University of Glasgow | School of Computing Science

Generating Individual Predictions from Aggregated Data

Tamit Halder

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

13th December, 2021

Abstract

Summarizing a collection of data obtained from different sources can be referred as data aggregation process. Data aggregation technique has proved to be effective for many purposes such as maintaining privacy or scalability. Aggregated level information extracted from a group of test objects which share similar attributes, can be treated as useful in many areas of modern science specially in healthcare advancements. However, it is a challenging and a novel area to explore.

The main aim of our study is to suggest multiple approaches to extract useful aggregated information from groups or clusters and using the same information for predicting classes for individual instances. Our proposed methodologies are quite unique and novel in nature if compared with existing studies on learning from aggregated data. Our methods focus on building pseudo training data set around mean or centroid location of each cluster and assigning them labels in efficient way. The results of our research work exhibit that our recommended approach is efficient in predicting on classification related problems. During experiments, our model achieved close accuracy score compared to traditional supervised machine learning approach where model is trained with complete information. Our models are also able to make significantly better predictions than naive clustering approaches where only mean or centroid information is used.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: Tamit Halder Signature: Tamit Halder

Acknowledgements

I want to thank my supervisor, Dr, Debasis Ganguly who has guided me with his knowledgeable advises throughout the project period and arranged meet up with me for discussions during his free time. He always motivated me to go extra mile and think out of box. I would also like to thank my family who supported me the whole time, and specially my mother who never fails to encourage me to pursue my dreams.

Contents

Chapter 1

Introduction

In this initial chapter, we will discuss about the problem area and the motivation factor to work on learning from aggregated data. We will also present our main objective and formulate our research questions which will eventually lead us to our goal of this thesis.

1.1 Motivation

Data Aggregation has become a topic of paramount importance in recent times. Despite the inventions of latest state of art machine learning models and information explosion in this era of data, there are still few restrictions in certain domains where data needs to be protected due to privacy concerns or ethical reasons [15]. Hence, ground truth data is often aggregated in such way so that it is quite impossible to trace back to the original individual from aggregated level information. One prime example can be healthcare sector where aggregated level data with some basic features such as age, gender, postal code, disease, symptoms etc can be made publicly available. At same time more sensitive information about an individual might be kept hidden.

One more important application area of prediction from aggregated information is Randomized Control Trials (RCT) in medical treatments domain, where mostly aggregated observations are obtained from these trials. During RCT trials, subjects are often assigned randomly one of the two groups where one (experimental) group receives the intervention which is under test, and other (control) group does not receive it [5] [10]. Using standard machine learning techniques can often lead to ecological fallacy if an individual element has quite different characteristics than other elements of group and eventually behave in a different manner compared to others.

Keeping in mind the importance and sensitivity of above mentioned application areas, we understand the caution that we need to exercise while handling the aggregated data. This challenge inspires us to work on this problem and propose some novel methods. To avoid pitfalls, we will use our proposed method and will compare the performance with models which are trained with full data.

1.2 Objectives and Research Questions

The main aim of the project work is to explore and implement different strategies on aggregated group level information and extract useful information from them to build pseudo vectors for further training of our model. After learning from these new data which are just the noisy representation of the original vectors, our model will be able to make predictions on unseen individual instances. So we can formulate our first research question as following-

RQ1. What pieces of information will be useful to extract from aggregated dataset based on which it is possible to make a reasonably effective classification prediction on individual instances?

This learning can be modeled using a traditional machine learning algorithm to predict on unseen individual elements. To achieve this, our model needs to be trained on individual instances. But due to ethical and legal restrictions, it might not be possible to train on actual individual instances. So finding effective ways to generate a useful training dataset will be the next part of our research.

RQ2: How to generate these individual pseudo-instances which are most similar in nature to the actual training instances?

To train a machine learning model, we need both training data and labels. For this research purpose, we assumed that we had access to true labels of actual training instances. While our previous research question focuses on generating pseudo vectors, now we need a way to assign labels to each of our pseudo instances. This problem analysis lays the foundation of our third research question.

RQ3: How to generate labels for these newly generated pseudo instances provided we have access the truth data? What strategy should we apply to predict most likely labels on each training instances?

Our final research question will focus on effectiveness and empirical validation of overall research work as we are dealing with uncertainty which arises after data aggregation.

RQ4: How effectively can we carry out a laboratory-based simulation of learning with uncertain data (uncertainty arising from aggregation over arbitrary groups)?

In subsequent chapters, we seek answers to our research questions. Once we are able to answer these questions, we will be able to generate a new pseudo training data set from aggregated data, which will closely match the characteristics and features of the actual vectors. In this project we will try to propose and build new models solving above stated challenges.

1.3 Outline of the thesis

In this chapter, we have described the problem from a high-level perspective. The rest of the thesis is organized as follows. In chapter 2 we will discuss about background and previous related research work done on Individual level prediction from aggregated data and present our research questions. We will give a very high level introduction of our proposed neighborhood models for data reconstruction. In chapter 3, we will discuss about the design of our proposed models along with implementation strategy. Evaluation metrics will also be discussed. Chapter 4 will focus on experimental setup for our model testing. We will also try to compare our model with other naive baseline to understand the effectiveness of our model. Models which were fully trained using only centroid or mean information of clustered groups of instances should be our standard benchmark. While models trained on complete information will be our apex benchmark. We will conclude our discussion in chapter 5 with overall outcome of our experiments and also future scope of our project work.

Chapter 2

Analysis

In this chapter we will discuss about the surveyed literature and will try to build the foundation on some of the key concepts which are being used in subsequent chapters of the report. We will analyse the data aggregation and neighborhood methods that we will be using.

2.1 Background and Related Work

For the survey purpose, we will focus on studies related to two different stages of our model. Initially, we will feature some existing binary classification focused study using Multiple Instance Learning (MIL). Then we will move on to multiclass classification based studies. Further, we are also interested in existing work on removing noise from data and labels. Finally, we will try to differentiate our proposed methods from existing methods of learning from aggregated data.

