Collaborative Classification from Noisy Labels

Lucas Maystre
Spotify

Nagarjuna Kumarappan
Spotify

Judith Bütepage
Spotify

Mounia Lalmas
Spotify

Abstract

We consider a setting where users interact with a collection of N items on an online platform. We are given class labels possibly corrupted by noise, and we seek to recover the true class of each item. We postulate a simple probabilistic model of the interactions between users and items, based on the assumption that users interact with classes in different proportions. We then develop a message-passing algorithm that decodes the noisy class labels efficiently. Under suitable assumptions, our method provably recovers all items' true classes in the large N limit, even when the interaction graph remains sparse. Empirically, we show that our approach is effective on several practical applications, including predicting the location of businesses, the category of consumer goods, and the language of audio content.

1 INTRODUCTION

Over the last two decades, online platforms have become ubiquitous. They let users access and interact with vast collections of restaurant and consumer goods reviews, audio and video content, and more. The success of these platforms comes, in part, from their ability to reach a large number of users and from their ability to offer access to a large number of items—typically, many more than any single user could reasonably interact with (Anderson, 2006; European Commission, 2016). A critical problem that these platforms face is the organization and categorization of information about items. For example, correctly identifying the location of a restaurant on a crowd-sourced review service or the language of a podcast on an audio streaming platform is crucial to providing a good experience to users across the world. This problem is particularly challenging when item metadata come from third-parties (e.g., on marketplaces) or from users themselves (e.g., on collaborative platforms), as the quality and consistency of that information might be low.

In this work, we consider a prototypical instance of this problem. We seek to classify each item as belonging to one of K different classes. We assume that we have access to a class label for each item, but that some of these labels are inaccurate (i.e., corrupted by noise). Furthermore, we assume that we are given a set of users and observe data about *who* interacts with *what*. We ask the question: Can we take advantage of user-item interactions to correct erroneous labels?

Our starting point is a natural assumption about users of online platforms, stating that users interact with classes in different proportions. For example, users tend to listen to podcasts mostly in languages they are fluent in, and they review (or consult reviews of) restaurants in a handful of cities they live in or visit. In consequence, two items that a given user interacts with are likely to be of the same class. Building on this assumption, we develop a structured probabilistic model that relates the network of user-item interactions and the noisy class labels to the items' true classes (Section 3). This model leads to a simple iterative message-passing inference algorithm. Informally, given an item, the algorithm considers the set of users having interacted with that item, and uses the label of other items that these users have interacted with in order to determine the class of the item. This links our work to well-known methods in statistical relational learning (Macskassy and Provost, 2007; Bhagat et al., 2011; Sen et al., 2008).

We aim to characterize the performance of our approach theoretically (Section 4). To this end, we consider an idealized generative model of online platforms. This model describes, among others, how users choose items to interact with and how class labels are corrupted by noise. Under this model, we identify necessary and sufficient conditions such that, asymptotically, our method perfectly recovers the true class of *all* items with high probability. These conditions are order-optimal, and they are surprisingly mild: they are essentially equiva-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

lent to having at least one user interacting with every item. These results set our method apart from generic statistical relational learning models, for which theoretical guarantees are usually difficult to obtain.

Next, we evaluate our algorithm empirically, using synthetic and real-world data (Section 5). We begin by illustrating the theoretical bounds and stress-testing their assumptions through simulations. We then study three classification tasks using public datasets from online platforms: a technical Q&A platform, a business review website, and an online retailer. We treat the labels provided in these datasets as ground truth and simulate noisy labels by corrupting some of the ground-truth labels at random. Our approach corrects a large fraction of the mistakes in all three cases, significantly outperforming competing heuristics. In addition, our approach is able to identify apparent mistakes in the ground-truth labels provided in the dataset. Finally, we investigate a language-identification task using a podcast dataset from Spotify, an audio streaming service. In addition to labels provided by show producers, we obtain high-quality ground-truth data. On this dataset, we show that applying our method results in a five-fold reduction in the error rate.

Our contributions are two-fold. First, we develop a probabilistic model and a matching inference algorithm that can be used to correct noisy item labels given user-item interactions. Our method is easy to implement and scales to millions of users and items effortlessly. Second, we derive tight bounds on our algorithm's sample complexity in the perfect recovery setting, under a natural model of user-item interactions. We show that our algorithm's sample complexity is order-optimal: No other algorithm can recover the true class of all items with fewer interactions (up to constant factors). Taken together, our results show that a simple assumption—users interact with classes in different proportions—leads to an effective "collaborative" classification method. Driven by theory, backed by favorable empirical results, we believe that our method will be valuable to machine-learning practitioners at large.

2 RELATED WORK

To the best of our knowledge, the exact problem we address in this paper, correcting noisy class labels using user-item interactions, has not been studied previously. Nevertheless, our approach builds upon existing methods and ties into a number of research areas. In this section, we present a brief survey.

Collective Classification. Perhaps the closest problem to ours is that of *collective classification*, where nodes in a network need to be jointly classified based on node features and the network's structure (Sen et al., 2008; Bhagat et al., 2011). Structural information can help, e.g., if nodes are more likely to be connected to other similar nodes, a property called homophily or assortativity (Newman, 2003). Common approaches to collective classification include local classifiers (Chakrabarti et al., 1998; Neville and Jensen, 2000; Lu and Getoor, 2003; Macskassy and Provost, 2007) and structured probabilistic models, both directed (Friedman et al., 1999; Taskar et al., 2001) and undirected (Taskar et al., 2002). Among those, the weighted-vote relational neighbor algorithm of Macskassy and Provost (2007) stands out as being simple and effective; we consider a variant in Section 3. Our work is closest in spirit to approaches based on graphical models, such as that of Taskar et al. (2002). In contrast to most of the existing work on collective classification, we consider a problem where the network has a bipartite structure and two distinct types of nodes (users and items), and we take advantage of this structure explicitly. Stankova et al. (2015) also consider bipartite networks, but in the context of binary classification, whereas we consider a multiclass problem.

