A Probabilistic Approach for Learning with Label Proportions Applied to the US Presidential Election

Tao Sun[1], Dan Sheldon[1,2], Brendan O'Connor[1]
[1]College of Information and Computer Sciences, University of Massachusetts Amherst
[2]Department of Computer Science, Mount Holyoke College
Email: {taosun, sheldon, brenocon}@cs.umass.edu

***Abstract*—*Ecological inference* (EI) is a classical problem from political science to model voting behavior of individuals given only aggregate election results. Flaxman et al. recently formulated EI as machine learning problem using distribution regression, and applied it to analyze US presidential elections. However, distribution regression unnecessarily aggregates individual-level covariates available from census microdata, and ignores known structure of the aggregation mechanism. We instead formulate the problem as *learning with label proportions* (LLP), and develop a new, probabilistic, LLP method to solve it. Our model is the straightforward one where individual votes are latent variables. We use *cardinality potentials* to efficiently perform exact inference over latent variables during learning, and introduce a novel message-passing algorithm to extend cardinality potentials to multivariate probability models for use within multiclass LLP problems. We show experimentally that LLP outperforms distribution regression for predicting individual-level attributes, and that our method is as good as or better than existing state-of-the-art LLP methods.**

I. Introduction

Ecological inference (EI) is the problem of making inferences about individuals from aggregate data [13]. EI originates in political science, where its history is closely intertwined with the specific application of inferring voting behavior of individuals or demographic groups from vote totals for different regions. EI maps onto a growing number of problem settings within machine learning—including distribution regression [7], [27], optimal transport [17], learning with label proportions [14], [18], [29], multiple-instance learning [4], [10], and collective graphical models [23], [24], [26]—where one wishes to perform supervised learning, but supervision is only available at an aggregate level, e.g., as summary statistics for "bags" of instances.

We consider the classical EI problem of analyzing voting behavior, motivated in particular by US presidential elections. Although there has been vigorous historical debate about the inherent limitations of EI [8], [12], [13], [22], work in machine learning makes it clear that, if one is careful to state assumptions and goals clearly, it is indeed possible to learn individual-level models from aggregate data [2], [18], [27], at least asymptotically. One must still be careful to map these results back to the application at hand; for example, a typical result may be that it is possible to consistently estimate the parameters of an individual-level model given enough group-level observations. This would not imply the ability to infer the outcome or behavior of a single individual.

We will focus on the EI voting problem formulated in [6], [7], where a collection of individual-level demographic covariates is available for each geographical region in addition to the region-level voting totals. In the US, individual-level demographic data for different regions is readily available from the Census Bureau [1]. In [6], [7], the EI problem is then formulated as *distribution regression*, where a function is learned to map directly from the distribution of covariates within each region to voting proportions. This is accomplished by creating kernel mean embedding vectors for each region, and learning a standard regression model to map mean embedding vectors to voting proprortions. Theoretical results about distribution regression support the ability of such an approach to correctly learn a model to make region-level predictions for new regions, assuming they are distributed in the same way as the regions used to train the regression model [27].

We argue that EI is more naturally modeled as a learning with label proportions (LLP) problem. Distribution regression treats voting proportions as generic labels associated with the covariate distributions for each region, which ignores a great deal of information about the problem. First, we know the precise aggregation mechanism: there is one vote per individual and these votes are added to get the totals. Second, in solving the problem, distribution regression unnecessarily aggregates the information we *do* know about individuals (the covariates) by constructing a mean embedding. In contrast, LLP acknowledges that the voting proportions come from counting the number of times each label appears in a bag of instances. It is able to use individual covariates and reason relative to the actual aggregation mechanism.

We posit a simple and natural probabilistic latent variable model for EI that places it within an LLP framework. In our model, each individual possesses an observed covariate vector $\boldsymbol{x}_i$ and an unobserved label y_i (the candidate they voted for), and, within each region, the total number of votes for each candidate is obtained by aggregating the y_i values. We then learn a logistic regression model mapping directly from covariates to individual-level labels. Because the individual labels are unobserved at training time, we use expectation maximization (EM) for learning. The key computational challenge is therefore to perform inference over the y_i values given the totals. We show that techniques for graphical models with *counting potentials* [9], [28] solve this problem exactly and efficiently. Furthermore, we develop a novel message-

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.54

passing algorithm to extend counting potentials to multivariate probability models, and thus multiclass classifiction. The result is the the first direct maximum-likelihood approach to LLP based on the "obvious" latent variable model.

To evaluate different EI methods, we design a realistic testbed for designing synthetic problems that mimic the voting prediction problem. We use geographic boundaries and individual-level covariates that match those used in analysis of the US presidential elections. We then design a variety of synthetic tasks where we withhold one covariate and treat this as the variable to be predicted. At training time, only aggregated per-region counts are provided for the withheld variable. Within this framework we control factors such as the number of individuals per region and the number of classes to obtain a variety of realistic EI tasks. For the task of learning models to make individual-level predictions, we show that LLP methods significantly outperform distribution regression, and that our fully probabilistic approach to LLP outperforms other existing state-of-the-art methods. We also assess the ability of different LLP methods as well as distribution regression to predict the voting behavior of different demographic groups in the 2016 US Presidential Election by making predictions using EI and then comparing the results with exit poll data. We find that EI methods do better on qualitative tasks, such as ordering subgroups by their level of support for Hillary Clinton, than they do in predicting precise voting percentages.

II. Related Work

LLP has been applied to many practical applications such as object recognition [14], ice-water classification [16], fraud detection [21], and embryo selection [11].

Early approaches to LLP do not atempt to infer individual labels: the Mean Map approach of [19] directly estimates the sufficient statistics of each bag (i.e., region) by solving a linear system of equations. The sufficient statistics summarize the information from each bag that is relevant for estimating model parameters. The Inverse Calibration method of [21] treats the mean of each bag as a "super-instance" (similar to the kernel *mean embedding* used in the distribution regression approach to EI [7]) and treats label proportions for each bag as target variables within a variant of Support Vector Regression. In contrast, our work explicitly models individual labels and the structural dependency between individual labels and their aggregate class counts.

Several recent LLP approaches reason explicitly about individual labels, but not in a fully probabilistic manner. [25] first clusters training data given label proportions, and classifies new instance using either the closest cluster label, or a new classifier trained from cluster-predicted training data labels. Alter-$\propto$SVM [29] poses a joint large-margin optimization problem over individual labels and model parameters, and solves it using alternating optimization. One step in the alternating optimization imputes individual labels. The Alter-CNN [16] and Alternating Mean Map (AMM) [18] methods also alternate between inferring individual lables and updating model parameters. However, all of these approaches infer "hard" labels for each instance (either 0 or 1). Alter-CNN is formulated probabilistically, but uses "Hard"-EM for learning, which is a heuristic approximate version of EM. In contrast, our method is conventional maximum-likelihood estimation in the straightforward probability model, and we conduct marginal inference instead of MAP inference over missing individual labels.

Several other papers formulate probabilistic models for LLP, but, unlike our method, resort to some form of approximate inference, such as "hard"-EM [16] or MCMC [11], [14]. The authors of [11] also propose an EM approach with exact probability calculations in the E step, but using brute-force algorithms that do not scale beyond very small problems; for larger problems they instead resort to approximate inference. In contrast to all previous work, we apply exact and efficient inference algorithms using counting potentials [9], [28] to directly maximize the likelihood in the natural probability model.