Our work has been inspired by traditional MIL or Multi Instance Learning method which is supervised learning in nature. In MIL, learner receives a group of instances combined in a cluster which is then provided a Binary label based on presence/absence of a certain class member [1]. While MIL only focuses on Binary classification using approximate bag probabilities for training purpose, Li and Xi-Lin developed a model where exact bag probabilities were used for estimating maximum likelihood model parameters [11]. There have been previous studies done where learner learns the model from proportions of the label (LLP Model) in each bag[6], [13]. On other hand, Bao et al. [2] proposed a learning method which focuses on pairwise similarities of two instances. In this method, a binary value is used to determine whether two instances belong to same category or not . However, the biggest difference with MIL and our proposed method is that the scope of our proposal is not limited to Binary classifications. Our proposed models are able to predict on multiclass classification problem.

In a recent study, which was focused on multiclass level classification problem, the concept of pairwise similarity was used. They proposed meta classification learning method, which can optimize a binary classifier for predicting pairwise similarity based on maximum likelihood estimation [8]. Zhang et al. has introduced a versatile framework consisting of pairwise similarity and triplet comparisons recently on top of the existing maximum likelihood estimation. They were able to achieve more than 90 percent accuracy on MNIST dataset with their proposed classification model [18].

Our study has been unique in another way because of the uncertainty that we are dealing with related to reconstructed sample data. Kendall et al. described uncertainties (Aleatoric and Epistemic) that are involved in Bayesian modelling. They explained an aleatoric uncertainty as something which captures the noises within the observations. In contrast, an epistemic is the uncertainty that exist in the model itself. They also made the observation that for most of the big data related problems, it is most effective to model aleatoric uncertainty because it can't be explained away. In contrast, epistemic uncertainty can be explained away with the large amount of data [9]. In this thesis, we focus on creating an effective model that aims to reduce noise and uncertainty while reconstructing training instances, which we tried to achieve by minimising the distance from true data points.

Zadrozny [17] has defined sample selection bias in terms of machine learning. Based on the assumption that the features are independent given a label y, they have introduced posterior probability with sample selection bias which is different from Naïve Bayes assumption. They also suggested that selection probabilities have to be estimated from data itself .

In a recent study, which focuses on both privacy preservation as well as data aggregation method such as K-Means clustering, Biswas et al. proposed a new variant of K-Means algorithm. This new privacy preserving K-means algorithm (PPK-means) was able in protecting privacy of original training instances or data vectors. The algorithm utilized a binary transformation method of original data and was able to process the incomplete information to predict at almost similar level of efficiency as encoded data [3].

Our proposed models are significantly different from existing studies considering the fact that we have emphasized on centroid information of each clustered set of data and aim to reconstruct samples from nearest neighbourhood region of cluster centre. We suggest two ways of multiclass prediction on individual instances from aggregated data. Our first proposed model reconstructs training samples using a user provided distance radius from centroid of the each clustered groups. The second model reconstructs data based on centroid vector and standard deviation across all features of every instance of the bag. Reconstructing training samples based on such information created some distortions in the training images. Ovadia et al. [12] showed in their studies that any such distortions can result in to decreased accuracy in various datasets, including the MNIST . Due to the presence of such noise and distortions in our sampled pseduo vectors, we understand the limitations of our model. We anticipate that our proposed model which is trained with noisy vectors and labels, may not perform at equal level to traditional models where complete information are used.

2.2 Data aggregation method : K-Means algorithm

To preserve privacy of the original data and other security/ethical reasons, we can not access the true vectors at any point of our model implementation. Rather, we implement a data aggregation strategy on our complete dataset which will help to form clustered groups of original instances, based on similar characteristics. For this task, we have decided to work on K-Means algorithm.

Despite being proposed several decades ago, K-Means algorithm is still one of the widely used clustering technique. In this algorithm, one initial cluster center is chosen. Every input is then assigned to it's nearest cluster center, and the cluster centroid value gets updated. This process keeps repeating until the algorithm reaches a maximum limit of iteration or cluster centers stop getting updated anymore[7].

Algorithm 1 K-Means Algorithm, reproduced from [7]

```
Input: $K$, the number of clusters
Input: $D$, A collection of a set of $N$ vectors ($|D| = N$)
Input: $M$, the maximum number of iterations
Output: A $K$-partition of $D$ such that $\bigcup_{k=1}^{K} D_k = D$
begin
    Randomly initialize $K$ cluster centres, $C_1 \dots C_K$. for $j = 1, \dots, M$ do
        for each $d \in D - \bigcup_{k=1}^{K} \{C_k\}$ do
            // Assign $d$ to its nearest cluster centre
            $k' \leftarrow_k sim(d, C_k)$  $D_{k'} \leftarrow D_{k'} \cup d$
        end
        Recompute cluster centres from the current partition $\{D_k\}_{k=1}^{K}$
    end
end
```

As mentioned earlier, the key focus in our project is to build optimum model which will be able to predict on individual instances from aggregated set of data. After implementing the K-Means algorithm, we will obtain our aggregated dataset. We will compare our proposed approaches with both naive K-means baseline and apex baseline, which in this case is prediction on individual data sets when complete information is available.

2.3 Analysing Neighborhood Methods

Our main objective is to determine a technique to reconstruct training data which are mostly similar to the actual training data, as we can not access original data due to privacy protection or ethical reasons. Most effective way to achieve this goal is by random sampling of vectors around a specified neighborhood of each cluster centroid or mean vector. For this, we have chosen two different methods for sampling around a cluster centre. We will give a high level overview of the techniques below.

2.3.1 Overview of Gaussian Neighborhood Method

A Gaussian distribution is considered as most common distribution function for real valued, independent and randomly generated distribution. A normal distribution function graph consists of two major attributes. First one is is the mean (μ) value of the distribution, while the other is the standard deviation σ which indicates the variation of the elements of the distribution. The greater the value of σ is, the more dispersed data points will be from mean [4]. Low σ value can ensure instances are centered around the mean, which will eventually give a better representation of that clustered group.