Statistical Relational Learning. Beyond collective classification, our work is linked to a number of ideas from the statistical relational learning (SRL) literature (Getoor and Taskar, 2007). For example, our problem could likely be modeled using a Markov logic network (Domingos and Lowd, 2009) or a relational dependency network (Heckerman et al., 2000; Neville and Jensen, 2007). In a sense, our work can be understood as an application of SRL to the specific problem of classifying items using user-item interactions. This specialization leads to *a*) an inference algorithm that is particularly simple and efficient, and *b*) theoretical guarantees on the algorithms's output, two features that general-purpose SRL models usually lack.

Parity-Check Codes. Our approach is also closely related to classic algorithms in channel coding, whose goal is to efficiently recover a sequence of symbols (e.g., bits) transmitted over a noisy communication channel (Cover and Thomas, 2006). In particular, *low-density parity check* (LDPC) codes share some similarities to our work (Gallager, 1962; MacKay and Neal, 1997). LDPC codes identify and correct corrupted symbols by using additional parity checks connected to some of the symbols, in a similar way to how we take advantage of users to identify and correct mislabeled items. What distinguishes our work from LDPC codes is a different probabilistic model relating parity checks / users to symbols / item classes.

Weakly Supervised Learning. A large body of machine-learning literature studies settings where supervision is imperfect, because of label noise (Frénay and Verleysen, 2014) or because some observations are unlabeled (Chapelle et al., 2010), among others (Zhou, 2018). Existing work on learning from noisy labels typically assume access to features and i.i.d. observations (Angluin and Laird, 1988; Brodley and Friedl, 1999; Natarajan et al., 2013; Ratner et al., 2016). In contrast, we study a structured problem without item features but where item classes become dependent through users' interactions. Semi-supervised learning addresses a setting where part of the data are unlabeled (Chapelle et al., 2010). Some methods, in particular those based on random walks (Zhu et al., 2003; Zhou et al., 2004), are closely related to collective classification (Macskassy and Provost, 2007; Bhagat et al., 2011) and, in turn, to our work. They exploit a network that captures similarities between data to overcome missing labels.

Truth Inference in Crowdsourcing. A well-known problem in crowdsourcing is that of classifying items by querying multiple crowdworkers, whose answers might be unreliable. In a seminal paper, Dawid and Skene (1979) introduce an approach that estimates users' skill and items' labels jointly. Since then, a number of improvements have been proposed (Whitehill et al., 2009; Welinder et al., 2010; Karger et al., 2014; Manino et al., 2019). Whereas crowdsourcing systems usually ask users to provide class labels explicitly, we study a setting where users' interactions carry information about item classes only implicitly. Since our users' goal is not to provide labels, we make different assumptions on the information contained in user-item interactions.

Collaborative Filtering. Our approach shares some similarities with collaborative filtering (CF), a popular class of methods for modeling user preferences (Ricci et al., 2011). Just like CF, we take advantage of user-item interactions to infer properties of users and items. Most CF methods, such as matrix factorization (Koren et al., 2009), model users and items as points in a low-dimensional vector space. In contrast, we assume that items belong to one of K discrete classes and we model user preferences by using a categorical distribution. Most importantly, however, our model addresses a different problem: We are not interested in predicting preferences *per se*, but instead we seek to correct corrupted item labels. We justify our specific modeling choices by showing, in Section 5, that our method outperforms an approach based on a standard CF model.

3 INTERACTION MODEL

In this section, we formally introduce our collaborative classification problem. Starting with a simple assumption about users' behavior, we postulate a probabilistic model relating item classes to user-item interactions and derive two inference algorithms.

3.1 Problem Definition & Model

We consider online platforms where M users interact with N items belonging to one of K classes. We denote the set of users by the consecutive integers $[M] = \{1, \ldots, M\}$. Likewise, we denote the sets of items and classes by $[N]$ and $[K]$, respectively. We are given a bipartite interaction graph $\mathcal{G} = ([M], [N], \mathcal{E})$, where $\mathcal{E} \subseteq [M] \times [N]$ and an edge $(i, j) \in \mathcal{E}$ indicates that user i has interacted with item j. The set of items that user i interacts with is given by $\mathcal{N}_i = \{j \in [N] : (i, j) \in \mathcal{E}\}$. Similarly, the set of users interacting with item j is given by $\mathcal{N}_j = \{i \in [M] : (i, j) \in \mathcal{E}\}$. The class of item j is denoted by $v_j \in [K]$, and we let $\boldsymbol{v} = (v_1, \ldots, v_N)$. We do not get to observe $\boldsymbol{v}$ directly; instead, we are given a noisy version $\hat{\boldsymbol{v}} = (\hat{v}_1, \ldots, \hat{v}_N)$. Our goal is to recover the true classes $\boldsymbol{v}$ using the noisy labels $\hat{\boldsymbol{v}}$ and the user-item interaction graph $\mathcal{G}$.