III. Background and Problem Statement

We now formally introduce the ecological inference problem and describe how it fits within an LLP context. Recall that we assume the availability of individual-level covariates and region-level voting totals for each voting region. In this section, we restrict our attention to binary prediction problems, i.e., we assume there are only two voting options (e.g., candidates from the two major US political parties). We will generalize to multiclass problems in Section IV-C.

Within a single voting region, let $\boldsymbol{x}_i$ denote a vector of demographic features for the ith individual, and let $y_i \in \{0, 1\}$ denote that individual's vote, e.g., $y_i = 1$ if the individual votes for the Purple party. Note that we never observe y_i, and we obtain a sample of $\boldsymbol{x}_i$ values from the Public Use Microdata Sample [1]. We also observe z, the total number of votes for the Purple party, and n, the total number of voters in the region.

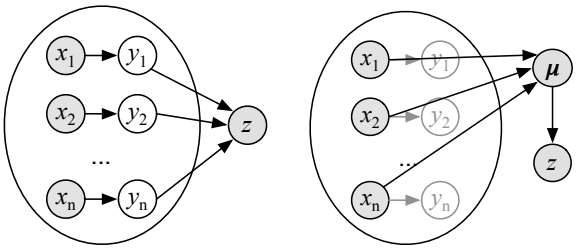
Assume there are B regions in total, and, following LLP terminology, refer to regions as "bags" of instances. The underlying data for bag b is $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{I}_b}$ where $\mathcal{I}_b$ is the index set for bag b. In the training data, instead of observing individual y_i values, we observe $z_b = \sum_{i\in\mathcal{I}_b} y_i$, the number of positive instance in the bag. The overall training data is

$$\big(\{\boldsymbol{x}_i\}_{i\in\mathcal{I}_1}, z_1\big), \ldots, \big(\{\boldsymbol{x}_i\}_{i\in\mathcal{I}_B}, z_B\big),$$

where each example consists of a bag of feature vectors and total number of positive votes for the bag. The goal is to learn a model to do one of two tasks. The first task is individual-level prediction: predict y_i given a new $\boldsymbol{x}_i$. The second task is bag-level prediction: predict z_b given a new bag $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}_b}$ without any labels.

A. Comparison Between LLP and Distribution Regression

The generative model for LLP is illustrated in Figure 1a. The figure shows a single bag, for which the feature vectors $\boldsymbol{x}_i$ and vote total z are observed, but the individual labels y_i are unobserved. The conditional distribution of z is specified by the deterministic relationship $z = \sum_i y_i$. In a probabilistic LLP



(a) LLP (b) Distribution Regression

Fig. 1. LLP and distribution regression models for EI. (a) In LLP, there is a latent variable y_i for each individual, and $z = \sum_i y_i$ is the number of positive instances. (b) In distribution regression, y_i is ignored; $\boldsymbol{\mu}$ is an aggregated summary of the $\boldsymbol{x}_i$'s, and a regression model is learned to map directly from $\boldsymbol{\mu}$ to z.

model, $p(y_i \mid x_i)$ is Bernoulli, and the modeler may choose any regression model for the success probability, such as logistic regression, as we do below, or a CNN [16].

For comparison, Figure 1b illustrates the mean embedding approach to distribution regression [7], [27]. Here, the y_i variables are ignored, as is the known relationship between the y_i variables and z. Instead, the (empirical) distribution of the $\boldsymbol{x}_i$ values in the bag is summarized by a mean embedding into a reproducing kernel Hilbert space. In practice, this means computing the average $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{x}_i)$ of expanded features vectors $\phi(\boldsymbol{x}_i)$ for each individual. Then, a standard regression model is learned to predict z directy from $\boldsymbol{\mu}$. Distribution regression introduces a tradeoff between the ability to preserve information about the individual-level covariates and complexity of the model. A feature expansion corresponding to a characteristic kernel preserves information about the distribution of the covariates, but is necessarily infinite and must be approximated; a very high-dimensional approximation will likely lead to increased variance in the following regression problem. If a simple feature expansion, such as a linear one, is used, it is clear the approach discards significant information about the individual-level covariates by simply computing their mean. LLP avoids this tradeoff by leveraging known structure about the problem.

IV. Our Approach

We now present our approach, which is based on the generative model in Figure 1a and the EM algorithm. We adopt the logistic regression model $p(y_i = 1 \mid \boldsymbol{x}_i; \theta) = \sigma(\boldsymbol{x}_i^T\theta)$ where $\sigma(u) = 1/(1+e^{-u})$. It is straightforward to consider more complex regression models for $p(y_i = 1 \mid \boldsymbol{x}_i; \theta)$ without changing the overall approach. This completes the specification of the probability model

We now turn to learning. A standard approach is to find θ to maximize the conditional likelihood:

$$p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x}; \theta) = \prod_{b=1}^{B} \left(\prod_{i=1}^{n_b} p(y_i \mid \boldsymbol{x}_i; \theta) \right) p(z_b \mid \boldsymbol{y}_b).$$

In this equation, let $\boldsymbol{y}_b = \{y_i\}_{i=1}^{n_b}$ denote the set of labels in bag b, let $\boldsymbol{x}_b = \{\boldsymbol{x}_i\}_{i=1}^{n_b}$ denote the set of feature vectors, and let $\boldsymbol{y}$, $\boldsymbol{z}$, and $\boldsymbol{x}$ denote the concatenation of the $\boldsymbol{y}_b$, z_b, and $\boldsymbol{x}_b$ variables from all bags. Also recall that $p(z_b \mid \boldsymbol{y}_b) = \mathbb{1}[z_b = \sum_{i=1}^{n_b} y_i]$. The obvious challenge to this learning problem is that the $\boldsymbol{y}$ variables are unobserved.

A. EM

EM is a classical approach to address the problem of missing variables within maximum-likelihood estimation [3]. A detailed derivation of EM for our model is given in Appendix A. In the tth iteration, we will select θ to maximize the following function, which is a constant plus a lower bound of the log-likelihood:

$$\begin{aligned} Q_t(\theta) &= \mathbb{E}_{\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x}}[\log p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x}; \theta)] \\ &= \sum_b \mathbb{E}_{\boldsymbol{y}_b|z_b,\boldsymbol{x}_b} \left(\log p(z_b \mid \boldsymbol{y}_b) + \sum_i \log p(y_i \mid \boldsymbol{x}_i; \theta) \right) \\ &= \sum_b \sum_i \mathbb{E}_{y_i|z_b,\boldsymbol{x}_b} \log p(y_i \mid \boldsymbol{x}_i; \theta) + \text{const} \end{aligned} \quad (1)$$

The expectation is taken with respect to the distribution parameterized by θ_t. The term $\log p(z_b \mid \boldsymbol{y}_b)$ is constant with respect to θ and is ignored during the optimization. Specializing to our logistic regression model, the lower bound simplifies to:

$$Q(\theta) = \sum_b \sum_i q_i \log \sigma(\boldsymbol{x}_i^T\theta) + (1-q_i)\log(1-\sigma(\boldsymbol{x}_i^T\theta)) \quad (2)$$

where $q_i := p(y_i = 1 \mid z_b, \boldsymbol{x}_b; \theta_t)$, and we have dropped an additional constant term from Equation (1).

