An Empirical Evaluation of Graph-based Semi-Supervised Learning Algorithms for Data Labeling.

Teodor Fredriksson, Jan Bosch
Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden
{teodorf, jan.bosch}@chalmers.se

Helena Holmström Olsson
Department of Computer Science and Media Technology
Malmö University
Malmö, Sweden
helena.holmström.olsson@mau.se

David Issa Mattos
Volvo Cars
Gothenburg, Sweden
david.mattos@volvocars.com

***Abstract*—The lack of labeled data is a major problem in both research and industrial settings since obtaining labels is often an expensive and time-consuming activity. In the past years, several machine learning algorithms were developed to assist and perform automated labeling in partially labeled datasets. While many of these algorithms are available in open-source packages, there is a lack of research that investigates how these algorithms compare to each other for different datasets of differet datatype and with different amounts of available labels. To address this problem, this paper empirically evaluates and compares thirteen graph-based semi-supervised learning algorithms for automated labeling in terms of their accuracy. We investigate how these algorithms using 24 different and well-known datasets with three different types of data, images, texts, and numerical values. We evaluate these algorithms under five different experimental conditions, with 10%, 25% 50%, 75%, and 90% of available labels in the dataset. Each algorithm, in each dataset for each experimental condition, is evaluated independently ten times with different random seeds.**

The results are analyzed and the algorithms are compared utilizing the Bayesian Bradley-Terry model and the Binomial model. The results indicte that Poisson MBO, Poisson MBO (old), sparse label propagation and Balanced Poisson are four best algorithms that achieve the highest accuracy. Furthermore, the three top algorithms that always increase the probability of achieving an accuracy higher than 90% are Poisson MBO, Balanced Poisson MBO and Sparse label propagation. These results help machine learning practitioners in choosing optimal machine learning algorithms to label their data.

***Index Terms*—Data Labeling, Automatic Labeling, Graph-based algorithms, Semi-Supervised learning, Bayesian Data Analysis, Bradley-Terry model, Inverse Logit Binomial model.**

I. Introduction

For the past decade, machine learning has beocome a data scientist's best tool and many companies have implemented or are in the process of implementing machine learning. Supervised learning is the most commonly used machine learning paradigm. However, there are problems with supervised learning and machine learning in general. The first problem is that machine learning requires often large amounts of data. Secondly, supervised learning needs labels in the data [1]. The first issue is not necessarily a problem for companies since they often do have data of sufficient quantity. The downside is that datasets are often incompletely labeled, which brings us to issue number two. According to the our experience with the industry, datasets are often missing labels partially or completely. In a case study performed with industry, several labeling issues were found [2]. Companies have to spend significant amounts of money and time into labeling techniques such as crowdsourcing or in-house labeling. This is time and money they would rather not spend and they prefer to use more automated approaches with as little human intervention as possible [3], [4]. A recent systematic literature review was conducted to see what type of machine learning algorithms exist to make the labeling easier [5]. In [5], the authors focused on investigating semi-supervised learning and active learning. From the results the authors concluded which semi-supervised learning algorithms were the most popular and which datatypes they can be used on. However, even if there has been work done on semi-supervised learning, these learning paradigms are still new for many companies and consequentially seldom used. Thus, data scientists have to spend time on testing and deciding which labeling algorithm is best suited for their specific situation [6]. This paper is an extension of a simulation study [7] where seven semi-supervised learning algorithms were evaluated in terms of *Performance* how accurate the algorithm is,and *Effort*, how much manual work has to be done from the data scientist. The previous simulation study evaluated seven semi-supervised learning algorithms on twelve datasets of different types, namely numerical, text and image data. A Bayesian Bradley Terry model was implemented to rank the algorithms

according to accuracy.

The contribution of this paper is to extend the taxonomy of semi-supervised algorithms by studying 13 different graph-based algorithms, study 12 additional datasets for a total of 24 datasets. The new empirical evaluation of the taxonomy's algorithms will add a third dimension. *Datatype*, Do the algorithms perform better on datasets of a different datatype? Furthermore, this paper will address whether using a certain algorithm will increase the probability of achieving a certain accuracy. This is analyzed by implementing the Inverse Logit Binomial model, a version of the binomial generalized linear model to include priors and random effects [8].

The remainder of this paper is organized as follows. In the upcoming section we provide an overview about semi-supervised and active learning algorithms and how they work. In section III we describe the method we used in our study, how we preformed the simulations, what datasets and source code we used, and what kind of metrics we used to evaluate performance, effort and applicability. In section IV we provide the results from the simulation study and finally, we will interpret the results and conclude the paper in section V.

II. Background

In this section, we provide an overview of how active and semi-supervised learning work.

A. Descriptive Statistics

Let X be a sample with n observations, $x_1, x_2, ..., x_n$. We define the descriptive statistics of X.

The **sample mean** $\overline{x}$ is defined as the arithmetic average of the sample

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

The following measurements describes the spread in of the sample

The **sample variance** measures the average value of the squares of the difference between the values in the sample and the sample mean,

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2.$$

The **sample standard deviation** measures the spread of the observerations in the sample.

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}.$$

Let $0 < \alpha < 1$, we say that q is the $\alpha\%$ quantile if $100\alpha\%$ of the observations in the sample are less than it.

If the samples are ordered from smallest to largest, If the n is odd, then the **sample median** is defined as the value in position $(n+1)/2$, if n is even, then the median is defined as the average of the values in position $n/2$ and $n/2+1$ The sample median is also known as the 50% quantile, because its the observation in the observation in the sample that splits the data in two groups of the same size.

B. Semi-Supervised Learning

Semi-supervised machine learning is a class of machine learning algorithms that utilizes both labeled and unlabeled data. Semi-supervised algorithms are then trained on both the unlabeled and the labeled data and in some cases it even outperforms supervised classifiers. For more information on semi-supervised learning we refer the reader to [9].

According to [5], one of the most popular semi-supervised learning algorithms are the *graph-based* algorithms. The idea of these algorithms is to build a graph from the training data. These graphs contains both labeled and unlabeled instances. Let each pair $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ represent each vertex and its corresponding label. Let the edge weight w_{ij} represent the weight of the edge between vertex i and vertex j. The larger w_{ij} becomes the more similar are the labels of both vertices. The question is then how to compute the weight w_{ij}.

C. The Bradley Terry model

The Bradley-Terry model [10], [11] is one of the most commonly used models when it comes to analysis of paired comparison data between two objects i and j for $j > i = 1, ..., n$. The comparison is done by a judge (subject), and the total number of possible paired comparisons is equal to $N = n(n-1)/2$. Let $y = (y_{1,2}, ..., y_{n-1,n})$ be the vector of outcomes of all paired comparisons. Each outcome $y_{i,j}$ in y are binary variables, either taking value 1 with probability $p_{i,j}$ and value 0 with probability $1 - p_{i,j}$. Here $p_{i,j}$ is the probability that i beats j. This means than the outcomes $y_{i,j}$ are Bernoulli distributed, in other words:

$$y_{i,j} \sim \text{Bernoulli}(p_{i,j}).$$

Furthermore, we assume that the outcomes are independent. Let $\mu_i \in \mathbb{R}, i = 1, 2, ..., n$ be the latent variable representing the "strength" of the algorithm being compared. Traditional ranking models such as the Bradley-Terry model assumes that the probabilities $p_{i,j}$ are dependent on the difference $\mu_i - \mu_j$ by some cumulative distribution function F:

$$P(i \text{ over } j) = F(\mu_i - \mu_j).$$

In the Bradley-Terry models we use the logistic cumulative distribution function, results in logistic regression [10]:

$$P(i \text{ over } j) = \text{logit}^{-1}(\mu_i - \mu_j).$$

By estimating the strength variable $\mu = (\mu_1, \mu_2, ..., \mu_n)$, we can rank the algorithms by calculating the probability of one algorithm beating the other.

The Bayesian Bradley-Terry model has several advantages over other types of model to compare rank data. It can estimate the relative strength of each algorithm to provide effect sizes measures in terms of probability of one algorithm beating the other, provide uncertainty estimation on the ranking and is able to be extended to include algorithm specific predictors and random effects to compensate for clustering and repeated measures.

D. Logit GLMM for model for binomial samples

The Generalized Linear Mixed Model for Binomial samples calculates the probability of success (an algorithm yields a specific accuracy). Let y_i be an observation, with

$$y_i = \begin{cases} 1 & \text{if success} \\ 0 & \text{if failure} \end{cases}$$

i.e, $y_i \sim \text{Bernoulli}(p)$.

For n samples $y_1, y_2, ..., y_n$, the sum of all outcomes will be binomial distributed,

$$y = \sum_{i=1}^{n} y_i \sim \text{Binomial}(n, p).$$

Hence, we will use the the binomial distribution as likelihood. The probability of success will be be modeled as

$$p = \text{logit}(P(y = 1)) = a + bx + u, \quad u \sim \text{Normal}(0, \sigma^2)$$

where a is the fixed effect, b is the log-odds ratio and u is the random effect.

The parameters have the following distributions:

$$\begin{aligned} a_{alg,i} &\sim N(0, 5) \\ b_{noise,i} &\sim N(0, 5) \\ a_{bm,j} &\sim N(0, s) \\ s &\sim \text{Exponential}(0.1). \end{aligned}$$

III. Research Method

In this section we present the details about the datasets that we used for our simulations, the experimental conditions and the algorithms we used.

The goal of this study is to show in detail how machine learning algorithms can be used to help with data labeling and to provide an in-depth comparison on how these different algorithms perform on different types of data. To achieve this, we performed an empirical evaluation of thirteen semi-supervised learning algorithms and evaluated them on twenty eight datasets under different conditions.

The main research questions that we use to evaluate the machine learning algorithms are the following.

- **RQ1:** *How can we rank different graph-based semi-supervised learning algorithms in terms of accuracy?*
- **RQ2:** *Do the algorithms rank differently according to a specific datatype?*
- **RQ3:** *How do we rank the accuracy of algorithms towards the number of manually labeled instances available?*
- **RQ4-a:** *What is the probability of of each algorithm yielding an accuracy* $\varepsilon \geq 0.9$
- **RQ4-b:** *What is the impact of noise in the probability of success of each algorithm at accuracy* $\varepsilon \geq 0.9$

To compare accuracy, each algorithm uses a fixed percentage of labeled and unlabeled instances. The goal is to predict the unlabeled instances. The semi-supervised learning algorithms uses the labeled instances to predict the labels of the unlabeled data

A. Simulations

As recognized in [5] co-training/multi-view learning is the most popular algorithms but is based on the assumption than we can observe an instance from multiple views. Graph-based algorithms are the second most common type of semi-supervised learning algorithm.

