network inference performance. Our intuition suggests that it is possible to do better by a wise selection of only the "valuable" nodes with lower labeling cost. The very field of selecting valuable instances for acquiring labels from an oracle (e.g., human) within a small budget for learning improved models with lower cost is widely known as *active learning* [1, 20]. Like active learning, *active inference* is a well known approach [3, 18] that targets to achieve higher accuracy with fewer labeled examples given a small budget, where the existing inference model can pose queries to choose most valuable instances which are labeled by an oracle. The difference between active learning and inference is that the former re-trains the model whenever new instances are obtained from the oracle, whereas the latter assumes that a model already exists and the new labeled instances by the oracle are used during the inference. The most challenging task of active inference is the selection of valuable instances for label acquisition.

A number of successful label acquisition approaches have been proposed in active learning as well as active inference literature [1, 3, 18, 20]. Selecting valuable instances for these label acquisition techniques can be divided into two main types based on the nature of the data, (i) using flat data (no network) [10, 12], and (ii) using relational information of the data (network structure) [3, 4, 18]. In this work, we address the problem of active inference for the opinion spam detection problem. Our goal is to achieve improved performance over random selection within a small budget. Our main contributions are as follows.

- We adapt several existing label acquisition approaches of active inference into our collective opinion spam detection framework called SpEagle [19], in order to wisely select valuable nodes to label.
- We present three important characteristics of a valuable node in a label acquisition strategy: (i) uncertainty of a node, (ii) density of the region it belongs to, and (iii) proximity of the node to other uncertain nodes. Intuitively, a node is more valuable if its uncertainty is high and it is close-by to other uncertain nodes in a dense region, hence, acquiring the label of this node helps other uncertain nodes, through diffusion of its label information within the neighborhood of this node.
- Based on the above characteristics, we devise a label acquisition strategy called Expected UnCertainty Reach (EUCR) that wisely selects valuable nodes within a small budget to improve performance.

We evaluate our method on two real-world datasets collected from Yelp.com, containing filtered (spam) and recommended (non-spam) reviews. To the best of our knowledge, this is the first work using active inference for opinion spam detection with a large-scale evaluation on real-world datasets. Our experiments shows that Expected Uncertainty Reach (EUCR) outperforms random sampling and several state-of-the-art active inference approaches.

2 Network Inference for Collective Classification

We consider a user–review–product tripartite network representation $G = (V, E)$, which contains N user nodes $U = \{u_1, \ldots, u_N\}$, M product nodes $P = \{p_1, \ldots, p_M\}$, and Q review nodes $R = \{r_1, \ldots, r_Q\}$, $V = U \cup P \cup R$, connected through two types of edges; the user–review edges $(u_i, r_k, t = \text{`}write\text{'}) \in E$ and the review-product edges $(r_k, p_j, t = \text{`}belong\text{'}) \in E$.

We then formulate the opinion spam detection problem as a network inference task on G, in which users are to be classified as $\mathcal{L}_U \in \{benign, spammer\}$, products as $\mathcal{L}_P \in \{non\text{-}targeted, targeted\}$, and reviews as $\mathcal{L}_R \in \{genuine, fake\}$. In addition, meta information (ratings, timestamps, and text) from review websites is utilized to extract indicative features of spam, which is incorporated into the network inference task as prior knowledge.

We address the network inference problem by considering the user–review–product network as a pairwise Markov Random Field (pMRF) [8]. A pMRF model consists of an undirected graph where the label (from a specific domain of class labels) of each node is dependent upon its neighbors only and independent of all other nodes in the graph. The joint probability of node labels is written as a product of individual and pairwise factors, parameterized over the nodes and the edges, respectively:

$$P(\mathbf{y}) = \frac{1}{Z} \prod_{Y_i \in V} \phi_i(y_i) \prod_{(Y_i, Y_j, t) \in E} \psi_{ij}^t(y_i, y_j) \tag{2.1}$$

where Z is the normalization constant. The individual factors ϕ_i are the initial class probabilities for each node i called the *priors*. The pairwise factors ψ_{ij}^t capture the likelihood of a node with label y_i to be connected to a node with label y_j through an edge with type t, are called the *compatibility* (or edge) potentials. Finding the best label assignments $\mathbf{y}$ to all the nodes, such that the joint probability $P(\mathbf{y})$ of the pMRF is maximized, is the inference problem which is computationally intractable and known to be NP-hard for general MRFs. To solve the inference, we use a computationally tractable (linear in the number of edges) approximate inference algorithm called Loopy Belief Propagation (LBP) [25].

LBP is based on iterative message passing between the connected nodes in the network. At every iteration, a *message* $m_{i \to j}$ is sent from each node i to each neighboring node j. The message captures the probability distribution over the class labels of j, and is computed as in Eqn. (2.2),

$$m_{i \to j}(y_j) = \alpha \sum_{y_i \in \mathcal{L}_{T_i}} \phi_i(y_i)\, \psi_{ij}^t(y_i, y_j) \prod_{Y_k \in \mathcal{Y}_{\mathcal{N}_i} \setminus Y_j} m_{k \to i}(y_i) \tag{2.2}$$

where $\mathcal{N}_i$ denotes the set of i's neighbors, $T_i \in \{U, R, P\}$ denotes type of node i and α is a normalization constant. These messages are exchanged iteratively over the edges until convergence, at which marginal probabilities are computed. The marginal probability is called the *belief* $b_i(y_i)$, of assigning each Y_i associated with a node of type $T_i \in \{U, R, P\}$ with the label y_i in label domain $\mathcal{L}_{T_i}$ (e.g. $\mathcal{L}_U \in \{benign, spammer\}$) as follows,

$$b_i(y_i) = \beta\, \phi_i(y_i) \prod_{Y_j \in \mathcal{Y}_{\mathcal{N}_i}} m_{j \to i}(y_i) \tag{2.3}$$

where β is the normalization constant. For ranking, the probability values $b_i(y_i)$ are sorted, where $y_i = spammer$ and $y_i = fake$ respectively for users and for reviews.