The M step, which requires maximizing $Q(\theta)$ given the q_i values, is straightforward. Equation (2) is the same cross-entropy loss function that appears in standard logistic regression, but the with "soft" labels q_i appearing in place of the standard 0-1 labels. It can be optimized with standard solvers.

The E step, however, is challenging. It requires computing the posterior distribution $p(y_i \mid \boldsymbol{x}_b, z_b; \theta_t)$ over a single y_i value given the observed data $\boldsymbol{x}_b$ and z_b and the current parameters θ_t. This corresponds exactly to inference in the graphical model shown in Figure 1a. Note that all variables are coupled by the hard constraint $\mathbb{1}[z_b = \sum_{i=1}^{n_b} y_i]$, and that this is a factor involving $n_b + 1$ variables, so it is not clear based on standard graphical model principles that efficient inference is possible.

B. Efficient Exact Inference with Cardinality Potentials

Tarlow et al. [28] showed how to perform efficient marginal inference for a set of n binary random variables $y_1, \ldots, y_n$ described by a probability model of the form:

$$q(y_1, \ldots, y_n) \propto \prod_i \psi_i(y_i) \phi\Big(\sum_i y_i\Big), \quad (3)$$

where each variable has a unary potential $\psi_i(y_i)$, and there is a single *cardinality potential* $\phi(\sum_i y_i)$, which couples all of the variables but depends only on the number that take a positive value. Our model fits in this form. Consider the model for a single bag, and dispense with the bag index, so that the variables are $\boldsymbol{x} = \{\boldsymbol{x}_i\}$, $\boldsymbol{y} = \{y_i\}$ and z. Our model for the bag has unary potentials $\psi_i(y_i) = p(y_i \mid \boldsymbol{x}_i; \theta)$ and counting

potential $\phi\big(\sum_i y_i\big) = \mathbb{1}[z = \sum y_i]$. The method of [28] can compute the marginal probability $q(y_i)$ for all i in $O(n \log^2 n)$ time. In our model $q(y_i) = p(y_i \mid \boldsymbol{x}, z; \theta)$ is exactly what we wish to compute during the E step, so this yields an E step that runs efficiently, in $O(n \log^2 n)$ time.

We now give details of the inference approach, but present them in the context of a novel generalization to the case when $y_1, \ldots, y_n \in \{0,1\}^k$ are binary *vectors* of length k. Such a generalization is necessary for the most direct extension of our LLP approach to multiclass classification, in which $p(y_i \mid \boldsymbol{x}_i; \theta)$ is a categorical distribution over three or more alternatives. In Section IV-C, we describe this approach in more detail as well as a different and faster approach to multiclass LLP.

Henceforth, assume that $y_1, \ldots, y_n$ are binary vectors that follow a joint distribution in the same form as Equation (3). To preview the meaning of the multivariate model, the binary vector y_i will be the "one-hot" encoding of the class for individual i, the unary potential is $\psi_i(y_i) = p(y_i \mid \boldsymbol{x}_i; \theta)$,[1] and the counting potential $\phi\big(\sum_i y_i\big) = \mathbb{1}[\sum_i y_i = \boldsymbol{z}]$ will encode the constraint that the total number of instances in each class matches the observed total, where $\boldsymbol{z}$ is now a vector of counts for each class. The description of the multivariate model is symbolically nearly identical to the scalar case.

The key observation of [28] is that it is possible to introduce auxiliary variables that are sums of hierarchically nested subsets of the y_i variables, and arrange the auxiliary variables in a binary tree with $z = \sum_i y_i$ at the root. Then, inference is performed by message-passing in this binary tree.

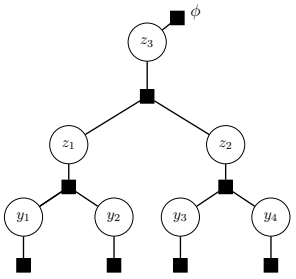


Fig. 2. Illustration auxiliary variables arranged in a binary tree factor graph for inference with a cardinality potential. Each non-leaf node is a deterministic sum of its children. The root node z is equal to $\sum_i y_i$.

Figure 2 illustrates the construction as a factor graph for an example with $n = 4$. The nodes are arranged in a binary tree. Each internal node z_p is connected to two children, which we will denote z_l and z_r (for left and right), by a factor which encodes that z_p is deterministically equal to the sum of z_l and z_p, i.e., $\psi(z_p, z_l, z_r) = \mathbb{1}[z_p = z_l + z_r]$. The unary factors at each leaf are the original unary potentials $\psi_i(y_i)$. The auxiliary nodes and factors enforce that the root node z satisfies $z = \sum_i y_i$. Then, the factor attached to the root node is the cardinality potential $\phi(z) = \mathbb{1}[z = z_{\text{obs}}]$, where z_{obs} is the observed total.

This model is a tree-structured factor graph. Hence, exact inference can be performed by message passing using a leaf-to-root pass followed by a root-to-leaf pass. Although there are only $O(n)$ messages, the support of the variables grows with height in the tree. A node z_l at height i is a sum over 2^i of the y_i values, so it is a vector with entries in $\{0, 1, \ldots, 2^i\}$. We will write this as $z_l \in [m]^k$ where $m = 2^i$ and $[m] = \{0, 1, \ldots, m\}$. Note that m is never more than n.

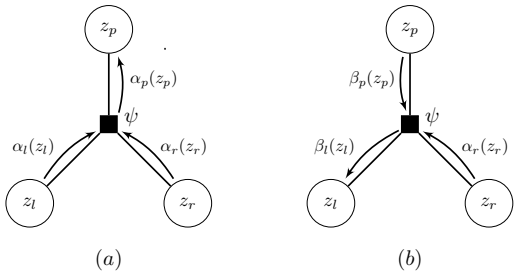


Fig. 3. Illustration of messages from the factor $\psi = \mathbb{1}[z_p = z_l + z_r]$. (a) The upward message $\alpha_p(z_p)$ is computed from $\alpha_l(z_l)$ and $\alpha_r(z_r)$; (b) The downward message $\beta_l(z_l)$ is computed from $\beta_p(z_p)$ and $\alpha_r(z_r)$, similarly for $\beta_r(z_r)$. See text for details.

The message passing scheme is illustrated in Figure 3. For any internal node z_u, let $\alpha_u(z_u)$ denote the incoming factor-to-variable message from the factor immediately below it. Similarly, let $\beta_u(z_u)$ be the incoming factor-to-variable message from the factor immediately above it. Because each internal variable is connected to exactly two factors, the variables will simply "forward" their incoming messages as outgoing messages, and we do not need separate notation for variable-to-factor messages.