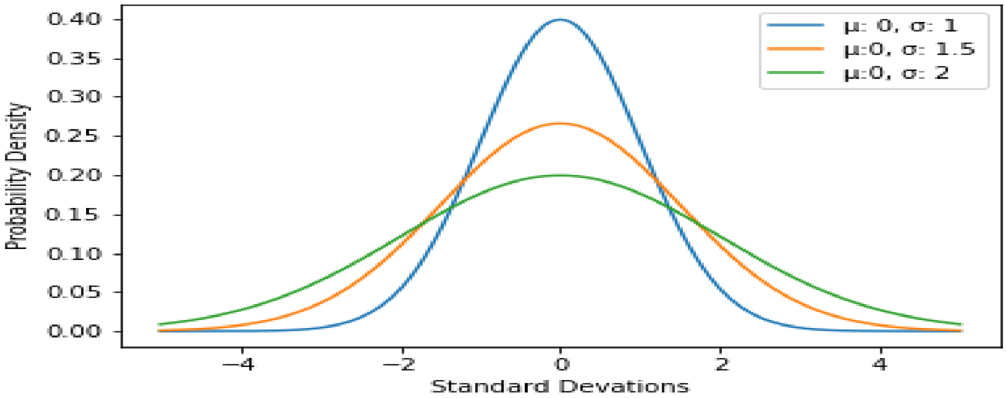

Figure 3.1: Normal Distribution curves with different mean and standard deviation values

2.3.2 Overview of ϵ Neighborhood Method

The simple ϵ neighbourhood is a method in which all nearest neighbors are clustered within a small radius of a given point. Pourbahrami [14] argued that choosing inappropriate or bigger ϵ value will lead to decreased efficiency.

In figure 3.2, If we consider a as a real number in a number plane and distance $\epsilon > 0$. Then from below image we can say point x lies within ϵ neighborhood of point a, while point y is outside of ϵ neighborhood area.

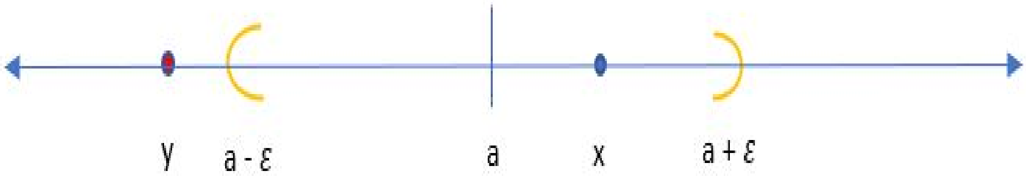

Figure 3.2: ϵ Neighbourhood of a real number

2.4 Summary

In this chapter, we covered the background research done for our thesis. We also presented the algorithm for data aggregation technique and introduced the neighborhood methods that we will be using in our codes.

Chapter 3

Proposed Methodolgy

This chapter describes our proposed methodologies based on analysis made on previous chapter. First, we will discuss about input dataset and the pre-processing steps, followed by a brief discussion on how we implemented two neighbourhood methods for reconstruction of new training vectors. Then we will describe our proposed method of sampling process for label generation based on prior probability and assigning them to reconstructed vectors. We will also discuss the machine learning models used to train our methods with newly constructed information which are noisy in nature. We will conclude the final bits of our model design which is evaluation metrics.

3.1 Datasets

We started with selecting the data sets focused on multi-class classification problem and also have sufficient training data to train our model well. Since the area of focus was working on novel method which can be implemented in various fields of everyday life, we assumed image classification problem would be a good starting point to experiment our proposed models. We have chosen to test our models on three different and widely available image classification sets.

- MNIST dataset
- CIFAR 10
- CIFAR 100

3.2 Input pre-processing and K-means implementation

As discussed earlier, we only considered multi class image classification related problems for our model. Though this model is capable of handling other tasks as well such as predicting from text data set. For all data sets that we have used, MNIST are black and white hand written digits while CIFAR-10 and CIFAR-100 are color image data set for various objects. MNIST images are (28,28,1) pixel in shape while CIFAR images are (32,32,3) RGB color images. Before passing the input data to our proposed model, we have decided to flatten our input images as vectors and also normalize them by dividing with 255 (pixel values ranges from 0 to 256) so that each feature value lies between 0 and 1.After applying this flatten step, each MNIST input image becomes a feature vector of length 784 and each CIFAR input image becomes a vector of 3072 features.

Our model can work after taking an input K from user which will tell us the number of clusters to use as part of aggregation step. The K value can range from anything between 50 to 1000. Using the value of K, we initiate the K-Means clustering. To save time in this step from next test onwards, we save the partition data for each specific K value in a tsv file.

Apart from K, our model can also take one optional input from user which is the ϵ value or simply the ϵ distance from each cluster centroid. This will be required when user wants to use ϵ Neighborhood method. New vectors will be sampled based on this ϵ distance. We will discuss more about this in chapter 3.5.

3.3 Estimating priors from a partition for label sampling

After implementing K-Means algorithm for data aggregation, our first task was to extract clustered information. For the task of Label sampling, we needed information such as number of elements present in each cluster and the ratio of the true classes present in each cluster partition. With the ratio information, we estimated the likelihood of different class labels within that cluster. We build this prior likelihood model based on the following hypotheses:

Hypotheses: If the number of true class labels for each class are determined in a partition, then we can calculate the likelihood by dividing the number of appearances of that class by total number of N instances within the same cluster.

These prior estimates were used later to sample labels for reconstructed vectors for the same cluster. The main idea was to generate exactly N number (where N is the number of elements inside a given cluster) of sampled labels and reconstructed vectors so that we do have equal number of training data, both at initial dataset and after implementing reconstruction steps.

3.4 Computing Gaussian Parameters

Our next step was to compute Gaussian parameters (Mean and Standard Deviation) for all elements present inside a particular cluster, across all features/dimensions. As we have already flattened our individual input vectors to one dimensional arrays, we scanned through every single dimension of each vector to calculate μ, or mean of a particular cluster group. Likewise, we also calculated σ or standard deviation for all features of K number of clusters. The mean or μ information obtained from this step are the cluster centroid vectors.

This functional step gave us two Gaussian parameters as output, μ and σ. If K is the number of clusters and F is number of features or dimensions of individual elements, then we will get a vector of length (K,F) for both μ and σ outputs. This information is a pre-requisite to reconstruct vectors using both Gaussian and ϵ Neighborhood method, which are our next steps.

3.5 Reconstruction using ϵ Neighbourhood

For our thesis, we have to reconstruct and randomly sample new vectors which should be within a certain radius (the ϵ value) from our target point, which is mean or cluster centers. So keeping ϵ value within a reasonable boundary is an effective approach. If we sample and reconstruct the vectors within a smaller radius from centre, then newly sampled vectors will more closely resemble the centroid vector of that cluster. While the cluster centroid itself reflects the average characteristics of all data points present inside that cluster. This is our first proposed neighbourhood vector generation method.

