Published in final edited form as:

Knowl Inf Syst. 2017 December ; 53(3): 749–765. doi:10.1007/s10115-017-1053-1.

Crowd labeling latent Dirichlet allocation

Luca Pion-Tonachini^{1,2,3,iD}, Scott Makeig², and Ken Kreutz-Delgado^{1,3}

¹Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA 92122, USA

²Swartz Center for Computational Neuroscience, University of California at San Diego, San Diego, CA 92122, USA

³Calit2/QI Pattern Recognition Laboratory, University of California at San Diego, San Diego, CA 92122, USA

Abstract

Large, unlabeled datasets are abundant nowadays, but getting labels for those datasets can be expensive and time-consuming. Crowd labeling is a crowdsourcing approach for gathering such labels from workers whose suggestions are not always accurate. While a variety of algorithms exist for this purpose, we present crowd labeling latent Dirichlet allocation (CL-LDA), a generalization of latent Dirichlet allocation that can solve a more general set of crowd labeling problems. We show that it performs as well as other methods and at times better on a variety of simulated and actual datasets while treating each label as compositional rather than indicating a discrete class. In addition, prior knowledge of workers' abilities can be incorporated into the model through a structured Bayesian framework. We then apply CL-LDA to the EEG independent component labeling dataset, using its generalizations to further explore the utility of the algorithm. We discuss prospects for creating classifiers from the generated labels.

Keywords

Crowd labelling; Generative model; Bayesian; Latent Dirichlet allocation; EEG

1 Introduction

Crowd labeling (CL), also referred to as crowd consensus, is a form of crowdsourcing with the purpose of labeling or categorizing the elements of a provided set of items [19]. Examples of such problems include identifying types of food from pictures and rating the emotion most representative of a sentence. Generating a label for a single example is typically easy and takes anywhere from a second to a minute depending on the task. However, with the large, unlabeled datasets that are so common nowadays, the number of labels needed is often far larger than any person has the time or inclination to produce. In such cases, crowdsourcing can be an effective solution as it greatly reduces the time required

Correspondence to: Luca Pion-Tonachini.

Luca Pion-Tonachini <http://orcid.org/0000-0002-4018-7776>

through sharing and parallelization of labor among volunteers or paid workers. Unfortunately, skill levels within the pool of workers are typically unknown beforehand. When monetary incentives are provided on a per-task basis, there may even be malicious workers who assign labels at random. This results in a collection of labels that are not entirely reliable, meaning that some labels would likely not match the opinion of a domain expert. CL algorithms exist specifically to estimate a label for each question, object, or feature (henceforth called an instance) that is more reliable than the worker inputs from which that label is generated.

The simplest crowdlabelling strategy, selecting the instance category by majority vote, assigns the most commonly submitted label for each instance. In many situations, this method is good enough. Given sufficient votes, low error rates, and lack of consistent bias among workers, the law of large numbers guarantees the average will be reliable. More complex algorithms learn a set of parameters to describe the skill or biases of each worker [19]. In the simplest case, the problem can be recast as learning a weighted average over worker submissions. From this perspective, the majority vote can be described as an averaging method assigning equal weight to all workers. Even more complex models can also learn parameters to describe the difficulty of each instance to account for disagreement between otherwise reliable workers. These algorithms can then be applied to binary classification, multiclass classification, multiple-choice classification (where classes vary for each question or task), or even to more free form paradigms in which workers are allowed to respond with unique, self-generated responses. Though many such algorithms already exist, they largely share the assumptions that each instance pertains to a single class, that responses relate to a single class, and that workers provide at most one response per instance.

After first formally posing the problem of CL, we introduce crowd labeling latent Dirichlet allocation (CL-LDA), an algorithm that generalizes the family of multiclass classification CL algorithms in four important ways:

1. Instance classes are viewed as compositional rather than categorical.
2. Workers may respond with any number of guessed possibilities if they cannot distinguish an obvious correct answer.
3. Responses that do not directly correspond to a class, and may have a different assumed meaning for each worker, are allowed as response options.
4. Prior information on workers can be incorporated in a structured and Bayesian manner.

While latent Dirichlet allocation (LDA) [2] is a well-known algorithm that has been applied to a wide variety of problems, including something approaching crowd labeling [1], as far as we are aware it has not been generalized in a way that renders it applicable to general CL problems. We provide a generalization that allows CL-LDA to be used in all the above applications of CL excluding multiple-choice classification. For simplicity, we restrain our analysis to binary and multiclass datasets. We then provide a comparison to prior CL methods. The notation used in the paper is presented in Table 1. Vector variables are

distinguished by boldface and matrix variables by underlined boldface. An additional subscript on vector variables indicates indexing over the scalar elements of the vector.

2 Problem description

The CL problem begins with the assumption that for each instance in a dataset, there is a true class label. CL algorithms then attempt to estimate the class label for each instance in a dataset. Estimates are made using the responses (henceforth referred to as votes) to those instances provided by a set of workers. A CL algorithm accomplishes this by comparing the votes provided by all workers, considering which workers agree or disagree with other workers and on which instances, thereby determining which label is most probable for each instance. More rigorously, CL requires a set of workers $\mathcal{U}$ indexed $u \in \{1, \dots, U\}$ who consider a set of instances $\mathcal{D}$ indexed $d \in \{1, \dots, D\}$ producing a set of votes $\mathcal{V} = \{v_{di} \in \{1, \dots, R\} \mid d \in \mathcal{D}, i \in \{1, \dots, N_d\}\}$ where U is the number of workers, D is the number of instances, R is the number of possible responses, and N_d is the number of votes on instance d as can be seen in Table 1. For each instance there assumed to exist an unknown, true class vector $\mathbf{y}_d \in \mathcal{Y}$ that relates the instance to C possible distinct classes. The goal of CL, provided this information, is to generate an estimate $\hat{\theta}_d \in \Theta$ as close to $\mathbf{y}_d$ as possible for each instance.

