Humans in the loop on subjective domains

Akash Venkatachalam
Graduate Student
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623
Email: av2833@rit.edu

Abstract—Labelling a data requires a lot of human intervention. Especially if the data is related to social media then many different opinions materialize, given the same set of data. These opinions and perspectives come through a worker’s personal experiences. When it comes to machine learning feedback, many of these labels with different views are often condensed to form a single “ground truth” label. This method of feedback hides the true, potentially rich and diverse interpretations of the data found across the social spectrum.

In this project, we explore different unsupervised learning approaches for clustering data around their true label distribution. We also aggregate labels over the range of data points, and not just based on individuals. This will help us enrich the costs of humans in the loop.

Index Terms—Twitter; Meteor Framework; Amazon Mechanical Turk; Human-in-the-loop active learning framework; Annotation; MMM, GMM, Labelling, Clustering

I. INTRODUCTION

A large amount of information on social media is not labelled. Labeling of the data is a challenging task as it requires human effort. Direct human annotation can be costly and a very tedious process. We have to look for and find new tools to make this labelling process much faster and cheaper. We aim to build a web application that allows crowd-sourcing for the annotation of job-related tweets. The data along with the labels are stored in MongoDB which is integrated with Meteor. This data is provided as an input to the learning algorithm in classification framework. On training this classifier with the labelled data, new incoming unlabelled data can easily be classified.

In this paper, we primarily discuss the theory of subjectivity for labelling data and also for exploring them with the help of clustering models. The theory of subjectivity is really a challenging task for validating the ground truth for many reasons. Some of them are: (a) the idea of ground truth is of primal importance to most of the machine learning algorithms, and here we propose to do away with it completely; (b) Noise and ambiguity could be another reason that makes it challenging to distinguish uncertainty due to subjectivity; (c) It is not always possible to strongly represent a fixed underlying distribution since, in many platforms the number of workers who label the task is too small (< 30) [1]. In Figure 1, the black dots represent the data items which is labelled by five worker each. When we pool together, we can see on the right side, that same cluster $k$ is made into a much larger sample for all items in that cluster. We are aiming to study about a distribution of beliefs about a data item, rather than a condensed “ground truth” label.

We also discuss about statistical tests to get the minimal number of votes or crowd-source worker opinions required to get the actual label. We will be performing clustering based on a probability mass function of the samples that were collected using Amazon Mechanical Turk. Through this process, we are expecting the distribution of probability mass functions to be clumped up around the cluster centroid. K-means clustering, Gaussian mixture model, Multinomial Mixture Model will be used for performing the statistical analysis. The evaluation of this analysis will also be executed.
The organization of this research paper is as follows: Section II discusses research and related work pertaining to the frameworks we use in this project. Section III briefly explains about the dataset we use for obtaining labels and all the collections built for developing this project. Section IV describes the methods used for analyzing cluster patterns and also the software that we used. Section V will focus on the discussion and our perspectives in this approach. Section VI explains the result of this project. Section VII will conclude the paper, by giving the current status of our research and what prospects the future holds.

II. RELATED WORK

Mahendran et al. [2] developed a human-in-the-loop open active model. The dataset is a collection of tweets over a one year period from Twitter website. This data was labelled with the help of Mechanical Turk, a crowd-sourcing platform developed by Amazon that enables humans to perform the annotation task. This labelled data was collected using CrowdFlower. At the server side, the labelled data was stored and accessed with the help of Meteor framework. This paper also talks about performing a statistical test to get the minimal number of labels required to get the actual label. For this a K-Means clustering and Gaussian mixture model having the euclidean distance as a distance metric was tested.

