*ECAI 2020*
*G.D. Giacomo et al. (Eds.)*

*doi:10.3233/FAIA200130*

# Neighborhood-Based Pooling for Population-Level Label Distribution Learning

**Tharindu Cyril Weerasooriya**[1] and **Tong Liu** and **Christopher M. Homan**

**Abstract.** Supervised machine learning often requires human-annotated data. While annotator disagreement is typically interpreted as evidence of noise, population-level label distribution learning (PLDL) treats the collection of annotations for each data item as a sample of the opinions of a population of human annotators, among whom disagreement may be proper and expected, even with no noise present. From this perspective, a typical training set may contain a large number of very small-sized samples, one for each data item, none of which, by itself, is large enough to be considered representative of the underlying population's beliefs about that item. We propose an algorithmic framework and new statistical tests for PLDL that account for sampling size. We apply them to previously proposed methods for sharing labels across similar data items. We also propose new approaches for label sharing, which we call *neighborhood-based pooling*.

## 1 Introduction

In supervised learning, the labels provided by a group of annotators are typically aggregated into a single label, which is regarded as ground truth. The underlying assumption that one label fits all is rarely questioned. However, the process of labeling is often subjective, i.e., based on the personal experiences of the humans who label the data [1], such as when the task is to predict beauty in images or rate movies [9]. Genuine disagreement is also common in seemingly "more objective" tasks, for instance, in assessing the mental state, beliefs, or other hidden states based on observable data [19, 13]. Nevertheless, the negative impacts of AI systems trained on too narrow a segment of a population are increasingly felt [23, 28, 30, 4, 26].

*Label distribution learning* (LDL) seeks to predict, for each data item, a probability distribution over the set of labels [8]. LDL can capture for each data item the diversity of opinions among the human annotators. In contrast, almost all of the prior work in LDL has taken the label distributions found in the training data to be accurate, when in fact annotations obtained from crowdsourcing sites or social media—a common though certainly not exclusive source of label distributions—are most often samples of a larger pool of annotators, who may not themselves be representative of the actual target populations' opinions. Furthermore, outside of a limited number of cases—such as movie ratings, where annotations are abundant and convenient—the sample size of annotations for each data item is far too small (often as small as five annotations per item) to reliably represent the true distributions of the annotator pool's beliefs and opinions.

Liu et al. [19] propose a method for sharing labels among items with similar label distributions, under the assumption that, relative to a specific labeling task, there are only a small number of possible interpretations of any data item, even at the population level. A goal in their work was to address the resource bottleneck of human annotation from a large sample by sharing the existing labels. They introduce the concept of population-level label distribution learning (PLDL), i.e., LDL that explicitly models label distributions as population samples. They explored PLDL by using different methods of clustering in the label space. However, they do not use population models to evaluate their algorithm's performance or select models.

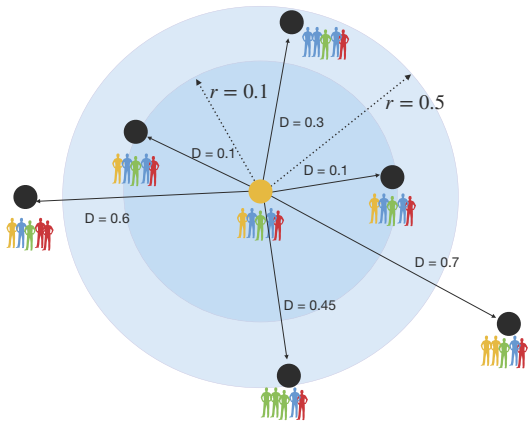


**Figure 1**: The neighborhood-based pooling strategy explored in this paper. The black dots represent data items. The human annotators are represented below each dot and the color of the person represents the label choice. In this example, five humans annotate each data item. $D$ represents the information theoretic divergence in each neighboring data item from the central data item and $r$ is the radius of the neighborhood. We explore the usefulness of refining the labels of the central data item for PLDL by pooling the labels of neighboring items with the central item's labels.

Our main contribution is to introduce a new model selection techniques based on population-level hypothesis tests. These techniques use traditional frequentist statistical approaches to hypothesize that items are similar if their labels can be regarded as samples from a common source. We use these methods to explore, from a sampling perspective, the clustering approach of Liu et al [19]. Additionally, we consider a new, bottom-up approach, called *neighborhood-based pooling* (NBP) (Figure 1), for improving the ground truth estimates of labels for PLDL. Our approach is based on the idea that items with very similar labels may have essentially the same meaning, relative to an annotation task, and thus may be shared, but that the space of label distribution is less clustered and smoother, rather than clustered, as Liu et al. assume.

[1] Rochester Institute of Technology, USA, email: cyriltcw@gmail.com

## 2 Related Work

Disagreement during labeling tasks is a well studied problem [21, 22, 1]. Snow et al. [27] observed using multiple crowdsourced workers that individuals (even experts) had personal biases when labeling. This can contribute to diversity among labels through disagreement, which nondistributional learning cannot account for. Aroyo [2] observed that when the human annotators agree with one another, they perform at a level comparable to experts, and when they disagree, it is often for a good reason. In conventional single label learning applications, researchers consider the majority label as the ground truth, disregarding previously discussed disagreements.

Geng [8] introduced label distribution learning, a learning paradigm where probability distributions are objects to be predicted. LDL takes into account the diversity, disagreement, ambiguity, and uncertainty between annotators for its approaches including, predictions. He and colleagues studied LDL using a variety of different learning algorithms, problem transformation, algorithm adaptation, and a specialized algorithm. The algorithm for LDL shares some similarities with *learning over a probability distribution*, which has a long history of research [25, 11, 6]. Both use probability distributions; LDL interprets them as ground truth while the others use them to model uncertainty. Geng and other researchers studied the applications of LDL in various settings such as predicting population level labels [9, 10, 24] while some do not [7, 17]. These studies acknowledge the need for a large number of labels to train on for improving their distributions. We use the natural scene and facial expressions datasets used by Geng [10] for our experiments (see Section 4).

In contrast to supervised learning, semi-supervised learning (SSL) is a combination of supervised and unsupervised learning [5]. SSL uses both labeled and unlabeled data to improve learning. SSL [14, 32] has a long history with a variety of methods of learning. Clustering is another semi-supervised learning method in which if two items belong to the same cluster, they are believed to share the same label [5].