2.1 Unsupervised Network Inference

The parameters of the aforementioned network inference framework include the compatibility potentials $\psi^t(y_i, y_j)$'s

and the priors $\phi_i(y_i)$'s. Since spam detection is a problem area in which labeled data is scarce, most often than not these parameters cannot be estimated from training data, but rather instantiated in an unsupervised fashion. Specifically, the SPEAGLE framework by Rayana et al. [19] leverages meta information and domain intuition to set these parameters, which we describe next.

Compatibility potentials. In SPEAGLE, the compatibility (or edge) potentials ψ_{ij}^t are initialized to enforce homophily [14]. In particular, it is assumed that all the reviews written by spammers (benign users) are fake (genuine), and that with high probability fake (genuine) reviews belong to targeted (non-targeted) products; although with some probability fake reviews may also belong to non-targeted products as part of camouflage and vice versa. Overall, the parameters are set as follows.

	User ($\psi^{t=\text{`write'}}$)		($\psi^{t=\text{`belong'}}$) **Product**	
Review	*benign*	*spammer*	*non-target*	*target*
genuine	1	0	$1-\epsilon$	ϵ
fake	0	1	ϵ	$1-\epsilon$

Priors. To estimate the prior potentials ϕ_i, the indicative features of spam are extracted from available metadata (ratings, time-stamps, review text) for all three types of nodes. These features can be divided into two main categories, (i) text based (review text), and (ii) behavioral (rating, time-stamp). Their framework utilize a total of 11 user, 11 product, and 16 review features, both text based and behavioral. To unify the features (having different scales) into a comparable range, SPEAGLE leverages the cumulative distribution function (CDF); in particular, the CDF values of all the features for each node i are combined into a spam score $S_i \in [0,1]$, that quantifies the suspiciousness of the node, such that its class priors can be initialized as $\phi_i = \{1-S_i, S_i\}$.

2.2 Semi-supervised Network Inference

One of the key advantages of the SPEAGLE framework is the seamless integration of node labels when available. This semi-supervised version, called SPEAGLE$^+$ [19], achieves improved performance, in which the labels of a *random* sample of nodes are consumed. Specifically, given the labels for any set of nodes (reviews, users, and/or products), the priors of the corresponding nodes are initiated as $\{\epsilon, 1-\epsilon\}$ for those that are associated with spam (i.e., fake, spammer, or target), and $\{1-\epsilon, \epsilon\}$ otherwise. The priors of unlabeled nodes are estimated based on the features extracted from metadata, that is as $\{1-S_i, S_i\}$. The inference procedure remains exactly the same. Since this integration of available labels does not require any model (re)training, it is extremely efficient and seamless. This is suitable even when the size of the labeled data is too small or imbalanced to learn from.

In this work, we extend the semi-supervised SPEAGLE framework with active inference. Our goal is to improve the detection performance significantly over the random selection. We describe our approach to *active* network inference in the following section.

3 Active Network Inference

Active inference addresses the problem of minimizing the labeling cost while maximizing the classification performance. The key idea is to achieve higher accuracy with fewer labeled examples given a budget, where the existing model of inference can pose queries to choose most "valuable" data instances which are labeled by an oracle (e.g., human). These labels are then used at inference time. The goal is to devise an effective strategy to identify such "valuable" nodes and a metric to quantify the "value" of instances (i.e., nodes). There exist a variety of active inference settings [1, 20]. In this work, we utilize the pool-based setting, in which the collective classifier is initially provided with a pool $\mathcal{P}$ of unlabeled nodes. At each iteration it selects the most informative node, adds it to the labeled set $\mathcal{L}$ and removes from $\mathcal{P}$ until the budget $\mathcal{B}$ (given) is exhausted.

Given a set of unlabeled nodes $\mathcal{U}$, we address the problem of finding the most valuable node to be labeled at each step iteratively, so as to improve the performance within a budget $\mathcal{B}$. We update the beliefs of all nodes each time a new label is acquired. In this work, we utilize some state-of-the-art label acquisition strategies, such as, random sampling, uncertainty sampling and query-by-committee approaches. We also adapt ALFNET [4] by modifying it to work with our network inference setting, using the metadata as well as relational information of the review network for label acquisition. Finally, we propose an efficient label acquisition strategy which we call Expected UnCertainty Reach (EUCR). We mainly build on uncertainty sampling, where our key insight is to select nodes that (i) exhibit high uncertainty, (ii) reside in a dense region, and (iii) are close-by to other uncertain nodes in the network (hence "reach"). We first describe how we adapt existing approaches to our setting in Sections 3.1 through 3.4, and later introduce our proposed approach in Section 3.5. We denote the most valuable node with x_A^*, where A is the label acquisition strategy. Here, we incorporate the label acquisition in our network inference framework for review nodes only. We consider Yelp.com to be our oracle, as they provide recommended and filtered reviews.