Our proposed model can work with different ϵ values to generate different set of pseudo vectors based on the ϵ distance from cluster centre. Generated vectors are generally blurred with poor boundaries as compared to original images of that class. In this step, we also pass the prior probability information obtained from chapter 3.3. With all these information, we are able to sample and reconstruct set of vectors and labels, equal to the number of original training data that were assigned in that cluster. Both output vectors and labels generated in this step are noisy in nature. These newly generated outputs will be new input for our training model.

3.6 Reconstruction using Gaussian Neighbourhood

For our thesis, we assumed each feature vector of any given cluster members, are independent of each other. In chapter 3.4, we described how we calculated μ and σ for all features of K number of clusters. These gave us equal number of mean and standard deviation vectors which is K. Using the Mean and Standard deviation information from each cluster, we sampled random vectors for N number of times. Here N is the number of elements present in that particular cluster.

Similar like ϵ neighborhood method, this functional step in our code also takes prior probability information in consideration while reconstructing sampled labels. As final output, this functional step reconstructs equal number of pseudo vectors and labels. This output will be the new input for our training model which is described in next step.

3.7 Training Model Architecture

We built our training model in Python using Keras with Tensorflow as backend. We have chosen Keras because it was easier to grasp compared to other machine learning libraries and it was also taught in our coure. We used combination of Jupyter Notebook as our development platform.

3.7.1 For MNIST Dataset

For MNIST dataset, our code loads the MNIST images, flattens them to convert into a vector and then computes our neighbourhood algorithm. Post neighborhood method implementation, it receives an output with a set of vectors and associated labels. The number of elements in this set will be equal to number of actual training instances passed at first step.

Our next step was to use these newly reconstructed vectors and labels generated from the prior probability method. These new vectors are not the actual training data from original dataset, but rather they are closely resembled noisy vectors. For this reason, using most probable labels for each of these vectors would make sense rather than giving same label to all members inside a single cluster based on its most common representing true label.

These input data are then fed into a three layered convolutional neural network. These layers work as feature extractors.The convolutional layers can break down images into smaller parts and model learns and detects the patterns for smaller parts of images. Hence it is quite useful in any Image classification task. We used dropout as our regularization technique. We used three dropout layers right after each convolutional layers to reduce overfitting of our model. Overfitting of a model occurs when a model is very well trained on given training data, rather than capturing some noises which is a an important essence in real world problem scenario. So the model may perform reasonably well with training data but may perform poorly on completely unseen data.

We also used two maxpooling2D layers after each of the first two dropout layers. A maxpooling layer helps to reduce the size of feature map and only extracts the most important feature of a block. Next, we have used a flatten layer before it passes through dense layers. Dense layers will perform the job of classification. They can not work with more than one dimensional data. Hence we used flatten layer before implementing it. Finally we used RELU and softmax activation functions. 'RELU' or 'Rectified Linear Unit' is most commonly used activation function. It needs much fewer computational resources than other activation functions and also adds sparsity to the overall model. This being a a multiclass classification problem, we have chosen softmax function which will help us generate probability distribution over the target classes.

The training model was compiled using Adam optimizer. We used a batch size of 1000 with 15 epochs for MNIST dataset. We have splitted the training and validation data into 80-20. We used consistent model architecture for our proposed neighborhood method experiments, K-Means baseline testing and also the apex line testing (Test with complete information) to make sure all models are tested fairly. We did this because our main goal is to evaluate our proposed methods compared to other baselines methods, so training model consistency and fairness across all methods was equally important. In figure 3.3 (a) and (b) respectively, we are providing accuracy and loss graphs of the model when it was trained with complete information, to understand the model effectiveness at apex line.

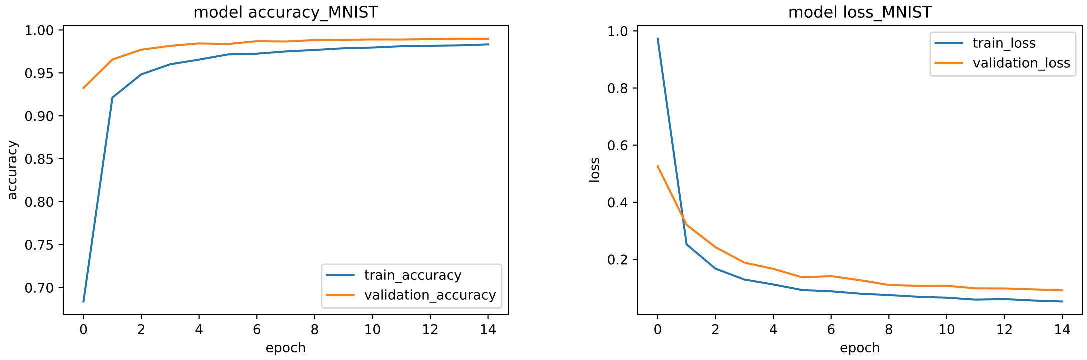

Figure 3.3 (a) and (b): Training model accuracy and loss when trained with complete data, on MNIST dataset

From the above figure 3.3 (a) and (b), we can observe that the training and validation curve looked quite similar after first 10 epochs. Looking at the loss graph for both seen (training) and unseen (validation) data, we can say the model was not over-fitting or under-fitting. This is because the validation loss was still higher than the training loss.

3.7.2 For CIFAR Dataset

We used a different structured multi layered convolutional network for training the model with CIFAR dataset. CIFAR images have more pixels per image than MNIST dataset and they are also in RGB format due to being color images. Hence we used a separate keras sequential model to handle these images.

Our CNN model for CIFAR data set consisted of 4 layers of convolutional neural network. We used one maxpool and one dropout(0.25) layer each after every two layers of convolutional network. Then we use a flatten layer before implementing dense layers and finally wrap up our model with a softmax activation function.

We used Adam optimizer for CIFAR dataset as well. Our model was run with a batch size of 64 and epoch of 30. Training and validation data had a split of 80-20. In figure 3.4 (a) and (b) respectively, we are providing accuracy and loss graphs of the model on CIFAR 10 dataset, when it was trained with complete information.