Tsoumakas et al. [3] demonstrates an approach to construct a ensemble multi label classifier called Label Powerset (LP) learning. LP learning had the advantage of taking label correlations into account, but it seems to suffer from the large number of label subsets. So this LP learning combined with an algorithm called RAKEL (RAndom k-labelSets) which extensively calculates label correlations and also it avoids the previously mentioned problems of LP. The RAKEL (RAndom k-LabELsets) algorithm monotonously develops an ensemble of m Label Powerset (LP) classifiers. At each epoch, i = 1..m, it arbitrarily picks a k-labelset, Y_i, from Lk without placing them back. The algorithm then learns an LP classifier hi : X \rightarrow P(Y_i). The authors performed this experiment on three different application domains. The drawback of this method is that the continuously changing arbitrary nature of RAKEL was driving the authors to include new models that strongly affect the performance of boosting algorithm like adaBoost in a negative way [4]. The drawbacks were also addressed in another paper written by Kawano et al. [5].

Madjarov et al. [6] presents an extensive experimental correlation of 12 multi-label learning methods which will be evaluated with 16 measures. Evaluations in data science are very important as describe the performance of our chosen model. The authors compare all their evaluation models using efficiency in the units needed to learn a classifier and the amount of time required by a model to produce a prediction for test data set. Quick weighted voting concept is against the thought that a set of top rank classes can be obliterated during voting early in the process [7]. The authors also prove that they cannot exceed the running maximum even if they reach the maximal voting mass in all the evaluations that are remaining. For choosing a class that has not received any vote, the authors used pairwise classifiers depending on the voting loss value. As the number of evaluations increase, the voting loss (starting from zero) increases monotonically. Through this process, the class with the current minimal loss is then chosen as best candidate for the top ranked class.

In this paper, the authors prove that for multi-label classification the best performing methods overall are hierarchy of multi-label classifiers (HOMER) and random forests of predictive clustering trees (RF-PCT). This is followed by binary relevance (BR) and classifier chains (CC) [6].

Liu et al. [8] discusses a way to precisely cluster the given document corpus, they introduce a much better feature set to their data for representing each document. They also performed Gaussian Mixture Model (GMM) together with the Expectation-Maximization (EM) algorithm to conduct an initial document clustering [9]. The authors develop a hierarchical structure which resembles a tree. This structure with one cluster has a root node that contains all the documents. This was performed with using hierarchical clustering method. The most commonly used method in this category is the hierarchical agglomerative clustering. Efficiency evaluations for the choice of model was conducted in the same way as they conducted for document classification.

Doan et al. [10] proposes a Grid-based clustering algorithms primarily for spatial data mining. The idea is to first reduce the image area into a fixed number of entities and then execute all operations on a quantized space. The authors of this paper, also discuss about density-based clustering algorithms which primarily focuses on the frequency of the pixels or small elements rather than shrinking them. An approach using Haar (digital image feature) is used in this project which seems to be advantageous in image processing tasks such as gesture detection and face detection. When it comes to clustering, the authors reject the idea of using Haar features since Haar features are so naive, that an image has plenty of Haar windows and also it fails to distinguish between subtle differences. Therefore, the authors conclude that without a labeled dataset it is almost impossible to choose the most optimal features for clustering. The future work of their project is develop a stronger framework that can predict the optimal features for clustering an image [11].

Chiang et al. [12] developed a model exclusively for spatio-temporal dynamics of taxi bookings. This model captures the demand for taxis and notifies the server. The authors conducted a research with less than a million bookings using three months of taxi booking data. This data was obtained from around hundreds and thousands of commuters in Singapore. A modification was made in multivariate Gaussian Mixture Model (GMM) and this approach or clustering technique was used to cluster bookings in two-dimensional space over D days into K clusters, each represented by a mean and covariance. They came up with several possible ways for detecting anomalous groups of bookings using GMM model. To verify this they tried to use a simple approach. It was to bin all taxi bookings by both space and time. For ease of
representation, each bin was made to have bookings in a grid cell over only one hour of the day. The authors claim that, although their motivation was to focus on taxi bookings, the models they developed can be used for modeling other spatio-temporal events.

III. DATA

The dataset is a collection of tweets over a one year period from Twitter website. There are more than 200,000 records in the dataset. Each tweet is considered as a record. A Meteor framework is being used for this project and so all the dataset will be stored in MongoDB running on the server. About 2000 records were labelled through an online workforce called Crowd Flower. This labelled data set is stored in the collection called ‘crowdflowerAnnotations’. This document will have 10000 documents in it, which typically stores fives times the copy of each record (i.e., $2000 \times 5 = 10000$). This collection will have seven tweets - one target tweet, three tweets before that and three tweet after the target tweet and also time stamp for each tweet [8].