To this end, we make the following key assumption: users interact with classes in different proportions. Formally, we assume that each user i is described by a vector of class proportions $u_i \in \Delta$, where $\Delta = \{x \in [0, 1]^K : \sum_k x_k = 1\}$ is the standard simplex, and we let $\boldsymbol{u} = (u_1, \ldots, u_M)$. This gives rise to the following probabilistic model. First, we endow u_i with a Dirichlet prior and v_j with a categorical prior. That is,

$$p_0(u_i) = \mathrm{Dir}(u_i \mid \alpha_i) \propto \prod_k u_{ik}^{\alpha_{ik}-1},$$
$$p_0(v_j) = \mathrm{Cat}(v_j \mid \beta_j) = \prod_k \beta_{jk}^{\mathbf{1}\{v_j = k\}},$$

independently for each $i \in [M]$ and $j \in [N]$, where $\alpha_i \in \mathbf{R}_{>0}^K$ and $\beta_j \in \Delta$. The parameter α_i can be used to encode prior beliefs about users' affinities towards different classes,[1] and, in the absence of such beliefs, we can use the flat prior $\alpha_{ik} \equiv 1$. The parameter β_{jk} captures the probability of item j belonging to class k after observing $\hat{v}_j$ (but before considering user interactions). In the absence of other information, a reasonable choice is to fix a noise level $\delta > 0$ and let

$$\beta_{jk} = \begin{cases} 1 - \delta & \text{if } k = \hat{v}_j, \\ \delta/(K-1) & \text{otherwise.} \end{cases}$$

[1] Consider, for example, a language classification application. In that case, the users' country of residence could be used to form an informative prior.

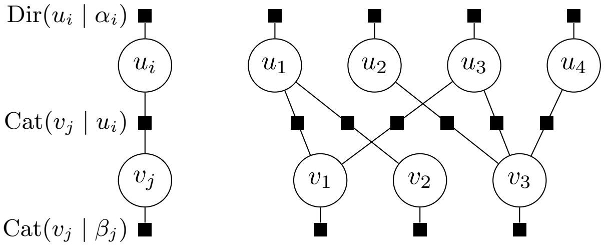

$\mathrm{Dir}(u_i \mid \alpha_i)$
u_i
$\mathrm{Cat}(v_j \mid u_i)$
v_j
$\mathrm{Cat}(v_j \mid \beta_j)$
u_1
u_2
u_3
u_4
v_1
v_2
v_3

Figure 1: Factor-graph representation of the model. Left: description of the factors. Dir and Cat represent Dirichlet and categorical factors, respectively. Right: example with 4 users and 3 items.

In experiments, we treat δ as a hyperparameter and find its optimal value by cross-validation. Next, for each edge $(i, j) \in \mathcal{E}$ we add a potential $f(u_i, v_j) = \mathrm{Cat}(v_j \mid u_i)$. This favors configurations where item j has a class for which user i has high affinity. Intuitively, this suggests that users choose items by first sampling from their class proportion vectors and then selecting an item within that class (we will consider a precise such generative model in Section 4). The resulting joint probability distribution is given by

$$p(\boldsymbol{u}, \boldsymbol{v}) \propto \Big[\prod_i \mathrm{Dir}(u_i \mid \alpha_i) \cdot \prod_j \mathrm{Cat}(v_j \mid \beta_j) \cdot \prod_{(i,j) \in \mathcal{E}} \mathrm{Cat}(v_j \mid u_i) \Big]. \tag{1}$$

We illustrate the model in Figure 1 with a concrete example, using the factor graph notation (Bishop, 2006).

3.2 Inference Algorithm

Ideally, given (1), we would like to obtain the marginal distribution $p(\boldsymbol{v})$. We would then be able to estimate the corrected item classes as $\arg\max_{\boldsymbol{v}} p(\boldsymbol{v})$. Unfortunately, computing this marginal distribution exactly is intractable. Instead, we start by approximating $p(\boldsymbol{u}, \boldsymbol{v})$ using a mean-field variational distribution,

$$q(\boldsymbol{u}, \boldsymbol{v}) \doteq \prod_i \mathrm{Dir}(u_i \mid \bar{\alpha}_i) \cdot \prod_j \mathrm{Cat}(v_j \mid \bar{\beta}_j),$$

where $\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_M)$ and $\bar{\boldsymbol{\beta}} = (\bar{\beta}_1, \ldots, \bar{\beta}_N)$ are variational parameters. Given such a distribution q, we can then easily compute the maximum-likelihood estimate of class of item j as $\bar{v}_j \doteq \arg\max_k \bar{\beta}_{jk}$. To find a distribution q that is a good approximation of p, we use the *coordinate-ascent variational inference* (CAVI) algorithm, also known as variational message-passing (Winn and Bishop, 2005; Blei et al., 2017). Starting from an arbitrary distribution q, the algorithm iteratively refines it by minimizing the divergence $\mathrm{KL}(q \| p)$, provably converging to a local minimum.