Thirteen different graph-based algorithms were implemented in this study. All algorithms were implemented in python, using the GraphLearning package. The algorithms in the GraphLearning package were all implemented with $w_{ij} = \exp(-4|x_i - x_j|^2/d_k(x_i)^2)$, $i \neq j$, $k = 10$. The weight matrix is made symmetric by $w = w^T + w$. For Poisson learning algorithms we set $w_{ii} = 0$ for all i. This does not affect the solution of the Poisson learning algorithm, but increases the convergence speed.

- **Centered kernel:**
- **Laplace Learning:**
- **Mean Shifted Laplace:**
- **Centered kernel method:**
- **Poisson Learning**
- **Poisson Learning, alternate version**
- **Balanced Poisson Learning**
- **Poisson MBO with vilume constraints:**
- **Balance Poisson MBO:**
- **Poisson learning with volume constraints:**
- **Random walk** is implemented with $\epsilon = 0.05$
- **Sparse Label Propagation:**
- **Weighted non-local Laplacian:**

We choose 24 enchmark datasets to be used in our experiments. Four numerical datasets, nine text datasets and ten image datasets. Due to the size of some datasets and to limited time and computational resources required we had to reduce the number of images used in our experiments. However, we made sure we used the same ratio for the classes to get a fair estimate.

- **Image data:**
  - **Caltech-256:** The Caltech-256 dataset contains 30607 images divided into 256 categories [12]. The dataset was downloaded from Kaggle [13] and the original dataset is located at [14].
  - **Cifar-10:** This dataset originally contains 60000 32x32 images that can be divided into ten classes, airplane, automobile, bird, car, deer, dog, frog, horse, ship and truck [15]
  - **Corel:** This dataset was taken from the COREL Datahase for Content based Image Retreival [16]. We choose ten classes of images: beaches, bus, dinosaurs, elephants, flowers, foods, horses, monuments, mountains and snow, people and villages in Africa. Each lass class contains 90 images. The dataset was downloaded from Kaggle [17]
  - **Digits:** This dataset contains 1797 samples of 8x8 images containing one digit each. There are ten

classes that represent which digits is contained in each image [18].

- **FashionMNIST:** The FashionMNIST [19] [20] dataset contains article images from Zalando. The dataset contains 60000 insrances, each instance is a 24×24 greyscale image. There are ten labels: T-shirt/top, Trouser, Pullover, Coat, Sandal, Shirt and Sneaker, Bag, and Ankle boot.
- **MNIST:** MNIST is a database for handwritten digits recognition [21]. It contains 60000 instances and is a subset of the larger NIST dataset.
- **MiniImageNet:** This dataset is a smaller version of the ImageNet dataset [22]. The dataset is constructed according to a hierachy provided by WordNet, and is used for object detection. The number of instances included is above fourteen million and the number of categories 20000. Mini ImageNet [23] is a smaller version of ImageNet, it contains 100 categories containing 600 images each.
- **TieredImageNet:** Like mini ImageNet, the tiered ImageNet [24] is a smaller version of ImageNet. It contains 608 categoris for a total of 779165 instances.

- **Text data:**
  - **20news:** This dataset contains 18846 instances divided into 20 classes that describes the 20 different types of news [25].
  - **Amazon:** This dataset contains reviews from Amazon. Originally it contains two features and 3 million instances but we have selected to only use 5000. The labels represents the review scores going from 1 to 5.
  - **DBworld:** This dataset contains 64 instances of emails collected from the DBWorld newsletter. The dataset was already pre-processed using a binary bag-of-words representation and stopword removal [26].
  - **Fake and true news:** This is a dataset containing 44594 instances and 5 features. The features are, ”title”, the title of the news article. ”text”, the text of the article, ”subject” the article subject and a column representing the label classes, ”False” or ”Truthful”. From this dataset we only extracted the ”text” column and used it as a features to predict the labels [27], [28].
  - **IMDB:** The IMDB dataset [29] contains 50000 movie reviews and their sentiment: postive or negative.
  - **Ohsumed:** This dataset is a subset of MEDLINE, the U.S National Library of Medicine premier bibliographic database consisting of references to journal articles in life science related to biomedicine. The original Ohsumed dataset consist of 23 medical subject headings categories. In this study we have selected the twelve most common categories for classification [30].
  - **Reuters:** Reuters is a benchmark dataset for document classification. It has a total of 90 classes and over 10000 instances [31].
  - **Spambase:** This dataset contains 4601 instances of emails labeled as ”spam” and ”non-spam”. The goal of the dataset is to classify the emails as either ”spam” or ”non-spam”.

- **Numerical data**
  - **German:** This datasets contains 1000 instances with 20 attributes and was used to is used to predict the credit score of germans based on demographic information as well as their financial information [32].
  - **Ionosphere:** The Ionosphere dataset [33] was collected 16 high-frequency antennas system in Goos Bay, Labrador. The purpose of the dataset is to classify wheter a radar return is ”good” or ”bad”. The dataset contains 351 instances and 34 features.
  - **Iris:** This dataset is a classic example for multi-class classification. It contains 150 instances across three classes [34] .
  - **MUSK:** This dataset contains 476 instances and is a subset of a larger MUSK dataset. The dataset describes molecules that are classified to be ”musk” or ”non-musks”. The label depends on 166 attributes that describes the shape of the molecule [35].
  - **PIMA:** The PIMA dataset [36] originates form the National Institute of Diabetes and Digestive and Kindey Diseases. Every participant in this study were 21 year old females of the Pima Indian heritage. The purpose of the dataset is to predict whether an individual haa diabetes or not. The features contains information such as MBI, insulin level, age and number of pregnancies.
  - **Sonar:** The Sonar dataset [37] contains 208 instances of sonar signals that bounces of a metal cylinder or rocks. The purpose of the dataset is to determine if the singal bounces of a metal cyliner of a rock. The data contains 208 features.
  - **WDBC:** The Breast Cancer Wisconsin (Diagnostics) [38], [39] dataset contains features from images of breast mass. The purpose of the dataset is to classify the cancer as malignant or benign. There are 569 instances and 32 features in total.
  - **Wine:** The wine dataset also a classic example of multi-class classification. It contains 178 instances across three classes [40].

Table I
SUMMARY TABLE FOR THE DATASETS.

Datatype	Dataset
Image	Caltech-256
	Cifar-10
	Coil-100
	COREL
	Digits
	FashionMNIST
	MNIST
	MiniImageNet
	TieredImageNet
Text	20news
	20news
	Amazon
	DBworld
	Fake and true news
	IMDB
	Ohsumed
	Reuters
	Spambase
Numeric	Ionosphere
	Iris
	German
	MUSK
	PIMA
	Sonar
	WDBC
	Wine

The training data is then prepared by splitting it into two parts, unlabeled and labeled. To assess the "initial manual effort", i.e. the initial amount of labels a practitioner requires before running a machine learning assisted labeling algorithm. The manual effort conditions are 10%, 25%, 50%, 75% and 90% of the labels.

For each dataset, each algorithm was run on all initial manual effort conditions (10%, 25%, 50%, 75% and the 90%) a total of ten times, each using a different random seed and stored in one .csv file. These .csv files contained

To answer the RQs, we measured accuracy by

$$\varepsilon = \frac{100}{n-m} \max\left(\sum_{i=1}^{n} I(y_i = \widehat{y}_i) - m, 0\right),$$

where $I(x)$ is **indicator function** defined as

$$I(x) = \begin{cases} 1 \text{ if } x \text{ is true} \\ 0 \text{ otherwise} \end{cases},$$

n is the number of both labeled and unlabeled instances and m is the number of labeled instances.

B. Threats to Validity

In this study we have not considered if the number of each label class is balanced or not. This might affect semi-supervised learning algorithms. [41].

IV. RESULTS

In this section we present the results. Tables A.II, A.IV, A.III all contain the descriptive statistics of the accuracy for aggregated, datatype and manual effort respectively. From left to right, the columns contain, model name (Models), sample mean (Mean), sample standard deviation (SD), sample median (Median), 5% quantile (5%) and 95% quantile (95%). Tables I,A.IV,A.III illustrates the descriptive statistics in the form of boxplots.

A. Bradley-Terry

This data was then expanded into paired comparisons for the use in the Bradley-Terry model [11], [42]. In this model, y is a binary variable that indicates which algorithm beats the other:

$$\begin{aligned} y &\sim \text{Bernoulli}(p), \\ p &= \text{logit}^{-1}(\mu_{\text{algo1}} - \mu_{\text{algo0}}), \\ \mu_i &\sim \text{Normal}(0, 5). \end{aligned}$$

The Bradley-Terry model is used to analyze RQ1-RQ3 and the Binomial Logit model is used to analyze RQ4. The models are written in Stan [43], which implements the No U-turn Hamiltonian Monte Carlo sampler [44]. We utilize the following configurations: 4 chains, warm-up of 200 iterations and a total of 2000 iterations. The data transformation, tables, plots and assessing the convergence of the chains are conducted in R together with package `rstan` and the collection of packages `tidyverse`.

The prior distributions of the μ_i parameters are adjusted to be weakly-informative distributions. The presented model estimates the posterior distribution of the latent strength parameters μ_i. In turn, sampling and ranking over the posterior distribution of the strength parameters allows us to obtain a posterior distribution of the ranks.

Tables A.V, A.VI, A.VIII, A.VII, A.IX, A.X, A.XI, A.XII, A.XIII contains the results concerning the posterior ranks. From left to right, each columns contains model name (Models), median rank (Median) and the variance of the rank estimates (Variance of the Rank). Figures 2,3,4,5,8,9,10,11,12 shows the HPD intervals for the strength parameters for aggregated, datatypes and manual effort.

B. Binomial model

For the binomial model y represents the sum of all binary outcomes, it is the number if tries that were successful.

$$\begin{aligned} y &\sim \text{Binomial}(n, p) \\ \text{logit}(p) &= a_{alg,i} + b_{noise} x_{noise} + a_{bm,j} \\ a_{alg,i} &\sim \text{Normal}(0, 5^2) \\ b_{noise,i} &\sim \text{Normal}(0, 2) \\ a_{bm,j} &\sim \text{Normal}(0, s^2) \\ s &\sim \text{Exponential}(0.1) \end{aligned}$$

where a_{alg} is the fixed effect (mean effect [intercept] of each algorithm), $a_{bm,j}$ random effect of each benchmark dataset, b_{noise} is the influence of noise in each dataset and $x_{noise} = 3$ is the actual noise.

Like the Bradley-Terry model, the prior distributions of $a_{alg}, a_{bm}, b_{noise}$ and s set to be weakly informative. The binomial model estimates the posterior distributions of the model parameters $a_{alg}, a_{bm}, b_{noise}$ and s.