3.1 Random Sampling (RS)

In random sampling strategy, we randomly pick a review node for labeling and simply set its prior as $\{\epsilon, 1-\epsilon\}$ if it belongs to the spam class (i.e., fake), and $\{1-\epsilon, \epsilon\}$ otherwise. This random selection is done iteratively until the budget $\mathcal{B}$ is exhausted. This is the strategy used in [19]. Our goal is to improve over this baseline with careful selection of "valuable" nodes to query the oracle.

3.2 Uncertainty Sampling (US)

Uncertainty sampling [9, 10] is perhaps the simplest and most commonly applied approach in active inference. In this framework, we select the node for which the model is most uncertain and label it by the oracle (i.e., Yelp.com). For example, while using SPEAGLE for binary classification of the network entities (users, reviews, and/or products), uncertainty sampling selects the node whose final belief is near 0.5. Hence, we utilize a general entropy-based uncertainty sampling approach [10], in which we compute the entropy of the final beliefs as the uncertainty measure given in Eq. (3.4):

$$x^*_{US} = \underset{x}{\text{argmax}} - \sum_i b_x(y_i) \log b_x(y_i) \tag{3.4}$$

Here $b_x(y_i)$ is the belief of node x to belong to class y_i, that is the marginal probability of assigning each node of type $\{R\}$ with the label y_i from label domain $\mathcal{L}_R = \{genuine, fake\}$. As such, at each iteration we select the review node with the highest uncertainty score to be labeled by the oracle. We incorporate the provided label by the oracle by initiating the review's priors as $\{\epsilon, 1-\epsilon\}$ if it is labeled as fake, and $\{1-\epsilon, \epsilon\}$ otherwise.

3.3 Query-by-Committee

Query-by-Committee (QBC) is an effective method of sampling for active inference where disagreement among different committee members is exploited to select nodes for labeling. QBC approach involves maintaining a committee $C = \{\theta^{(1)}, \ldots, \theta^{(|C|)}\}$ of models which represent competing hypotheses. Each committee member is then allowed to vote on the labeling of candidate nodes (i.e., reviews). The most informative candidates are those about which the committee members disagree the most. The key requirements of the QBC approach are (i) constructing a committee of models that represent different regions of input space and (ii) a measure of disagreement among the committee members.

(i) Committee Building: In this work, we build the committee by selecting 4 features out of 16 review features at random without replacement four times. This gives us a committee of four members each utilizing 4 review features to compute their priors (see Section 2.1).

(ii) Disagreement measure: We utilize the average *Kullback-Leibler (KL) Divergence* proposed by MacCallum et al. [12] as our disagreement measure which is an information-theoretic approach to calculate the difference between two probability distributions. This strategy is called the *soft voting (SV)* and represented by Eq. (3.5):

$$x^*_{QBC-SV} = \underset{x}{\text{argmax}} \frac{1}{|C|} \sum_{c=1}^{|C|} D(b_x^{\theta^{(c)}} || b_x^C) \tag{3.5}$$

where, $D(b_x^{\theta^{(c)}} || b_x^C) = \sum_i b_x^{\theta^{(c)}}(y_i) \log \frac{b_x^{\theta^{(c)}}(y_i)}{b_x^C(y_i)}$.

Here $\theta^{(c)}$ represents a particular member model in the committee and C represents the whole committee. $b_x^C(y_i) = \frac{1}{|C|} \sum_{c=1}^{|C|} b_x^{\theta^{(c)}}(y_i)$ is the average belief that y_i is the correct label for node x. This soft voting measure considers the node as highest informative which has the largest average difference between the label distributions of any one committee member and the whole committee.

For a budget $\mathcal{B}$, at each iteration we select the review node with the highest disagreement score to be labeled by the oracle. We leverage the provided label during the inference in the next step, by initiating priors as $\phi_{x^*} = \{\epsilon, 1-\epsilon\}$ if selected review is labeled fake, and $\{1-\epsilon, \epsilon\}$ otherwise.

In addition to the above, we build two strategies (i) most-sure disagreement and (ii) least-sure disagreement, above the soft voting based QBC approach motivated by Sharma et al. [21]. Our approach is different from [21] in a sense that they use uncertainty of different features, whereas, we use disagreement of different committee members. Most-sure disagreement occurs if the committee members have *strong* and *conflicting* evidence about an instance and least-sure disagreement occurs if the committee members have *no conclusive* evidence about an instance. For example, when half of the committee members vote $fake$ and the other half vote $genuine$ for the same node, then in most-sure disagreement the committee members are more certain about their decision (e.g., beliefs $[0.01\ 0.99]$ for $fake$ and $[0.99\ 0.01]$ for $genuine$), whereas, in least-sure disagreement the committee members are less certain about their decision (e.g., beliefs $[0.45\ 0.55]$ for $fake$ and $[0.55\ 0.45]$ for $genuine$). For the review network, we classify a node x based on the ratio $\frac{b_x(+)}{b_x(-)}$, where $b_x(+)$ ($b_x(-)$) is the belief of node x belonging to spam or positive class (non-spam or negative class):

$$y_x = \begin{cases} + & \text{if } b_x(+) > b_x(-), \\ - & \text{otherwise} \end{cases} \tag{3.6}$$

From the above equation, it follows that for a node x the committee member $\theta^{(c)}$ provides evidence for positive class if $\frac{b_x^{\theta^{(c)}}(+)}{b_x^{\theta^{(c)}}(-)} > 1$, and it provides evidence for negative class otherwise. Let P_x and N_x denote two sets, such that P_x contains committee members that provide evidence for positive class and N_x contains committee members that provide evidence for negative class:

$$P_x = \{\theta^{(c)} | \frac{b_x^{\theta^{(c)}}(+)}{b_x^{\theta^{(c)}}(-)} > 1\} \tag{3.7}$$

$$N_x = \{\theta^{(c)} | \frac{b_x^{\theta^{(c)}}(-)}{b_x^{\theta^{(c)}}(+)} > 1\} \tag{3.8}$$