Algorithm 1 CAVI

```
1: repeat
2:     for $i = 1, \ldots, M$ do                          ▷ Update users.
3:         $\bar{\alpha}_i \leftarrow \alpha_i + \sum_{j \in \mathcal{N}_i} \bar{\beta}_j$
4:     for $j = 1, \ldots, N$ do                          ▷ Update items.
5:         for $k = 1, \ldots, K$ do
6:             $z_k \leftarrow \log \beta_{jk} + \sum_{i \in \mathcal{N}_j} \psi(\bar{\alpha}_{ik})$
7:         $\bar{\beta}_j \leftarrow \mathrm{softmax}(z)$
8: until has converged
```

Algorithm 2 wvRN

```
for $i = 1, \ldots, M$ do
    for $k = 1, \ldots, K$ do
        $x_{ik} \leftarrow \sum_{j \in \mathcal{N}_i} \mathbf{1}\{\hat{v}_j = k\}$
for $j = 1, \ldots, N$ do
    for $k = 1, \ldots, K$ do
        $z_k \leftarrow \sum_{i \in \mathcal{N}_j} (x_{ik} - \mathbf{1}\{\hat{v}_j = k\})$
    $\bar{v}_j \leftarrow \arg\max_k z_k$
```

Applying CAVI to our probabilistic model (1) results in Algorithm 1. For conciseness, we defer its full derivation to Appendix A in the supplementary file, but we briefly discuss the algorithm's simple structure. CAVI repeatedly updates user and item marginals (parametrized by $\bar{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\beta}}$ and initialized to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively) using closed-form expressions. Line 3 updates the distribution over a user's class proportions by considering soft class assignments of the items the user interacted with. Similarly, line 6 updates the distribution over an item's class by allocating larger probability mass to high-affinity classes of users that interacted with the item (note that the digamma function, $\psi(x) \doteq \Gamma'(x)/\Gamma(x)$, is monotonically increasing).

Running Time. One iteration of the algorithm entails passing messages of size $O(K)$ over the edges twice, and thus runs in time $O(K|\mathcal{E}|)$. The loops over users and items (lines 2 and 4) are trivially parallelizable, leading to a further speed-up. By way of example, on a standard laptop, a reference implementation[2] in the Python programming language takes approximately two minutes per iteration on a dataset with 120 M edges.

3.3 Alternative Heuristic

We develop Algorithm 1 using a principled approximation to a precise probabilistic model, but we observe

[2] The implementation is available at: `https://github.com/spotify-research/collabclass`.

that, at a high-level, it follows an intuitive idea. For a given item j, the algorithm tends to assign high weight to class k if other items that co-occur with j (i.e., other items that users interacting with j interact with) are of class k.

This suggests the following simpler heuristic. For all $j, \ell \in [N]$, let $w_{j\ell}$ denote the number of users that interact with both items j and ℓ. Predict the class of j using the majority class of all items weighted by the number of co-occurrences with j, i.e., $\bar{v}_j = \arg\max_k \sum_{\ell \neq j} w_{j\ell} \mathbf{1}\{\hat{v}_\ell = k\}$. This recovers the *weighted-vote relational neighbor* (wvRN) algorithm, a collective-classification procedure that has been demonstrated to work well on a wide range of problems (Macskassy and Provost, 2007; Stankova et al., 2015).

Algorithm 2 presents a computationally-efficient implementation tailored to our use-case, with running time $O(K|\mathcal{E}|)$. Note that wvRN does not take an item's own label into account when predicting that items' class (this can be observed by expanding line 3 into line 6). As such, we expect CAVI to outperform wvRN at least in settings where some items are very sparsely connected—we verify this experimentally in Section 5. Nevertheless, wvRN remains an important baseline.

4 THEORETICAL RESULTS

In this section, we consider an idealized generative model of online platforms and characterize the sample complexity of the two inference algorithms presented in Section 3, CAVI and wvRN. We defer the full proofs of our results to Appendix B in the supplementary file.

We extend (1) into a full generative model of online platforms as follows. Each item $j \in [N]$ draws a class $v_j \in [K]$ uniformly at random. Each user $i \in [M]$ first draws a vector of class proportions u_i from a symmetric Dirichlet with concentration parameter α. Then, they draw S class labels from a multinomial with probabilities $u_i \in \Delta$, resulting in a count vector $n_i \in \mathbf{N}^K$. Finally, for each k, they draw n_{ik} items uniformly at random among items of class k, resulting in a set $\mathcal{N}_i \subset [N]$ of items they interact with. In other words,

$$\begin{aligned} v_j &\sim \mathrm{Unif}\left([K]\right), & u_i &\sim \mathrm{Dir}(\alpha), \\ n_i &\sim \mathrm{Mult}(S, u_i), & \mathcal{N}_i &\sim \bigcup_k \mathrm{Unif}\left[\binom{\mathcal{V}_k}{n_{ik}}\right], \end{aligned}$$

independently for all $i \in [M]$ and $j \in [N]$, where $\mathcal{V}_k = \{j \in [N] : v_j = k\}$. Lastly, we assume that the observed class labels are corrupted independently and identically with probability $\delta \in [0, 1]$. That is, for a given j, the observed label $\hat{v}_j$ is equal to v_j with probability $1 - \delta$ and takes a value uniformly at random in $[K] \setminus \{v_j\}$ with probability δ. We call the resulting model the *sparse & biased interaction model* (SBM), and we denote a random dataset sampled from this generative model by $\mathcal{D} \sim \mathrm{SBM}$, where the parameters M, N, K, S, α and δ are omitted for conciseness. This model is clearly simplistic (e.g., it is unlikely that all users interact with the same number of items S), but it captures essential properties of the problem. We discuss and relax the assumptions in Appendix C.

First, we derive a consequence of our assumption that users interact with classes in different proportions. Lemma 1 formalizes the fact that this assumption leads to a notion of assortativity: two items connected through a user in the graph are more likely to be of the same class. This key property captures the essence of *why* learning from user interactions is effective.