Tables A.XIV,A.XV,A.XVI,A.XVII,A.XVIII,A.XIX,A.XX A.XXI,A.XXII contains descriptive statistics of a_{alg}, b_{noise} and s. rom left to right the columns contain the of parameter name (Parameter), the mean of each estimate (Mean), the mean odds ratio of the parameters, n_{eff} and $\widehat{R}$. Figures 13,14,15,16,17,18,19,20,21 contains the HDPI intervals of a_{alg}, b_{noise} and s.

C. Aggregated results, RQ1

To answer RQ1 we combined all of the 364 data frames containing our results into one big data frame by stacking all the rows, we also added a column where we labeled each instance by which algorithm it resulted from. The columns containing the information regarding whether the algorithm was run using 10%, 25%, 50%, 75% or 90% of available labels and datatype were dropped, thus we could analyze the accuracy of every algorithm without regards to amount of available labels and datatype.

From left to right, the columns in Table A.II contains the sample mean, standard deviation, median as well as 5% and 95% quantiles for each algorithm. Figure 1 illustrates the descriptive statistics of Table A.II as a boxplot.

Based on the Bradley-Terry model described above, the parameter strength is computed for each algorithm. Figure 2 illustrates the distribution of the parameter strengths along with their High Posterior Density interval. To rank the algorithms, we sample over the posterior distribution of the strength parameters 1000 times. The median ranks and their corresponding variances is displayed in Table A.V.

D. Datatype comparison RQ2

For RQ2 we again combined all of the 364 datasets like for RQ1, only this time we did not drop the column containing the datatype. Hence we could analyze the accuracy with respect to the datatype of each dataset. From left to right, the columns in Table A.IV contains the mean, standard deviation, median as well as 5% amd 95% quantiles for each algorithm. Figure 6 illustrates the descriptive statistics of Table A.IV as a boxplot.

Based on the Bradley-Terry model described above, the parameter strength is computed for each algorithm. Figures 3, 4, 5 illustrates the distribution of the parameter strengths along with their High Posterior Density intervals for image, text and numeric datatypes respectively. To rank the algorithms, we sample over the posterior distribution of the strength parameters 1000 times. The median ranks and their corresponding variances are displayed in Table A.VI, A.VII, A.VIII for image, text and numeric datatypes.

E. Manual effort, RQ3

To answer RQ3 we performed the same operations on the 364 data frames as with RQ1 and RQ2, except this time we only dropped the column telling whether the algorithm was run on 10%, 25%, 50%, 75% or 90% of the data. Thus we could analyze the accuracy of the algorithms with respect the amount of available labels.

Table A.III contains the mean, standard deviation, median as well as 5% and 95% quantiles for each algorithm. Figure 7 provides descriptive statistics in the form of tfive boxplots, one for 10%, one for 25%, one for 50%, one for 75% labels, and one for 90% labels

Based on the Bradley-Terry model described above, the parameter strength is computed for each algorithm. Figures 8, 9, 10, 11, 12 illustrate the distribution of the parameter strengths along with their High Posterior Density interval for 10%, 25%, 50%, 75% and 90% labels respectively. To rank the algorithms we sample over the posterior distribution of the strength parameters 1000 times. The median ranks and their corresponding variances are displayed in Table A.IX, A.X. A.XI, A.XII, A.XIII for 10%, 25%, 50%, 75% and 90% labels respectively.

F. Probability of success, RQ4

This research question was answered w.r.t to both aggregated results, datatype comparison, and manual effort. Hence, to answer RQ4 we performed the following operations on all three of the datasets that were used to answer the previous RQs. First we made a copy of the dataset. In both copied and original variants, we added a new column called "SD" (for standard deviation). In the original dataset we put $SD = 0$ to indicate the absence of noise. The copied dataset we put $SD = 3$ to indicate that there is noise in the data. To add noise for each instance we replace the accuracy y with a simulated value of normal distribution with mean y and standard deviation 3.

Based on the Inverse Logit Binomial model described above, the odds ratios for the intercept a_{alg}, and noise b_{noise}, of each algorithm is computed. For aggregated data, Table A.XIV contains the estimated posterior parameters and Figure 13 contains the HDPI intervals of the estimated OR for the intercept, noise and standard deviation. For image, text and numeric datatypes the estimated posterior parameters are located in Table A.XV,A.XVI,A.XVII and the HDPI intervals of the estimated OR for the intercept, noise and standard deviation are illistrated in Figures 14,15 and 16 respectively.

V. Discussion

According to Table A.V, sparse label propagation is the highest ranking algorithm followed by Poisson MBO, Poisson MBO (old), and Balanced Poisson MBO . The uncertainty intervals of the posterior distribution are shown in Figure 2. The large overlap between the top three algorithms strength parameters indicates the uncertainty in rank between them (which can be observed in the large variance of each rank).

According to Table A.VI the highest ranking algorithm for image datasets is Centered kernel, second highest ranking algorithm is Poisson MBO, third ranking is shared with Laplace learning and Sparse label propagation and fourth highest algorithm is Balance Poisson. The high variance of

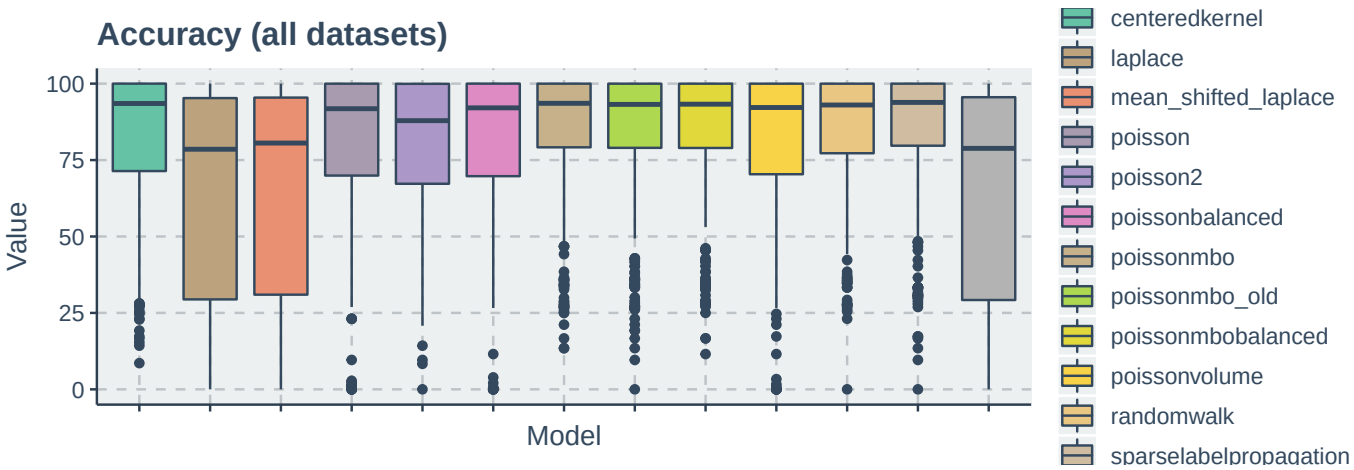

Figure 1. Boxplot illustrating the accuracy of all algorithms

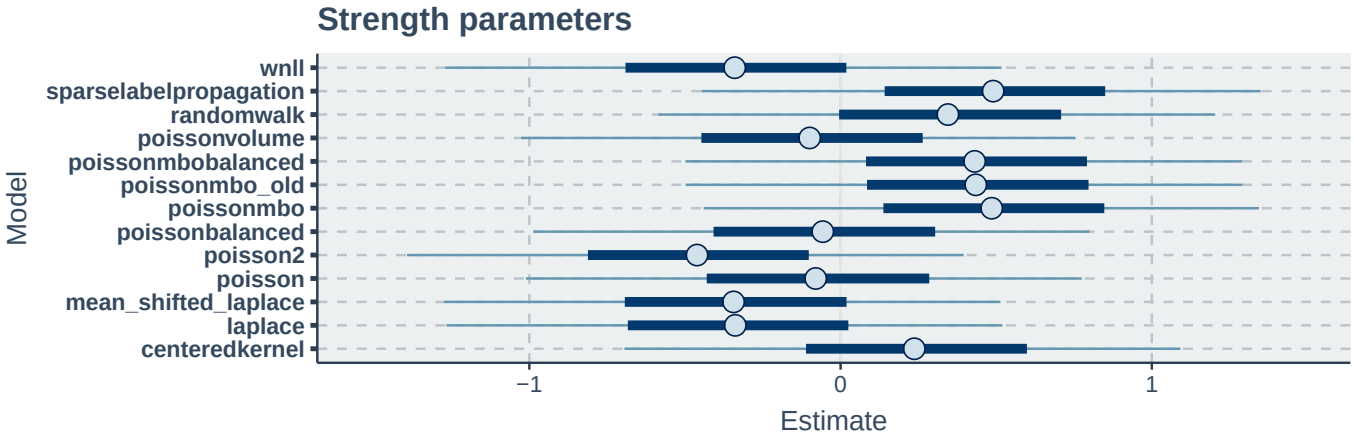

Figure 2. The credible interval of the estimated strength parameters. The thick blue line correspond to 50% probability and the thin blue line represent the 90% probability.

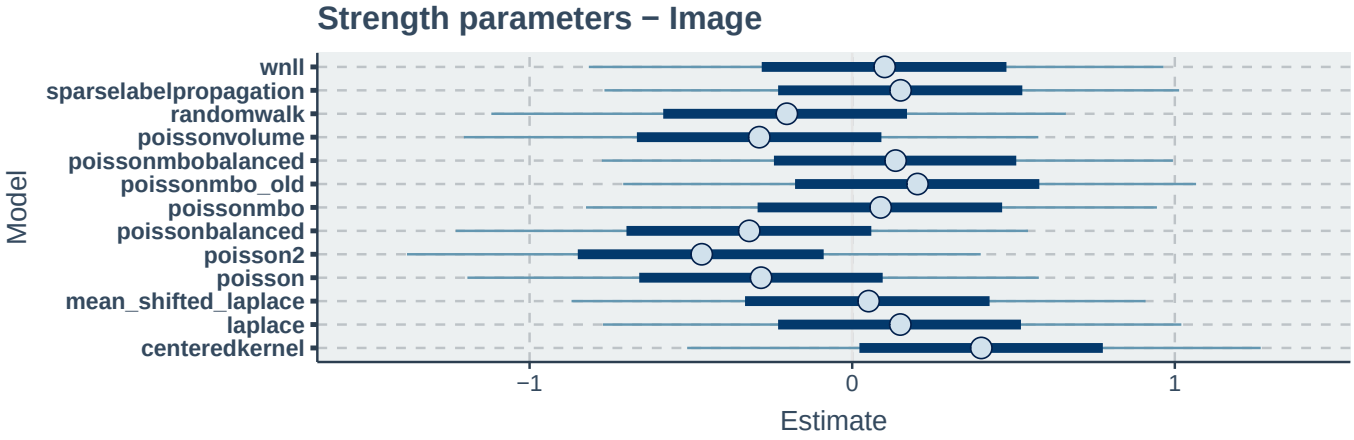

Figure 3. Boxplot illustrating the accuracy of all algorithms of image datatype

the third highest ranking algorithm indicates the uncertainty in their ranks.