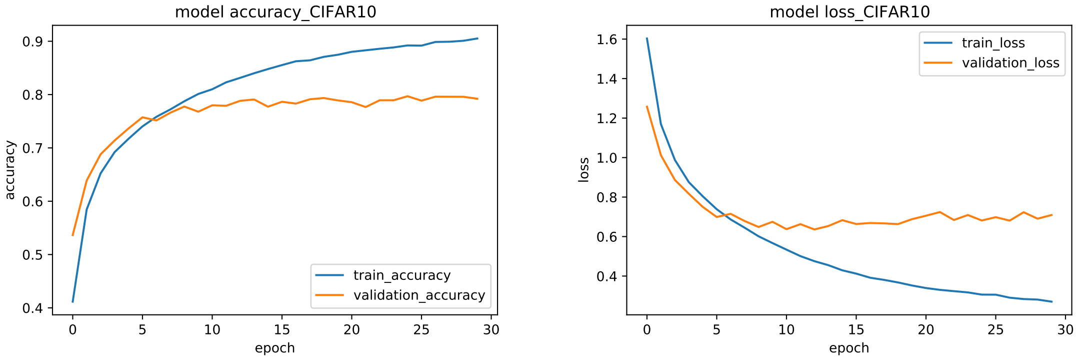

Figure 3.4 (a) and (b): Training model accuracy and loss when trained with complete data, on CIFAR 10 dataset

It ca be observed from the above figure 3.4 (a) and (b), that the training loss was decreasing at an exponential rate, hence accuracy was increasing. While validation loss stabilized only after 10 to 15 epochs. The validation loss was higher than training loss, which tells us that our model for CIFAR dataset was not over-fitting.

3.8 Evaluation Metrics

We used some other standard Machine Learning metrics such as precision, recall and f1-score to measure the effectiveness of our proposed models. We are describing the role of each metrics at a high level below.

Accuracy. Accuracy tells us the number of correctly predicted classes out of total predictions made. This is the most widely used metric. Sometimes accuracy as a metrics can be quite deceiving. It doesn't always reflect the true effectiveness of a machine learning model if data set is not well balanced[16].

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Precision We use Precision metrics to check how good a model is determining the true positive class, out of all positive classification. If Precision is high, then chances of false positive will be low.

Precision = $\frac{TP}{TP+FP}$

Recall Recall metrics is used to find out how well a model performs while determining true positive values against total actual positive values.

Recall = $\frac{TP}{TP+FN}$

F1 Score F1 score metrics is a good measure to balance out results between Recall and Precision. It is useful if class distribution is imbalanced.

F1 = $\frac{2*Precision*Recall}{Precision+Recall}$

3.9 Summary

In this chapter, we discussed the overall design and implementation process of our proposed neighborhood methods to reconstruct data and labels for further training if our model. We also discussed our training model effectiveness with visualisation technique. A high level overview of the evaluation metrics used are also discussed in this chapter.

Chapter 4

Experiments

In this chapter, we will discuss different experiment strategies that we followed to evaluate our model with existing baseline and apex line models. We will compare the models using tables and visualisation and also analyse the outcomes of the testing.

4.1 Experiment setup

For the purpose of experimenting our two models, we tried to compare them with baseline performance. For baseline estimation, first we implemented standard K-Means algorithm on actual data using K number of cluster partitions and then considered K number of new training points using each cluster centroid vector. We followed this strategy for different K values (ranging 50 to 1000) and across all three of our test data sets to determine K-means baseline performance. The only drawback we experienced in this naive K-means baseline testing was that the number of training instances got reduced. So model did not have enough data to train on.

Our second experiment strategy was to evaluate our ϵ Neighbourhood Model at different ϵ value ranging from 0.01 to 1. Our hypothesis for the testing was:

Hypothesis: The greater the value of ϵ will be, the more noisy vectors will be generated. Hence model training quality and accuracy of the prediction will keep decreasing.

Since we will be dealing with noisy and uncertain training vector data, we do not expect the model to perform better than any apex line model where the model is trained on individual instances with complete information. But anything close to that benchmark score is considered a promising outcome when model is trained on aggregated data.

4.1.1 Strategy 1: Comparison with Baseline

Our baseline comparisons between K-Means algorithm with centroid information and two of our proposed model at a fixed value of K, shown below. For baseline calculation, we first implemented K-Means algorithm on our actual data. We extracted the cluster centroid vectors or mean information from all K cluster instances and inferred an appropriate label for each of those vectors. The most appropriate strategy to achieve this was to calculate the most common class inside a cluster group and label that specific cluster centroid vector with that most commonly occurred class. In next step, we trained our model with these new K number of centroid vectors and the inferred labels. We then tried to predict on individual instances using this model which only used mean data. We wanted to verify how much information can be extracted from each centroid vector after applying K-Means aggregation method. Since our goal was to make predictions from aggregated data, we aimed to improve this standard K-Means clustering performance by extracting more information from these clustered groups.

MNIST Dataset				
K Value	K Means Baseline	Gaussian Neighbourhood Method	$\epsilon(0.01)$ Neighbourhood Method	Complete Information (No Clustering Applied
50	0.2645	0.8011	0.7745	0.9914
100	0.4910	0.8719	0.8774	
200	0.7558	0.9188	0.9241	
500	0.8546	0.9365	0.9362	
1000	0.9116	0.9582	0.961	

Table 4.1 : Comparison of our models with Naive Baseline on MNIST Dataset

CIFAR-10 Dataset				
K Value	K Means Baseline	Gaussian Neighbourhood Method	$\epsilon(0.01)$ Neighbourhood Method	Complete Information (No Clustering Applied
50	0.2506	0.3154	0.2645	0.7833
100	0.2523	0.3372	0.3089	
200	0.2908	0.3699	0.2841	
500	0.3412	0.3505	0.3121	
1000	0.315	0.3953	0.3453	

Table 4.2 : Comparison of our models with Naive Baseline on CIFAR-10 dataset

CIFAR-100 Dataset				
K Value	K Means Baseline	Gaussian Neighbourhood Method	$\epsilon(0.01)$ Neighbourhood Method	Complete Information (No Clustering Applied
50	0.0294	0.0718	0.0667	0.4414
100	0.0268	0.0919	0.0928	
200	0.0531	0.1039	0.0979	
500	0.0876	0.1175	0.1106	
1000	.074	0.13	0.1077	