Liu et al. [19] use a similar approach for PLDL (they also formally introduce the PLDL problem). They explored clustering as an unsupervised learning method to improve the quality of ground truth estimation. They [19] collected a crowdsourced dataset in which five-to-ten human annotators labeled each data item. Then they "[clustered] together semantic similar data items, and then [pooled] together all the labels in each cluster into a single, larger sample" [19], under the assumption that this sample represented the population-level beliefs about each item in the cluster. They compared and contrasted the performance of a variety of clustering methods. Their approach is more typically applied to items with no or unreliable labels, where similarity between data items is necessarily determined in the feature space of the data items, not in the label space. However, Liu et al. [19] observe that, if all the items already have labels, one can determine similarity in the label space alone. Their work has two limitations: (1) though their approach was based on statistical principles, they do not exploit this connection in their analysis and (2) they consider only a relatively small number of clusters per learning problem (under the assumption that for classification problems the number of distinct answer distributions is necessarily limited).

Zhang [31] introduces multi-instance-multi-learning (MIML)-$k$-NN as a non-parametric learning method that used the $k$-nearest neighbor for multiple labels. However, $k$-NN selects the closest $k$ neighbors around the item regardless of how close or further away they are from the data point. The drawback of this approach is not taking into account the similarity between the item and its neighbors.

## 3 Methods

### 3.1 Label Distribution Learning

Wang and Geng [29] developed a theory for label distribution learning, proving a number of theorems about error functions for LDL. Their theory presents label distributions as ground truth objects, without considering that they may be merely estimates of an underlying ground distribution. Here, we present a theory for LDL for the special case of when the label distribution is a sample of an underlying population, which explicitly accounts for this population, along with noise in the sampling process and level of reliability of the annotators. First, we introduce some notation. For any probability distribution $\mathcal{D}$, and any set $\mathcal{X}$ let $X_{\mathcal{D}}$ denote a random variable in $\mathcal{X}$ over $\mathcal{D}$. If context is clear we will drop the "$\mathcal{D}$" subscript. Let $\mathcal{D}|X$ denote the distribution conditioned on $X$, and similarly for $x \in \mathcal{X}$ and $\mathcal{D}|x$. Thus, $Y_{\mathcal{D}|x}$ is a random variable in $\mathcal{Y}$ over the distribution $\mathcal{D}|x$. Finally, for any set $\mathcal{X}$, let $\mathcal{P}_{\mathcal{X}}$ denote the space of all probability distributions over $\mathcal{X}$.

**Algorithm 1:** Sampling procedure.

```
S ← ()
for i ← 1 to n do
    choose x_i ~ X_D
    for j ← 1 to m(x_i) do
        choose a_{i,j} ~ P_D
        choose y_{i,j} ~ Y_D | x_i, a_{i,j}
        add (x_i, a_{i,j}, y_{i,j}) to S
return S
```

Now, to formally define PLDL, let $\mathcal{D}$ be a probability distribution over $\mathcal{X} \times \mathcal{A} \times \mathcal{Y}$, where $\mathcal{X}$ is the *feature space* of a data set of interest, $\mathcal{A}$ is the *agent (or annotator) space*, and $\mathcal{Y}$ is a *label space*. Let $X_{\mathcal{D}}$, $A_{\mathcal{D}}$, and $Y_{\mathcal{D}}$ be random variables representing the marginal distributions of $\mathcal{X}$, $\mathcal{A}$, and $\mathcal{Y}$ respectively. We assume that $X_{\mathcal{D}}$ is independent from $A_{\mathcal{D}}$.

Let $\mathcal{S}$ be a sample from $\mathcal{D}$, drawn according to Figure 1. We can thus consider the indexed set $(x_i)_{p(X_{\mathcal{S}}=x_i)>0}$. and let $\mathcal{H}$ be a *hypothesis space*, where each $h \in \mathcal{H}$ is a function $h : \mathcal{X} \to \mathcal{P}_{\mathcal{Y}}$. The *empirical risk minimization (ERM) problem* for PLDL is to find a hypothesis $h_{\mathcal{S}} \in \mathcal{H}$ that minimizes $L_{\mathcal{S}}$, the *loss function* applied to $\mathcal{S}$ on $h$:

$$h_{\mathcal{S}} \quad \in \quad \arg\min_{h \in \mathcal{H}} L_{\mathcal{S}}(h) \tag{1}$$

(Digression: in the multi-label setting one can take $\mathcal{Y}' = 2^{\mathcal{Y}}$ and treat it as a single label learning problem; however, it is often beneficial to exploit the set structure of the multi-label setting in designing a machine learning solution.) $L_{\mathcal{S}}$ is a function that is small when each $h(x_i)$ is close to $Y_{\mathcal{S}|x_i}$ and zero whenever $h(x_i) = Y_{\mathcal{S}|x_i}$. For the sake of discussion, we will take $L_{\mathcal{S}}$ to be the expected Kullback-Liebler (KL) divergence, $E_{\mathcal{S}}[\mathrm{KL}(h(x_i)||Y_{\mathcal{S}|x_i})]$, where for any two probability distributions $\mathcal{P}$ and $\mathcal{Q}$ with random variables over $\mathcal{Y}$,

$$\mathrm{KL}(P||Q) \quad = \quad \sum_{y \in \mathcal{Y}} P(y) \log \left( \frac{P(y)}{Q(y)} \right) \tag{2}$$

KL divergence is widely used in machine learning, especially in belief modeling, and as a loss function in many settings. Here, it can be roughly interpreted as the expected number of bits per item needed to

correct whenever a sample from $P$ is mistaken for a sample from $Q$, and this seems to capture intuitively the notion of error when comparing two probability distributions.

## 3.2 Estimating Ground Truth via Label Pooling

A common problem that occurs in population-based label distribution is that datasets frequently only have a small number of annotations per data item, i.e., each $x$ occurs at most $m$ times in $\mathcal{S}$, where $m$ is typically too small to estimate the ground truth sample $f(x)$. For simplicity's sake we will assume hereafter that each item occurs exactly $m$ times.