Lemma 1 (Assortativity)**.** *Let $\mathcal{D} \sim \mathrm{SBM}$, and for any i, let $j, \ell \in \mathcal{N}_i$. Then, for any $k' \neq k$,*

$$p(v_\ell = k \mid v_j = k) = (1 + 1/\alpha) \cdot p(v_\ell = k' \mid v_j = k).$$

Proof. By construction, the probability that user i interacts with a first item of class k and a second item of class k' is $u_{ik} u_{ik'}$. Letting $k' \neq k$ and marginalizing over $u_i \sim \mathrm{Dir}(\alpha)$, we have

$$\begin{aligned} p(v_\ell = k, v_j = k) &= \int u_{ik}^2 \mathrm{Dir}(u_i \mid \alpha) du_i \\ &= (\alpha + 1)/(K^2\alpha + K), \\ p(v_\ell = k', v_j = k) &= \int u_{ik} u_{ik'} \mathrm{Dir}(u_i \mid \alpha) du_i \\ &= \alpha/(K^2\alpha + K). \end{aligned}$$

□

Next, we derive upper bounds on the sample complexity for perfect recovery. We consider a setting where the number of items N grows and the number of users M is a function of N, and we assume that the remaining parameters stay fixed. We say that an event $\mathcal{A}(N)$ holds *with high probability* (w.h.p.) if $\mathbf{P}[\mathcal{A}(N)] \to 1$ as $N \to \infty$. We first study wvRN. Theorem 1 states that $O(N \log N)$ users are sufficient to recover the true label of all items.

Theorem 1. *Let $\mathcal{D} \sim \mathrm{SBM}$, and let $\bar{\boldsymbol{v}}$ be the output of Algorithm 2 on $\mathcal{D}$. If $\delta < \frac{K-1}{K}$ and $M \geq \max\{16, 40\frac{(K\alpha+1)^2}{S}(1 - \frac{K}{K-1}\delta)^{-1}\} \cdot N \log N$, then for all $j \in [N]$, $\bar{v}_j = v_j$ w.h.p.*

Sketch of proof. Consider line 6 of Algorithm 2 Fixing j, letting $k \doteq v_j$ and $y_{i\ell} \doteq x_{i\ell} - \mathbf{1}\{\hat{v}_j = \ell\}$ for all $\ell \in [K]$, we have that $\bar{v}_j = k$ iff $\sum_{i \in \mathcal{N}_j} (y_{ik} - y_{ik'}) > 0$ for all $k' \neq k$. By Lemma 1 and by properties of our noise model, we can bound $\mathbf{E}[y_{ik} - y_{ik'}]$ by a positive function of S, K, α and δ. Additionally, we can show that if $M = O(N \log N)$ then $|\mathcal{N}_j| = O(\log N)$ w.h.p. The random variables $\{y_i\}$ are not independent, but we can control their dependencies, and we use a Chernoff bound for sums of dependent variables due to Janson (2004) to show that $\sum_{i \in \mathcal{N}_j}(y_{ik} - y_{ik'}) > 0$ w.h.p. The claim then follows by a union bound on j. □

The analysis of CAVI is more difficult due to the non-linearities on line 6. Nevertheless, we are able to derive a similar upper-bound on the number of users needed for perfect recovery. We formalize this in the next theorem.

Theorem 2. *Let $\mathcal{D} \sim \text{SBM}$, and let $\bar{\boldsymbol{\beta}}$ be the output of Algorithm 1 on $\mathcal{D}$ after one iteration. There exist C_1, C_2, C_3, C_4 independent of N such that if $M \geq C_1 N \log N$, $\alpha < C_2$, $\delta < C_3$, $S > C_4$, then for all $j \in [N]$, $\arg\max_k\{\bar{\beta}_{jk}\} = v_j$ w.h.p.*

Theorems 1 and 2 show that $O(N \log N)$ users are sufficient to recover the true classes of all items. But how many users are *necessary*? Our next theorem shows that this upper bound is order-optimal: If there are less than $\Omega(N \log N)$ users, then w.h.p. a growing number of items are disconnected in the interaction graph. There is no way to correct the noisy labels of disconnected nodes, and therefore the δ-fraction of these nodes that are corrupted by noise remain corrupted in the output. As such, this lower bound applies to *any* algorithm estimating $\boldsymbol{v}$ based on $\hat{\boldsymbol{v}}$, and not only to CAVI and wvRN.

Theorem 3. *Let $\mathcal{D} \sim \text{SBM}$. If $M \leq \frac{1}{5KS} N \log N$, then w.h.p. there exists a set of items $\mathcal{B} \subseteq [N]$ such that $|\mathcal{B}| \geq \log \log N$ and $\mathcal{N}_j = \varnothing$ for all $j \in \mathcal{B}$.*

Sketch of proof. We begin by viewing the bipartite interaction graph as a hypergraph over the N items, where the M edges are sets of size S corresponding to users' interactions. We then adapt a result due to Poole (2015) that extends the connectivity theory of Erdős-Rényi random graphs to hypergraphs. Poole's result states that, if $M < (1 - \varepsilon)S^{-1} N \log N$ edges are sampled uniformly at random, then w.h.p. at least $\lceil \log \log N \rceil$ nodes have degree 0, for any $\varepsilon > 0$. In our setting, however, edges of the hypergraph are not sampled uniformly at random, and our bound reflects the necessary adaptations. □

To summarize, under our generative model, both Algorithm 1 and Algorithm 2 are able to correct all the erroneous class labels, essentially as soon as the interaction graph is connected.