Note that these two sets are defined around a particular node x. The total evidence for node x of belonging to the positive class and the negative class are calculated using the

following Eq. (3.9) and (3.10) respectively:

$$E^{+}(x) = \prod_{\theta^{(c)} \in P_x} \frac{b_x^{\theta^{(c)}}(+)}{b_x^{\theta^{(c)}}(-)} \tag{3.9}$$

$$E^{-}(x) = \prod_{\theta^{(c)} \in N_x} \frac{b_x^{\theta^{(c)}}(-)}{b_x^{\theta^{(c)}}(+)} \tag{3.10}$$

Our investigation shows that we have to optimize several objectives at the same time to make this evidence framework work on top of the QBC approach:

- Committee members should disagree on node x (i.e., high average KL divergence score).
- For most-sure disagreement, both $E^{+}(x)$ and $E^{-}(x)$ need to be large.
- For least-sure disagreement, both $E^{+}(x)$ and $E^{-}(x)$ need to be small.

We define the overall evidence of node x as:

$$E(x) = E^{+}(x) + E^{-}(x) \tag{3.11}$$

This aggregation makes sense as the overall evidence $E(x)$ is large if both $E^{+}(x)$ and $E^{-}(x)$ are large and close to each other. Similarly, $E(x)$ is small when both $E^{+}(x)$ and $E^{-}(x)$ are small. Furthermore, selecting an unlabeled node x for which $E(x)$ is largest (or smallest) will not guarantee that the committee members have disagreement on x. To guarantee the disagreement of committee members, we first rank the nodes in decreasing order of their soft voting score x_{QBC-SV} (measured by equation (3.5)) and take the top k nodes. Let S be the set of top k nodes on which the committee members disagree the most. The most-sure disagreement approach selects the node with the maximum overall evidence:

$$x^{*}_{QBC-MS} = \underset{x \in S}{\operatorname{argmax}}\ E(x) \tag{3.12}$$

and, the least-sure disagreement approach selects the node with the minimum overall evidence:

$$x^{*}_{QBC-LS} = \underset{x \in S}{\operatorname{argmin}}\ E(x) \tag{3.13}$$

Most of the existing works on active inference do not use any relational information for query selection during label acquisition. There exist some recent works which utilize the relational information among the instances to improve the selection strategy [3, 4]. However, requirement of an initial labeled training graph or non-scalable greedy approach [3] makes some existing techniques inapplicable in our spam detection setting. We utilize both metadata and relational information for label acquisition using two techniques, one of them is a modified version of ALFNET [4] and the other is Expected UnCertainty Reach (EUCR) that we propose in this work. We describe the relational label acquisition approaches in the following sections.

3.4 ALFNET

Proposed by Bilgic et al. [4], ALFNET is an active learning algorithm for collective classification. This algorithm uses two learners called CO (content-only) and CC (collective classifier), and combines their decision in order to select nodes for labeling. In particular, this algorithm considers those nodes to be informative for which the decisions of the two classifiers differ. It is assumed that the labels are acquired iteratively in batch size of k. At first, this algorithm clusters the nodes in the graph into C clusters using the network structure of the data. Then, it selects the k clusters which satisfy two important properties: (i) the decisions of CO and CC differ the most, and (ii) the decisions of the classifiers do not match with the already observed labels in the cluster. Based on these two properties, an overall score is computed for each cluster $c \in C$ and for a batch of size k, the top k clusters $C_k \subset C$ are selected. From each of these top k clusters a node $x \in c_i$ $(i = 1, \ldots, k)$ is randomly selected for labeling. For further details on this algorithm, we refer the readers to the original paper [4].

We modify this algorithm to work with our spam detection framework. Specifically, instead of using Iterative Classification Algorithm (ICA), we utilize SpEagle as the collective classifier CC and logistic regression (as original work) as the content-only classifier CO. The CO is trained based on the features extracted from meta-data. Furthermore, we construct a review–review network on which to perform the clustering, by connecting two reviews with an edge if they share at least one reviewer. Initially, we have no labels to train the CO classifier. Therefore, we first sort the clusters by size and select one random review node to query from each of the top k clusters with largest size. The acquired labels constitute the initial training set for CO. We also incorporate these labels to CC for inference. Following the initial selection, in each iteration we compute a score for each cluster based one the disagreement between CO and CC as well as the estimated labels in the clusters. We then select top k clusters based on these scores to draw a node x for querying from each cluster randomly.

The main constraints of ALFNET are that (i) it needs several iterations to acquire enough labels to be able to learn an effective CO classifier, (ii) CO is susceptible to high class-imbalance, as most of the acquired labels are non-spam, and (iii) it needs to re-train CO at every iteration. As a result, it requires more labels to improve performance over random sampling in our opinion spam detection setting.

3.5 Expected UnCertainty Reach (EUCR)

Finally, we propose a label acquisition approach which considers both the uncertainty of a node as well as the uncertainty of other nodes close-by to it in the network structure. Our main objective is to find "islands" of uncertainty, in which we aim to obtain correct classification for all the nodes

by acquiring labels for only a few. Following our intuition, we find three important characteristics of a valuable node for label acquisition, those are (i) uncertainty of a node, (ii) density of the region the node belongs to, and (iii) its proximity to other uncertain nodes. As such, a node is more valuable to query it exhibits uncertain beliefs and resides close-by to other uncertain nodes in a dense region, such that acquiring its label could help the nodes in its proximate neighborhood.