Liu et al. [19] explore the idea that similarly-labeled data items may have similar meanings at the population level and could thus be seen as samples of a common underlying source. We generalize the idea of pooling and extend it to neighborhood-based approaches, but first we need to formally define pooling: A *pooling* is: an integer $p \in \mathbb{N}$, a collection of sets $K_1, \ldots, K_p \subseteq \mathcal{X}$ such that $K_1 \cup \cdots \cup K_p = \mathcal{X}$, and a mapping $k : \mathcal{X} \to \{1, \ldots p\}$. After learning a model $(p, \{K_1, \ldots, K_p\}, k)$ that best fits the data, each data item $x \in \mathcal{X}$ is then associated with the marginal label distribution $Y_{K_{k(x)}}$ (which Liu et al. call the *refined label distribution*) of $x$'s cluster $K_{k(x)}$.

### 3.2.1 Cluster-Based Pooling

Liu et al. [19] introduce pooling methods based on generative hierarchical probabilistic models. Each of these models assumes that the empirical label distributions were generated by choosing a cluster according to a distribution $\pi$ (or in the case of latent Dirichlet allocation (LDA) [3], each data item $x_i$ has its own distribution $\pi_i$) over the pools, and then choosing the labels via a distribution $\phi_j$ associated with the chosen cluster (or in the case of LDA, choosing a new "pool" for each label).

For comparison purposes, we consider four of the models used by Liu et al. [19]: a (finite) multinomial mixture model (**F**) with a Dirichlet prior over $\pi \sim \text{Dir}(p, \gamma = 75)$, where each cluster distribution $\pi_j$ is a multinomial distribution with Dirichlet priors $\text{Dir}(d, \gamma = 0.1)$, a Gaussian mixture model (**G**) without Dirichlet priors, k-means (**K**) and LDA (**L**).

### 3.2.2 Neighborhood-based Pooling

Neighborhood based pooling (NBP) creates for each data item $x \in \mathcal{S}$ one pool $K_x = \{x' \mid D_{KL}(Y_{\mathcal{S}|x} || Y_{\mathcal{S}|x'}) < r\}$, where $r > 0$ is a hyperparameter called the *neighborhood radius* (see Algorithm 2).

We also considered Euclidean distance, Chebyshev distance, and the Canberra metric, however our results using these methods were similar enough to our results with KL divergence (Figure 2). Additionally, KL divergence has a meaningful interpretation in the context of statistical estimation. It is the expected number of bits per item needed to make a multinomial sample from one distribution appear to be from the other.

## 3.3 Hyperparameter Selection for (and Evaluation of) Pooling Models

To evaluate our label pooling methods and select hyperparameters (the number of pools $p$ for the clustering or the neighborhood radius $r$ for the NBP methods), we consider two loss functions: first, we use the mean KL divergence between the empirical and label distributions of each item the evaluation set.

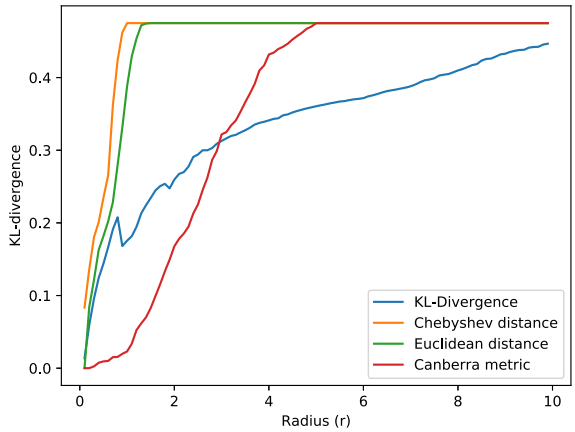


**Figure 2**: The KL-divergence for each $r$ value for the *JQ1* dataset with various distance/divergence measures.

**Algorithm 2:** Neighborhood based pooling (NBP)

***Inputs:***
Empirical label distributions $(Y_{\mathcal{S}|x_i})_{i \in \{1,\ldots,n\}}$
Radius of the neighborhood $r$
Information theoretic measure $D$
***Output:***
A pooling $(p, (K_1, \ldots, K_n), k)$
**Function** *Neighborhood Based Pooling*
  **forall** $i \in \{1, \ldots, n\}$ **do**
    let $K_i = \{x \in \mathcal{X}_{\mathcal{S}} \mid D(Y_{\mathcal{S}|x} || Y_{\mathcal{S}|x_i}) \leq r\}$
    let $k(x_i) = i$
  **end**
  return $(p, (K_1, \ldots, K_n), k)$
**end**

$$\mathcal{L}_S(p, (K_1, \ldots, K_p), k) = \sum_{i \in \{1,\ldots,n\}} D_{KL}(Y_{K_{k(x_i)}} || Y_{\mathcal{S}|x_i})/n \quad (3)$$

Second, we consider the probability of the given label set, given the pools. Additionally, the clustering methods all rely on stochastic optimization. To avoid overfitting, for a range of cluster sizes $p \in \{1, \ldots 40\}$ (based on the range selected by Liu et al. [19]), we ran 100 trials on the training set (since we are using clustering here for sample estimation and not prediction, it is valid and proper to test on the training set) picked the model for each value of $p$ with the median loss. Table 2 shows the number of clusters selected on each label set.

Selecting the best hyperparameter by the raw loss function (Eq 3) may not be the best choice here, in part because it only measures which hyperparameter fits the model best, but also because ground truth is for us unobservable. Sometimes, even the best models are still not adequate. In unsupervised problems such as pooling, there are few settings where widely agreed-upon numerical measures are sufficient to judge the quality of a model [12].

However, in the case of PLDL we have a population of annotators $\mathcal{A}$ to work with. This enables us to use frequentist statistical methods such as hypothesis testing to assess the quality of our pooling models. The basic idea is to generate from each given model random samples that are the same size as our training sample. If the model is a good fit for our training sample, then statistics from the training sample should be in line with those generated directly by the model.