Table 4.3 : Comparison of our models with Naive Baseline on CIFAR-100 dataset

4.1.2 Strategy 2: Experiments with ϵ Neighborhood Model

Our second strategy was to experiment classification performance done on aggregated data, if it's reconstructed using an user given ϵ value. For testing purpose we used ϵ values ranging from 0.01 to 1.0 and used K=1000 for all the tests. As mentioned earlier in this chapter, we were expecting lower accuracy for higher ϵ values. We tested this same strategy on all three of our test data sets and the result is shown below:

MNIST Dataset	
ϵ Value	Accuracy
0.01	0.961
0.05	0.9609
0.1	0.9557
1.0	0.9237

Table 4.4 : Model performance at different values of ϵ on MNIST Dataset, when K=1000

CIFAR-10 Dataset	
ϵ Value	Accuracy
0.01	0.3453
0.05	0.332
0.1	0.3669
1.0	0.3398

Table 4.5 : Model performance at different values of ϵ on CIFAR-10 Dataset, when K=1000

CIFAR-100 Dataset	
ϵ Value	Accuracy
0.01	0.1077
0.05	0.1113
0.1	0.1077
1.0	0.0919

Table 4.6 : Model performance at different values of ϵ on CIFAR-100 Dataset, when K=1000

4.1.3 Metrics visualisation of ϵ Neighborhood model with MNIST data set

Since we used various standard Machine Learning metrics in our model, we summarized the results for different experiments and incorporated them in a comparison table for each of these metrics. By using visualisation technique, it is easy to compare model performance at different K and ϵ values focused on a particular performance metric.

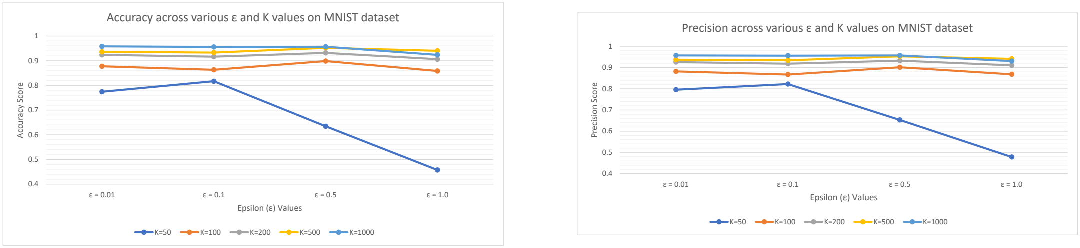

Figure 4.7 (a) and (b): Accuracy and Precision across various ϵ and K values on MNIST dataset

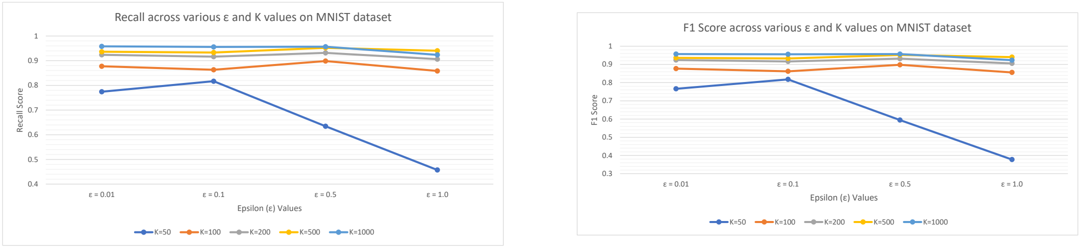

Figure 4.7 (c) and (d): Recall and F1 score across various ϵ and K values on MNIST dataset

4.2 Evaluation

We will divide our evaluation of models in two different parts like we did for experiments. We will discuss both of our model performance with respect to baseline and will elaborate the outcome of experiments. Secondly, we will dig deeper in to our proposed ϵ neighbourhood method to analyze how it performed across three different data sets and why the performance sharply dropped in classification task of color images.

4.2.1 Evaluating proposed models classification performance with baseline 'K-Means'

MNIST : From the Table 4.1, we can see the performance of both of our models beating the standard K-Means baseline at different values of K(Number of clusters the data was aggregated on). For reference, if we look at K=1000, both Gaussian and ϵ Neighbourhood models performed at par. (achieved around 96 percent accuracy). Our standard K-Means baseline model achieves around 91 percent accuracy here. We can explain that our model worked better in this scenario because we have used more information from each clusters, rather than relying only on Mean values. Another reason was we had more training data to train our model in both our proposed models, which is equal to the original training set.

One more interesting observation we have made from experiments is that the performance of all three models (baseline and both of our proposed models) had dropped consistently as we tested with smaller K values. For example, our model's accuracy score dropped by nearly 2 percent when we aggregated all our training data with 500 clusters, instead of 1000. We didn't use any Elbow-Curve method in our code to determine the optimal number of clusters, as it was not our point of interest. We wanted to compare all three models at any given value of K to find out the best performing one.

CIFAR Data sets : Table 4.2 and 4.3 describes our models performance on CIFAR-10 and CIFAR-100 image data sets respectively along with K-Means baseline performance. We used Keras as well as a 4 layered Convolutional Neural Network for image classification problem in these data sets, CIFAR images being color images with image having (32,32,3) pixels, the training had taken a longer duration and accuracy was relatively lower than MNIST dataset. For example, Our K-Means baseline accuracy was 31.5 percent for K=1000. Our proposed ϵ Neighborhood (0.01) model achieved 34.5 percent at same number of K value, while Gaussian Neighborhood model performed much well (around 39.5 percent).

Also, CIFAR-100 datasets have higher number of classes, which is 100. So, for this data set all model performances including baseline was significantly lower than CIFAR-10 dataset. We used simple CNN based models for testing. The results could be significantly better with other state of the art machine learning models such as Residual Neural Network or ResNet.