Based on our intuition about the characteristics of a valuable node, we propose a scoring measure called Expected UnCertainty Reach (EUCR). In this method, to address the first two characteristics of a valuable review node, we first calculate the weighted uncertainty score WUC_x for all review nodes $x \in R$ using Eq. (3.14).

$$WUC_x = -w_x \sum_i b_x(y_i) \log b_x(y_i) \tag{3.14}$$

where w_x is calculated from the user-degree of the corresponding review node as

$$w_x = \frac{UD_x - \min_{UD}}{\max_{UD} - \min_{UD}} .$$

In particular UD_x denotes the degree (total number of reviews) of the user who posted review x and $\min_{UD}$ and max_{UD} denote the minimum and maximum degree of a user node in the network. The weighted uncertainty (WUC) score gives higher values to those (review) nodes which are more uncertain and also reside in a denser region (i.e., have many other review nodes nearby).

We then rank the review nodes based on their WUC score and take the top k nodes. Let S be the set of top k nodes that we pick by the uncertainty of the nodes and density of the region they belong to. We want a review node x to be not only uncertain itself, but also close-by to many other uncertain review nodes. Importantly, due to the existence of homophily it is considered that the neighboring nodes have similar labels, thus acquiring label of a node from a dense region helps all nodes in that region. To quantify proximity, we leverage the review–review network (denoted by G_R) where two reviews are connected if they share the same reviewer, on which we compute the Random Walk with Restart probability vector p_x for x. $p_x(j)$ depicts the probability of reaching node j through a (infinitely long) random walk, with occasional restarts to x. As such, this probability captures proximity of j to x. Then for each node $x \in S$ we calculate the probability of reaching a node j for all $j \in R$ under random walk with restart as Eq. (3.15).

$$p_x = cWp_x + (1-c)e_x \tag{3.15}$$

where, $1-c$ ($c = 0.85$) is the teleportation probability, e_x is a unit vector containing 1 for node x and 0's for all other nodes, and W is the column normalized adjacency matrix of G_R. Eq. (3.15) defines a linear system problem, where we can write p_x as

$$p_x = (1-c)(I - cW)^{-1} e_x \tag{3.16}$$

The most informative node is then the one with the maximum total uncertainty reach as weighted by proximity, i.e.,

$$x^*_{EUCR} = \underset{x \in S}{\operatorname{argmax}} \sum_{j \in R} p_x(j) \times WUC_j \tag{3.17}$$

In summary, we assume that a node is more valuable not only for its level of uncertainty and residence in a dense region but also with respect to the existence of other uncertain nodes in its proximity (or "reach").

4 Evaluation

We evaluate our approach on two real world datasets from Yelp.com. These datasets consist of recommended as well as filtered reviews from Yelp. We describe the datasets and evaluation metrics, followed by performance results.

Dataset Description In this work, we evaluated our method on two datasets from Yelp.com, a summary of which is given in Table 1. Our first dataset, called `YelpChi` has been collected and used by [16] and contains a list of reviews from the hotels and restaurants in Chicago. The second dataset, called `YelpNYC` that we collected and used in [19], contains reviews of restaurants in New York City.

Yelp has its own proprietary filtering algorithm, which filters out reviews. Yelp has made all its *recommended* as well as *filtered* reviews public. We consider them as *genuine* and *fake* respectively. We also separate the users in two classes, benign vs. spammer, where spammers are those users with at least one filtered review.

Table 1: *Review datasets used in this work.*

Data	**#Reviews** (filtered %)	**#Users** (spammer %)	**#Prod.** (rstr.)
`YelpChi`	67,395 (13.23%)	38,063 (20.33%)	201
`YelpNYC`	359,052 (10.27%)	160,225 (17.79%)	923

Evaluation Metrics We use three evaluation metrics to measure the performance of our approach and all other compared approaches. We generate the precision vs. recall (PR) curve for different thresholds and calculate the area under the curve to get Average precision (AP). For spam detection in an imbalanced dataset (fake class represents minority), often the top positions in the ranking are more important. Therefore, we also calculate (i) *precision@k* and (ii) *NDCG@k*, for $k = 100, 200, \ldots, 1000$ to provide the performance on the top positions of the ranking.

Performance Results We compare the performance of different label acquisition approaches described in Section 3. Fig. 1 provides the AP curves with varying budget (e.g., budget = $0, 1, 2, \ldots, 500$) of compared approaches for both user and review ranking. In review ranking, EUCR outperforms all the competing approaches for both the datasets, except for ALFNET in `YelpNYC`. However, ALFNET starts

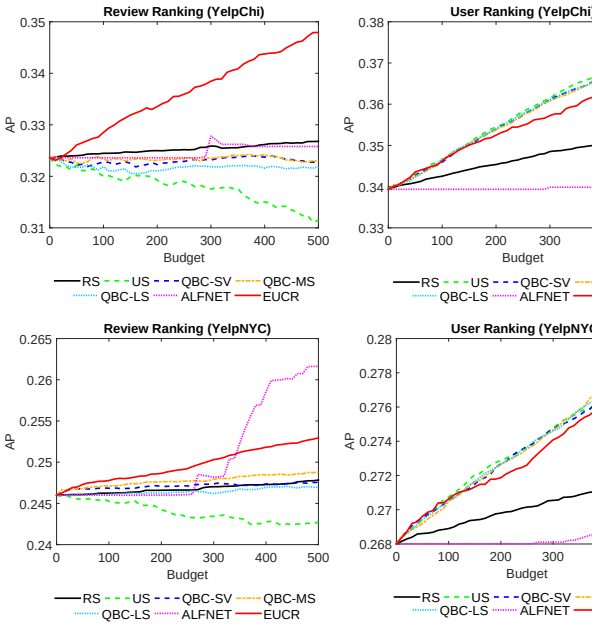

Figure 1: *AP of compared methods on* `YelpChi` *(top),* `YelpNYC` *(bottom) for both user and review ranking.*