In previous methods, $\mathbf{y}_d$ was assumed to be a discrete value indicating the true class. To accommodate generalization 1 for compositional data, it is assumed that $\mathbf{y}_d \in \mathbb{S}^{C-1}$ where $\mathbb{S}^n$ is the n -dimensional probability simplex in $\mathbb{R}^{n+1}$. This means that $\sum_{l=1}^C y_{dl} = 1$; i.e., the elements of $\mathbf{y}_d$ sum to one. The assumption of previous methods that $\mathbf{y}_d$ is discrete is a special case solution under this generalization, wherein $\mathbf{y}_d$ is a binary indicator vector with only the element related to the true class being equal to one and all others zero.

3 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) is a generative hierarchical Bayesian mixture model for unsupervised data clustering based on unobserved similarities or themes common throughout a dataset [2]. It is most well known for topic modeling in documents [2], but has many other applications such as recommendation systems [2,13], object detection in images [27], and image annotation [14]. In this section we provide a brief review of LDA for context before presenting the generalization to CL-LDA. While the model described here is referred to in the original paper [2] as smoothed-LDA, the smoothed-LDA model is also commonly called LDA as we do here. For simplicity, we describe LDA from the perspective of document topic modeling to maintain a consistent analogy between the intuitions behind LDA and CL-LDA.

LDA (Fig. 1) applied to document topic modeling learns a probabilistic generative model for a corpus, $\mathcal{D}$, comprised of D documents. Each document, d , contains N_d words. The probabilistic generation of these documents and words begins with a Dirichlet prior over topics in the corpus with parameter vector α , from which each document's topic distribution, θ_d , is drawn. For each document, N_d samples are taken from a multinomial distribution with parameter vector θ_d . These samples are word-topics within the document,

denoted as z_{di} , which make explicit the topic of the context in which a word is used. For example, the word “rash” could be a symptom in the context of medical literature, but might also describe a decision in the context of a political commentary. A second Dirichlet prior has parameter vector $\vec{\beta}$ over word distributions given topics from which each topic-dependent word distribution, $\vec{\phi}_k$, is sampled. For each word-topic z_{di} , a word is drawn from a multinomial distribution with parameter vector $\vec{\phi}_{z_{di}}$. Effectively, $\vec{\phi}_k$ parameterizes the vocabulary used by topic k . In summary:

$$\vec{\theta}_d \sim \text{Dirichlet}(\vec{\alpha}) \quad \forall d \in \mathcal{D} \quad (1)$$

$$\vec{\phi}_k \sim \text{Dirichlet}(\vec{\beta}) \quad \forall k \in \{1, \dots, C\} \quad (2)$$

$$z_{di} \sim \text{Multinomial}(\vec{\theta}_d) \quad \forall d \in \mathcal{D}, \quad \forall i \in \{1, \dots, N_d\} \quad (3)$$

$$v_{di} \sim \text{Multinomial}(\vec{\phi}_{z_{di}}) \quad \forall d \in \mathcal{D}, \quad \forall i \in \{1, \dots, N_d\} \quad (4)$$

The only information provided to the model are the words, $\mathcal{V}$, in the corpus and the priors, $\vec{\alpha}$ and $\vec{\beta}$. In the original derivation N_d is treated as a random variable drawn from a Poisson distribution, but as it is independent from the other data generating variables, $\vec{\theta}_d$ and z_{di} it may be treated as deterministic.

4 Crowd labeling latent Dirichlet allocation

Though LDA may not immediately appear applicable to the problem of CL, it can be shown to be analogous given two generalizations (Fig. 2). Where before the data were a corpus of documents containing words, for CL we analyze a set of instances on which workers have voted. There is a clear relation between documents in a corpus and instances from CL in which the words in documents are analogous to votes on those instances. The missing component in the topic modeling paradigm is some complement to the relation between workers and their votes. Such a relationship can be easily added by generalizing the topic-dependent word distribution in classical LDA, $\vec{\phi}_k$, to a class and worker-dependent vote distribution, $\vec{\phi}_k^u$. Likewise, the prior over vote distribution, $\vec{\beta}$, is generalized on a per-worker and per-class basis as $\vec{\beta}_k^u$. This generalized prior can be thought of as a per worker prior on confusion matrices. While the possibility exists for a unique prior assigned to each worker, a more apt model would use the worker-dependent priors to describe known populations of workers. Worker-dependent priors need not be different, as the worker and class-dependent

vote distributions will still be learned from the data. Therefore, that flexibility should only be employed when prior information supports its use.

As a third generalization, vote-classes are given weights, m_{dv} , such that each voting worker has equal weight on an instance, independent of how many votes they submit on that instance. This is an adaptation of term weighting schemes [31] using a different formula for the value of weights and serves to allow multiple responses within a single vote, CL generalization 2 from Sect. 1, without biasing the result in favor of users who do so more often.

4.1 Meaning of vote-classes

When compared to word-topics in LDA, vote-classes in CL-LDA are less intuitive because when a worker submits a vote, that vote often appears obvious in its intention. A distinction can nevertheless be made between equivalent votes, one of which is made explicit through the vote-class latent variables. Suppose there is an instance from a dataset with two possible classes. One worker submits a vote for the first class, and a second worker submits a vote for the second class. Assuming the first worker is estimated as trustworthy, the vote-class matches the vote by that user. If the second worker is estimated as inaccurate, as they often misidentify the first class as being the second, then the vote-class will likely still be for the first class and the instance will be estimated as strongly first class. If, instead, the second worker is estimated to be accurate, the second vote-class will follow the vote as being of the second class and the instance will be estimated to be evenly split. In the case that the second worker votes both classes, each vote has a unique vote-class and so can vary the interpretation of this instance from either fully class one due to a misunderstanding of class two on the workers behalf, a mixture of both classes as the worker correctly identified similarities to both, or fully class two due to a misunderstanding of class one. Vote-classes estimate the best interpretation of votes, as they might differ from the obvious intention due to a misunderstanding of response options or misinterpretation of instances by the worker.