So for each model, we run two simulation-based statistical tests. First we generate 1000 synthetic label sets, each the size of our training set, based on the pooling model we are testing, we then compare

the loss function on our test data to the distribution of loss functions on the synthetic sets. As an additional test, we compare these to another sample of 1000 synthetic label sets based on a bootstrap sample from the actual data. See Algorithms 3–4.

$$L_{\mathcal{S}}(p,(K_1,\ldots,K_p),k) = \log n! + \sum_{i=1}^{n}\sum_{y\in\mathcal{Y}} y^{\#}_{\mathcal{S}|x_i}\log y_{K_{k(x_i)}} - \log y^{\#}_{\mathcal{S}|x_i}!,$$

where $y^{\#}_{\mathcal{S}|x_i}$ denotes $|\{y \in \mathcal{Y} \mid \exists x, a((x,a,y)\in\mathcal{S}\}|$.

We consider a range of reasonable values for $r$ based on our NBP and bootstrap sampler. Table 3 shows the neighborhood sizes ($r$) selected on each of the label sets. The selected $r$ is the elbow point of a piecewise linear regression line (See Figure 6 and Figure 7).

**Algorithm 3:** Model Selection/Evaluation for Pooling

***Inputs:*** A sample of data items and empirical label distributions $\mathcal{S}$
A pooling of the data $(p,(K_1,\ldots,K_p),k)$
A stochastic sample generator $G$ for random label samples
A statistic $L$ defined over labeled data items and poolings of data
The number votes $m$ per item.
The number $b$ of samples used for testing
***Output:*** The fraction of samples $\mathcal{B}$ such that $L_{\mathcal{B}}(p,(K_1,\ldots,K_p),k) > L_{\mathcal{S}}(p,(K_1,\ldots,K_p),k)$
**Function** *Model Evaluation*
  $count \leftarrow 0$
  **for** $i \in \{1,\ldots,b\}$ **do**
    generate $\mathcal{B} \sim G((p,(K_1,\ldots,K_p),k),n,m)$ according to one of the sampling procedures from Algorithm 1
    **if** $L_{\mathcal{B}}(p,(K_1,\ldots,K_p),k) > L_{\mathcal{S}}(p,(K_1,\ldots,K_p),k)$ **then**
      $count \leftarrow count + 1$
    **end**
  **end**
  return $count/n$
**end**

**Algorithm 4:** Sampling approaches

$G =$ cluster sampler
**for** $i \in \{1,\ldots,n\}$ **do**
  choose a cluster $K_j \sim \pi$
  choose $Y_{\mathcal{B}|x_i} \sim \mathcal{M}(Y_{K_j},m)$ where $\mathcal{M}(Y_{K_j},m)$ denotes the multinomial distribution over all size $m$ i.i.d. samples of $Y_{K_j}$
**end**

$G =$ NBP sampler
**for** $i \in \{1,\ldots,n\}$ **do**
  choose $j \in \{1,\ldots n\}$ uniformly at random
  choose $Y_{\mathcal{B}|x_i} \sim \mathcal{M}(Y_{K_j},m)$
**end**

$G =$ Bootstrap sampler
**for** $i \in \{1,\ldots,n\}$ **do**
  choose $j \in \{1,\ldots n\}$ uniformly at random
  choose $Y_{\mathcal{B}|x_i} \sim \mathcal{M}(Y_{\mathcal{S}|x_j},m)$
**end**

# 4 Experiments

## 4.1 Data and Labels

All of the datasets we consider have labels that were produced by humans, and in many cases so was the data, we consulted with our institutional review board (IRB), who determined that the data was both secondary and publicly available, and thus exempt from IRB review. For each dataset, we used a 50/25/25 train/dev/test split. These datasets have been used in prior LDL related research [19, 9].

**Jobs dataset - JQ1, JQ2, and JQ3** We used a set of 2,000 job-related annotated tweets data set collected by Liu et al. [18] for this research. The dataset contains responses from three annotation tasks. Originally five crowdsourced annotators each from Mechanical Turk and FigureEight (10 annotators total) labelled the data. The task for JQ1 was to identify the point of view of the tweet (i.e., 1st person, 2nd person, 3rd person, unclear, or not job related). JQ2 was to capture the employment status of the subject in the tweet (i.e., employed, not in labor force, not employed, unclear, and not job-related). The final task was to identify if there was any mention of a employment transition event in the tweet (i.e., getting hired/job seeking, getting fired, quitting a job, losing job some other way, getting promoted/raised, getting cut in hours, complaining about work, offering support, going to work, coming home from work, none of the above but job related, and not job-related).

**Natural Scenes (NS) dataset**[2] We also use the natural scenes dataset by [8]. This set contains 2,000 images of natural scenes (NS). Each image has a label distribution over nine labels (i.e., plant, sky, cloud, snow, building, desert, mountain, water, and sun) and with labels collected from ten human annotators. These images have 36 features associated with them.

**BU-3DFE Facial Expression (FE) dataset**[2] This dataset contains 2,500 facial expression images, each of these images are associated with a label distribution over 6 label categories (i.e., happiness, sadness, surprise, fear, anger, and disgust). The dataset was collected by Yin et al [16]. For each image, the labels come from 23 human annotators.

## 4.2 Experiments on Label Pooling

In evaluating our pooling results according to the simulation-based methods described in Section 3.3 we noticed that, frequently, the values of the loss function on the synthetic data were either all greater or all less than their corresponding values in the training data (recall that we evaluated the model on the training data because, given its purpose to estimate population statistics from samples, rather than predict individual values on new items, held-out data was not necessary). This made our test defined by Algorithm 3 meaningless. Thus, rather than use it, we instead, for each parameter value $\phi$ to be tested, and corresponding pooling $(p,(K_1,\ldots,K_p),k)$ subtract the loss on the training data $\mathcal{L}_{\mathcal{S}}(p,(K_1,\ldots,K_p),k)$ from the mean loss on the synthetic data $\mathcal{L}_{\mathcal{B}}(p,(K_1,\ldots,K_p),k)$, divided by the standard deviation of the synthetic data loss:

$$\frac{\mu(\mathcal{L}_{\mathcal{B}}(p,(K_1,\ldots,K_p),k)) - \mathcal{L}_{\mathcal{S}}(p,(K_1,\ldots,K_p),k)}{\sigma(\mathcal{L}_{\mathcal{B}}(p,(K_1,\ldots,K_p),k))} \tag{4}$$

and seek the parameter $\phi$ that minimizes this quantity. The optimal number of clusters ($p$) based on the observation for $\phi$ is given in Table 1 and Table 2. For the JQ1, JQ2, and JQ3 datasets, **F** clustering models outperformed other clustering methods based on these results. It was followed by **L** clustering for the same set of labels. As expected, **K** and **G** come last. Figure 3 shows the histograms of the loss functions on the JQ1 dataset for **L** clustering. While the standard difference for the value in Eq. 4, is shown in Figure 4 and Figure 5 for bootstrap and cluster samplers. In contrast to the jobs dataset (JQ1, JQ2, and JQ3), for NS and FE, **G** outperformed other models and it was followed by either **F** (for NS) or **K** (for FE). These results are related to the structure of the label distributions of the dataset.