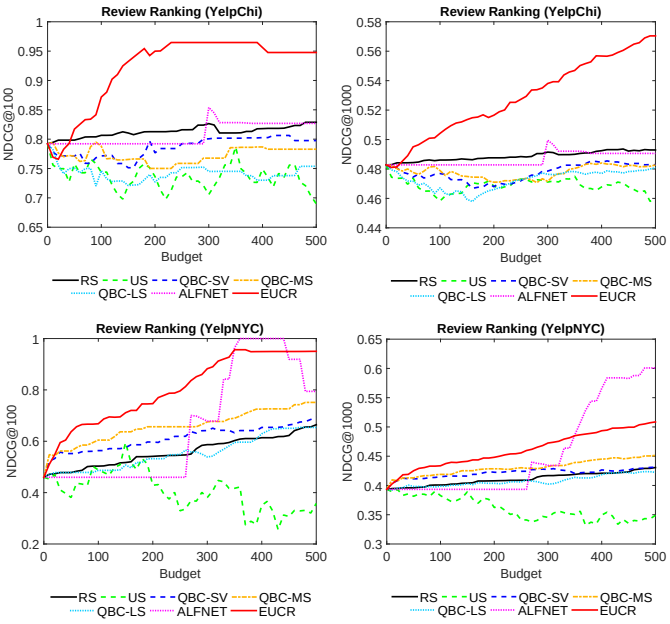

Figure 2: *NDCG@100 (left) and NDCG@1000 (right) of compared methods on* `YelpChi` *(top),* `YelpNYC` *(bottom) for review ranking with varying budget* $(0, 1, \ldots, 500)$.

performing well only after around 350 labels are acquired, whereas, EUCR does better with fewer labels. Experiments show that the sudden performance increase of ALFNET is due to acquiring more $fake$ labels only after 300 labels, which improves the learning of CO and also helps the inference of CC. Specifically, out of the first 300 acquired labels only 26 are from $fake$ class, whereas, for the next 200 acquired labels 123 are $fake$. Thus acquiring some balance between the number of $fake$ and $genuine$ labels improves ALFNET's performance. However, balance is not always guaranteed—even after 500 labels for `YelpChi`, ALFNET performs worse than RS. Fig. 2 shows the NDCG curves of the compared methods with varying budget (e.g., budget = $0, 1, 2, \ldots, 500$) and fixed k (e.g., $k = 100, 1000$) to better depict the performance of review ranking. Again, the *NDCG* curves for compared methods show similar trend as the AP curves for review ranking.

Our analysis shows that the US and QBC approaches, which do not consider the network structure in label acquisition end up selecting nodes which may be most uncertain (or most disagreed upon), however, not representative (i.e., close-by) of some or many other nodes. Although acquiring the labels for such review nodes are useful for the classification of corresponding users, they are not assisting other review nodes. As a result, both US and QBC (SV, MS, LS) have significant performance improvement for user ranking, but same is not true for review ranking as depicted in Fig. 1 and Fig. 2. For user ranking these baseline approaches provide very close results. Our EUCR approach also shows a comparable trend to those baselines.

Our analyses show that when a label acquisition method selects reviews written by different users then it is likely

Table 2: *Summary of* 500 *labeled reviews of compared method for* `YelpChi` *and* `YelpNYC` *datasets.*

	`YelpChi`		`YelpNYC`	
Methods	#unique users	#fake/#gen. labeled	#unique users	#fake/#gen. labeled
RS	494	62 / 438	500	53 / 447
US	500	231 / 269	500	205 / 295
QBC-SV	500	186 / 314	476	164 / 336
QBC-MS	500	183 / 317	494	154 / 346
QBC-LS	500	184 / 316	495	163 / 337
ALFNET	483	19 / 481	335	149 / 351
EUCR	498	139 / 361	500	131 / 369

that more users get correct labels (i.e., spammer or benign), hence, performance improves for user ranking. Again, label acquisition of more $fake$ reviews which are also representative of neighboring $fake$ reviews improves the performance of review ranking. In Table 2, we provide the statistics of the labeled reviews and their corresponding users for different compared label acquisition approaches on both `YelpChi` and `YelpNYC` datasets with budget 500. This summary shows that our proposed label acquisition approach EUCR provides labels to the reviews of different users, resulting approximately as many correct user classification as the budget size. On the other hand, ALFNET acquires labels for multiple reviews of the same user, allowing label passing to fewer number of unique users, hence, reducing user ranking performance. EUCR also has some balance between number of $fake$ and $genuine$ reviews compared to RS and ALFNET. As our datasets are imbalanced (fake being minority), RS gets the most imbalanced number of $fake$ vs. $genuine$ label passing. Although the summary

shows better statistics for US and QBC, these approaches do not achieve the expected performance due to selfishly selecting uncertain (or disagreed) nodes without considering the neighborhood information of the corresponding nodes in the network structure.