4.2 Inference on CL-LDA

We use collapsed Gibbs sampling [8] to perform inference on the CL-LDA model. For reference, the word-topic probabilities in LDA are calculated as:

$$P\left(z_{di} = k \mid \mathcal{X}^{-di}, \mathcal{V}, \vec{\alpha}, \vec{\beta}\right) \propto \left(n_{jk*}^{-di} + \alpha_k\right) \frac{n_{*kv}^{-di} + \beta_v}{n_{*k*}^{-di} + \beta_*} \quad (5)$$

and the marginalized distribution parameters reconstructed as:

$$\theta_{dk} = \frac{n_{dk*} + \alpha_k}{n_{d**} + \alpha_*} \quad \phi_{kl} = \frac{n_{*kl} + \beta_l}{n_{*k*} + \beta_*} \quad (6)$$

where $\mathcal{L}^{-di}$ are all the word-topics excluding z_{di} and n_{jkl} is the number of times word l appears in document j with topic k . A $*$ indicates a summation over the index it occupies; e.g., n_{jk*} is the number of words in document j that have topic k .

Accounting for workers and weighting votes to equalize the influence of each worker, the instance-class probabilities in CL-LDA are calculated as:

$$P\left(z_{di} = k | \mathcal{L}^{-di}, \mathcal{V}, \vec{\alpha}, \underline{\beta}^u\right) \propto \left(n_{d*k*}^{-di} + \alpha_k\right) \frac{n_{*kv}^{-di} + \beta_{kv}^u}{n_{*k*u}^{-di} + \beta_{k*}^u} \quad (7)$$

and the marginalized distribution parameters reconstructed as:

$$\theta_{dk} = \frac{n_{dk*} + \alpha_k}{n_{d**} + \alpha_*} \quad \phi_{kl}^u = \frac{n_{*klu} + \beta_{kl}^u}{n_{*k*u} + \beta_{k*}^u} \quad (8)$$

Everything is the same as before except that z , w , and n are additionally indexed by workers and β is additionally indexed by workers and classes. Although not evident from the equation, the counts are also changed to summations of weights, m , such that $n_{dz_{di}^v}^{-di} = n_{dz_{di}^v} - m_{du}$ rather than simply subtracting one as before. The result is that computational complexity remains unchanged from LDA.

4.3 Effect of priors in CL-LDA

Just as in the original derivation of LDA, priors can be interpreted as pseudo-votes that act to smooth the solution toward prior beliefs. Selection of adequate parameters for the prior distributions is essential for CL-LDA to correctly infer instance classes. This is especially true for each $\vec{\beta}_k^u$. If $\vec{\beta}_k^u$ is set to be uniform, then the class-dependent distributions that CL-LDA finds will be associated with an unknown class and the resulting solution would have to be analyzed to determine the meaning of each “class,” thereby negating the utility and autonomy of the algorithm. By choosing $\vec{\beta}_k^u$ such that each possible vote is favored by the most associated class, the results are guided to a known distribution of classes. If it is assumed a priori that the workers are highly skilled, and given that each vote has a one-to-one correspondence to a single, unique class, then $\underline{\beta}^u$ is a scaled identity matrix with scaling equivalent to the strength of the assumption of worker competence. Workers are usually not perfect, so it instead makes sense to set $\underline{\beta}^u$ to a positive, linear combination of an identity matrix and a matrix of uniform values. If a possible vote does not have a one-to-one correspondence to a class, then that value can be set however best fits prior assumptions.

The class distribution prior does not require such premeditated structure but is also important. This distribution is entirely analogous to that of LDA and should incorporate any prior knowledge on the distribution of instance classes within the dataset. The smoothing

effect of the class distribution prior leads to a concern when choosing the scaling for $\vec{\alpha}$ as there are typically far fewer responses per instance in CL than there are words in a document. Care should therefore be taken that α_* is significantly less than the typical number of worker responses on an instance of the dataset or else the smoothing effect of the α will overpower most votes.

4.4 Bayesian prior estimation

Imposing a prior can benefit inference by favoring estimates that are closer to solutions that are believed to be more likely, especially when there are minimal data available. Conversely, if there is a large discrepancy between the belief guiding an imposed prior and the true distribution, that prior can be equally detrimental when there are not enough data to overcome its influence. Bayesian prior estimation (BPE) optimizes the prior distribution so as to maximize the data evidence. Following the analysis by Wallach et. al. of estimating non-symmetric Dirichlet priors in LDA [26], an extension is possible to CL-LDA with Bayesian prior estimation (CL-LDA-BPE). While Wallach suggests the use of her derived estimator [25], this is not possible with CL-LDA as Wallach's derivation assumes that counts are integer valued, which is not applicable here as a result of vote weighting. Therefore, CL-LDA-BPE uses Minkas fixed-point iteration [16]. It is adapted to CL-LDA-BPE as:

$$\hat{\alpha}_k = \alpha_k \frac{\sum_{d=1}^D \Psi(n_{dk*} + \alpha_k) - \Psi(\alpha_k)}{\sum_{d=1}^D \Psi(n_{d**} + \alpha_*) - \Psi(\alpha_*)} \quad (9)$$

and

$$\hat{\beta}_{kl}^u = \beta_{kl}^u \frac{\Psi(n_{*klu} + \beta_{kl}^u) - \Psi(\beta_{kl}^u)}{\Psi(n_{*k*} + \beta_{k*}^u) - \Psi(\beta_{k*}^u)} \quad (10)$$

where $\hat{\alpha}_k$ is the updated k th element of $\vec{\alpha}$, $\hat{\beta}_{kl}^u$ is the updated l th element of β_k^u , and $\Psi(\cdot)$ is the Digamma function. BPE is a single example of the extensive literature of LDA modifications that can possibly be adapted to CL-LDA. Other advantageous modifications include parallelizing inference across processors [28], parallelizing inference on a graphical processing unit [32], and adaptation for online inference when receiving streaming data [4,10].