According to Table A.VII the highest ranking algorithm for text datasets is random walk, second highest algorithm is Poisson Balanced, third highest is Poisson MBO, and fourth highest is Poisson MBO (old). According to Table A.VIII the

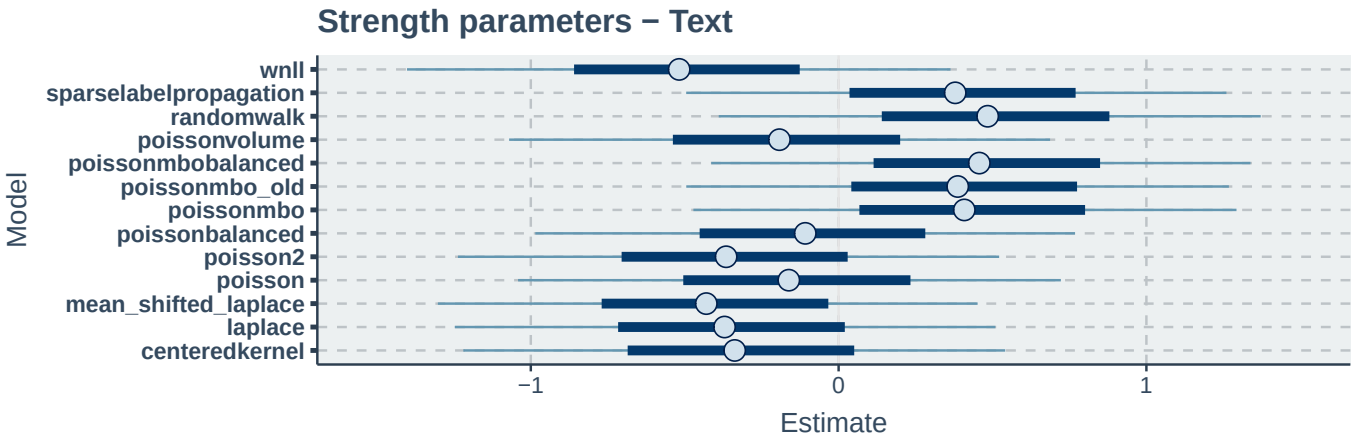

Figure 4. Boxplot illustrating the accuracy of all algorithms of text datatype

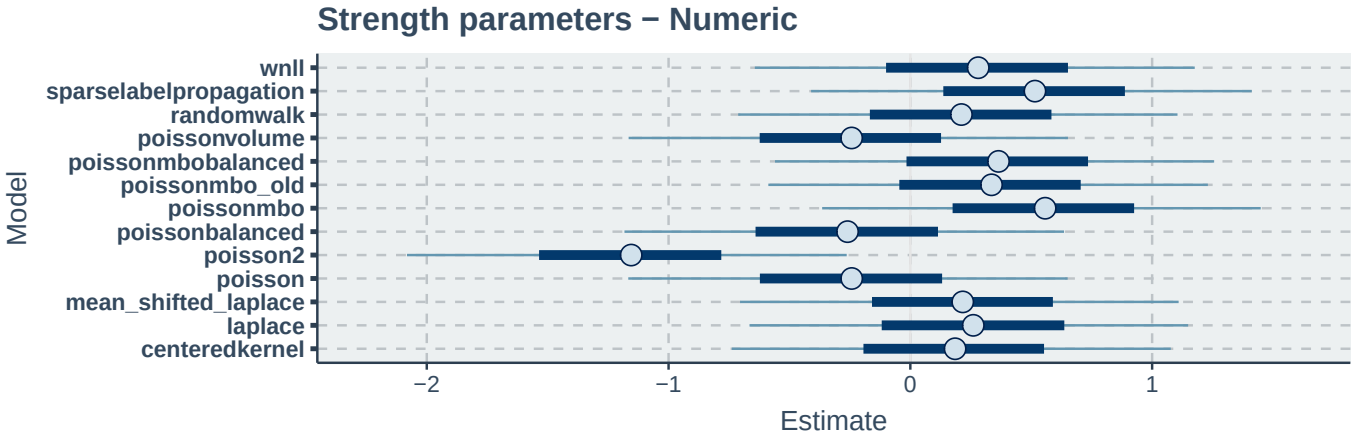

Figure 5. Boxplot illustrating the accuracy of all algorithms of numeric datatype

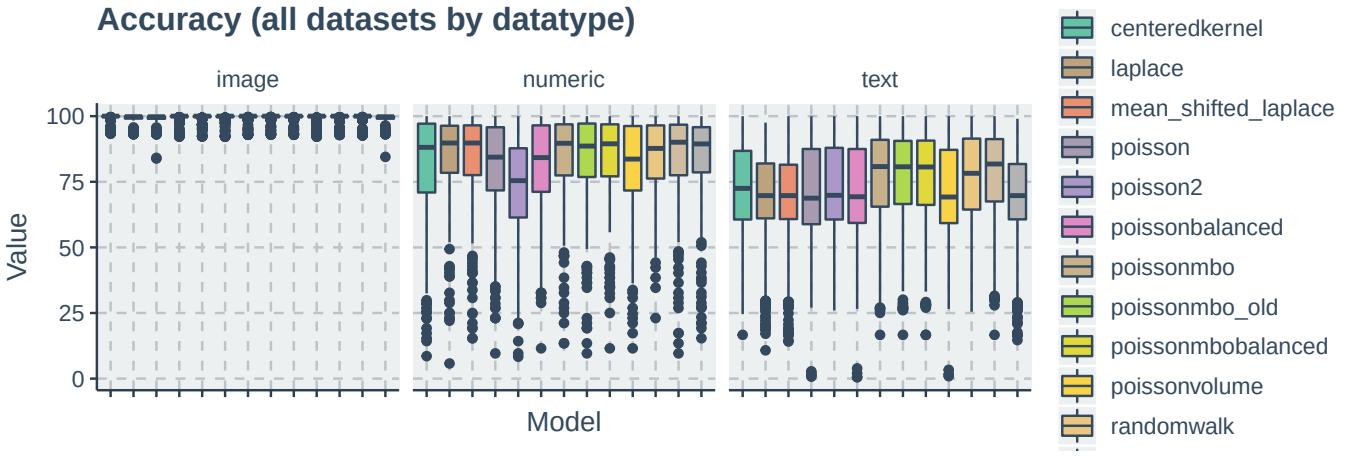

Figure 6. Boxplot illustrating the accuracy of all algorithms based on datatype

highest ranking algorithm is Poisson MBO, second highest is Sparse label propagation, third highest is Balanced Poisson and fourth highest is Poisson MBO (old). The uncertainty in invervals of the posterior distribution are shown i Figures 3, 4 and 5.

According to Table A.IX, the highest ranking algorithm when having access to 10% available labels is Poisson MBO,

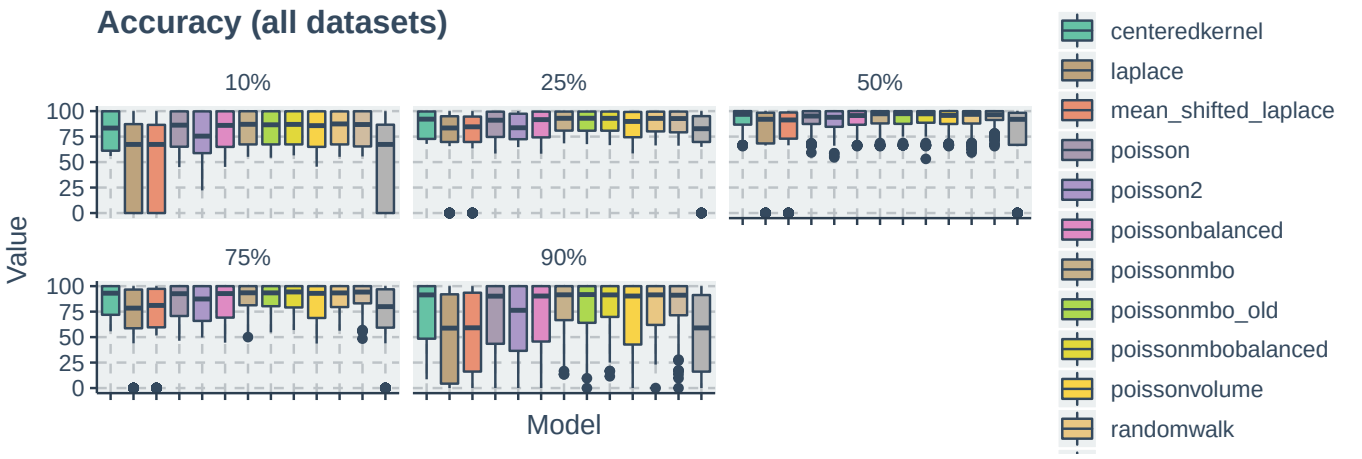

Figure 7. Boxplots illustrating the accuracy of all algorithms. From left to right in the upper row, we have boxplots for 10%,25% and 50%. From left to right in the lower right we have 75% and 90% available labels

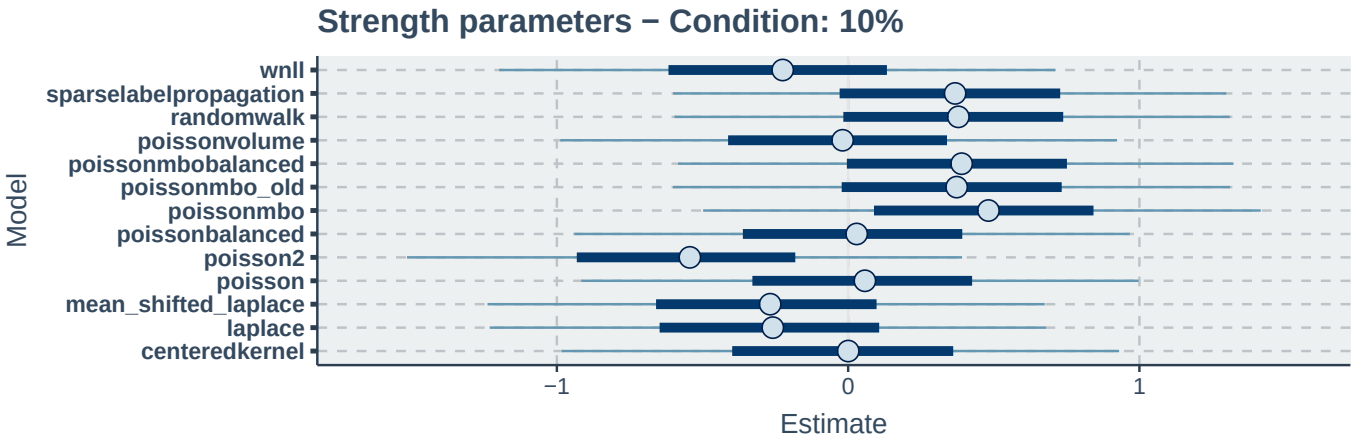

Figure 8. The credible interval of the estimated strength parameters of the algorithms with 10% available labels. The thick blue line correspond to 50% probability and the thin blue line represent the 90% probability.