In light of these results, a reasonable question to ask is: How often is $M \gg N$ in practice? For most online platforms this typically holds in the following practical sense. Even when, strictly speaking, the number of items N exceeds the number of users M, there is almost always a much smaller number of items $N' \ll M$ that cover most of the user-item interactions. Thus, the regime $M = O(N \log N)$ is realistic if we restrict ourselves to "reasonably popular" items. Independently of these considerations, we will demonstrate in the next section that, empirically, our method works well even when $M < N$.

5 EXPERIMENTAL EVALUATION

Next, we study our approach empirically using synthetic and real-world datasets. By using synthetic datasets, we illustrate the theoretical results of Section 4 and study our model's robustness in a controlled setting. By using real-world datasets, we evaluate its performance on realistic interaction networks and on a broad set of practical classification problems.

5.1 Synthetic Datasets

Taken together, the results of Section 4 suggest a phase transition in the performance of our algorithms under SBM. Perfect recovery is highly probable or highly improbable if the number of users is above $CN \log N$ or below $C'N \log N$, respectively, for some constants C, C'. To illustrate this, we set $N = 1000, S = 5, K = 5, \alpha = 0.5$, and draw 200 realizations of $\mathcal{G}, \boldsymbol{u}, \boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ from SBM for different values of M and δ. For each sample, we run CAVI and wvRN on the noisy labels $\hat{\boldsymbol{v}}$ and record whether the output $\bar{\boldsymbol{v}}$ matches the ground-truth labels $\boldsymbol{v}$ perfectly. Figure 2 (left) shows the fraction of realizations for which we recover the true classes perfectly as a function of M, for three different values of δ. We observe a sharp transition from "never recover" to "always recover", as suggested by the theory. We also note that, as expected, the critical threshold for M increases with δ.

In Section 4, we assume that, within a class, users choose items to interact with uniformly at random. In practice, some items are likely more popular than others. To investigate the effect of a popularity bias on the empirical performance of our algorithms, we modify SBM as follows. For each item $j \in [N]$, we draw a popularity score from a Pareto Type II distribution with shape parameter 2 (Onnela et al., 2007). Conditioned on a particular class, each user then chooses items to interact with probability proportional to the items' popularity score (as opposed to uniformly at random). As a result, the degree distribution of the item nodes is highly skewed: A handful of items are connected to a large number of users, while most items are only connected to zero, one or a few users. We draw 50 random realizations of this model, using $N = 1000, S = 5, K = 5, \alpha = 0.5$ and $\delta = 0.1$, for

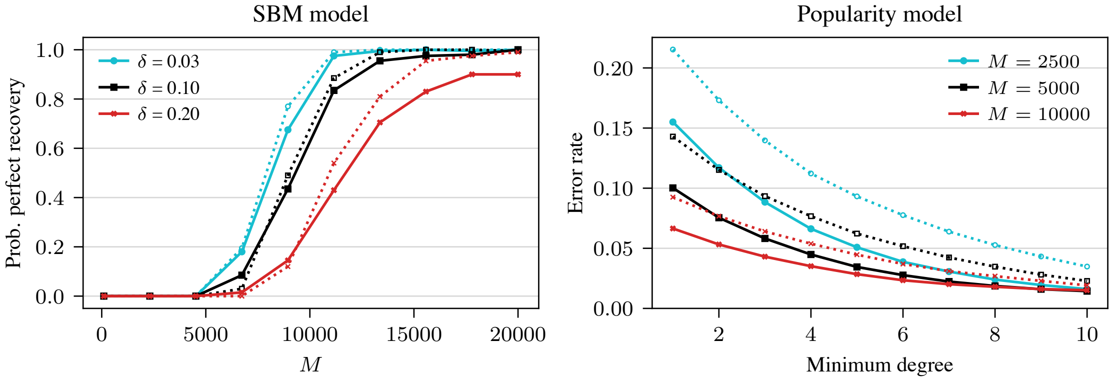


Figure 2: Experiments on synthetic data, fixing $N = 1000, S = 5, K = 5, \alpha = 0.5$. Solid lines are used for CAVI, dotted lines for wvRN. Left: empirical probability of perfect recovery as a function of M under SBM, for different δ. Right: error rate as function of minimum node degree under the Popularity model, for $\delta = 0.1$ and different M.

three different values of M. For each realization, we run CAVI and wvRN and record, for each item, whether the prediction matches the actual class. We also record the degree of the node corresponding to each item.

In Figure 2 (right) we plot the error rate (i.e., the fraction of mistakes) as a function of the items' minimum degree, averaged over the realizations In this popularity model, it is unlikely that we can recover the class of all items perfectly, as it is likely that some items are not connected to any users. However, if we disregard low-degree items, both algorithms are still successfully able to correct a large fraction of the mistakes—with a clear advantage to CAVI over wvRN.

5.2 Real-World Datasets

We begin by considering data[3] from three platforms: *a*) Stack Overflow, an online Q&A platform for programmers, *b*) Yelp, a crowd-sourced business review service, and *c*) Amazon, an e-commerce platform. In each case, users interact with items, be it questions, businesses or products. We seek to classify, respectively, questions by programming language, businesses by location, and products by category. Table 1 provides summary statistics for the datasets, and we provide additional details in Appendix D. Note that all three datasets are sparse and present heavy popularity biases. In the Amazon dataset, for example, over 40% of the items are connected to a single user, and over half of the users are connected to a single item. Thus, in addition to average performance, we also report the performance of items in the 50th and 90th percentiles of the node-degree distribution, denoted P_{50} and P_{90}, respectively. Finally, all three datasets provide class labels. We treat these labels as ground truth and artificially generate noisy versions, by corrupting every item's label independently with probability δ.