4.5 Similar prior methods

CL-LDA can be interpreted as part of a larger family of CL algorithms based upon the confusion matrix approach of Dawin and Skene (DS) [5]. DS models each worker using a confusion matrix across classes and votes with the assumption that each instance is of a particular discrete class. This approach can be, and has been, extended many times. Many extensions make the model Bayesian by incorporating hierarchical prior distributions over the classes, workers, or both. For example, the independent Bayesian classifier combination

[12] and its variations [17] extend DS by imposing a Dirichlet prior over the class distribution, Dirichlet priors over the rows of the confusion matrices, and exponential priors on the Dirichlet parameters. Latent confusion analysis (LCA) [20] also extends this method using confusions matrices by imposing normalized gamma priors on the confusion matrices as well as assuming that voting patterns are structured and shared throughout populations of workers. LCA can also be framed from the perspective of LDA but is changed substantially to incorporate shared voting patterns and latent variables for instance difficulty. As previously described, preserving more similarities to LDA, as CL-LDA does, provides many additional benefits, such as access to a rich literature and the use of efficient collapsed Gibbs sampling for inference.

5 Experimental evaluation

To quantify the performance of CL-LDA when applied to CL problems, we use SQUARE [21, 22]: a toolbox that applies CL algorithms to publicly available datasets as well as simulated datasets under various conditions ranging from unsupervised to fully supervised. As CL-LDA is an unsupervised method, only the unsupervised methods of SQUARE are utilized.

5.1 Datasets

To compare CL-LDA against other CL methods, we are required to use data that are compatible with those other methods. As a result, datasets that require CL-LDA's added capabilities cannot be used in these comparisons. An exception is made for multiclass datasets, in which case algorithms that only accept binary classes are excluded. The data in this experiment are therefore in the form of workers voting for a single discrete label per instance.

5.1.1 Found data—All non-simulated CL datasets used in these tests are publicly available on the internet. Each varies in response types, overall number of responses, and distributions of responses across workers and instances. Details of these datasets are shown in Table 2. AC2 [11] consists of collected worker ratings on Web sites ranging from child-friendly to pornographic on a four-point scale. In BM [18], workers rate the sentiment of tweets as positive or negative. CSv3B is a series of judgments on whether statements are true or false and is a binarized version of CSv3 [9] in which there was a very rare “skip” response accounting for less than 0.15% of all responses. In HC [3,24], workers rate search results as either not-, somewhat-, or very relevant, while HCB combines the somewhat-relevant and very relevant ratings into a single response. WVSCM [30] has workers discriminate images of genuine smiles from images of forced smiles. All datasets are either binary or multiclass except for AC2 which is ordinal.

5.1.2 Simulated data—To compare the resilience of CL-LDA to spammers as compared to other methods, we generate simulated data. The simulated datasets are again made to conform with the requirements of the other algorithms. Every simulated dataset consists of 16,000 instances which pertain to one of seven classes. The prior probabilities of the i th class are $i/28$. To explore the effects of low-accuracy workers, we model those workers as

random spammers who vote uniformly at random over all classes. Smart spammers are modeled as trying to avoid detection while minimizing effort by always voting with the class having the highest prior probability. Such workers might appear when monetary rewards are offered for every instance completed. Here, smart spammers always vote for the seventh class. Each worker votes on 300 separate instances at random within the dataset. Non-spamming workers are modeled purely as an accuracy so as to not explicitly favor any family of methods over any other. Each dataset contains 16 workers with 98% accuracy and 160 with 70% accuracy. For each type of spammer, five additional datasets are created adding 32–160 spammers in increments of 32. For each condition, ten datasets are generated with different random seeds to provide a measure the stability on each solution.

5.2 Other CL algorithms

Sheshadri et. al. [22] provide a suitable review of each method compared, which is summarized here. The majority vote (MV) takes the most commonly voted class for instance as the correct answer without any regard for which workers produced those votes. ZenCrowd (ZC) [7] generalizes MV by adding a scalar parameter for each worker's ability which can be either positive for helpful workers or negative for adversarial. Dawin and Skene (DS) [5] estimates a confusion matrix for each worker to provide a more detailed model of worker ability. Naive Bayes (NB) [23] also employs a confusion matrix, but with Laplace smoothing. The Generative model of Labels, Abilities, and Difficulties (GLAD) [30] models workers with a scalar parameter and additionally models each instance with a parameter estimating its difficulty. The algorithm for GLAD used in SQUARE is only compatible with datasets containing binary-valued votes. Caltech UCSD Binary Annotation Model (CUBAM) [29] estimates workers with parameters for skill and bias while also estimating instance difficulty. The implementation of CUBAM used here is also only applicable to binary data.

5.3 Implementation details

For all the experiments in this paper, CL-LDA uses four Gibbs sampling chains with a burn-in of 200 samples and then takes the average over vote-classes associated with each instance for the next 300 samples of each chain combined. CL-LDA-BPE has a longer burn-in of 4000 samples to allow the BPE to converge and then averages the next 1000 samples. Equations 9 and 10 are applied until convergence after every 20 complete Gibbs sampling iterations during the burn-in period, beginning after the first 100 samples.