The collection ‘activeTweet’ stores ten tweet set in each document along with a HIT ID. So when each HIT is created, a document is added to this collection. This document with ten tweet set will be displayed to the user on Amazon’s Mechanical turk. When a worker agrees to work on this HIT, twelve tweets will be displayed. This is about randomly adding two extra tweets from the same set of ten tweets. The collection ‘centuryLabels’ was created from ‘crowdflowerAnnotations’ and it would do the work of ‘activeTweet’ collection. The collection ‘centuryLabels’ will have only hundred selected tweets. The way each tweet was selected is explained in section Discussion.

The collection ‘label’ is the one in which all the final labelled data from Mechanical turk is stored. Each record in this collection is added when the crowd worker submits a HIT. Each document in this collection will contain information like HIT ID, Worker ID, time stamp of that submitted HIT, answers for question 1,2 and 3. The response for question 1 and 2 will be a string type of choice from the radio button. The response for question 3 (with check boxes) will be a +1 if the option choice is checked by the worker (by default it is set to -1).

The collection ‘hit’ is used to store the blueprint of details about a HIT. Each document in ‘hit’ is added once the worker submits the work on the questions. So the documents in ‘hit’ will contain Worker ID, HIT ID and the set of tweets as a list that was displayed to the worker in order. The collection ‘worker’ is used for storing just the details of worker and which HIT ID the person was working on.

IV. METHODS

For creating the ‘centuryLabels’ collection, we chose data items that were closely placed to the cluster centroid which had high entropy and low information gain. For instance, assume there were 3 best clusters under a model with the number of centroids as 3, 4 and 5. Our target was to get 50 tweets, in that case we would choose $50/(3 + 4 + 5)$ tweets that is 4 tweets around each cluster centroid. In this way, we created ‘centuryLabels’ collection. Figure 9 gives a model of how this collections works. Initially, we have the filesystem containing text files with the best clusters from MMM models for all the three questions. The tweet IDs are extracted from these files using the above mentioned procedure. Now these
specific tweet IDs are compared with 'crowdflowerAnnotation' collection and inserted into 'centuryLabels' collection. After this process, createHIT file is run on the newly generated 'centuryLabel' collection and publish on the Turk. Figure 11 gives an overview of the words that are most frequently used. Evaluation of our cluster model based on the labels provided is one of the most important part of software development process. This process will give an idea if the model we chose is strong enough or if we have to look for a better model. An evaluation system was designed for evaluating whether the labels given by the crowd-source workers coincide with the clusters given by the entire set of workers.

For the start of this evaluation process, labels were represented in a JSON format. This JSON file will contain its key as a tweet ID and then value as a list of probabilities to which each cluster this particular tweet belongs to. Since it is a probability of cluster, the sum of probabilities in the value list should be one. A python script was written to create a dictionary for storing the information of which cluster a particular tweet belongs to. The key of this dictionary is used to pull data from 'crowdflowerAnnotation' collection. A new collection was created in the database and it was named as ‘clusterSet’. Firstly, the documents matching the key are inserted into the new collection, ‘clusterSet’. Next a new field of cluster and to which cluster it belongs to is updated to each and every document. So the new collection will give information about which cluster a tweet belongs to.

A. Clustering Strategies

1) Multinomial Mixture Model (MMM): Given a multiple choice labeling question $Q$, with $d$ labels $\{1, \ldots, d\}$, a collection of data items $x_1, \ldots, x_n$, and a population of individuals $P$, let $z_1, \ldots, z_n$ denote for each corresponding data item $x_i$. Then the probability distribution of answers over $P$ for question $Q$, i.e., $z_j$ is a $d$-dimensional vector, where for $j \in \{1, \ldots, d\}$, $z_{ij}$ is the fraction of $P$ who would give answer $j$ if asked $Q$ about $x_i$ [13].