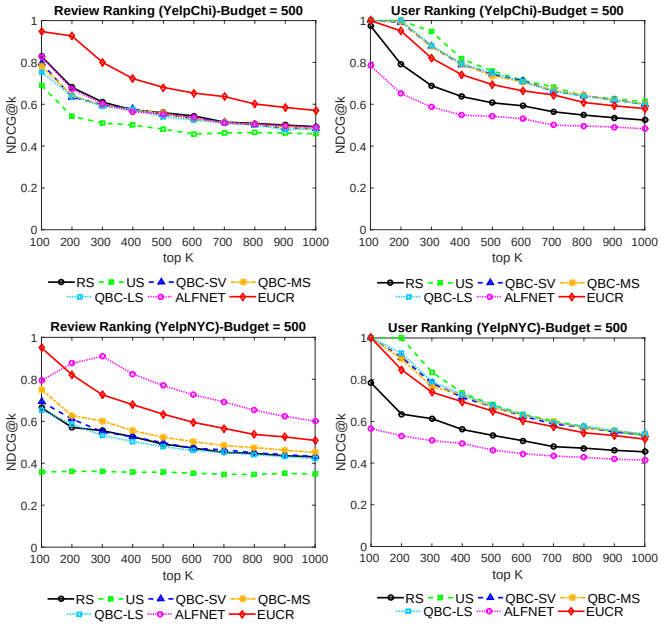

Figure 3: *NDCG@k on* `YelpChi` *(top),* `YelpNYC` *(bottom) for both user and review ranking with budget 500.*

We further provide the *NDCG@k* curves in Fig. 3 for varying top k (e.g., $k = 100, 200, \ldots, 1000$) nodes and budget = 500, to better describe the ranking performance of compared methods, for both user and review ranking. Our EUCR approach outperforms all other compared approaches significantly on `YelpChi` for review ranking. However, ALFNET performs better than EUCR on `YelpNYC` with budget 500. The same arguments for Fig. 1 hold here as well. Having smaller budget (e.g., budget = 300), EUCR outperforms ALFNET (due to imbalanced labeled data) as well as other approaches, as depicted in Fig. 4. Recall that besides a larger budget (i.e., training data), ALFNET requires a balanced labeled set to perform well and re-trains its local feature-based classifier CO at every step.

In Table 3 we also show the *precision@k* values for review ranking under fixed budget = 300, to provide further evidence of the ranking performance of the compared methods. Once again, our EUCR approach outperforms RS as well as all other label acquisition approaches on both `YelpChi` and `YelpNYC` datasets.

In conclusion, our analyses show that most state-of-the-art approaches perform better than random sampling for only one type of ranking, users or reviews. In contrast, our EUCR approach achieves significant improvement over random selection for both user and review ranking. Specifically, with a small budget of 300 it improves *NDCG@100* by 20–34%,

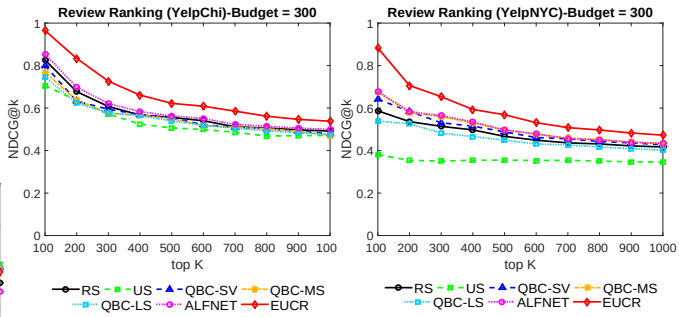

Figure 4: *NDCG@k on* `YelpChi` *(left),* `YelpNYC` *(right) for review ranking with budget 300.*

and *precision@100* by 14–30% for review ranking, and for user ranking it improves *NDCG@100* by 9–23%, and *precision@100* by 12–28%, over random sampling used in [19].

5 Related Work

Although both active learning and active inference are widely explored in the literature (detailed survey in [1, 20]), there are only a few recent works on active learning (or inference) applied to network data. In network data, often the label of a node is influenced by its neighborhood. Hence, the common intuition is that knowing the label of a particular node can help inferring labels of other nodes in its neighborhood. In [18], Rattigan et al. proposed to select the most central nodes for labeling in order to get more significant impact. However, exploiting network structure sometimes becomes disadvantageous. In collective classification, wrong labels can be propagated throughout the network, misclassifying other unlabeled nodes. Bilgic et al. [3] proposed a collective classification approach called *Reflect and Correct (RAC)* to find islands of misclassification to correct the labels of a few nodes to improve performance. However, RAC requires an initial labeled training graph to find misclassification. They also proposed in [3] a greedy approach called *AIGA* which acquires the labels by minimizing the expected error (e.g., log loss). In AIGA, the expected error is minimized by considering all possible labels for each network instance, which is intractable to be applied on large-scale graphs. A similar approach to AIGA called Expected Risk Minimization (ERM) has been proposed by Macskassy [13] which provides significant speed up over AIGA by leveraging the graph structure (e.g., betweenness centrality) to initialize good candidates for labeling. Moreover, there are some scenarios where both meta information and network structure are available (like our spam detection problem). The work in [4] proposed an algorithm ALFNET where they utilize two classifiers (i) content-only, and (ii) collective classifier to combine their prediction for label acquisition. In particular, the nodes on which the two classifiers have disagreement are considered to be good candidates. While this approach has been tested on document classification, we

Table 3: *Precision@k for review ranking on* YelpChi *and* YelpNYC *with budget 300.*