For both methods, β^u is set to $3 \cdot (0.9 \cdot \mathbf{1}^{W \times W} + 0.1 \cdot \mathbf{1}^{W \times W} / W)$, which weakly assumes that workers are proficient. α is set to a uniform vector with total sum of 0.5 which effectively adds a pseudo-vote with weight of 0.5 to each instance. Each vote-class is initialized to the class with the highest probability to produce the relevant vote given the workers prior, i.e., $\arg \max_l \beta_{kl}^u$.

5.4 Results

Both implementations of CL-LDA generally perform well as other CL methods. Each class of algorithms performs better or worse on any given dataset based on the characteristics specific to that dataset and those variations can be seen in the following results.

5.4.1 Found datasets—The performance of both versions of CL-LDA on the SQUARE datasets is shown in Tables 3 and 4. All performance metrics are calculated using the mean across class-specific performance. Accuracies among all algorithms are similar on AC2, BM, and CSv3B. Only on HC, HCB, and WVSCM is there a wider range of accuracies, and for two of those three, CL-LDA is the top performer. For precision, both CL-LDA methods display a wider range in performance. When compared to DS, which is heavily cited and uses a very similar model, CL-LDA and CL-LDA-BPE do better on almost all datasets in both accuracy and precision.

5.4.2 Simulated datasets—The simulated data experiments indicate that CL-LDA performs better than MV under the influence of many spammers, as shown in Fig. 3. CL-LDA-BPE and DS are also affected much less than MV and also perform better than CL-LDA in this case. ZC strangely appears to perform better with the influence of spammers though suffers from high variability with random spammers. These results should be interpreted with some skepticism as DS and CL-LDA-BPE did not perform in such a superior manner on the found CL datasets as compared to CL-LDA. Still, these experiments provide some insight into each algorithm's robustness to poor workers.

5.4.3 Effects of Bayesian prior estimation—CL-LDA-BPE does perform better than CL-LDA in certain cases, but it also does worse in other as demonstrated by some of the non-simulated datasets. In effect, BPE becomes beneficial when the provided priors greatly misalign with the true class distributions. This claim is supported by the increased efficacy of CL-LDA-BPE when applied to the simulated data as the initial priors impose the beliefs that all classes are equally likely and that all workers are somewhat competent. Once an initial estimate has been formed, BPE can relearn priors to support the data. It follows that in cases when some idea of the data distribution and worker capability is already known, CL-LDA is sufficient. Applying BPE to only the class distribution prior or only to the workers, even to just a subset of workers, is not only possible, but very easy as well. A disadvantage to CL-LDA-BPE is that the BPE requires significantly more Gibbs sampling iterations, and therefore more time, to converge than CL-LDA.

6 Application to the independent component labeling dataset

We demonstrate our methodology on the challenging problem of labeling unmixed, independent components (ICs) of multidimensional electroencephalography (EEG) data [15], making full use of all four generalizations listed in Section 1. We apply CL-LDA to crowdsourced label suggestions to provide training labels for a subset of the independent component labeling (ICL) dataset, a collection of millions of EEG ICs which currently has no such labels. Generating labels for a subset of ICL allows the use of semi-supervised learning algorithms on the entire dataset, enabling the creation of an automated EEG IC

classifier to aid neuroscientists in analyzing large collections of datasets, to help and teach those who do not know how to distinguish ICs manually, and for applications which require automation such as certain real-time brain–computer interfaces.

ICs are separated into seven classes based on the estimated source of the EEG component signal. The classes are “brain” for signals originating from a subject’s cerebral cortex, “muscle” for signals generated by muscle activity, “eye” for electrical potentials produced by the retina, “heart” for components that account for the electrical activity from the heart, “line noise” for components following external electrical fields produced by nearby power fixtures or electronics, “channel noise” for artifacts resulting from poor electrode quality or loose electrode contacts, and “other” as an amalgamation of additional rare classes and poorly unmixed or otherwise uninterpretable signals which provide little or no usable information to neuroscientists. Information regarding ICL can be seen in Table 5.

The task of labeling ICs is difficult enough that when asking experts to evaluate the same components, it is not uncommon for them to disagree on the correct labels for a significant number of those components. Because of this difficulty and the occasionally imperfect unmixing that may result from the algorithms used, compositional labels on ICs provide a more informative model when ascribing meaning or origin to an IC. By describing ICs as a composition, as CL-LDA allows with generalization 1, labels can express similarity to multiple classes while maintaining the capacity to define an IC as primarily from a single class. Pursuant to the compositional label model, votes are also not limited to a single class. Instead, using generalization 2, workers may select any number of classes they find to be applicable to a given IC. In cases when a worker feels significant doubt in his or her assessment of an IC, the worker can indicate that uncertainty through an additional “?” response which may be used in addition to uncertain guesses or alone as a way to abstain from voting, as provided for by generalization 3. Finally, there is a subset of workers whom we, a priori, deem to be experts. This information is incorporated into the CL model to counteract any biases in the general population of workers [6]. As “expert” does not mean infallible in this context, their individual votes cannot be treated as ground truth. Information regarding expert skill is instead incorporated into the model using generalization 4, i.e., worker matrix priors $\vec{\beta}_k^u$.

6.1 Implementation details

When applying CL-LDA to ICL, the matrix prior for experts and non-experts is set to $30 \cdot [\mathbf{I}^{7 \times 7} \mathbf{1}^{7 \times 1/7}]$ and $30 \cdot [(0.4 \cdot \mathbf{I}^{W \times W} + 0.6 \cdot \mathbf{1}^{W \times W/W}) \mathbf{1}^{7 \times 1/2}]$, respectively. These priors effectively add 30 pseudo-votes to each possible vote value for a total of 240 pseudovotes per worker which strongly assumes a skill gap between experts and non-experts, but not so strong that those assumptions cannot be overcome by workers who cast many votes. While such a strong prior may appear excessive, it is helpful in overcoming many common biases among non-expert workers for this task. The class prior vector is set to $[0.15 \ 0.1 \ 0.05 \ 0.025 \ 0.025 \ 0.025 \ 0.15]$ for ICs of type “brain,” “muscle,” “eye,” “heart,” “line noise,” “channel noise,” and “other,” respectively. As in Sect. 5.3, CL-LDA is run with a 200 sample burning and an average is taken over the following 300 samples.