[2] The datasets are to download on the website http://ldl.herokuapp.com/download

The hyperparameter for NBP is the neighborhood size ($r$). To build the synthetic dataset for NBP, we use the bootstrap and NBP based samplers defined in Algorithms 3–4. Figure 6 and Figure 7 shows the standard difference with linear piecewise fitting for bootstrap and NBP samplers. Table 3 gives the optimum neighborhood sizes ($r$) based on each sampling method. Due to the nature of NBP, we also report $N_{Median}$, the median of all the neighborhood sizes. The $r$ sizes identified using the NBP sampler outperformed the sizes identified using the bootstrap sampler. In FE, both the values identified utilized the entire dataset which was available for pooling, while in contrast other datasets utilized approximately a quarter of the entire set.

**Table 1**: We achieve optimal label aggregation models on each label set with the presented number of clusters ($p$) and KL-divergence for the datasets using the cluster sampler with Multinational distribution as the loss function. *The lowest KL per dataset is highlighted in blue.*

| Model | | JQ1 | JQ2 | JQ3 | NS | FE |
|---|---|---|---|---|---|---|
| **F**MM | $p$ | 29 | 12 | 11 | 36 | 16 |
| | $D_{KL}$ | 0.201 | 0.151 | 0.347 | 0.340 | 0.080 |
| **G**MM | $p$ | 2 | 3 | 5 | 6 | 37 |
| | $D_{KL}$ | 0.700 | 0.639 | 1.416 | 0.215 | 0.008 |
| **K**-Means | $p$ | 5 | 16 | 36 | 6 | 22 |
| | $DKL$ | 0.716 | 0.809 | 1.215 | 0.654 | 0.049 |
| **L**DA | $p$ | 2 | 12 | 7 | 4 | 35 |
| | $D_{KL}$ | 0.428 | 0.201 | 0.587 | 0.443 | 0.064 |

**Table 2**: We achieve optimal label aggregation models on each label set with the presented number of clusters ($p$) and KL-divergence for the datasets using the cluster sampler with KL-divergence as the loss function. *The lowest KL per dataset is highlighted in blue.*

| Model | | JQ1 | JQ2 | JQ3 | NS | FE |
|---|---|---|---|---|---|---|
| **F**MM | $p$ | 14 | 7 | 35 | 7 | 16 |
| | $D_{KL}$ | 0.193 | 0.170 | 0.269 | 0.935 | 0.080 |
| **G**MM | $p$ | 4 | 2 | 6 | 17 | 37 |
| | $D_{KL}$ | 0.819 | 0.777 | 1.225 | 0.285 | 0.008 |
| **K**-Means | $p$ | 10 | 15 | 33 | 12 | 22 |
| | $DKL$ | 0.712 | 0.797 | 1.126 | 0.675 | 0.049 |
| **L**DA | $p$ | 7 | 5 | 5 | 4 | 35 |
| | $D_{KL}$ | 0.243 | 0.237 | 0.625 | 0.602 | 0.064 |

**Table 3**: We achieve optimal neighborhood size for NBP on each label set with the given KL-divergence and the median neighborhood size at each level. *The lowest KL per dataset is highlighted in blue.*

| | JQ1 | JQ2 | JQ3 | NS | FE |
|---|---|---|---|---|---|
| $r_B$ | 5 | 6 | 8 | 4 | 4.5 |
| $D_{KL}$ | 0.358 | 0.358 | 0.704 | 0.568 | 0.080 |
| $N_{Median}$ | 967 | 922 | 863.5 | 548.5 | 1,250 |
| $N_{Maximum}$ | 1,000 | 1,000 | 1,000 | 1,000 | 1,250 |
| $r_{NBP}$ | 3 | 2 | 2 | 2 | 3 |
| $D_{KL}$ | 0.317 | 0.257 | 0.409 | 0.444 | 0.080 |
| $N_{Median}$ | 875 | 645 | 293 | 265 | 1,250 |
| $N_{Maximum}$ | 1,000 | 1,000 | 1,000 | 1,000 | 1,250 |

## 4.3 Supervised Learning Experiments

Following the same experimental process as Liu et al [19] we use the refined labels to train and test supervised learning models. Here, we tested our held-out data on two supervised learning models (CNN on Table 4 and LSTM on Table 5) from Liu et al., with the goal of predicting the pooled label distribution of each data item. Note that the features for the supervised learning models to learn were different among the datasets. The JQ1, JQ2, and JQ3 datasets are text-based [19] and the NS and FE features are the vectorized representations used by Geng [9]. We evaluated the predictions against the refined labels generated from the clustering and NBP. We evaluated them: (1) as a traditional supervised learning problem by measuring the accuracy (using $argmax(Y_{K_{k(x)}})$), (2) as a probability distribution problem using the KL-divergence.

### 4.3.1 *Results from Label Clustering*

Looking at the KL-divergence results, the CNNs in Table 4 outperformed the LSTM models. In all the instances, the labels refined through clustering outperformed the baseline. The labels refined using the **G** models outperformed the other models for the JQ1 and JQ2 datasets while for JQ3 and NS, **K** outperformed the others. For the FE dataset, the labels refined by the **F** model performed better than the other models. These results could be related to the size of the label spaces. JQ3 had the largest label space, while others were within a similar range. Moreover, in JQ3, human annotators were allowed to provide multiple labels per data item, in contrast to others where only one label choice was allowed per item. In the LSTMs, the observations from the CNNs were common, in contrast to JQ1, where **L** outperformed other models.