In addition to CAVI and wvRN, we also evaluate a baseline method inspired from Brodley and Friedl (1999) and denoted by MF+LR. Given a dataset, we proceed as follows. We learn a feature representation $x_j \in \mathbf{R}^D$ for each item j by using BPR, a matrix factorization model that predicts user-item interactions (Rendle et al., 2009). Given these learned features and the noisy labels $\hat{\boldsymbol{v}}$, we can train a classifier that predicts an item's class from its features. We partition the dataset into $L = 10$ folds. For each fold ℓ, we train a multinomial logistic regression classifier on the remaining $L-1$ folds, and use that classifier to predict on ℓ. We compare the predicted label to the label provided in the dataset: if they disagree and the classifier's confidence is above a threshold, we change the label to the one predicted by the classifier. We choose empirically optimal values the confidence threshold (and other hyperparameters) for each dataset.

In Table 2, we report the error rate, that is, the fraction of labels that differ from the ground-truth, for $\delta = 0.1$. Unsurprisingly, wvRN does not perform well on these difficult datasets, where most items are connected to only a few users. On the other hand, both CAVI and MF+LR systematically reduce the error rate below its baseline value of 0.1, with CAVI significantly outperforming MF+LR. These results highlight that CAVI remains effective even on network structures that deviate substantially from those studied in Section 4. On the Stack Overflow dataset, for example, CAVI reduces the error rate by 2.2× (3.3× when considering items in the 90th percentile, i.e., connected to 6 users or more). In Figure 3a, we show how CAVI performs with increasing levels of noise. We observe that the method remains effective in the high-noise regime as well. As

[3] The datasets are publicly available at `https://archive.org/details/stackexchange`, `http://jmcauley.ucsd.edu/data/amazon/`, and `https://www.yelp.com/dataset`, respectively.

Table 1: Summary statistics of the datasets considered in Section 5.2. P_{50} and P_{90} correspond to the 50th and 90th percentiles of the items' degree distribution, respectively.

Dataset	Class type	K	M	N	$\lvert\mathcal{E}\rvert$	P_{50}	P_{90}
Stack Overflow	Programming language	10	644 443	704 982	2 554 436	3	6
Yelp	Location	10	1 962 440	207 974	7 990 277	10	81
Amazon	Product category	5	14 216 570	4 849 549	43 065 188	2	15
Podcasts	Language	43	43 731 473	114 889	121 086 529	81	1099

Table 2: Empirical error rate on three datasets, setting the noise level to $\delta = 0.1$. The best performance is highlighted in bold.

Dataset	Method	All	P_{50}	P_{90}
Stack Overflow	MF+LR	0.067	0.058	0.048
	wvRN	0.185	0.157	0.149
	CAVI	**0.046**	**0.037**	**0.030**
Yelp	MF+LR	0.006	0.002	0.001
	wvRN	0.026	0.012	0.005
	CAVI	**0.005**	**0.001**	**0.001**
Amazon	MF+LR	0.087	0.078	0.061
	wvRN	0.282	0.228	0.162
	CAVI	**0.074**	**0.062**	**0.039**

long as the labels carry even a little bit of information, the network structure can correct a significant fraction of the errors. On the Yelp dataset, for example, CAVI's error rate is 3.15% even when 80% of the labels are corrupted.

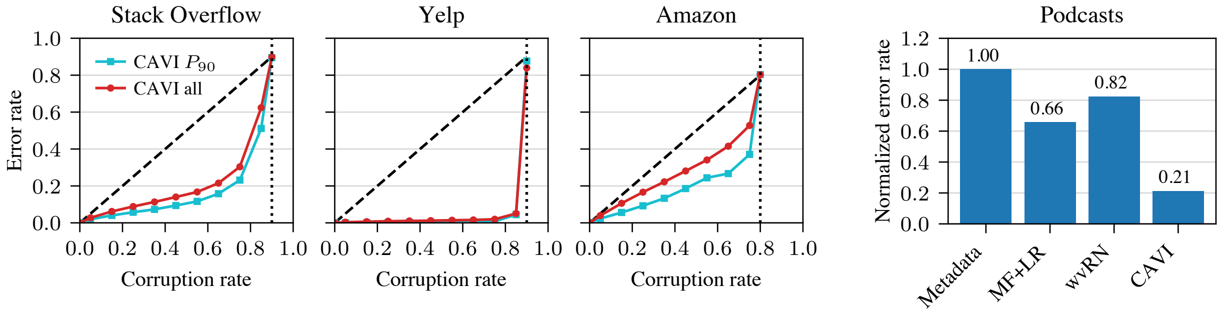
To conclude our analysis of these datasets, we now revisit the labels we have used so far as ground truth. The quality of these labels is unknown, and it is possible that they also include errors. We thus ask the question: Can we use CAVI to find and correct mistakes in these so-called "ground-truth" labels? To this end, we run the algorithm without adding any artificial noise to the labels, and we manually examine the items that are most confidently estimated as mislabeled. Qualitatively, we find that a significant fraction of these appear to be indeed mislabeled. On the Yelp dataset, for example, we look at the top 20 most confident predictions that do not match the provided label. By manually searching for information about each business online, we find that the provided labels of *all* of these businesses are incorrect, and that CAVI's prediction is correct. At the time of writing, these businesses' listings on Yelp.com appear to be fixed.