6.2 Results

Without ground-truth labels, evaluation is difficult on ICL. The occasional ambiguity between EEG component class types adds to the challenge as well. To overcome these difficulties, we inspect “heart” components as they are very clearly defined while still necessitating a level of skill to identify. “Heart” components are rare, allowing for a full manual analysis of all components with any “heart” votes. For this evaluation, classes are separated by taking the maximum class contribution to each compositional label, solely to simplify manual analysis. The manual classification of “heart” components was done with the help of a cardiologist. Figure 4 shows all components that CL-LDA estimates as primarily “heart” component. Of the 23 instances selected, 16 are actually “heart,” while the other five are not. Of all 91 ICs that have any “heart” votes, 21 are actually “heart.” Combining this information provides an accuracy of 89%. For comparison, MV only accurately labels 12 heart components and achieves an overall accuracy of 82%. More importantly, the results in Fig. 4 are almost linearly separable. As the proportion “heart” component (PHC) decreases and the variance of PHC increases, the less likely a component is to be an actual “heart” component.

6.3 Label variance

Label variances can potentially provide an important benefit to the broader goal of the ICL dataset and generally to other CL datasets as well. CL-LDA separates the estimate into a composition over classes and generates variance measures on fractions of the composition as an analog to confidence. By having an explicit measure of the estimate credibility, heavy voting overlap (which implicitly ensures a level of label confidence) is no longer a requirement and therefore should allow workers to label more unique components, thereby increasing the expected number of low-probability components in the labeled dataset. To fully utilize this approach in CL, the classifier has to incorporate confidence values during training by assuming label heteroscedasticity. An exemplary method that makes this assumption is generalized least squares (GLS) which scales errors according to the noise covariance using Mahalanobis distance. The effect is to diminish the penalty of misclassifying an instance along dimensions with high label variance. Therefore, if a class estimate is the product of a single unreliable worker, the resulting penalty for misclassification will be minuscule and the classifier will not be heavily skewed as a result of the inaccurate label. In fact, CL-LDA can just as easily provide an estimate of the full label covariance which matches the GLS example more closely. This benefit is clearly applicable to the ICL results shown in Fig. 4 as all but one of the incorrectly labeled heart components have high label variance in addition to lower PHC. Figure 4 is indicative of how label variances provides an additional layer of information that can aid the generation of the EEG IC classifier and other CL-dependent applications and comprehensive validation of the claim will be presented in future work.

7 Conclusion

In this paper we have presented a new CL algorithm, CL-LDA: a generalization of the well-known latent Dirichlet allocation. Using SQUARE, we show that CL-LDA performs comparable to or better than many other CL algorithms, depending upon the dataset, while

allowing for four useful generalizations to the CL problem. These generalizations allow CL-LDA to be used with datasets that require or would benefit from compositional labels, multiresponse votes, class-agnostic responses, and structured Bayesian incorporation of prior knowledge regarding worker abilities. Furthermore, CL-LDA provides variance estimates on each class proportion assigned to an instance as a measure of confidence. We discuss the convenience of using a method based upon LDA as it provides access to an extensive literature with which to easily extend CL-LDA, a fact which we exploit by incorporating Bayesian prior learning of all priors in CL-LDA-BPE. We show CL-LDA-BPE to be better in cases when true class distributions and worker abilities vary strongly from uninformed guesses.

We then apply CL-LDA to the ICL dataset which uses all four stated generalizations. EEG components that capture heart signals demonstrate the utility of variance on class proportions to separate poor class labels estimates from those that are more likely to be true. These class label variances can be incorporated into the error function during classifier training making the classifier robust to unreliable labels rather than discarding or suffering from those labels. In future work, such a classifier can be used to further validate the efficacy of CL-LDA by comparing the performance of a classifier trained on labels generating according to the majority vote against that of a classifier trained with labels from CL-LDA.

Acknowledgments

The authors would like to thank Paul D. Pion, DVM, DipACVIM (Cardiology), for his input on which EEG components contain signals originating from heart activity. This work was supported in part by a gift by the Swartz Foundation (Old Field, NY) and by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1144086.

References

1. Agarwal D, Chen B-C. Proceedings of the third ACM international conference on web search and data mining. ACM; 2010. fLDA: matrix factorization through latent Dirichlet allocation; 91–100.
2. Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. J Mach Learn Res. 2003; 3:993–1022.
3. Buckley C, Lease M, Smucker MD. Overview of the TREC 2010 Relevance Feedback Track (Notebook); The nineteenth text retrieval conference (TREC) notebook; 2010.
4. Canini KR, Shi L, Griffiths TL. Online inference of topics with latent Dirichlet allocation. International conference on artificial intelligence and statistics (AISTATS). 2009; 9:65–72.
5. Dawid AP, Skene AM. Maximum likelihood estimation of observer error-rates using the EM algorithm. Appl Stat. 1979; 28:20–28.
6. Della Penna N, Reid MD. Crowd and prejudice: an impossibility theorem for crowd labelling without a gold standard. arXiv preprint arXiv:1204.3511. 2012
7. Demartini G, Difallah DE, Cudré-Mauroux P. Proceedings of the 21st international conference on World Wide Web. ACM; 2012. Zencrowd: leveraging probabilistic reasoning and crowd sourcing techniques for large-scale entity linking; 469–478.
8. Griffiths TL, Steyvers M. Finding scientific topics. Proc Natl Acad Sci. 2004; 101:5228–5235. [PubMed: 14872004]
9. Orr D. [Accessed 01 May 2016] 50,000 Lessons on How to Read: a Relation Extraction Corpus. 2013. <https://research.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html>
10. Hoffman M, Bach FR, Blei DM. Online learning for latent Dirichlet allocation. Advances in neural information processing systems. 2010:856–864.