4.2.2 Evaluating ϵ Neighborhood model classification performance for different ϵ values

We introduced the ϵ Neighborhood Sampling model which sampled vectors based on a certain distance ϵ from each cluster centroids. The further we moved away from origin to regenerate our pseudo vectors for training, more blurry the images has become. The hypothesis we presented at chapter 4.1, has been tested in these experiments. For every data set, we got our best performance at ϵ=0.01, which was the lowest ϵ value that we tested with. The performance kept decreasing slowly as we further moved away from centroid vectors. CIFAR color image data sets also showed same tendency. From ϵ value 0.01 to 1.0, the classification accuracy dropped around 3 percent for MNIST data set, nearly 1 percent for CIFAR-10 data set and around 1.5 percent for CIFAR 100 data set.

Below we are presenting some MNIST data set images post implementation of ϵ Neighborhood sampling process. It was evident from the images that images developed a lot of noise as the ϵ value kept getting bigger and it also established our hypothesis to be true. We will also show the images that we have received post implementation of Gaussian Distribution method.

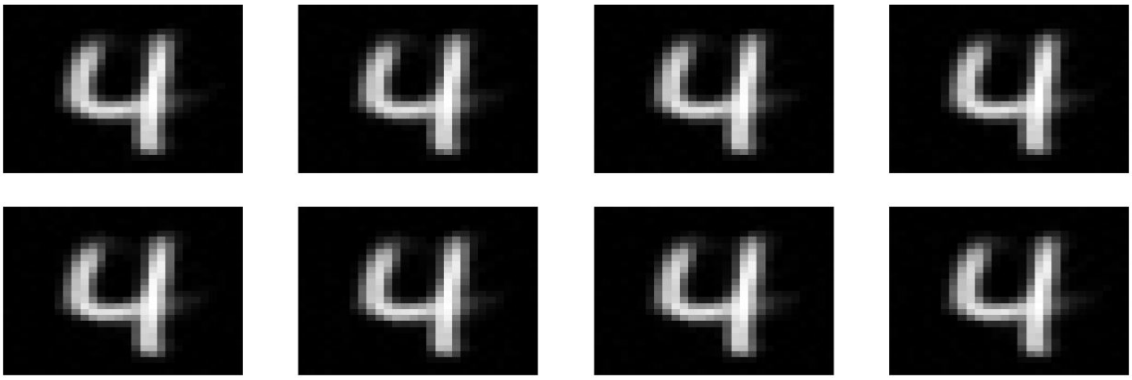

Figure 4.11: Processed images after implementing ϵ (0.01) Neighborhood sampling model on MNIST dataset

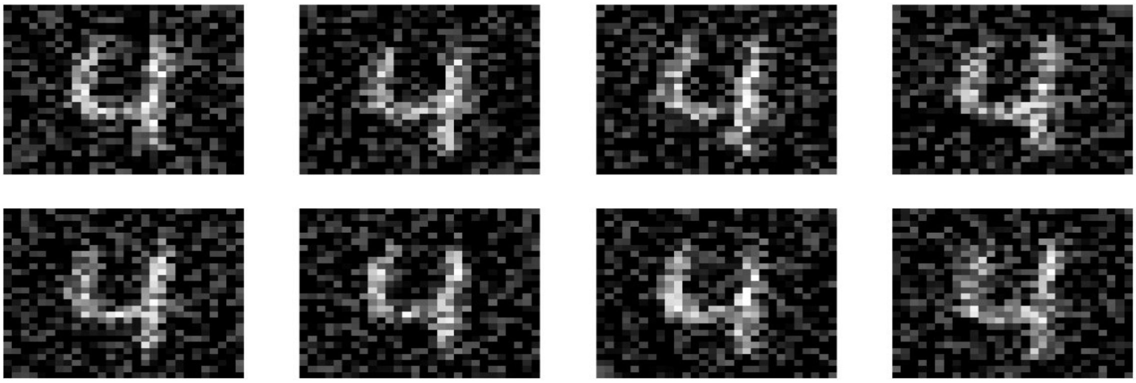

Figure 4.12: Processed images after implementing ϵ (0.5) Neighborhood sampling model on MNIST dataset

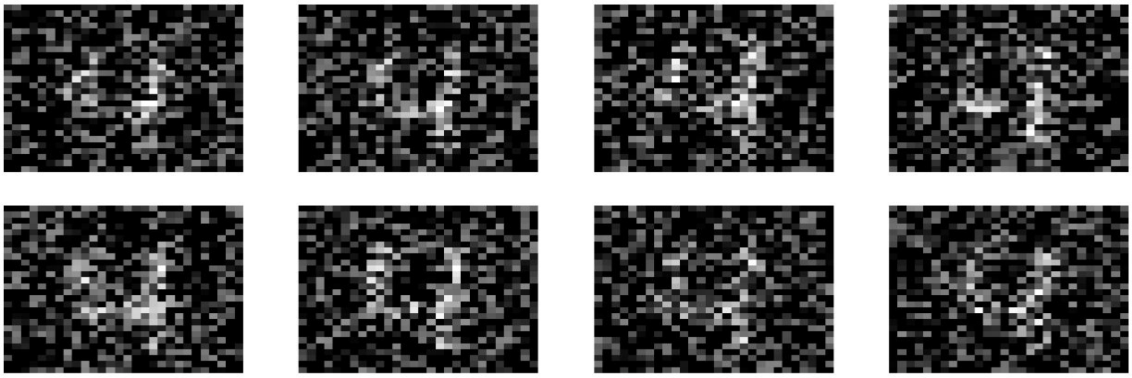

Figure 4.13: Processed images after implementing ϵ (1.0) Neighborhood sampling model on MNIST dataset

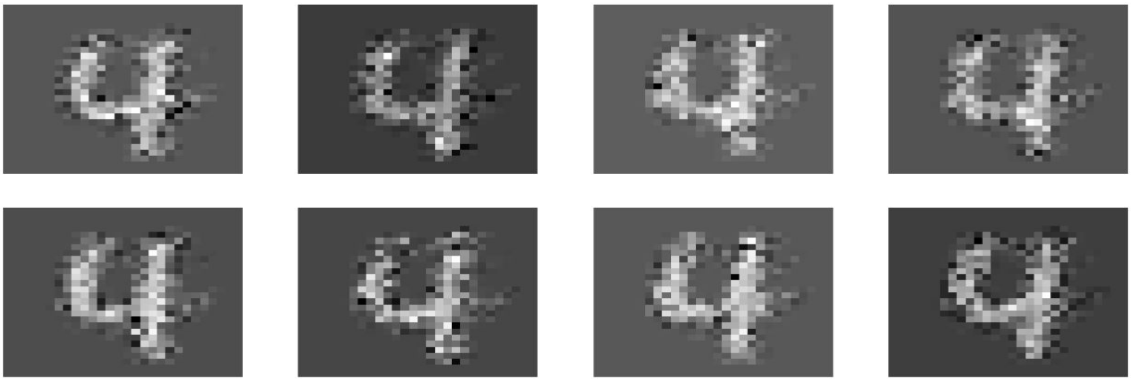

Figure 4.14: Processed images after implementing Gaussian Neighborhood sampling model on MNIST dataset