### 4.3.2 *Results from Label Pooling*

In our experiments, we used the two radii picked using bootstrap and NBP sampling. Looking at the mean KL-divergence, for the CNNs, the NBP outperformed the clustering models. The accuracy results obtained did supplement the results obtained through the KL test in majority of the cases other than FE dataset. Similar to the clustering approaches, the results from CNNs outperformed the results obtained through the LSTMs. These observations will be discussed further in the next section.

## 5 Discussion

Discrepancies in the results obtained from different clustering models correspond to discrepancies in the label distributions in different datasets and in the behaviors of different clustering models. For instance, the label distributions in the jobs dataset (Figure 8) tend to be skewed towards one label choice, while this was not the case for NS (Figure 9) and FE (Figure 10). This skewing in the jobs dataset label distributions may be due to the nature of human annotation with respect to human interpretations of the questions asked about the data.

In contrast to the clustering methods, NBP as a whole showed promising results throughout the experiments. On the NS and FE datasets, it showed a pattern of increasing KL divergence as the radius increased. This behavior seems to depend on how the label distributions are structured. For instance, the skewed label distributions in the jobs dataset results in some items having no neighbors at smaller radii.

The jobs dataset by Liu et al. [19] contains labels obtained from from two crowdsourcing platforms. Manual inspection suggests there exist population-level disparities in the distributions provided by each. This phenomenon could be studied more rigorously by modeling user behaviors and traits to improve PLDL. One approach, for instance, could involve inter-annotator agreement using measures such

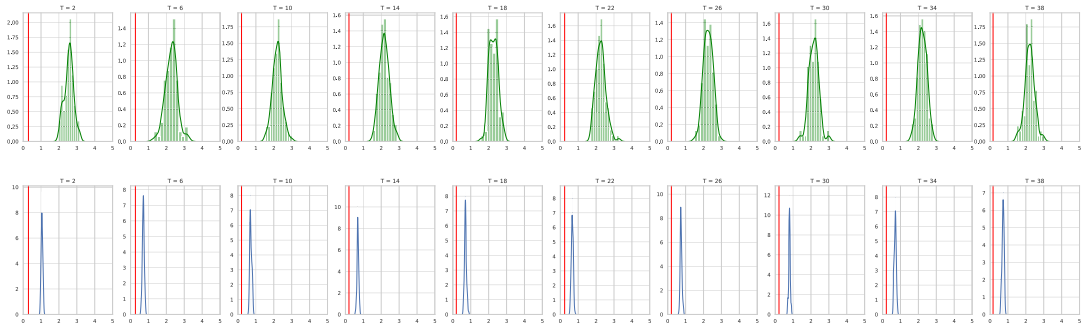

**Figure 3**: Comparison of the $\mathcal{L}_{\mathcal{B}}(p,(K_1,\ldots,K_p),k)$ (synthetic dataset) distributions obtained from the sampling techniques and clustering methods on the *JQ1* dataset for LDA. Reference line: $\mathcal{L}_{\mathcal{S}}(p,(K_1,\ldots,K_p),k)$ value of the training dataset. Top: *Cluster sampler in green.* Bottom: *Bootstrap sampler in blue.*

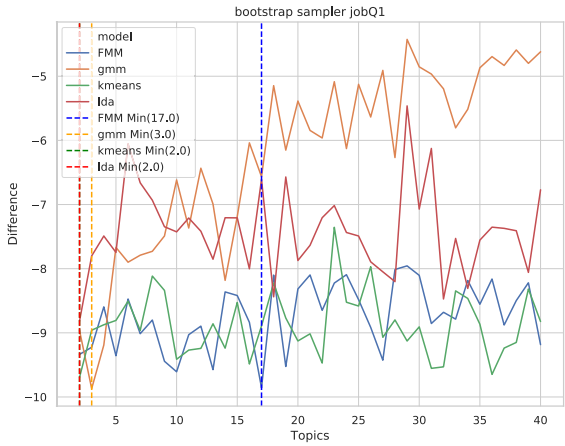


**Figure 4**: The difference between the average KL-divergence of the sample generated by the bootstrap sampler and the predicted data for the JQ1 dataset with the bootstrap sampler.

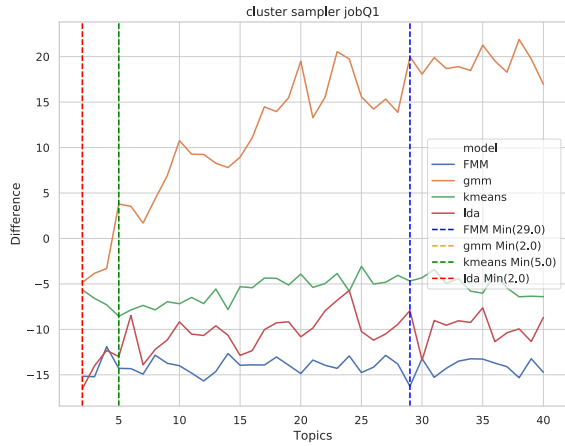


**Figure 5**: The difference between the average KL-divergence of the sample generated by the cluster sampler and the predicted data JQ1 dataset with the cluster sampler.

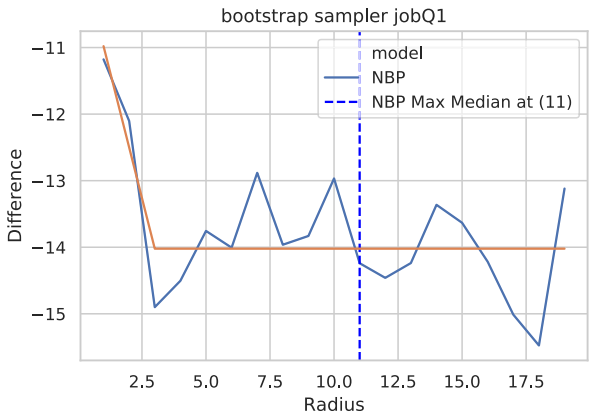


**Figure 6**: The difference between the average KL-divergence of the sample generated by the bootstrap sampler and the predicted data for the JQ1 dataset with the bootstrap sampler for NBP. *We included for reference the piece-wise linear regression line.*