Podcast Language Identification. Finally, we study a realistic use-case where high-quality ground-truth labels are available. We consider a dataset from Spotify, an audio streaming service providing access to music and podcasts. We seek to classify the language of each podcast. The dataset contains podcasts in 43 different languages, ranging from `ar` (Arabic) to `zh` (Chinese). Further statistics on the dataset are given in Table 1. Podcast producers provide metadata that include language, but that information is not always reliable. Thus, in addition to the metadata, we obtain high-quality ground-truth language labels for 10 000 podcasts through manual annotation by domain experts. These labels enable us to assess *a*) the quality of the metadata provided by producers, and *b*) the accuracy of any predictive model in a consistent and unbiased way. We initialize CAVI and wvRN using the metadata language, and we compare their output to the ground-truth class labels. In Figure 3b, we report the error rate of each method. For confidentiality reasons, we normalize each error rate by that of the metadata. In this application, CAVI achieves a 4.8× reduction in error rate over the metadata provided by podcast producers.

6 CONCLUSION

In this work, we present a probabilistic model for correcting mislabeled items on online platforms by taking advantage of user-item interactions. The model builds on the assumption that users interact with classes in different proportions, and it lends itself to a computationally-efficient approximate Bayesian inference algorithm. We first analyze our model theoretically and characterize its sample complexity under a natural generative model. We have then evaluated its performance empirically and found that it outperforms alternative approaches on multiple real-world applications with difficult network structures.

We have focused on a simple classification setting, but we envision multiple extensions in future work. One such extension would be to tailor our approach to class-conditional noise models, a setting where label noise is *not* uniformly random (Natarajan et al., 2013). For example, in the podcasts dataset, we informally observe that languages seem more likely to be mislabeled to another language of the same language group (e.g., Sundanese to Malay). Another extension consists of further exploiting the probabilistic nature of our approach. Indeed, our model can take advantage of flexible user and item-dependent priors through the parameters $\boldsymbol{\alpha}$



(a) Error rate of CAVI as a function of the noise level. The dashed lines represent the baseline error rate. The vertical dotted lines indicate the corruption rate at which the mutual information between observed and actual labels is zero.

(b) Normalized error rate on a podcast language prediction task.

Figure 3: Empirical performance on real-world datasets.

and $\boldsymbol{\beta}$. Similarly, CAVI's output is a probability distribution, and preliminary results show that this can lead to useful confidence scores. For example, uncertain predictions could be flagged for expert review.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback. We also thank Dmitrii Moor, Federico Tomasi, Zhenwen Dai and Rishabh Mehrotra for constructive discussions while developing the method.

References

C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More.* Hyperion, 2006.

D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988.

S. Bhagat, G. Cormode, and S. Muthukrishnan. Node classification in social networks. In C. C. Aggarwal, editor, *Social Network Data Analytics*, pages 115–148. Springer, 2011.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518): 859–877, 2017.

C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of SIGMOD 1998*, Seattle, WA, USA, June 1998.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning.* MIT Press, 2010.

T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, second edition, 2006.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

P. Domingos and D. Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence.* Morgan & Claypool Publishers, 2009.

European Commission. Online platforms, May 2016. Commission Staff Working Document SWD(2016) 172 final.

N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of IJCAI 1999*, Stockholm, Sweden, Aug. 1999.

B. Frénay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5): 845–869, 2014.

R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.

L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning.* MIT Press, 2007.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1(Oct): 49–75, 2000.

S. Janson. Large deviations for sums of partly dependent random variables. *Random Structures & Algorithms*, 24(3):234–248, 2004.

D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42 (8):30–37, 2009.

Q. Lu and L. Getoor. Link-based classification. In *Proceedings of ICML 2003*, Washington, DC, USA, Aug. 2003.

D. J. MacKay and R. M. Neal. Near Shannon limit performance of low density parity check codes. *Electronics letters*, 33(6):457–458, 1997.

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

E. Manino, L. Tran-Thanh, and N. Jennings. Streaming Bayesian inference for crowdsourced classification. In *Advances in Neural Information Processing Systems 32*, Vancouver, BC, Canada, Dec. 2019.

N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems 26*, Lake Tahoe, CA, USA, Dec. 2013.

J. Neville and D. Jensen. Iterative classification in relational data. In *Proceedings of AAAI 2000 Workshop on Learning Statistical Models from Relational Data*, Austin, TX, USA, Aug. 2000.

J. Neville and D. Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8 (Mar):653–692, 2007.

M. E. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336, 2007.

D. Poole. On the strength of connectedness of a random hypergraph. *The Electronic Journal of Combinatorics*, 22, 2015.

A. Ratner, C. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems 29*, Barcelona, Spain, Dec. 2016.

S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI 2009*, Montreal, QC, Canada, June 2009.

F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

M. Stankova, D. Martens, and F. Provost. Classification over bipartite graphs through projection. Technical report, Faculty of Applied Economics, University of Antwerp, Jan. 2015.

B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proceedings of IJCAI 2001*, Seattle, WA, USA, Aug. 2001.

B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of UAI 2002*, Edmonton, AL, Canada, Aug. 2002.

P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23*, Vancouver, BC, Canada, Dec. 2010.

J. Whitehill, R. Paul, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, Vancouver, BC, Canada, Dec. 2009.

J. Winn and C. M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(Apr):661–694, 2005.

D. Zhou, O. Bousquet, T.-N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 17*, Vancouver, BC, Canada, Dec. 2004.

Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of ICML 2003*, Washington, DC, 2003, Aug. 2003.