The message operation for the upward pass is illustrated in Figure 3(a). The factor ψ connects children z_l and z_r to parent z_p. We assume that $z_l, z_r \in [m]^k$ for some m, and therefore $z_p \in [2m]^k$. The upward message from ψ to z_p is

$$\begin{aligned}\alpha_p(z_p) &= \sum_{z_l \in [m]^k} \sum_{z_r \in [m]^k} \alpha_l(z_l)\alpha_r(z_r)\mathbb{1}[z_p = z_l + z_r] \\ &= \sum_{z_l \in [m]^k} \alpha_l(z_l)\alpha_r(z_p - z_l). \end{aligned} \tag{4}$$

Upward message computation:

$$\alpha_p(z_p) = \sum_{z_l \in [m]^k} \alpha_l(z_l)\alpha_r(z_p - z_l)$$

Downward message computation:

$$\beta_l(z_l) = \sum_{z_p \in [2m]^k} \beta_p(z_p)\alpha_r(z_p - z_l)$$

$$\beta_l(z_r) = \sum_{z_p \in [2m]^k} \beta_p(z_p)\alpha_l(z_p - z_r)$$

Fig. 4. Summary of message passing for cardinality potentials. Each message operation is a convolution; the entire message can be computed in $O(m^k \log m)$ time by the multidimensional FFT. The overall running time to compute all messages is $O(n^k \log^2 n)$.

[1]Note: this factor has 2^k entries indexed by the binary values $y_{i1}, \ldots, y_{ik}$. In this particular model, the binary vector is a one-hot vector, so $\psi_i(y_{i1}, \ldots, y_{ik})$ is nonzero if and only if there is a single nonzero y_{ij}. The inference technique also applies to arbitrary distributions over binary vectors, for which potentials would not have this structure.

Similarly, the downward message to z_l, illustrated in Figure 3(b), has the form

$$\beta_l(z_l) = \sum_{z_p \in [2m]^k} \sum_{z_r \in [m]^k} \beta_p(z_p)\alpha_r(z_r)\mathbb{1}[z_p = z_l + z_r]$$
$$= \sum_{z_p \in [2m]^k} \beta_p(z_p)\alpha_r(z_p - z_l). \quad (5)$$

Note that the upward and downward message operations in Equations (4) and (5) both have the form of a convolution. Specifically, if we let $*$ denote the convolution operation, then $\alpha_p = \alpha_l * \alpha_r$, and $\beta_l = \beta_p * \hat{\alpha}_r$, where $\hat{\alpha}_r(z_{r1}, \ldots, z_{rk}) = \alpha_r(m - z_{r1}, \ldots, m - z_{rk})$ is the factor with the ordering of entries in every dimension reversed. While a direct iterative convolution implementation can compute each message in $O(m^{2k})$ time, a more efficient convolution using multidimensional fast Fourier transform (FFT) takes only $O(m^k \log m)$ time.

The maximum computation time for a single message is $O(n^k \log n)$, for the messages to and from the root. It can be shown that the total amount of time to compute all messages for each level of the tree is $O(n^k \log n)$, so that the overall running time is $O(n^k \log^2 n)$. The upward and downward message-passing operations are summarized in Figure 4.

C. Multiclass Classification

We explore two different methods to extend our LLP approach to multi-class classification: softmax or multinomial regression and one-vs-rest logistic regression. Consider the case when there are $C \geq 3$ classes. It is convenient to assume y_i is encoded using the "one-hot" encoding, i.e., as a binary vector of length C with $y_{ic} = 1$ if and only if the label is c. For each bag, we now observe the vector $z = \sum_i y_i$; the entry z_c is the total number of instances of class c in the bag.

a) Softmax regression: The obvious generalization of our logistic regression model to multiclass classification is multinomial or softmax regression. In this case, $p(y_i \mid \boldsymbol{x}_i; \theta)$ is a categorical distribution with probability vector $\mu_i = \mathbb{E}[y_i]$ obtained through a regression model. The entry μ_{ic} is the probability that y_i encodes class c, and is given by:

$$\gamma_{ic} = \exp(\theta_c^T \boldsymbol{x}_i), \qquad \mu_{ic} = \gamma_{ic} / \Big(\sum_{c'} \gamma_{ic'} \Big).$$

The parameters θ of the model now include a separate parameter vector θ_c for each class c.

Our EM approach generalizes easily to this model. The M step remains a standard softmax regression problem. The E step requires computing the posterior probability vector $q_i = \mathbb{E}[y_i \mid \boldsymbol{x}, z; \theta]$ for every instance in the bag. This is exactly the problem we solved in the previous section for cardinality potentials over binary vectors. Since each y_i, μ_i, and q_i vector sums to one, we may drop one entry prior to performing inference, and complete the E step for a bag with n instances in $O(n^{C-1} \log^2 n)$ time.

This approach is appealing because it is follows a widely used multiclass classification model, which is the natural generalization of logistic regression. However, a drawback is that the running time of the E step grows exponentially with the number of classes, which may be too slow in practice when the numbers of instances or classes grows large.

b) One-vs-Rest Classification: An obvious alternative to softmax regression is one-vs-rest logistic regression, in which a separate logistic regression model is learned for each class c by training on the binary task of whether or not an instance belongs to class c. At test time, the class that predicts the highest probability is selected. This model has been observed to work well in many settings [20]. For our LLP model, it has a significant running-time advantage: each classifier can be trained in the binary prediction setting, so the E step will always run in $O(n \log^2 n)$ time, regardless of the number of classes.

V. Experiments

A. Overview

Our approach is designed for the setting where the learner has access to individual-level covariates, but the outcome variable has been aggregated at the bag level. We apply our model to the problem of inferring how demographic subgroups voted in an election—from voting results and Census demographic data, we would like to, for example, infer what proportion of a particular minority group voted for a particular candidate. In this setting,

- The outcome is aggregated at the bag-level: voting is anonymous, and proportions of how people voted are known only at coarse aggregations by geographic region, from officially released precinct-level vote totals.
- Demographics are individual-level: the U.S. Census releases anonymized "microdata," which consists of covariate vectors of demographic attributes of individuals. It is not known how individuals voted, of course.

Flaxman et al. [6], [7] apply distribution regression for this problem, performing aggregation on microdata demographics as a preprocessing step. In order to test our hypothesis that individual-level modeling can improve these inferences, we conduct a series of experiments:

- (§V-B): Synthetic experiments. We follow [30] and hide a known attribute from the individual-level data, and at training time our model accesses it only as aggregated proportions per region. We evaluate models in their ability to accurately predict the hidden attribute for individuals.
- (§V-C): 2016 presidential elections. Here, we look at the same task as in [6]: trying to infer the individual-level conditional probability $p(\text{vote Dem} \mid f)$ for any arbitrary feature $f(x)$ of an individual's covariates x (e.g., "person is Hispanic/Latino and lives in Illinois"). We train the model with official per-geography voting tallies for the outcome, and a large set of Census demographic covariates for individuals, and perform custom aggregations of the model's inferences to analyze $f(x)$ selections. Quantitative performance is evaluated by comparing to separate survey evidence (exit polls) for the same election.

Individual-level census data (x) is obtained from American Community Survey's Public Use Microdata Sample (PUMS), which covers all of the United States (including D.C. and Puerto Rico).[2] Its millions of records each represents a single person or demographically typical person, along with a survey weight representing how many people are represented by that record, based on the Census' statistical inferences to correct for oversampling, non-response, and to help preserve privacy. We use Flaxman et al.'s open-source preprocessor (used for [6])[3] to select and process the Census data. It merges PUMS data from 2012–2015, resulting in 9,222,638 records, with an average survey weight of 24.2.