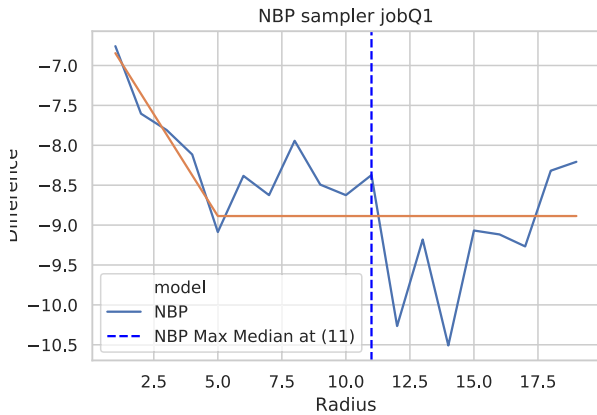


**Figure 7**: The variation of the standard difference between the average KL-divergence of the sample generated by the NBP sampler and the predicted data JQ1 dataset with the NBP sampler. *We included for reference the piecewise linear regression line.*

**Table 4**: The predictions with KL-divergence ($\mathbf{D_{KL}}$) when used with supervised learning (CNN) and unsupervised learning (clustering and NBP). The *lowest* KL and the *highest* accuracy for NBP and clustering is highlighted in blue.

| | KL-divergence | | | | | | | Accuracy | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **Raw** | **Clustering** | | | | **NBP** - KL | | **Raw** | **Clustering** | | | | **NBP** - KL | |
| | **Labels** | **F** | **G** | **K** | **L** | $r_B$ | $r_{NBP}$ | **Labels** | **F** | **G** | **K** | **L** | $r_B$ | $r_{NBP}$ |
| Jobs dataset - Supervised Learning Classification (CNN) | | | | | | | | | | | | | | |
| JQ1 | 1.088 | 0.346 | 0.265 | 0.270 | 0.370 | 0.021 | 0.028 | 0.537 | 0.747 | 0.575 | 0.677 | 0.727 | 1.000 | 0.974 |
| JQ2 | 1.072 | 0.483 | 0.148 | 0.306 | 0.738 | 0.064 | 0.032 | 0.477 | 0.720 | 1.000 | 0.652 | 0.648 | 0.916 | 1.000 |
| JQ3 | 1.440 | 0.772 | 0.366 | 0.341 | 1.033 | 0.206 | 0.117 | 0.271 | 0.313 | 0.467 | 0.553 | 0.330 | 0.528 | 0.600 |
| Natural Scenes dataset - Supervised Learning Classification (CNN) | | | | | | | | | | | | | | |
| NS | 0.985 | 0.589 | 0.469 | 0.235 | 0.551 | 0.117 | 0.192 | 0.282 | 0.816 | 0.186 | 0.340 | 0.828 | 0.529 | 0.044 |
| Facial Expressions dataset - Supervised Learning Classification (CNN) | | | | | | | | | | | | | | |
| FE | 0.081 | 0.0009 | 0.071 | 0.081 | 0.081 | 2.927 | 1.045 | 0.259 | 0.353 | 0.315 | 0.259 | 0.227 | 1.000 | 1.000 |

**Table 5**: The predictions with KL-divergence ($\mathbf{D_{KL}}$) when used with supervised learning (LSTM) and unsupervised learning (clustering and NBP). The *lowest* KL and the *highest* accuracy for NBP and clustering is highlighted in blue.

| | KL-divergence | | | | | | | Accuracy | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **Raw** | **Clustering** | | | | **NBP** - KL | | **Raw** | **Clustering** | | | | **NBP** - KL | |
| | **Labels** | **F** | **G** | **K** | **L** | $r_B$ | $r_{NBP}$ | **Labels** | **F** | **G** | **K** | **L** | $r_B$ | $r_{NBP}$ |
| Jobs dataset - Supervised Learning Classification (LSTM) | | | | | | | | | | | | | | |
| JQ1 | 0.907 | 0.874 | 0.918 | 0.788 | 0.770 | 0.550 | 0.533 | 0.865 | 0.874 | 0.794 | 0.861 | 0.844 | 0.987 | 1.000 |
| JQ2 | 0.989 | 1.038 | 0.624 | 0.746 | 1.084 | 0.659 | 0.669 | 0.841 | 0.861 | 1.000 | 0.855 | 0.832 | 0.964 | 1.000 |
| JQ3 | 1.567 | 1.358 | 1.003 | 0.910 | 1.456 | 0.789 | 0.748 | 0.643 | 0.621 | 0.642 | 0.718 | 0.612 | 0.782 | 0.821 |

as Cohen's Kappa [15] to either resolve conflicts during annotation or use them for refining the distribution estimates. However, a challenge would be to acquire a dataset that contains annotator information, as they are generally removed in publicly available datasets.

Table 3 raises questions about hyperparameter selection for NBP. As our main goal is to share labels between neighbors, looking at the median number of neighbors per each data item ($N_{Median}$) presents some insights. All the datasets had a $N_{Median}$ of at least a quarter of the entire dataset except for FE. One reason could be that FE is, among the datasets we consider, the one that has the largest population of annotators.

While single label learning has a number of established performance measures, such measures are not so well-established in label distribution learning. Geng [9] analyzed 41 different measures and identified five (KL-divergence, Chebyshev, Clark, Canberra, Cosine similarity, and Intersection) as most effective. In our work, we used KL-divergence (one of the five measures identified). KL-divergence is used to measure information loss specifically for probability distributions. The use of our sampling techniques to evaluate the models and hyperparameter selection contributes to the establishment of standard procedures for LDL evaluation.

## 6 Conclusion

Gathering labeled data is an evident resource bottleneck for population-based label distribution learning (PLDL). We introduced and studied new methods for refining the label distributions of data items for PLDL based on the labels of their neighbors. Neighborhood-based pooling (NBP) is semi-supervised learning approach that uses information theoretic measures to pool similar label distributions. We compared NBP as a pooling technique for supervised learning to clustering methods introduced in prior research. We also introduced new methods based on population hypothesis testing for selecting models for label refinement. Our results show that NBP is a feasible approach for refining label distribution estimates.