One important observation we have made here is that, despite the images looks significantly blurry at ϵ 0.5 or 1.0, classification accuracy didn't dropped hugely for any of these data set. We believe it happened because the model was still able learn and distinguish the outlines of each images from all the noises, mostly due to feature extraction capability of CNN even at smallest blocks.

4.3 Summary

In this chapter, we analysed and evaluated our proposed methods effectiveness against baseline methods. We showed that our model showed promising prediction capability and accuracy scores were very close when compared with learning with complete data. Our model even outperforms naive K-means baseline in all of our data sets.

Chapter 5

Conclusion

5.1 Discussions

The main aim of our project was to propose and develop models capable of making classification predictions on aggregated data. We focused on image classification tasks for this project and we believe both models proposed by us showed some great promise. While working on this project, we explored some novel areas of machine learning and sampling processes. We were able to answer our first research question through extracting mean and standard deviation across all dimensions of all cluster members, which proved to be an effective information set.

We proposed two novel neighbourhood sampling techniques to reconstruct training elements, which were pseudo in nature. Both Gaussian Sampling and ϵ Neighborhood sampling method generated closely resembled vectors. We showed that both models performed on par at various K cluster numbers. This effectively answered our second research question.

We then shifted our focus to generating label by sampling method, using prior probabilities. As we had access to true labels, we were able to build an effective and scientific label sampling method, which fitted our training vectors well. This method worked really well along with the vector generation model of ours which we can see by overall performance of our model, This was our third research question to answer.

We conducted our experiment using a complete dataset. We first employed an aggregation strategy and then restricted ourselves from accessing true vectors at any point of entire experiment. In real life scenario, this model can fit into different application areas which can be simulated in lab based environment. The model would be equally effective under those circumstances. This answers our last research question.

Overall, predicting on individual instances just by learning from aggregated level features was a challenging task. We tried to produce methods which will perform decently in aggregated image classification problems and may also work on other data sets with some fine tuning.

5.2 Future Scope

In this thesis, we presented two neighborhood sampling strategies which showed great promise. However, We believe that, despite being a novel area of study and a new interest in machine learning studies, commercial usage of learning from aggregated data is going to be widely adopted in different aspects of society. We are still not aware of any study or model where aggregated level information has achieved almost identical or beating the benchmark performance of individual instance level learning. But we still thrive to get closer to the benchmark score. As part of the future scope of our project, there is still room for improving these proposed strategies to make even better predictions at individual level. We might be also interested to explore the area where ground truth data labels are also unavailable to us. This scenario will require more extensive study, better aggregation technique and newer information extraction strategies from clustered group to achieve a decent outcome. It would be interesting to delve deeper into this unexplored horizon.

Appendix A

Appendix

A.1 Useful Links

MNIST Dataset

http://yann.lecun.com/exdb/mnist/

CIFAR Datasets

https://www.cs.toronto.edu/ kriz/cifar.html

A.2 Our Code Location

Main source code

https://github.com/tamit90/Generating-Individual-Predictions-from-Aggregated-Data

Bibliography

[1] Boris Babenko. Multiple instance learning: algorithms and applications. *View Article PubMed/NCBI Google Scholar*, pages 1–19, 2008.

[2] Han Bao, Gang Niu, and Masashi Sugiyama. Classification from pairwise similarity and unlabeled data. In *International Conference on Machine Learning*, pages 452–461. PMLR, 2018.

[3] Chandan Biswas, Debasis Ganguly, Dwaipayan Roy, and Ujjwal Bhattacharya. Privacy preserving approximate k-means clustering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1321–1330, 2019.

[4] George Casella and Roger L Berger. *Statistical inference*. Cengage Learning, 2021.

[5] Sofia Dias, Alex J Sutton, AE Ades, and Nicky J Welton. Evidence synthesis for decision making 2: a generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Medical Decision Making*, 33(5):607–617, 2013.

[6] X Yu Felix, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. Psvm for learning with label proportions. 2013.

[7] Debasis Ganguly. A fast partitional clustering algorithm based on nearest neighbours heuristics. *Pattern Recognition Letters*, 112:198–204, 2018.

[8] Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*, 2019.

[9] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[10] Jason Kendall. Designing a research project: randomised controlled trials and their principles. *Emergency medicine journal: EMJ*, 20(2):164, 2003.

[11] Xi-Lin Li. A multiclass multiple instance learning method with exact likelihood. *arXiv preprint arXiv:1811.12346*, 2018.

[12] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

[13] Giorgio Patrini, Richard Nock, Paul Rivera, and Tiberio Caetano. (almost) no label no cry. *Advances in Neural Information Processing Systems*, 27:190–198, 2014.

[14] Shahin Pourbahrami, Leyli Mohammad Khanli, and Sohrab Azimpour. A novel and efficient data point neighborhood construction algorithm based on apollonius circle. *Expert Systems with Applications*, 115:57–67, 2019.

[15] Hua Shen, Mingwu Zhang, and Jian Shen. Efficient privacy-preserving cube-data aggregation scheme for smart grids. *IEEE Transactions on Information Forensics and Security*, 12(6):1369–1381, 2017.

[16] Koo Ping Shung. Accuracy, Precision, Recall or F1? `https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9`, 2018. [Online; accessed 08-December-2021].

[17] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114, 2004.

[18] Yivan Zhang, Nontawat Charoenphakdee, Zhenguo Wu, and Masashi Sugiyama. Learning from aggregate observations. *arXiv preprint arXiv:2004.06316*, 2020.