We also assume that first the probability mass function (PMF) is selected and then each $y_i$ is generated. Our PMF is $\theta_k \in \{\theta_1, \ldots, \theta_p\}$ over $d$ alternatives, according to a probability distribution $T = (\tau_1, \ldots, \tau_p)$ (in keeping with the spirit of a parameterless approach, we assume a uniform prior over $T$), and then selecting the answers by drawing them independently and identically from the same $\theta_k$. Let $z = (z_1, z_2, \ldots, z_n)$ indicate the (hidden) PMF that generated each $y_i$, i.e., each $z_i \in \{\theta_1, \ldots, \theta_p\}$ [13].

Under these assumptions, $(y_i|\theta_j)$ is the following multinomial function

$$f(y_i; \theta_j) = \frac{(\sum_k y_{ik})!}{y_{i1}! \ldots y_{id}!} \theta_{j1}^{y_{i1}} \ldots \theta_{jd}^{y_{id}},$$

and the (a posteriori) likelihood function is

$$L(\theta; y, z) = (y, z|\theta) = \prod_{i=1}^{n} \prod_{j=1}^{p} f(y_i; \theta_j)^{I(z_i=j)}.$$

An advantage of the multinomial mixture model is it can easily capture the degree of confidence we have in the label. Given a scenario where we have only few labels, the design of the model is in such a way that the likelihood of the data item being chosen in a particular cluster is closer to uniform (higher entropy). As we aggregate more labels, a much picture of which cluster PMF $\theta_j$ the distribution of labels most represents emerges and the cluster likelihood distribution becomes less uniform (lower entropy) [13]. We can use this to help determine whether we have enough labels per cluster for the model to be useful. In our MMM model, the cluster chosen as best is expected to come from an empirical label distribution, then we select all samples from that one chosen cluster.

2) Gaussian Mixture Model (GMM): Gaussian mixture model is a probabilistic unsupervised approach to model the normally distributed data in population. It is extensively used to fit a vector of unknown attributes. The GMM model is parameterized by two types of values with $K$ components: the mixture component weights ($\phi_k$ for component $C_k$ where
\[ \Sigma_k \prod_{0_k = 1}^{K} \theta_k = 1 \] and the component means and covariances (\( \mu_k \) and \( \sigma_k \) for the univariate case, \( \vec{\mu}_k \) and \( \Sigma_k \) for the multivariate case, respectively). The expectation-maximization (EM) clustering algorithm is also used for finding out the mixture model’s parameters through an iterative process until the algorithm with the maximum-likelihood estimation converges[10] [13].

B. Software Used

1) Meteor Framework: Meteor Framework integrated with MongoDB is used in the project for setting up connectivity to Amazon’s Mechanical Turk. Our idea is to get a wide distribution of labels from different people. Otherwise, we can label all the data on our own. To get different perspective and understand the multitude of thoughts we need to raise these tweets or phrases to a crowd-sourcing platform. The crowd-sourcing platform we’re using here is Amazon Mechanical Turk. For designing the front-end and controlling the tweets publishing and receiving the labels, we need a framework. This fully integrated task is supported by Meteor framework. It is a full-stack development framework [2].

2) Scikit Learn: Scikit Learn, a machine learning library for Python’s programming language is used for building algorithms like MMM, GMM and K-Means. In this project, Scikit Learn was used mostly for performing K-Means clustering algorithm, for calculating the entropy and for executing probability distribution.

3) Mechanical Turk: Amazon Mechanical Turk is a crowd sourcing platform, allowing the requester to make use of human intelligence for performing specific tasks. Usually tasks that are difficult to comprehend by the machines at this time are sent to the Mechanical Turk. In this platform, the concept of HIT or Human Intelligence Task is a representation of a single task that a worker will work on and get monetary reward. Mechanical Turk has two type of users: requester and worker. The requester is the user who wants their data labelled and so they create an account and publish their questions or HITs on the Turk. The worker is the user who completes these tasks and gets paid for it. Amazon Mechanical Turk has two modes of operation: Sandbox mode and Production mode. Initially, to test the working of HITs we develop our implementation on the sandbox environment. If it works satisfactorily then we move our implementation to the production mode. In production mode, our HITs are published to Mechanical Turk site and the workers can submit their response.