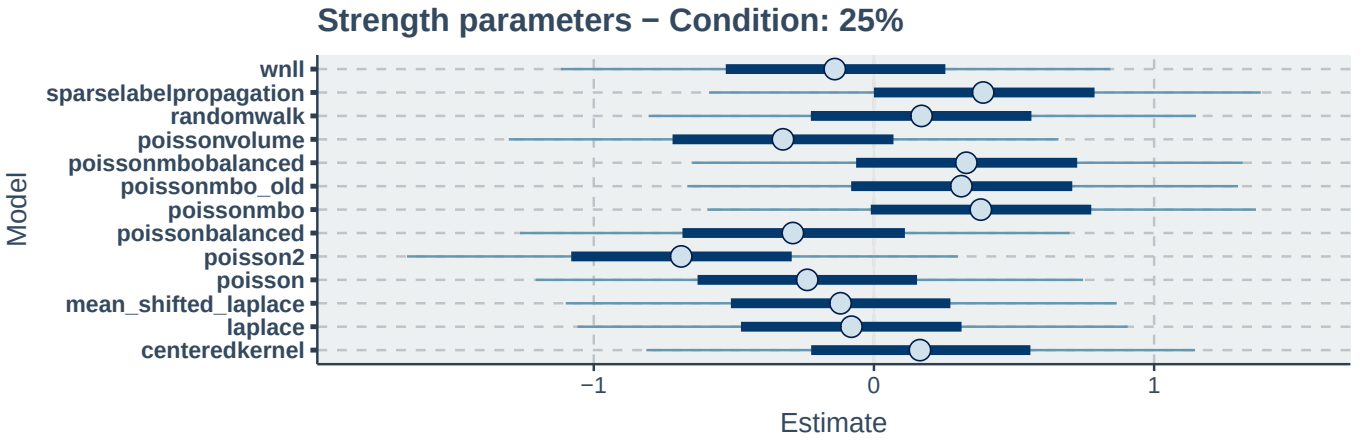

Figure 9. The credible interval of the estimated strength parameters of the algorithms with 10% available labels. The thick blue line correspond to 50% probability and the thin blue line represent the 90% probability.

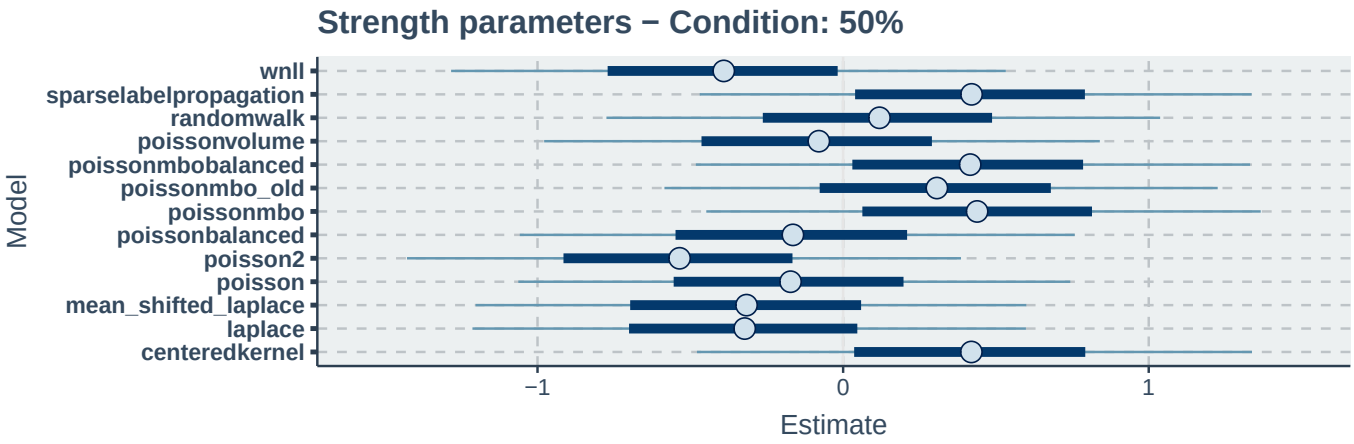

Figure 10. The credible interval of the estimated strength parameters of the algorithms with 50% available labels. The thick blue line correspond to 50% probability and the thin blue line represent the 90% probability.

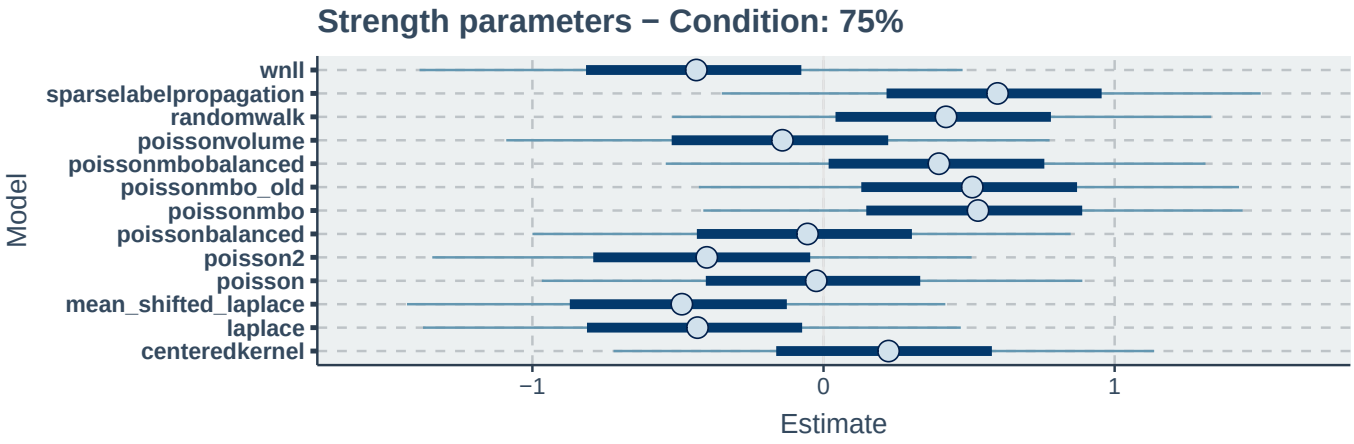

Figure 11. The credible interval of the estimated strength parameters of the algorithms with 75% available labels. The thick blue line correspond to 50% probability and the thin blue line represent the 90% probability.

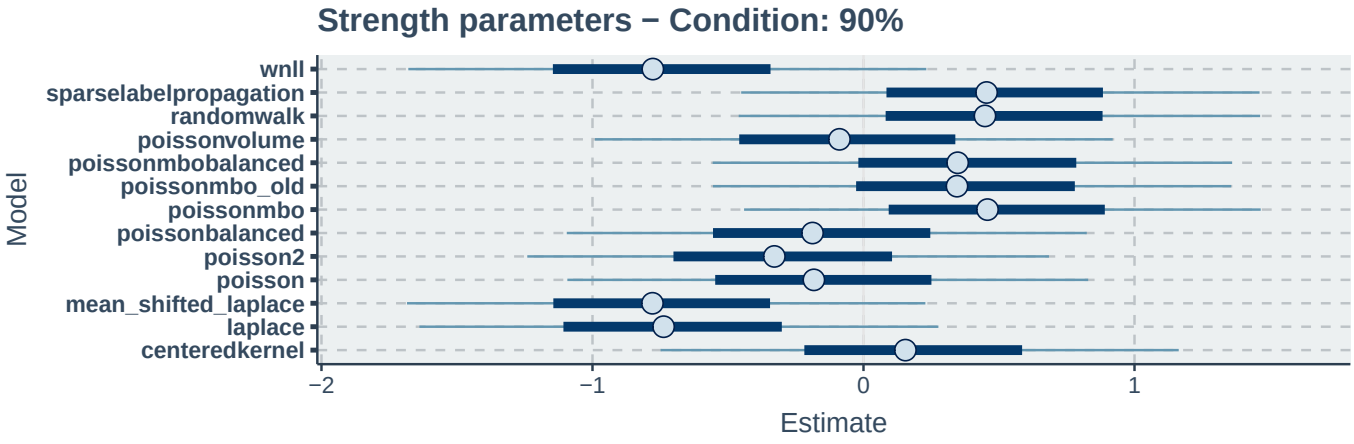

Figure 12. The credible interval of the estimated strength parameters of the algorithms with 90% available labels. The thick blue line correspond to 50% probability and the thin blue line represent the 90% probability.

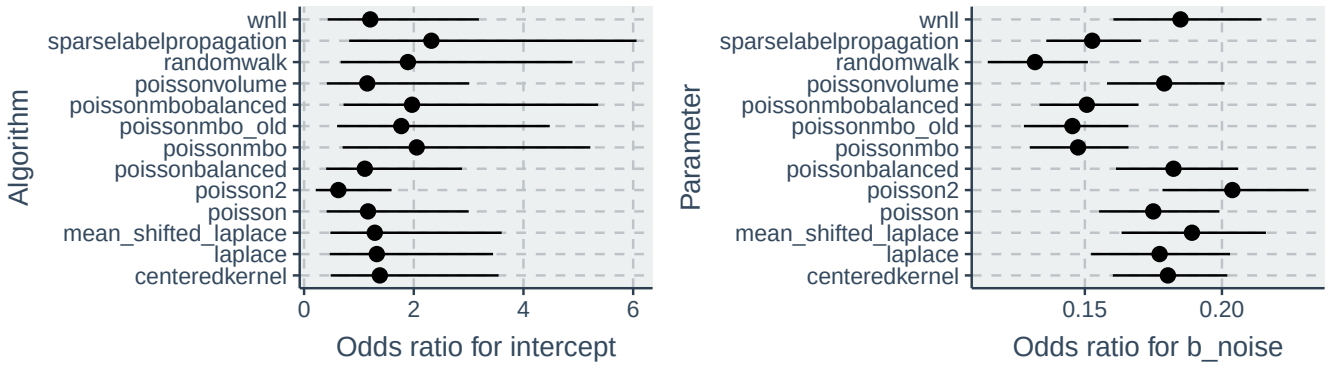

Figure 13. Odds rations for the HPDI interval for each algorithm intercept, influence of noise and standard deviation for aggregated data.