11. Ipeirotis PG, Provost F, Wang J. Proceedings of the ACM SIGKDD workshop on human computation. ACM; 2010. Quality management on Amazon mechanical turk; 64–67.
12. Kim HC, Ghahramani Z. Bayesian classifier combination; International conference on artificial intelligence and statistics (AISTATS); 2012. 619–627.
13. Krestel R, Fankhauser P, Nejdl W. Proceedings of the third ACM conference on recommender systems. ACM; 2009. Latent Dirichlet allocation for tag recommendation; 61–68.
14. Lienou M, Maître H, Datcu M. Semantic annotation of satellite images using latent Dirichlet allocation. IEEE Geosci Remote Sens Lett. 2010; 7(1):28–32. DOI: 10.1109/LGRS.2009.2023536
15. Makeig S, Bell AJ, Jung TP, Sejnowski TJ, et al. Independent component analysis of electroencephalographic data. Advances in neural information processing systems. 1996:145–151.
16. Minka T. Estimating a Dirichlet distribution. Tech. rep. 2000
17. Moreno PG, Teh YW, Perez-Cruz F, Artés-Rodríguez A. Bayesian nonparametric crowdsourcing. arXiv preprint arXiv:1407.5017. 2014
18. Mozafari B, Sarkar P, Franklin MJ, Jordan MI, Madden S. Active learning for crowd-sourced databases. arXiv preprint arXiv:1209.3686. 2012
19. Muhammadi J, Rabiee HR, Hosseini A. Crowd labeling: a survey. arXiv preprint arXiv:1301.2774. 2013
20. Sato I, Kashima H, Nakagawa H. Latent confusion analysis by normalized gamma construction; Proceedings of the 31st international conference on machine learning (ICML-14); 2014. 1116–1124.
21. Sheshadri A. Master's thesis. The University of Texas at Austin; 2014. A collaborative approach to IR evaluation.
22. Sheshadri A, Lease M. SQUARE: a benchmark for research on computing crowd consensus; First AAAI conference on human computation and crowdsourcing; 2013.
23. Snow R, O'Connor B, Jurafsky D, Ng AY. Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics; 2008. Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks; 254–263.
24. Tang W, Lease M. Semi-supervised consensus labeling for crowdsourcing. ACM SIGIR workshop on crowdsourcing for information retrieval (CIR). 2011:36–41.
25. Wallach HM. PhD thesis. University of Cambridge; 2008. Structured topic models for language.
26. Wallach HM, Mimno DM, McCallum A. Rethinking LDA: why priors matter. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, Culotta A, editorsAdvances in neural information processing systems 22. Curran Associates Inc; 2009. 1973–1981.
27. Wang X, Grimson E. Spatial latent Dirichlet allocation. In: Platt JC, Koller D, Singer Y, Roweis ST, editorsAdvances in neural information processing systems 20. Curran Associates Inc; 2008. 1577–1584.
28. Wang Y, Bai H, Stanton M, Chen WY, Chang EY. PLDA: parallel latent Dirichlet allocation for large-scale applications. In: Goldberg AV, Zhou Y, editorsAlgorithmic aspects in information and management. Springer; Berlin, Heidelberg; 2009. 301–314.
29. Welinder P, Branson S, Belongie S, Perona P. Proceedings of the 23rd International Conference on Neural Information Processing Systems. Curran Associates Inc; USA; 2010. The multidimensional wisdom of crowds; 2424–2432.
30. Whitehill J, fan Wu T, Bergsma J, Movellan JR, Ruvolo PL. Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, Culotta A, editorsAdvances in neural information processing systems. Vol. 22. Curran Associates Inc; Rostrevor; 2009. 2035–2043.
31. Wilson AT, Chew PA. Human language technologies: the 2010 annual conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics; 2010. Term weighting schemes for latent Dirichlet allocation; 465–473.
32. YanXu N, Qi Y. Parallel inference for latent Dirichlet allocation on graphics processing units. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, Culotta A, editorsAdvances in neural information processing systems 22. Curran Associates Inc; 2009. 2134–2142.

Biographies

Luca Pion-Tonachini received the B.S. and M.S. degrees in electrical engineering with a focus intelligent systems from the University of California, San Diego (UCSD), in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree in electrical engineering, also at UCSD. As part of his doctoral program, he works at the Swartz Center for Computation Neuroscience (SCCN) developing tools for EEG processing and analysis using crowd labeling, deep neural networks, and semi-supervised machine learning and is also directly involved in EEG and EMG analysis projects. In addition to his work at the SCCN, he is associated with the new UCSD Calit2/QI Pattern Recognition Laboratory (PRLab).

Scott Makeig completed a bachelor's degree, self in experience, at the University of California, Berkeley, in 1972 and a Ph.D., music psychobiology, from the University of California, San Diego (UCSD), in 1985. He became a psychobiologist at UCSD and then a research psychologist at the Naval Health Research Center, San Diego. In 1999, he moved to the Salk Institute, La Jolla, as a Staff Scientist and then in 2001 to UCSD as a Research Scientist to develop the Swartz Center for Computational Neuroscience (SCCN) that he now directs. His primary research interests are in developing and applying high-resolution 3-D functional EEG imaging, and recording and analysis of concurrent high-density human EEG and behavior, termed mobile brain/body imaging (MoBI). He and colleagues pioneered the use of independent component analysis (ICA) in biomedical signal processing and continue to develop and support the open-source EEGLAB software environment for electrophysiological signal processing (www.sccn.ucsd.edu/eeglab).