PUMS is coded and partitioned geographically by several hundred Public Use Microdata Areas (PUMAs), which are contiguous geographic units with at least 100,000 people each. Since PUMAs do not exactly coincide with counties for which official electoral statistics are reported, the processing scripts merge them with overlapping counties (taking connected components of the bipartite PUMA-county overlap graph), resulting in 979 geographical regions. On average each region contains 9,420 PUMS records, representing on average 228,342 (stdev 357,204) individuals per region, when accounting for survey weights.

Each raw individual record x is comprised of 23 continuous covariates such as income and age, and 97 categorical covariates such as race, education, and spoken language. [7] used FastFood [15] to approximate a kernel map $\phi(x)$, then averaged $\phi(x)$ vectors to produce per-region mean embeddings for distribution regression. Materials posted for later work [6] suggest that linear embeddings may perform similarly as nonlinear, random Fourier-based embeddings. To simplify our current investigation, we only consider linear embeddings in this work.

We use the same preprocessing routine to further process the covariates; it standardizes continuous covariates to z-scores, binarizes categorical covariates, and adds regions' geographical centroids, resulting in 3,881 dimensions for x in the model.

For reference, descriptions of several covariates are shown in Table I, including ones inferred for the synthetic prediction experiments as well as exit poll analysis.[4] In some cases, the number of categories results from coarsening the original Census categories (e.g. SCHL has 25 categories in the original data, but is coarsened to 4 for prediction purposes), using the same preprocessing rules as in previous work. (The 3,881 dimensions for modeling use the original non-coarsened covariates.)

Finally, for computational convenience, we perform two final reductions on the x data before model fitting. First, for most experiments we subsample a fixed number of individuals per region, sampling them with replacement according to the PUMS survey weights. Second, we use PCA to reduce the covariates to 50 dimensions, which preserves approximately 80% of the variance in x.

[2]https://www.census.gov/programs-surveys/acs/data/pums.html

[3]https://github.com/dougalsutherland/pummeler

[4]Details of all variables are available at: https://www2.census.gov/programs-surveys/acs/tech_docs/pums/data_dict/PUMSDataDict15.txt

TABLE I
COVARIATES

Num. Classes	Covariate	Description
2	SEX	Gender
2	DIS	With or without disability
3	WKL	When last worked
4	SCHL	Educational attainment (high school or less, some college/assoc degree, college graduate, postgraduate study)
5	RAC1P	Race (White, Black, Asian, Hispanic/Latino, Other)

B. Synthetic experiments

1) Partial synthetic data: We create partial synthetic data following the same procedure as [30]: we hide a known discrete variable from x and try to predict it from the other variables. At training time, we supply it as supervision only as an aggregated count by region ($z_b = \sum_{i \in \mathcal{I}_b} y_i$). We evaluate models in their ability to predict the hidden variable for individuals in held-out data.

The training data is prepared by sub-sampling either 10 or 100 individual records per region (as described in §V-A); we test both settings since prior literature has occasionally examined performance as a function of bag size. The test set is constructed to include 10,000 records sampled from all regions (by survey weight), from records that that were never selected for the training set.

For certain hidden response variables, some covariates are duplicates or near-duplicates, which makes the prediction problem too easy. We make the problem harder by removing attributes in x that have at least one value with Pearson correlation higher than 0.7 with the response (hidden attribute). For example, if NATIVITY (native or foreign born) was the response variable, it has high absolute correlations to two different values of CIT (citizenship status): binarized values CIT_4 (US citizen by naturalization) and CIT_1 (born in the US) have Pearson correlations of 1 and 0.85, respectively. Furthermore, DECADE_nan (Decade of entry is N/A, meaning born in the US), and WAOB_1 (world area of birth is US state) also have high absolute correlations (both 0.85). Thus all CIT, DECADE, and WAOB attributes are removed. Depending on which hidden attribute is used as response and how many covariates are highly correlated, the number of covariates (out of 3,881) we removed ranges from 0 for HICOV (health insurance available) to 965 for WKL (when last worked).

2) Models: We test a series of logistic regression models, all of which can make predictions for individuals.

- `individual`: an oracle directly trained on the individual labels. This is expected to outperform all other methods, which can only access aggregated counts.
- `mean-emb`: logistic regression trained with mean embedding vectors and label proportions for each

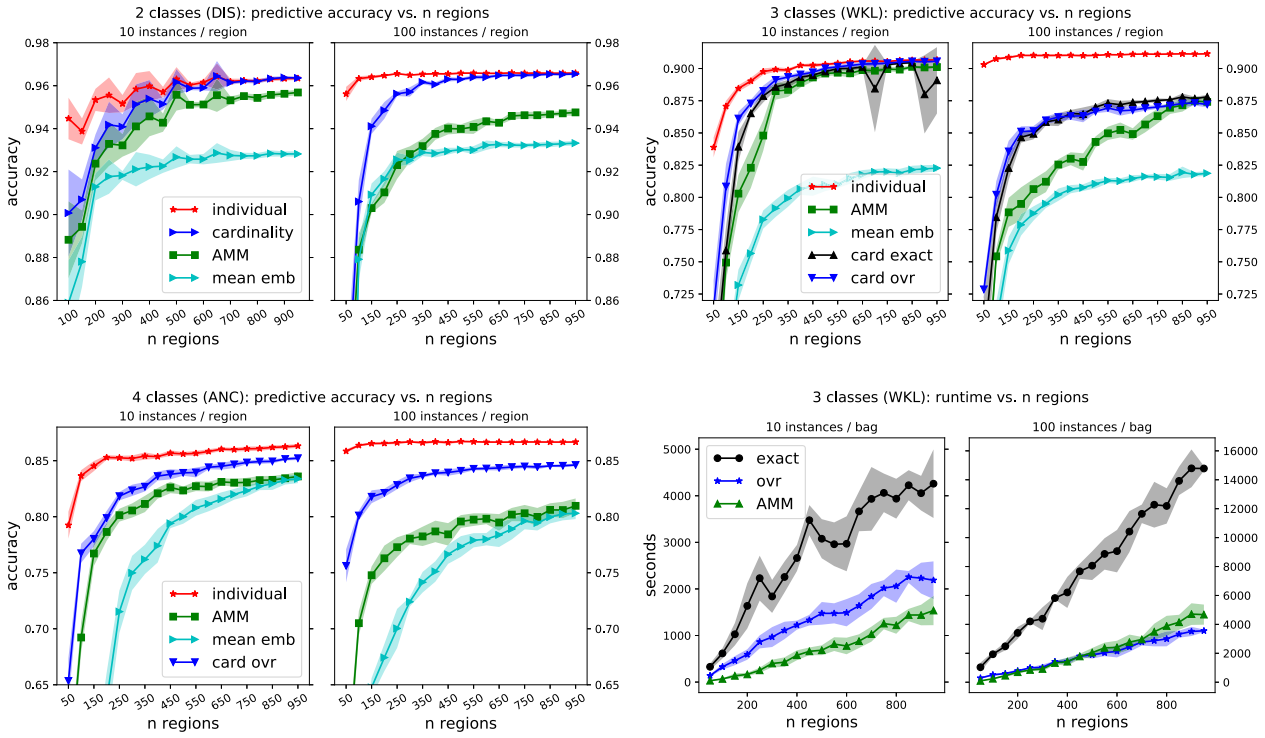


Fig. 5. Predictive accuracies of trained models for 2, 3, or 4 labels classification tasks. The hidden attributes we chosen are Disability (DIS), When last worked (WKL), and Ancestry recode (ANC). We consider a small bag and a larger bag (10 or 100 instances per bag)) for each hidden attribute. Shaded error bars are 95% confidence intervals computed from 10 repeated trials.

region. (Since the sampling already accounts for survey weights, the mean embedding vector for one region is the simple average $\hat{\boldsymbol{\mu}}_b = \frac{1}{n}\sum_{i\in\mathcal{I}_b} \boldsymbol{x}_i$.)