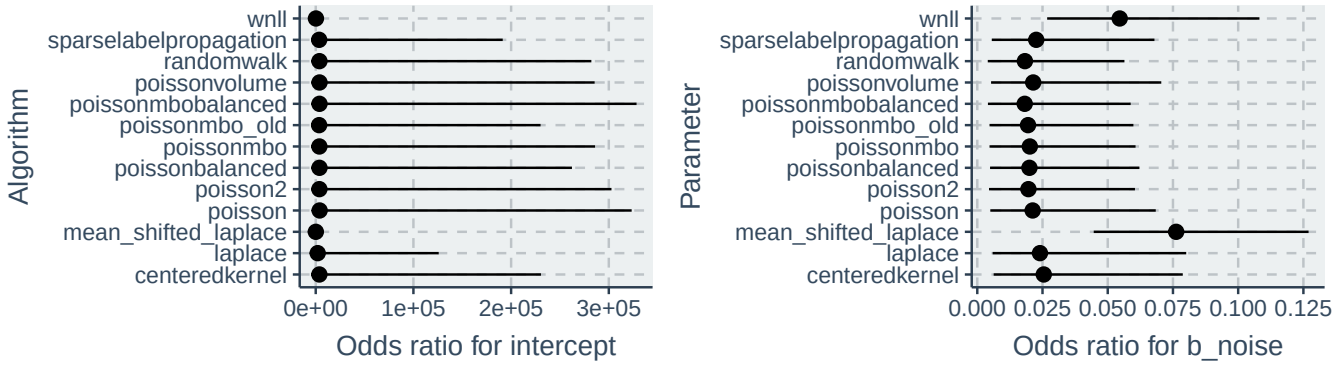

Figure 14. Odds rations for the HPDI interval for each algorithm intercept, influence of noise and standard deviation for image datatype.

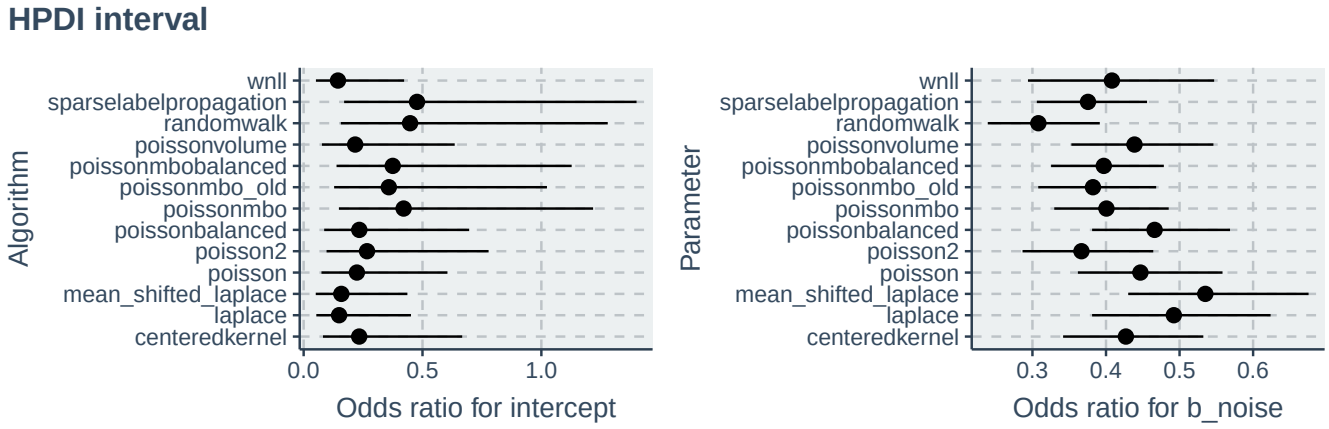

Figure 15. Odds rations for the HPDI interval for each algorithm intercept, influence of noise and standard deviation for text datatype.

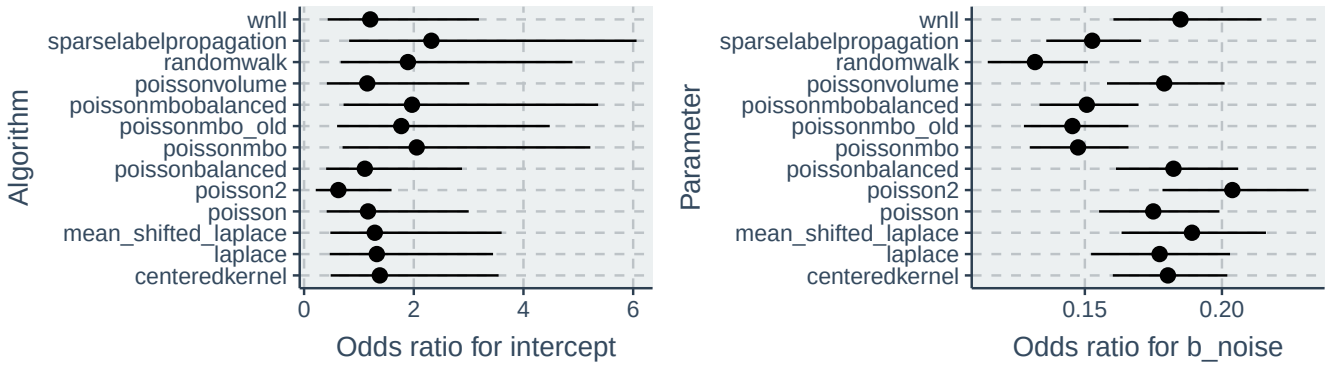

Figure 16. Odds rations for the HPDI interval for each algorithm intercept, influence of noise and standard deviation for numerical datatype.

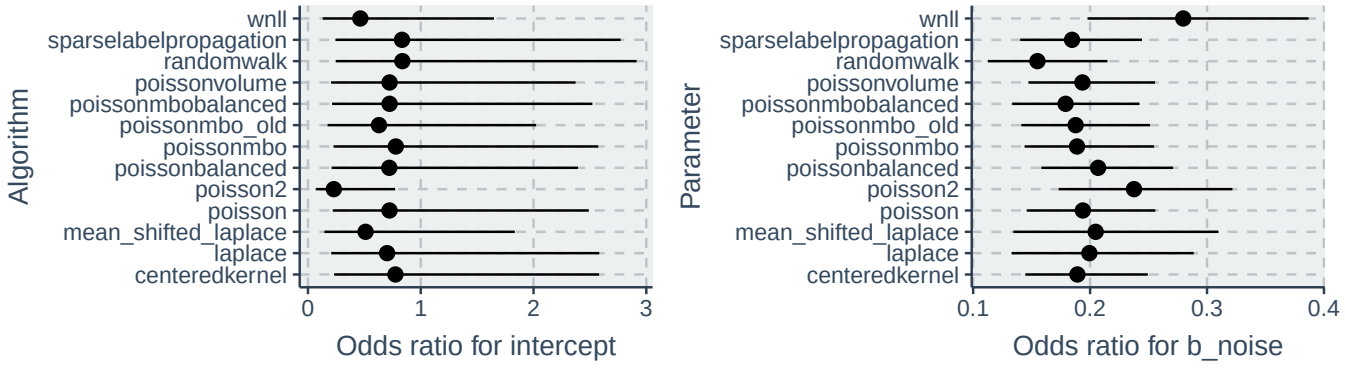

Figure 17. Odds rations for the HPDI interval for each algorithm intercept and influence of noise for 10% available labels.

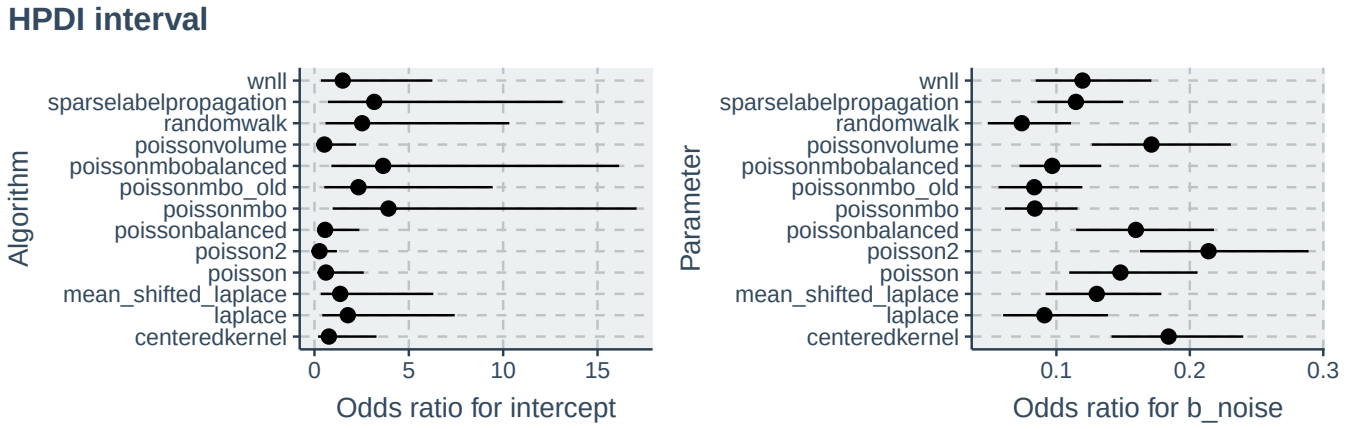

Figure 18. Odds rations for the HPDI interval for each algorithm intercept and influence of noise for 25% available labels.

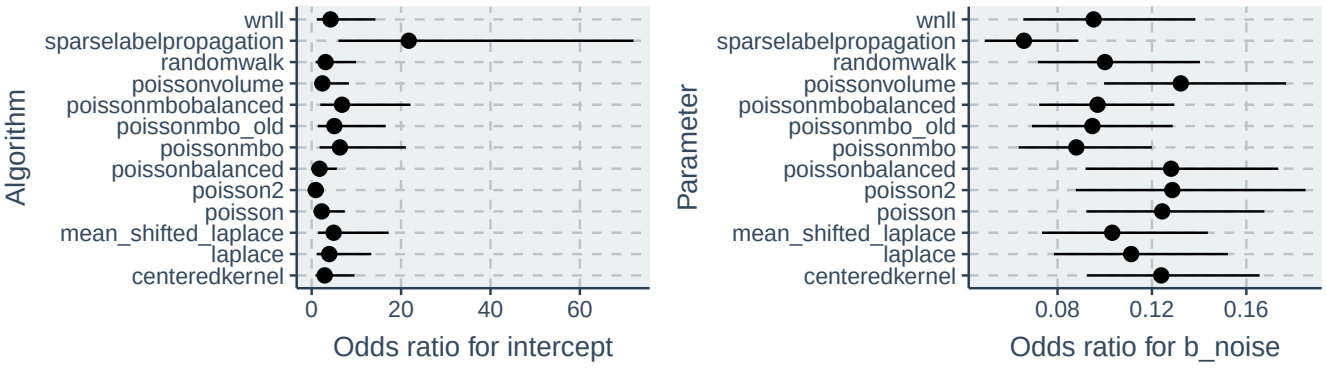

Figure 19. Odds rations for the HPDI interval for each algorithm intercept and influence of noise for 50% available labels.