4) Anaconda: Anaconda is an open source software for python language. In this project we use Anaconda for making virtual environment utility, for creating isolated python environment that works with pip. Anaconda comes with an in-built conda environment, which is useful for managing installations of various packages.

V. DISCUSSION

In this section, we would like to give a number of data items, that we have analyzed primarily from job series datasets. We are considering to get a better perspective when the MMM model fails and also to get insights of the reasons that MMM could not outperform the other models, by exploring the probabilistic labels and semantic meanings of our data items.
Fig. 11. Word Cloud model for question 3 with nine clusters

Q1. Which of the following items could best describe the point of view of job/employment-related information in the target tweet?
- 1st person
- 2nd person
- 3rd person

Q2. Which of the following items could best describe the employment status of the subject in the tweet?
- Employed
- Not Employed
- Not in Labor Force

Q3. Does the subject specifically mention any job/employment transition event in the tweet? (Choose all that apply)
- Getting hired/job seeking
- Getting Fired
- Quitting a job
- Losing job some other way
- Getting promoted/raised
- Getting cut in hours

Fig. 12. An annotation task contains three questions Q1, Q2, and Q3. Q1 and Q2 accepts only one answer while Q3 is a check-box question where the crowd workers can select more than one choice.

For a tweet, “Really want a job where I get tips”, Table I gives the label probability distributions for Q1 (see Figure 12) provided by MMM, GMM and crowdworkers, respectively. It can be seen that, “I”—a common singular first-person subject pronoun is present in the tweet. Our model, MMM’s probabilistic prediction of “1st person” point of view is more accurate than GMM’s prediction (0.79 > 0.72). It can also be noticed that MMM performed much better that the probability distribution obtained from crowdworkers.

| MMM  | 0.79, 0.05, 0.04, 0.06, 0.06 |
| GMM  | 0.72, 0.08, 0.12, 0.05, 0.02 |
| jobQ1BOTH | 0.93, 0.00, 0.07, 0.00, 0.00 |

TABLE I

Probability distribution of Q1 for the tweet “Really want a job where I get tips”.

In another tweet for question 1, ““@SOMEONE Got the job’ congrats!!” MMM, GMM and crowdworkers provided labels as Table II shows. The user was quoting another user’s message (“@SOMEONE Got the job’) and add a comment ("congrats!!") to congratulate his friend who got a job. This tweet is job-related so MMM distinguishes it from “Unclear” and “Not job-related” categories better than GMM does (0.05 < 0.11 and 0.06 < 0.12 in Table II) [13]. The actual tweet in here is “Congrats”, it can be viewed in two different perspectives. One, the worker can think that ‘Congrats’ is just a word of praising and it may not have anything to do with 1st or 2nd person. On the other hand, a worker can think of this as a retweeted message and understand the context of it. As we can see, MMM ranked higher probabilities to “2nd person” and “3rd person” and a relatively lower probability to “1st person” choice than GMM does, which describes the target tweet more accurately because the user was talking about another one’s job-related information instead of his/her own matter.

| MMM  | 0.13, 0.43, 0.33, 0.05, 0.06 |
| GMM  | 0.29, 0.21, 0.27, 0.11, 0.12 |

TABLE II

Label distributions of Q1 for “@SOMEONE Got the job’ congrats!!”.

There’s another example with uncertainty and ambiguity, highlighted in Figure 13 with number 0. It is to be noted that Five CrowdFlower workers unanimously labeled the target tweet as “Not job-related” (see Table III). This is mainly because the judgment can be made fairly easily with the help of the contextual information. But MMM and GMM gave almost completely opposite predictions (between “1st person” and “Not job-related”), which could be, because of our inputs to both models which are pure probabilistic vectors without any textual information [13].

| MMM  | 0.20, 0.03, 0.05, 0.05, 0.67 |
| GMM  | 0.64, 0.11, 0.00, 0.00, 0.25 |

TABLE III

Label distributions of Q1 for “Not saying that’s a problem”. For a more complex question Q3 in Figure 12, MMM shows the capability to provide a more accurate probability distribution than GMM for the tweet “My mom made me a.
huge bubble bath after I got off work. That was so sweet of her". It can be observed that for both the models (trained with 17 clusters/components), as Table IV, MMM assigns a higher probability which is closer to that provided by crowdworkers to the 10th class (Coming home from work) than GMM does.