- `AMM`: logistic regression trained on bags of instances and label proportions. For multiclass problems, we use a one-vs-rest (ovr) approach [18].
- `cardinality`: our method, trained on bags of instances and label proportions, for binary labels (§IV-A,IV-B).
- `card-exact`: our method for multiclass problems, with exact inference (§IV-C).
- `card-ovr`: our method for multiclass problems, with an alternative one-vs-rest formulation, using binary cardinality inference.

Following [18], we initialize the LLP methods (AMM, cardinality, card-exact, card-ovr) from `mean-emb`'s learned parameters.

3) Results: Figure 5 shows predictive accuracies of all trained models on the test set, for several hidden attributes (with 2, 3, and 4 categories each), with the mean and standard deviation of performance across 10 different trials. Each trial consists of one sampled training set, shared between all models (all trials use the same test set). Results are broadly similar for other hidden attributes (omitted for space). The results show:

1) LLP outperforms mean embeddings: AMM and the cardinality models substantially improve on `mean-emb`, presumably because they can exploit individual-level covariate information and the structure of the aggregation mechanism.
2) Our cardinality method is the most accurate LLP method: It outperforms AMM, the previous state-of-the-art for LLP with statistically significant differences for most sample sizes of regions and individuals. The cardinality method is a little slower than AMM, though the difference is minimal in the larger bag setting. Asymptotically, the running time for the "E"-steps of `AMM` and `card-ovr` are nearly the same: $O(kn\log n)$ and $O(kn\log^2 n)$, respectively.
3) For multiclass, `card-ovr` is performs similarly as `card-exact`, and is computationally advantageous: `card-ovr` takes $O(kn\log^2 n)$ in runtime and only requires a 1D FFT, versus `card-exact`'s $O(n^{k-1}\log^2 n)$ runtime and additional memory usage for a multidimensional FFT. The exact method also has some precision issues (numerical underflow of downward messages) when running message passing in larger binary trees. Future work could pursue improvements to exact multiclass cardinality inference; in the meantime, we recommend one-vs-rest as an effective practical solution.
4) General observations: as expected, the oracle individual model outperforms all others. Also note that the larger per-bag samples (100) result in a harder problem than smaller

(10) per-bag samples, since the training-time aggregation hides more information when bags are larger.

C. 2016 US Presidential Election Analysis

For real-world experiments, our goal is to infer an individual-level conditional probability $p(\text{vote Dem} \mid f)$ for any arbitrary feature $f(x)$ of an individual's covariates x. We will compare our predictions for such subgroups to an alternative, well-established source of information, exit polls at state and national levels.

1) Experiment: This experiment requires exit polls for validation, and voting data for model training. Exit polls are surveys conducted on a sample of people at polling stations, right after they finish voting; we use the widely reported 2016 exit poll data collected by Edison Research.[5]

Voting data (z) is based on county-level official results,[6] aggregated to the 979 regions. This results in a tuple of vote totals $(v_D, v_R, v_{\text{oth}})$ for each region: how many people voted for Clinton (D), Trump (R), or another candidate. Since the PUMS data includes information on all people—including nonvoters—we add in nonvoters to the third category, resulting in the following count vector for each region:

$$z = (v_D, v_R, S - v_D - v_R)$$

where S is the PUMS estimate of the number of persons in the region (sum of survey weights). This is a somewhat challenging setting for the model, requiring it to learn what type of people vote, as well as their voting preference.

We test the mean embedding model, AMM, and the one-vs-rest cardinality model, using all 3,881 covariates. Unlike the previous section, we give the mean embedding model additional access to *all* instances in the data (mean embeddings are constructed from a weighted average of all PUMS records per region), following previous work. By contrast, for the LLP models we sample 1000 individuals per region. PCA is again applied for all models in the same manner as before, and the LLP models are again initialized with `mean-emb`'s learned parameters.

For evaluation, we prepare a held-out dataset with a 1% subsample from all regions in the 28 exit poll states. After training each model, we make predictions on held-out records and aggregate them to state-level and nation-level statistics, so they can be compared against exit polls. We specifically infer fraction of the two-party vote

$$p(\text{vote D} \mid \text{vote D or R}, f(x)) = \frac{n_{D,f}}{n_{D,f} + n_{R,f}}$$

where $n_{D,f}$ and $n_{R,f}$ are counts of the model's (hard) predictions for individuals in the test set with property $f(x)$: for example, $n_{D,f}$ (and $n_{R,f}$) might be the number of Clinton (and Trump) voters among Hispanics in Illinois. These quantities are calculated from exit polls as well for comparison.

[5]This questionnaire was completed by 24,537 voters at 350 voting places throughout the US on Election Day, from 28 states intended to be representative of the U.S. We use data scraped from the CNN website, available at: https://github.com/Prooffreader/election_2016_data.

[6]We use [6]'s version of the data, scraped from NBC.com the day after the election: https://github.com/flaxter/us2016

2) Results: The scatter plots in Figure 6 show predictions made by different methods vs. the exit poll results. The columns correspond to methods, and the rows correspond to the feature used to define subgroups. Each data point represents the subgroup for one feature value (e.g., males) in one state. There are up to 28 points per feature value; there may be fewer due to missing statistics in some state-level exit polls. For example, only 1% of respondents in Iowa exit polls were Asian, and the Iowa exit polls do not report the voting breakdown for Asians.

The scatter plots show that EI methods are indeed able to make correct inferences, but also make certain mistakes. For most methods and feature values (e.g., `mean-emb`, males), the state-level predictions are strongly linearly correlated with the exit polls—that is, the method correctly orders the states by how much males in the state supported Clinton. However, subgroups are often clustered at different locations away from the 1:1 line, indicating systematic error for that group—this is especially prominent for SCHL, where all methods tend to overestimate support for Clinton among college graduates, and underestimate support among individuals with high school or less education or with some college. In other examples, such as `mean-emb` for ETHNICITY=white, the overall positioning of the subgroup is correct and the state-level predictions are well *correlated* with the exit polls, but the slope is wrong. This suggests that the model has not correctly learned the magnitude of other features that vary by state. Overall, the LLP methods appear qualitatively better (predictions more clustered around the 1:1 line) for SEX and ETHNICITY, while there is no clear "winner" for SCHL.

It is also interesting to aggregate the state-level predictions to national-level predictions. Table II shows national-level predictions as well exit polls for subgroups defined by gender (SEX), race (RAC1P), and educational attainment (SCHL). We see here that the models make mostly correct qualitative comparisons, such as: Which subgroup has a higher-level of support for Clinton? Does the majority of a subgroup support Clinton or Trump? However, the models make notable errors predicting the majority among men and women. Moreover, the models have a difficult time predicting the exact percentages even when the qualitative comparisons are correct.