	YelpChi							YelpNYC						
k	**RS**	**US**	**Q'-SV**	**Q'-MS**	**Q'-LS**	**A'NET**	**EUCR**	**RS**	**US**	**Q'-SV**	**Q'-MS**	**Q'-LS**	**A'NET**	**EUCR**
100	0.78	0.64	0.77	0.75	0.70	0.81	**0.98**	0.51	0.41	0.57	0.60	0.49	0.60	**0.85**
200	0.62	0.59	0.58	0.60	0.57	0.64	**0.81**	0.49	0.36	0.54	0.52	0.50	0.52	**0.65**
300	0.55	0.52	0.55	0.53	0.53	0.56	**0.68**	0.47	0.35	0.48	0.51	0.45	0.52	**0.60**
400	0.51	0.48	0.53	0.53	0.53	0.53	**0.61**	0.46	0.36	0.48	0.49	0.44	0.49	**0.54**
500	0.51	0.46	0.52	0.52	0.50	0.51	**0.57**	0.43	0.36	0.45	0.45	0.42	0.45	**0.52**
600	0.50	0.46	0.48	0.48	0.48	0.51	**0.56**	0.42	0.36	0.42	0.44	0.41	0.44	**0.48**
700	0.47	0.45	0.47	0.48	0.47	0.48	**0.54**	0.41	0.36	0.42	0.42	0.40	0.42	**0.46**
800	0.47	0.43	0.47	0.47	0.46	0.47	**0.52**	0.40	0.35	0.41	0.42	0.40	0.42	**0.45**
900	0.46	0.44	0.46	0.46	0.45	0.47	**0.50**	0.39	0.35	0.41	0.41	0.39	0.40	**0.44**
1000	0.46	0.44	0.45	0.44	0.45	0.46	**0.50**	0.39	0.35	0.40	0.40	0.38	0.40	**0.43**

adapt it to our opinion spam detection setting and compare to our proposed approach EUCR. To the best of our knowledge, ours is the first work on active inference for collective opinion spam detection, where we utilize all of network, meta-data, and active label acquisition simultaneously.

6 Conclusion

In this work we extend the semi-supervised opinion spam detection framework SPEAGLE [19] with active inference, by carefully selecting valuable nodes (for acquiring labels from the oracle) within a small budget based on the network structure. Our main contributions are:

- We show how to adapt existing general label acquisition techniques of active inference for the semi-supervised relational inference setting.
- We present three useful characteristics of a valuable node for querying: (i) uncertainty, (ii) neighborhood density, and (iii) proximity to other uncertain nodes.
- We propose a new label acquisition approach called Expected UnCertainty Reach (EUCR) for relational active inference, which selects uncertain nodes from dense regions within reach to other uncertain nodes.

We evaluate our method on two large datasets from Yelp.com, where EUCR outperforms random selection as well as other state-of-the-art approaches.

Acknowledgments The authors thank the anonymous reviewers for their useful comments. This material is based upon work supported by the ARO Young Investigator Program under Contract No. W911NF-14-1-0029, NSF CAREER 1452425, IIS 1408287 and IIP1069147, DARPA Transparent Computing Program under Contract No. FA8650-15-C-7561, a Facebook Faculty Gift, an R&D grant from Northrop Grumman Aerospace Systems, and Stony Brook University Office of Vice President for Research. Any conclusions expressed in this material are of the authors' and do not necessarily reflect the views, either expressed or implied, of the funding parties.

References

[1] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and P. S. Yu. Active learning: A survey, 2014.

[2] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In *ICWSM*, 2013.

[3] M. Bilgic and L. Getoor. Effective label acquisition for collective classification. In *ACM KDD*, pages 43–51, 2008.

[4] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. In *ICML*, pages 79–86, 2010.

[5] S. Feng, L. Xing, A. Gogar, and Y. Choi. Distributional footprints of deceptive product reviews. In *ICWSM*, 2012.

[6] N. Jindal, L. Bing, and E.-P. Lim. Finding unusual review patterns using unexpected rules. In *CIKM*, 2010.

[7] N. Jindal and B. Liu. Opinion spam and analysis. In *WSDM*, pages 219–230, 2008.

[8] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. 1980.

[9] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *ICML*, pages 148–156, 1994.

[10] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *ACM SIGIR*, pages 3–12, 1994.

[11] M. Luca. Reviews, reputation, and revenue: The case of yelp.com. In *Working Paper 12-016, Harvard Bus. Sch.*, 2011.

[12] A. MacCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *ICML*, 1998.

[13] S. A. Macskassy. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In *ICML*, pages 385–392, 2011.

[14] P. V. Marsden. Homogeneity in confiding relations. *Social Networks*, 10(1):57–76, Mar. 1988.

[15] A. Mukherjee, L. Bing, and N. S. Glance. Spotting fake reviewer groups in consumer reviews. In *WWW*, 2012.

[16] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance. What yelp fake review filter might be doing? In *ICWSM*, 2013.

[17] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *ACL*, pages 309–319, 2011.

[18] M. Rattigan, M. Maier, and D. Jensen. Exploiting network structure for active inference in collective classification. In *ICDM*, pages 429–434, 2007.

[19] S. Rayana and L. Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*, 2015.

[20] B. Settles. Active learning. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, pages 1–114, 2012.

[21] M. Sharma and M. Bilgic. Most-surely vs. least-surely uncertain. In *ICDM*, 2013.

[22] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. In *ICDM*, 2011.

[23] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In *KDD*, 2012.

[24] J. Ye and L. Akoglu. Discovering opinion spammer groups by network footprints. In *ECML/PKDD*, pages 267–282, 2015.

[25] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding BP and its generalizations. In *Explor. AI in New Millen.* 2003.