<table>
<thead>
<tr>
<th>MMM</th>
<th>GMM</th>
<th>JobQ3BOTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.02, 0.00, 0.00, 0.01, 0.01, 0.02, 0.07, 0.02, 0.07, 0.40, 0.26, 0.10]</td>
<td>[0.03, 0.00, 0.04, 0.01, 0.03, 0.06, 0.06, 0.08, 0.03, 0.12, 0.47, 0.07]</td>
<td>[0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.08, 0.00, 0.58, 0.33, 0.00]</td>
</tr>
</tbody>
</table>

LABEL DISTRIBUTIONS OF Q3 FOR "My mom made me a huge bubble bath after I got off work. That was so sweet of her".

Similarly, “Drunk idiots at work are probably the main reason I hate my job” is described better by MMM (see Table V), particularly about the 7th class (Complaining about work).

<table>
<thead>
<tr>
<th>MMM</th>
<th>GMM</th>
<th>JobQ3BOTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.01, 0.00, 0.01, 0.01, 0.01, 0.02, 0.58, 0.01, 0.09, 0.02, 0.17, 0.07]</td>
<td>[0.00, 0.03, 0.00, 0.00, 0.07, 0.00, 0.15, 0.05, 0.14, 0.00, 0.30, 0.27]</td>
<td>[0.00, 0.00, 0.00, 0.00, 0.10, 0.00, 0.70, 0.10, 0.00, 0.00, 0.10, 0.00]</td>
</tr>
</tbody>
</table>

LABEL DISTRIBUTIONS OF Q3 FOR “Drunk idiots at work are probably the main reason I hate my job”.

When the domain of labels were changed from job to suicide, neither MMM or GMM predicted the label distributions well for this tweet “happy birthday bro! I miss having u by my side in that net, hopefully soon we’ll get to share it again, love you bro @SOMEONE”:

<table>
<thead>
<tr>
<th>MMM</th>
<th>GMM</th>
<th>Suicide</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.05, 0.08, 0.70, 0.16]</td>
<td>[0.20, 0.29, 0.29, 0.23]</td>
<td>[0.00, 0.00, 0.29, 0.71]</td>
</tr>
</tbody>
</table>

LABEL DISTRIBUTIONS FOR “happy birthday bro! I miss having u by my side in that net, hopefully soon we’ll get to share it again, love you bro @SOMEONE” IN SUICIDE DATASET.

VI. RESULTS

Figure 14 depicts the histogram of labelled distribution for 50 job related tweets having 50 labels each. In this figure, we can notice that Group 1 (red colored plots) have most of their probability distribution on choices Getting hired/job seeking and None of the above, but job-related. In this group, all the tweets are talking about plans or ways to get a job (e.g., really want a job, dont put that on ur resume for a minimum wage job). Group 2 (cyan colored plots) has most their probability distribution on Getting hired/job seeking (e.g., got the job). The third group (brown colored plots) has more weightage on Complaining about work and Going to work, suggesting a topic about complaining about having to go to work. Group four (green color plots) are a set of tweets complaining about work while at work. Group five (blue colored plots) (where

VII. CONCLUSION

In this project, we have studied about the underlying problems that were considered as trivial and neglected in supervised learning domain. We also considered explaining the problem with the help of clustering to pool and aggregate...
labels. Our results suggest that such methods are effective to an extent. However, the reason may have to do with the underlying sources of subjectivity being limited. More research work need to be done in this point of view to make the most out of humans-in-the-loop.

As a future work we need to focus on developing an algorithm that is much more powerful than our proposed MMM model. We also want to extend our research by building a supervised classifier like Convolutional Neural Network (CNN) to effectively use it across a wide range of corpora.

REFERENCES