## 7 Acknowledgments

We thank the anonymous reviewers for their helpful feedback and suggestions. Many thanks to Ifeoma Nwogu, Cecilia O. Alm, and Victoria Maung for their contributions and conversations.

## REFERENCES

[1] Cecilia Ovesdotter Alm, 'Subjective Natural Language Problems: Motivations, Applications, Characterizations, and Implications', in *Proceedings of the 49th Annual Meeting of the ACL : Human Language Technologies*, pp. 107–112, (jun 2011).

[2] Lora Aroyo and Chris Welty, 'The Three Sides of CrowdTruth', in *Journal of Human Computation*, volume 1, pp. 31–34, (2014).

[3] David M Blei, Andrew Y Ng, and Michael I Jordan, 'Latent dirichlet allocation', *Journal of machine Learning research*, (2003).

[4] Joy Buolamwini and Timnit Gebru, 'Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification', in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, (2018).

[5] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien, *Semi-Supervised Learning*, The MIT Press, 1st edn., 2010.

[6] A. P. Dawid and A. M. Skene, 'Maximum likelihood estimation of observer error-rates using the em algorithm', **28**(1), 20–28, (1979).

[7] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng, 'Deep label distribution learning with label ambiguity', volume 26, pp. 2825–2838. IEEE Press, (June 2017).

[8] Xin Geng, 'Label Distribution Learning', in *IEEE Transactions on Knowledge and Data Engineering*, volume 28, pp. 1734–1748, (2016).

[9] Xin Geng and Peng Hou, 'Pre-release prediction of crowd opinion on movies by label distribution learning', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-15*, (2015).

[10] Xin Geng, Qin Wang, and Yu Xia, 'Facial age estimation by adaptive label distribution learning', in *Proceedings - International Conference on Pattern Recognition*, (2014).

[11] Tanya Goyal, Tyler Mcdonnell, Mucahid Kutlu, Tamer Elsayed, and Matthew Lease, 'Your Behavior Signals Your Reliability: Modeling Crowd Behavioral Traces to Ensure Quality Relevance Annotations', in *Sixth AAAI Conference on Human Computation and Crowdsourcing*, (2018).

[12] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis, 'On Clustering Validation Techniques', *Journal of Intelligent Information Systems*, **17**(2), 107–145, (2001).
[13] Christopher Homan, Ravdeep Johar, Tong Liu, Megan Lytle, Vincent Silenzio, and Cecilia Ovesdotter Alm, 'Toward macro-insights for suicide prevention: Analyzing fine-grained distress at scale', in *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pp. 107–117. Association for Computational Linguistics, (June 2014).
[14] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum, 'Label propagation for deep semi-supervised learning', (June 2019).
[15] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser, 'Chapter 11 - analyzing qualitative data', in *Research Methods in Human Computer Interaction (Second Edition)*, eds., Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser, 317 – 319, Morgan Kaufmann, Boston, second edition edn., (2017).
[16] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M. J. Rosato, 'A 3d facial expression database for facial behavior research', in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 211–216, (April 2006).
[17] Miaogen Ling and Xin Geng, 'Soft video parsing by label distribution learning', in *Frontiers of Computer Science*, pp. 1331–1337, (2018).
[18] Tong Liu, Christopher Homan, Cecilia Ovesdotter Alm, Megan Lytle, Ann Marie White, and Henry Kautz, 'Understanding discourse on work and job-related well-being in public social media', in *Proceedings of the 54th Annual Meeting of the ACL*, (2016).
[19] Tong Liu, Akash Venkatachalam, Pratik Sanjay Bongale, and Christopher M. Homan, 'Learning to Predict Population-Level Label Distributions', in *Seventh AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pp. 68–76, (2019). A preliminary version appears in [20].
[20] Tong Liu, Akash Venkatachalam, Pratik Sanjay Bongale, and Christopher Homan, 'Learning to predict population-level label distributions', in *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, pp. 1111–1120. ACM, (2019).
[21] Andrea Malossini, Enrico Blanzieri, and Raymond T. Ng, 'Detecting potential labeling errors in microarrays by data perturbation', volume 22, pp. 2114–2121, (2006).
[22] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz, 'Building a large annotated corpus of English: The Penn Treebank', volume 19, pp. 313–330, (1993).
[23] Gina Neff and Peter Nagy, 'Talking to bots: Symbiotic agency and the case of Tay', in *International Journal of Communication*, volume 10, pp. 4915–4931, (2016).
[24] Yi Ren and Xin Geng, 'Sense beauty by label distribution learning', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, (2017).
[25] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis, 'Get another label? improving data quality and data mining using multiple, noisy labelers', in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 614–622. ACM, (2008).
[26] Tom Simonite, 'When it comes to gorillas, google photos remains blind', (Jan 2018). https://bit.ly/2FzeNM7.
[27] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng, 'Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, Stroudsburg, PA, USA, (2008). Association for Computational Linguistics.
[28] James Vincent. Twitter taught Microsoft's friendly AI chatbot to be a racist asshole in less than a day, 2016. https://bit.ly/2u8m6qB.
[29] Jing Wang and Xin Geng, 'Theoretical analysis of label distribution learning', 5256–5263, (2019).
[30] Yingzhi Yang, 'China chatbot goes rogue: 'do you love the communist party?' 'no'', (Aug 2017). https://on.ft.com/2vp8Qi5.
[31] M. Zhang, 'A k-nearest neighbor based multi-instance multi-label learning algorithm', in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pp. 207–212, (Oct 2010).
[32] Xiaojin Zhu, 'Semi-supervised learning literature survey', Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, (2005).

# A Appendix

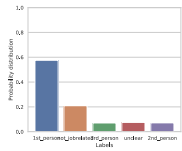

**Figure 8**: Overall label distribution of the JQ1 dataset [19].

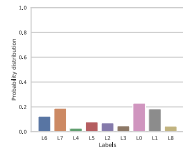

**Figure 9**: Overall label distribution of the NS dataset [9].

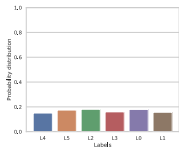

**Figure 10**: Overall label distribution of the FE dataset [16].