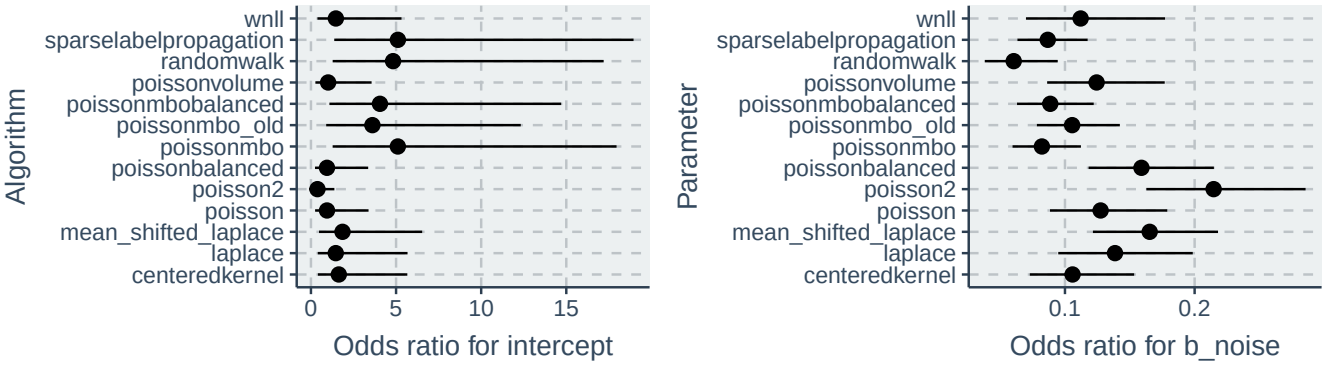

Figure 20. Odds rations for the HPDI interval for each algorithm intercept and influence of noise for 75% available labels.

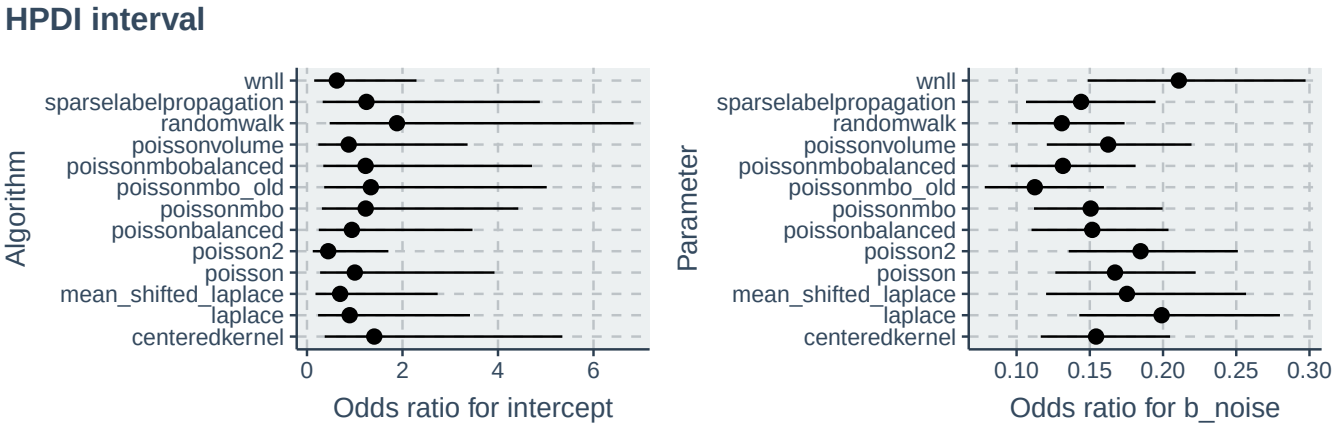

Figure 21. Odds rations for the HPDI interval for each algorithm intercept and influence of noise for 90% available labels.

second highest rank is shared with Balanced Poisson and random walk, third highest rank is shared with Poisson MBO (old) and sparse label propagation. The the four aforementioned algorithms have high variance in their rank which explains the ranks uncertainty. According to A.X the highest ranking algorithm when having access to 25% available labels is sparse label propagation followed by Poisson MBO, Balance Poisson MBO and Poisson MBO (old).

According to Table A.XI the highest ranking algorithm when having access to 50% available labels is Poisson MBO, the second highest algorithms spot is shared between centered kernel, Balance Poisson and Sparse label propagation. Third highest is Poisson MBO (old) and fourth highest algorithm is random walk. The second highest ranking algorithms have higher variance in their ranks, which is the reason for the uncertainty in their ranks.

According to Table A.XII the highest ranking algorithm when having access to 75% available labels is sparse label propagation followed by Poisson MBO, Poisson MBO (old) and random walk. According to Table A.XIII the highest ranking algorithm when having access to 90% labels is shared by Poisson MBO, random walk and sparse label proagation. The algorithm with the second highest ranking is Balanced Poisson MBO, the highest is Poisson MBO (old) and fourth highest algorithm is centered kernel. The highest ranking algorithms have much higher varince in their ranks, hence the uncertainty in their median rank. The uncertainty intervals of the posterior distribution are shown in Figures 8, 9, 10, 11, and 12 for 10%, 25%, 50%, 75% and 90% respectively. The overlap between the top algorithms strength parameters indicates the uncertainty in their estimates, this can also be observed in their variance.

According to Table A.XIV all algorithms have high mean OR above 1 except for Poisson2. This means that Poisson2 is the only algorithm whose intercept parameter does not increase the probability of success. When it comes to the noise parameters, they all have odds ratio$\approx$ 0.2 or less, hence the every algorithm perform poorly in the presence of noise. This is especially true for randomwalk, poissonmbo and poissonnmbo_old

For image datatypes, Table A.XV shows that all algorithms have high mean OR above 1. This means that they all increase the probability of success. Every algorithm has an odds ratio less than 0.09, hence all algorithms perform poorly in the presence of noise. For text datatypes, Table A.XVI show that all algorithms have OR less than 1, hence every algorithm does not increase the probability of success. The mean OR of the noise parameters are all around 0.4, thus all algorithms perform poorly in the presence of noise. For numeric datatypes, Table A.XVII shows that laplace, mean_shiften_laplace, poissonmbo, poissonmbobalanced, sparselabelpropagation and wnll, all have mean OR greater than 1, hence these algorithm increases the probability of success. The mean OR of the noise parameters are around 0.2 hence every algorithm perform poorly in the presence of noise.

When having access to 10% available labels, Table A.XVIII shows that all algorithms have mean OR less than 1, every algorithm do not increase the pro ability of success. Every algorithm have mean OR around 0.2, hence every algorithm perform poorly in the presence of noise. When having access to 25% of available labels, Table A.XIX centeredkernel, poisson, poisson2, poissonbalanced and poissonvolume are the only algorithms that does not have mean OR greater than 1. Hence, these algorithms does not increase the probability of success. All noise parameters are less than 1, most of them around 0.1, hence every algorithm perform poorly in the presence of noise.

When having access to 50% of available data labels, Table A.XX shows that every algorithm except poisson2 have mean OR above 1, hence every algorithm except poisson2 increases the probability of success. Every noise parameter have mean OR around 0.1, hence every algorithm perform poorly in the presence of noise. When having access to 75% of available data labels, Table A.XXI shows that poisson, poisson2 and poissonbalance are the only parameters with mean OR less than 1. Hence these althe only algorithms that does not increase the probability of success. Every noise parameter have mean OR around 0.1 hence every algorithm perform poorly in the presence of noise.

When having access to 90% of available labels, Table A.XXII shows that laplace, mean_shiften_laplace, poisson2, poissonbalanced, poissonvolume and wnll have mean OR less than 1. Hence, all of these algorithms do not increase the probability of success. Every noise parameter have mean OR close to 0, most of them around 0.15 so every algorithm perform poorly in presence of noise.

VI. Conclusion

The goal of this study is to provide a detailed overview of what machine learning algorithm should be used for automatic labeling of data in industrial contexts. Based on the results, we have found the algorithms that are ranked top four. In eight out of nine scenarios Poisson MBO and Poisson MBO (old). In two out of three scenarios Sparse label propagation and in seven out of nine scenarios Balanced Poisson is in the top four algorithms. Poisson MBO is not in the top for image datatypes and Poisson MBO (old) is not in the top for 90% available labels. Sparse label propagation is missing for Text data, 10% available labels and 50% available labels. Finally, the Balanced Poisson is missing for image data and 75% available labels. In 78% of the scenarios, Poisson MBO, Balanced Poisson MBO and Sparse label propagation are the top three algorithms. These three algorithms all increase the probability achieving at least 90% accuracy in every simulation except for when we analyse text datatypes, and when we evaluate manual effort with 10% available labels.

Thus this paper contributes in assisting machine learning practitioners to choose the optimal machine learning algorithm for automatic labeling. In future simulation studies we wish to examine these algorithms using other statistical models [45] to answer other related RQs, as well as examining other types of

semi-supervised learning algorithms. Another interesting topic is to compare semi-supervised learning to transfer learning.

Acknowledgment

This work was partially supported by the Wallenberg AI Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Fundation and by the Software Center.

References

[1] AzatiSoftware, *AzatiSoftware Automated Data Labeling with Machine Learning*, 2019. https://azati.ai/automated-data-labeling-with-machine-learning.

[2] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "Data labeling: An empirical investigation into industrial challenges and mitigation strategies," in *Product-Focused Software Process Improvement* (M. Morisio, M. Torchiano, and A. Jedlitschka, eds.), (Cham), pp. 202–216, Springer International Publishing, 2020.

[3] J. C. Chang, S. Amershi, and E. Kamar, "Revolt: Collaborative crowdsourcing for labeling machine learning datasets," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 2334–2346, 2017.

[4] J. Zhang, V. S. Sheng, T. Li, and X. Wu, "Improving crowdsourced label quality using noise correction," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1675–1688, 2017.