To quantify these issues further and to gain a better comparison between the methods, Table III evaluates the methods based on national-level predictions according to three different metrics for each feature:

1) *Binary prediction* is the number of subgroups for which the method correctly predicts which candidate receives the majority (e.g., "a majority of males supported Trump", "a majority of women supported Clinton").
2) *AUC* measures the ability of the methods to order subgroups by their level of support for Clinton (e.g., "females showed higher support for Clinton than males"). It is measured by ordering the groups by *predicted* support for Clinton, and then measuring the fraction of pairs of groups that are in the correct order relative to the exit polls; this is related to the area under the ROC curve [5].

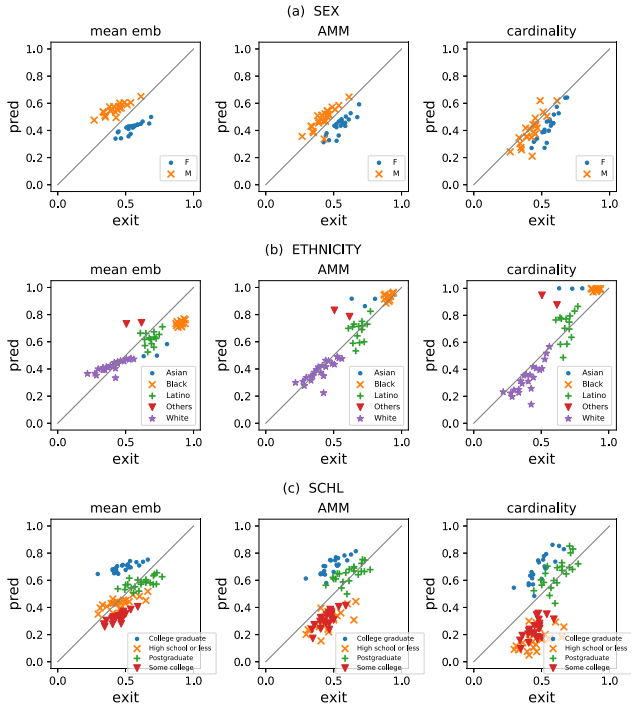


Fig. 6. Model predictions versus exit polls, by demographic group and state. Each color (demographic category) has up to 28 points for its subpopulation per state (for subgroups large enough such that the exit poll results show voting numbers).

3) *Weighted RMSE* measures the numerical accuracy of the predictions. It is the square root of the weighted mean-squared error between the predicted and exit poll percentages, with weights given by the size of each subgroup.

The results show that the models are indeed generally good at the comparison tasks, as shown by the binary prediction and AUC metrics. However, they have considerable error (RMSE more than 5% in all cases) predicting percentages. There is no clear winner among the methods across all metrics. The mean embedding model has the lowest AUC for two out of three variables, and is tied on the third variable.

TABLE II
NATIONAL-LEVEL VOTING PREDICTIONS FOR CLINTON PER DEMOGRAPHIC GROUP

		mean-emb	**AMM**	**card**	**exit**
SEX	M	0.57	0.51	0.41	0.44
	F	0.44	0.47	0.45	0.57
RAC1P	Latino/Hispanic	0.63	0.69	0.77	0.70
	White	0.42	0.38	0.31	0.39
	Black	0.74	0.93	0.99	0.92
	Asian	0.58	0.91	1.00	0.71
	Others	0.77	0.83	0.94	0.61
SCHL	High school or less	0.44	0.32	0.18	0.47
	Some college	0.34	0.34	0.26	0.46
	College graduate	0.71	0.73	0.71	0.53
	Postgraduate	0.60	0.68	0.68	0.61

TABLE III
DEMOGRAPHIC-LEVEL MODEL ACCURACY IN PREDICTING VOTING PROPORTIONS, COMPARED TO EXIT POLLS.

		Binary prediction	AUC	Weighted RMSE
SEX	embed	0/2	0	0.13
	AMM	0/2	0	0.09
	card	1/2	1	0.09
RAC1P	embed	5/5	0.7	0.08
	AMM	5/5	0.9	0.06
	card	5/5	0.8	0.11
SCHL	embed	4/4	0.83	0.12
	AMM	4/4	0.83	0.15
	card	4/4	0.83	0.20

VI. CONCLUSION

In this paper we formulated the ecological inference problem, motivated by analysis of US presidential elections, in the framework of learning with label proportions. Compared with previous approaches, this allows us to use more known structure of the problem, and preserve information in individual-level covariates available to us from Census microdata. We contributed a novel, fully probabilistic, LLP method that outperforms distribution regression and a state-of-the-art LLP method on a range of synthetic tasks. Our probabilistic approach is enabled by adapting message-passing inference algorithms for counting potentials to a natural latent-variable model for LLP. We also applied several methods to analyze the 2016 US presidential election, and found that models frequently make correct comparisons among different choices or groups, but may not predict percentages accurately. Here, no method was a clear winner, but state-level results suggest that LLP methods are in closer agreement with exit polls.

A direction for further exploration is the potential of non-linear methods to improve performance. Previous work used non-linear feature embeddings, and, to a lesser extent, non-linear kernels for classification [7]. In this work we have focused on linear embeddings and linear classifiers. However, our method can support arbitrary non-linear regression models, e.g., neural networks, for the individual-level model to predict y_i from $\boldsymbol{x}_i$. Exploration of such models is a worthwhile avenue for future research.

ACKNOWLEDGMENTS

This material is based on work supported by the National Science Foundation under Grant No. 1522054. Thanks to Seth Flaxman and Dougal Sutherland for feedback.

REFERENCES

[1] "Public Use Microdata Sample (PUMS), 2010-2014 ACS 5-year PUMS and 2015 ACS 1-year PUMS." U.S. Census Bureau; American Community Survey (ACS). Available from https://www.census.gov/programs-surveys/acs/data/pums.html.