Ken Kreutz-Delgado received the Ph.D. degree in engineering systems science from the University of California, San Diego (UCSD), in 1985. From 1985 through 1990 he was a member of the technical staff at the NASA/Caltech Jet Propulsion Laboratory (JPL) where he was a researcher of intelligent robotic systems. Currently he is a professor in the Department of Electrical & Computer Engineering at UCSD. Prof. Kreutz-Delgado has worked in brain-inspired computation, machine intelligence, and robotics in collaboration with JPL, the UCSD Institute for Neural Computation (INC), the UCSD Swartz Center for Computational Neuroscience (SCCN), and at the new UCSD Calit2/QI Pattern Recognition Laboratory (PRLab), for which he is the Founding Director. His current research includes developing statistical signal processing algorithms for sparse source separation and localization of real-time EEG and fMRI, and deep machine learning algorithms instantiated on non-von Neumann processing architectures.

Fig. 1.
Graphical model for LDA

Fig. 2.
Graphical model for CL-LDA

Fig. 3.

Effects of varying numbers and types of spammers on CL-LDA, CL-LDA-BPE, and other CL algorithms. The *black dashed line* is the expected performance for the proficient workers, while the *black dot-dashed line* is the expected performance for the medium performance workers. *Central lines* indicate mean performance over 10 datasets. *Colored area around central lines* indicates the standard deviation of performances over 10 datasets

Fig. 4.

EEG components from the ICL dataset that CL-LDA estimated as mostly capturing signals generated from a subject's heart. The *horizontal axis* shows how strong the estimate is, while the *vertical axis* shows how stable that estimate is. The *blue parabola* represents the maximum estimate variance which would only occur in the case of estimates drawn from multinomial distributions. *Green dots* represent actual "heart" components, while *red dots* are false positives. A clear trend can be seen with false positives having lower strength and higher variance estimates. On the *right*, exemplar EEG components are shown. The cartoon head visualizes the resulting pattern of electrical potentials at the scalp resulting from activity of that EEG component. *Red, green, and blue* represent positive, neutral, and negative polarity, respectively, with recording electrode positions shown as black dots. The graph to the *right* of each head shows a segment of time series activity from that component. The only high strength false positive has time series activity that closely resembles the QRS complex that is highly characteristic of heart activity

Table 1

Notation

Symbol	Meaning
$\mathbf{I}^{N \times N}$	Identity matrix of size N by N
$\mathbf{1}^{N \times M}$	Matrix of all ones of size N by M
D	Number of instances in the dataset
C	Number of classes
R	Number of possible responses
N_d	Number of votes submitted on sample d
$\mathcal{V}$	All votes
U	Number of workers contributing votes
$\mathcal{U}$	All workers contributing votes
$\vec{a}$	Prior class distribution in the dataset
$\vec{\beta}_k^u$	Prior vote distribution given class k and worker u
$\mathbf{B}^u$	Prior vote distribution matrix on all classes for worker u
$\vec{\theta}_d$	Class distribution of instance d
$\vec{\phi}_k^u$	Vote distribution given class k and worker u
v_{di}	Value of vote i in document d
z_{di}	Class of vote i in document d
$\mathcal{Z}^{-di}$	All vote-classes excluding that of vote i in document d
m_{du}	Weight of a vote on sample d by worker u
n_{jklu}	Combined weight of votes with value l by worker u that are assigned class k on instance j
n_{jklu}^{-di}	Same as above excluding the weight of the i th vote on instance d

Table 2

Benchmark data

Dataset name	Classes	Instances	Evaluation labels	Workers	Votes
AC2	4	11,040	333	825	89,948
BM	2	1000	1000	83	5000
CSv3B	2	42,624	550	57	214,665
HC	4	20,232	4459	766	97,164
HCB	2	20,026	3277	762	90,564
WVSCM	3	2134	159	64	19287

Table 3

Unsupervised accuracy on found data

Algorithms	Datasets					
	AC2	BM	CSv3B	HC	HCB	WVSCM
CL-LDA	0.876	0.813	0.960	0.947	0.664	0.757
CL-LDA-BPE	0.885	0.812	0.972	0.938	0.631	0.786
CUBAM	–	0.804	0.942	–	0.646	0.671
DS	0.850	0.812	0.958	0.926	0.643	0.772
GLAD	–	0.811	0.972	–	0.325	0.826
MV	0.867	0.812	0.973	0.884	0.506	0.710
RY	–	0.818	0.972	–	0.493	0.809
ZC	0.833	0.815	0.975	0.684	0.271	0.818

Bold values indicate the highest performance across CL algorithms on each dataset

Table 4

Unsupervised precision on found data

Algorithms	Datasets						
	AC2	BM	CSv3B	HC	HCB	WVSCM	
CL-LDA	0.507	0.636	0.919	0.647	0.722	0.716	
CL-LDA-BPE	0.540	0.578	0.943	0.654	0.758	0.717	
CUBAM	–	0.613	0.882	–	0.736	0.746	
DS	0.491	0.613	0.915	0.664	0.734	0.690	
GLAD	–	0.573	0.943	–	0.773	0.700	
MV	0.481	0.576	0.944	0.595	0.746	0.729	
RY	–	0.586	0.943	–	0.778	0.694	
ZC	0.405	0.530	0.950	0.666	0.771	0.318	

Bold values indicate the highest performance across CL algorithms on each dataset

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 5

Independent component labeling dataset

Classes	ICs	Experts	Votes	Non-experts	Votes
7	4375	3	2596	19	8754