[2] G. Bernstein and D. Sheldon, "Consistently estimating markov chains with noisy aggregate data," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 1142–1150.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[4] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1, pp. 31 – 71, 1997.
[5] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
[6] S. Flaxman, D. Sutherland, Y.-X. Wang, and Y. W. Teh, "Understanding the 2016 US presidential election using ecological inference and distribution regression with census microdata," *stat.AP arXiv:1611.03787*, 2016.
[7] S. R. Flaxman, Y.-X. Wang, and A. J. Smola, "Who supported Obama in 2012?: Ecological inference through distribution regression," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 289–298.
[8] D. A. Freedman, S. P. Klein, M. Ostland, and M. Roberts, "On "solutions" to the ecological inference problem," *Journal of the American Statistical Association*, vol. 93, no. 444, pp. 1518–22, 1998.
[9] R. Gupta, A. A. Diwan, and S. Sarawagi, "Efficient inference with cardinality-based clique potentials," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 329–336.
[10] H. Hajimirsadeghi and G. Mori, "Multi-instance classification by max-margin training of cardinality-based markov networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
[11] J. Hernández-González, I. Inza, and J. A. Lozano, "Learning bayesian network classifiers from label proportions," *Pattern Recognition*, vol. 46, no. 12, pp. 3425–3440, 2013.
[12] G. King, "The future of ecological inference research: A reply to freedman et al." *Journal of the American Statistical Association*, vol. 94, pp. 352–355, March 1999.
[13] ——, *A solution to the ecological inference problem: Reconstructing individual behavior from aggregate data*. Princeton University Press, 2013.
[14] H. Kück and N. de Freitas, "Learning about individuals from group statistics," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2005, pp. 332–339.
[15] Q. Le, T. Sarlós, and A. Smola, "Fastfood-approximating kernel expansions in loglinear time," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.
[16] F. Li and G. Taylor, "Alter-CNN: An approach to learning from label proportions with application to ice-water classification," in *NIPS 2015 Workshop on Learning and privacy with incomplete data and weak supervision*, 2015.
[17] B. Muzellec, R. Nock, G. Patrini, and F. Nielsen, "Tsallis regularized optimal transport and ecological inference." in *AAAI*, 2017, pp. 2387–2393.
[18] G. Patrini, R. Nock, T. Caetano, and P. Rivera, "(Almost) no label no cry," in *Advances in Neural Information Processing Systems*, 2014, pp. 190–198.
[19] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le, "Estimating labels from label proportions," *Journal of Machine Learning Research*, vol. 10, no. Oct, pp. 2349–2374, 2009.
[20] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of machine learning research*, vol. 5, no. Jan, pp. 101–141, 2004.
[21] S. Rueping, "SVM classifier estimation from group probabilities," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 911–918.
[22] A. A. Schuessler, "Ecological inference," *Proceedings of the National Academy of Sciences*, vol. 96, no. 19, pp. 10 578–10 581, 1999.
[23] D. Sheldon, T. Sun, A. Kumar, and T. G. Dietterich, "Approximate inference in collective graphical models," in *International Conference on Machine Learning (ICML)*, vol. 28, no. 3, 2013, pp. 1004–1012.
[24] D. R. Sheldon and T. G. Dietterich, "Collective graphical models," in *Advances in Neural Information Processing Systems*, 2011, pp. 1161–1169.
[25] M. Stolpe and K. Morik, "Learning from label proportions by optimizing cluster model selection," *Machine Learning and Knowledge Discovery in Databases*, pp. 349–364, 2011.
[26] T. Sun, D. Sheldon, and A. Kumar, "Message passing for collective graphical models," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 853–861.
[27] Z. Szabó, B. K. Sriperumbudur, B. Póczos, and A. Gretton, "Learning theory for distribution regression," *Journal of Machine Learning Research*, vol. 17, no. 152, pp. 1–40, 2016.
[28] D. Tarlow, K. Swersky, R. S. Zemel, R. P. Adams, and B. J. Frey, "Fast exact inference for recursive cardinality models," in *Proceedings of the Twenty-Eighth Conference Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2012.
[29] F. Yu, D. Liu, S. Kumar, J. Tony, and S.-F. Chang, "$\propto$SVM for learning with label proportions," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 504–512.
[30] F. X. Yu, K. Choromanski, S. Kumar, T. Jebara, and S.-F. Chang, "On learning from label proportions," *arXiv preprint arXiv:1402.5902*, 2014.

APPENDIX

A. EM Derivation

The model is:

$$p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x}; \theta) = \prod_{b=1}^{B} \left(p(z_b \mid \boldsymbol{y}_b) \prod_{i=1}^{n_b} p(y_i \mid \boldsymbol{x}_i; \theta) \right) \quad (6)$$

where $p(z_b \mid \boldsymbol{y}_b) = \mathbb{1}[z_b = \sum_{i=1}^{n_b} y_i]$.

We wish to maximize the marginal log-likelihood of the observed data $\boldsymbol{z}$ conditioned on $\boldsymbol{x}$, i.e., maximize

$$\begin{aligned}\ell(\theta) &= \log p(\boldsymbol{z} \mid \boldsymbol{x}; \theta) = \log \prod_b p(z_b \mid \boldsymbol{x}_b; \theta) \\ &= \sum_b \log \sum_{\boldsymbol{y}_b} p(z_b, \boldsymbol{y}_b \mid \boldsymbol{x}_b; \theta)\end{aligned}$$

We may introduce a variational distribution $\mu_b(\boldsymbol{y}_b)$ and apply Jensen's inequality to obtain a lower bound $Q(\theta)$ for $\ell(\theta)$:

$$\begin{aligned}\ell(\theta) &= \sum_b \log \sum_{\boldsymbol{y}_b} \mu_b(\boldsymbol{y}_b) \frac{p(z_b, \boldsymbol{y}_b \mid \boldsymbol{x}_b; \theta)}{\mu_b(\boldsymbol{y}_b)} \\ &\geq \sum_b \sum_{\boldsymbol{y}_b} \mu_b(\boldsymbol{y}_b) \log \frac{p(z_b, \boldsymbol{y}_b \mid \boldsymbol{x}_b; \theta)}{\mu_b(\boldsymbol{y}_b)} := Q(\theta)\end{aligned}$$

It is well known that, for a particular value θ_t, this lower bound is maximized when $\mu_b(\boldsymbol{y}_b) = \boldsymbol{p}(\boldsymbol{y}_b \mid z_b, \boldsymbol{x}_b; \theta_t)$. After making this substitution and dropping the term $\sum_b \sum_{\boldsymbol{y}_b} -\mu_b(\boldsymbol{y}_b) \log \mu_b(\boldsymbol{y}_b)$,which is constant with respect to θ, we may rewrite $Q(\theta)$ as:

$$\begin{aligned}Q(\theta) &= \sum_b \mathbb{E}_{\boldsymbol{y}_b \mid z_b, \boldsymbol{x}_b} \log p(z_b, \boldsymbol{y}_b \mid \boldsymbol{x}_b; \theta) \\ &= \sum_b \mathbb{E}_{\boldsymbol{y}_b \mid z_b, \boldsymbol{x}_b} \Big[\log p(z_b \mid \boldsymbol{y}_b) + \sum_i \log p(y_i \mid \boldsymbol{x}_i; \theta) \Big] \\ &= \sum_b \sum_i \mathbb{E}_{y_i \mid z_b, \boldsymbol{x}_b} \log p(y_i \mid \boldsymbol{x}_i; \theta) + \text{const}\end{aligned}$$

In these equations, the expectation is with respect to the distribution $p(\boldsymbol{y}_b \mid z_b, \boldsymbol{x}_b; \theta_t)$ parameterized by the fixed value θ_t. In the last line, $\log p(z_b \mid \boldsymbol{y}_b)$ is constant with respect to θ and is ignored. Specializing now to our logistic regression model, we have:

$$\begin{aligned}Q(\theta) &= \sum_b \sum_i \mathbb{E}_{y_i \mid z_b, \boldsymbol{x}_b} \Big[y_i \log \sigma(\boldsymbol{x}_i^T \theta) + (1 - y_i) \log \big(1 - \sigma(\boldsymbol{x}_i^T \theta)\big) \Big] \\ &= \sum_b \sum_i q_i \log \sigma(\boldsymbol{x}_i^T \theta) + (1 - q_i) \log \big(1 - \sigma(\boldsymbol{x}_i^T \theta)\big)\end{aligned}$$

where $q_i := p(y_i = 1 \mid z_b, \boldsymbol{x}_b; \theta_t)$ is the posterior probability of y_i given the observed data, and $Q(\theta)$ remains a cross-entropy loss with the "soft" labels q_i taking the place of y_i.