[5] T. Fredriksson., J. Bosch., and H. H. Olsson., "Machine learning models for automatic labeling: A systematic literature review," in *Proceedings of the 15th International Conference on Software Technologies - Volume 1: ICSOFT,*, pp. 552–561, INSTICC, SciTePress, 2020.

[6] A. Kumar, M. Boehm, and J. Yang, "Data management in machine learning: Challenges, techniques, and systems," in *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1717–1722, 2017.

[7] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "An empirical evaluation of algorithms for data labeling," in *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 201–209, IEEE, 2021.

[8] A. Agresti, *Categorical data analysis*, vol. 482. John Wiley & Sons, 2003.

[9] X. J. Zhu, "Semi-supervised learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2005.

[10] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[11] M. Cattelan, "Models for paired comparison data: A review with emphasis on dependent data," *Statistical Science*, pp. 412–433, 2012.

[12] P. Kleinschmidt and R. Mock, "Sensor systems in industrial applications," in *COMPEURO 89 Proceedings VLSI and Computer Peripherals.*, pp. 3–9, IEEE Computer Society, 1989.

[13] https://www.kaggle.com/jessicali9530/caltech256.

[14] http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

[15] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., 2009.

[16] "Corel database for content-based image retrieval." https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval.

[17] https://www.kaggle.com/elkamel/corel-images.

[18] A. K. Seewald, "Digits-a dataset for handwritten digit recognition," *Austrian Research Institut for Artificial Intelligence Technical Report, Vienna (Austria)*, 2005.

[19] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[20] https://github.com/zalandoresearch/fashion-mnist.

[21] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[23] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016.

[24] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017.

[25] Newsgroup dataset, "20 newsgroups dataset," 2012.

[26] M. Filannino, "Dbworld e-mail classification using a very small corpus," *The University of Manchester*, 2011.

[27] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pp. 127–138, Springer, 2017.

[28] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Security and Privacy*, vol. 1, no. 1, p. e9, 2018.

[29] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Portland, Oregon, USA), pp. 142–150, Association for Computational Linguistics, June 2011.

[30] W. Hersh, C. Buckley, T. Leone, and D. Hickam, "Ohsumed: an interactive retrieval evaluation and new large test collection for research," in *SIGIR'94*, pp. 192–201, Springer, 1994.

[31] empty, "Reuters-21578 dataset," empty.

[32] O. Ekin, P. L. Hammer, A. Kogan, and P. Winter, "Distance-based classification methods," *INFOR: Information Systems and Operational Research*, vol. 37, no. 3, pp. 337–352, 1999.

[33] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Technical Digest*, vol. 10, no. 3, pp. 262–266, 1989.

[34] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[35] C. C. Aggarwal and S. Sathe, "Theoretical foundations and algorithms for outlier ensembles," *Acm Sigkdd Explorations Newsletter*, vol. 17, no. 1, pp. 24–47, 2015.

[36] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, and R. S. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the annual symposium on computer application in medical care*, p. 261, American Medical Informatics Association, 1988.

[37] R. P. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural networks*, vol. 1, no. 1, pp. 75–89, 1988.

[38] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," in *Biomedical image processing and biomedical visualization*, vol. 1905, pp. 861–870, International Society for Optics and Photonics, 1993.

[39] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Operations Research*, vol. 43, no. 4, pp. 570–577, 1995.

[40] M. Forina, S. Lanteri, C. Armanino, *et al.*, "Parvus-an extendible package for data exploration, classification and correlation, institute of pharmaceutical and food analysis and technologies, via brigata salerno, 16147 genoa, italy (1988)," *Av. Loss Av. O set Av. Hit-Rate*, 1991.

[41] B. N. De França and G. H. Travassos, "Reporting guidelines for simulation-based studies in software engineering," 2012.

[42] H. L. Turner, J. van Etten, D. Firth, and I. Kosmidis, "Modelling rankings in r: The plackettluce package," *Computational Statistics*, pp. 1–31, 2020.

[43] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of statistical software*, vol. 76, no. 1, 2017.

[44] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo.," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, 2014.

[45] D. I. Mattos, J. Bosch, and H. H. Olsson, "Statistical models for the analysis of optimization algorithms with benchmark functions," *IEEE Transactions on Evolutionary Computation*, 2021.

APPENDIX

Table A.II
SUMMARY STATISTICS FOR THE ACCURACY AGGREGATED DATA

Model	Mean	SD	Median	5%	95%
centeredkernel	84.811	18.740	93.507	48.545	100.000
laplace	63.547	38.022	78.544	0.000	99.938
mean_shifted_laplace	63.991	37.973	80.559	0.000	99.926
poisson	83.266	20.161	91.800	43.589	100.000
poisson2	81.150	20.508	87.886	35.677	100.000
poissonbalanced	83.276	20.102	92.082	45.234	100.000
poissonmbo	87.078	16.449	93.568	57.681	100.000
poissonmbo_old	86.861	16.804	93.208	57.483	100.000
poissonmbobalanced	86.996	16.483	93.277	57.899	100.000
poissonvolume	83.255	20.155	92.194	42.830	100.000
randomwalk	86.755	16.245	93.011	57.799	100.000
sparselabelpropagation	87.542	16.037	93.861	58.110	100.000
wnll	63.765	37.928	78.827	0.000	99.929

Table A.III
SUMMARY STATISTICS FOR THE ACCURACY AGGREGATED DATA

Model	Mean	SD	Median	5%	95%
10% labels available					
centeredkernel	81.233	17.586	83.393	56.777	100.000
laplace	55.813	37.672	67.327	0.000	99.814
mean_shifted_laplace	55.174	37.807	67.327	0.000	99.814
poisson	81.169	18.336	86.249	50.963	100.000
poisson2	77.586	20.17	75.475	48.146	100.000
poissonbalanced	81.149	18.383	86.020	50.787	100.000
poissonmbo	83.338	16.415	87.068	58.098	100.000
poissonmbo_old	83.439	16.150	86.592	57.957	100.000
poissonmbobalanced	83.413	16.203	86.986	58.102	100.000
poissonvolume	81.067	18.361	85.773	51.223	100.000
randomwalk	83.427	16.164	87.439	58.110	100.000
sparselabelpropagation	82.978	16.779	86.865	57.236	100.000
wnll	55.686	37.791	67.327	0.000	99.814
25% labels available					
centeredkernel	87.166	11.999	92.147	69.730	100.000
laplace	72.369	34.048	83.475	0.000	99.926
mean_shifted_laplace	72.395	34.075	84.615	0.000	99.852
poisson	86.457	12.579	91.101	66.487	100.000
poisson2	84.819	12.172	83.682	67.854	100.000
poissonbalanced	86.446	12.634	91.672	66.844	100.000
poissonmbo	89.441	10.366	92.982	69.730	100.000
poissonmbo_old	89.225	10.466	93.019	69.890	100.000
poissonmbobalanced	89.236	10.522	92.964	69.730	100.000
poissonvolume	86.138	12.627	89.886	67.028	100.000
randomwalk	88.752	10.874	92.879	69.890	100.000
sparselabelpropagation	88.780	10.964	92.726	69.745	100.000
wnll	72.343	34.068	82.749	0.000	99.852
50% labels available					
centeredkernel	92.839	8.996	96.842	76.836	100.000
laplace	72.095	38.026	92.008	0.000	99.870
mean_shifted_laplace	72.307	38.074	91.346	0.000	99.949
poisson	91.607	10.236	95.182	67.327	100.000
poisson2	90.470	11.133	93.953	66.639	100.000
poissonbalanced	91.598	10.147	95.674	67.381	100.000
poissonmbo	92.995	9.277	97.403	71.719	100.000
poissonmbo_old	92.941	9.131	97.491	75.000	100.000
poissonmbobalanced	93.003	9.358	97.366	68.651	100.000
poissonvolume	91.766	10.115	95.789	67.558	100.000
randomwalk	91.959	10.277	96.842	66.801	100.000
sparselabelpropagation	94.264	7.317	96.431	78.108	100.000
wnll	71.909	38.036	91.970	0.000	99.957
75% labels available					
centeredkernel	86.703	15.228	93.079	59.997	100.000
laplace	65.847	36.664	78.444	0.000	99.995
mean_shifted_laplace	67.575	36.234	81.078	0.000	99.989
poisson	85.006	16.289	92.600	59.728	100.000
poisson2	83.167	16.291	87.412	59.493	100.000
poissonbalanced	84.580	16.911	92.720	51.538	100.000
poissonmbo	89.012	13.438	93.470	60.564	100.000
poissonmbo_old	88.995	13.152	93.400	61.115	100.000
poissonmbobalanced	88.933	13.451	94.318	61.220	100.000
poissonvolume	84.702	16.662	92.958	58.346	100.000
randomwalk	88.853	13.312	93.369	61.082	100.000
sparselabelpropagation	89.568	12.851	94.193	62.500	100.000
wnll	66.842	36.316	79.902	0.000	99.989
90% labels available					
centeredkernel	76.112	28.864	91.414	27.476	100.000
laplace	51.611	39.035	58.804	0.000	99.933
mean_shifted_laplace	52.503	38.950	59.130	0.000	99.914
poisson	72.092	31.012	90.200	23.077	100.000
poisson2	69.708	30.295	76.316	26.912	100.000
poissonbalanced	72.605	30.668	90.300	26.614	100.000
poissonmbo	80.605	24.763	91.500	27.689	100.000
poissonmbo_old	79.705	26.065	91.733	27.137	100.000
poissonmbobalanced	80.395	24.891	91.414	28.689	100.000
poissonvolume	72.600	30.973	90.300	20.962	100.000
randomwalk	80.782	24.063	91.429	33.129	100.000
sparselabelpropagation	82.120	23.838	91.114	30.623	100.000
wnll	52.048	38.843	59.022	0.000	99.917

Table A.IV
SUMMARY STATISTICS FOR THE ACCURACY AGGREGATED DATA

Model	Mean	SD	Median	5%	95%
Image					
centeredkernel	99.579	1.289	100.000	98.378	100.000
laplace	47.769	49.549	0.000	0.000	99.977
mean_shifted_laplace	48.674	49.499	0.000	0.000	99.971
poisson	99.489	1.495	100.000	97.281	100.000
poisson2	99.414	1.533	100.000	96.557	100.000
poissonbalanced	99.489	1.465	100.000	97.345	100.000
poissonmbo	99.552	1.339	100.000	98.359	100.000
poissonmbo_old	99.569	1.308	100.000	98.364	100.000
poissonmbobalanced	99.553	1.328	100.000	98.327	100.000
poissonvolume	99.477	1.495	100.000	97.331	100.000
randomwalk	99.525	1.368	100.000	97.976	100.000
sparselabelpropagation	99.545	1.419	100.000	98.268	100.000
wnll	48.233	49.537	0.000	0.000	99.983
Numeric					
centeredkernel	81.695	19.320	88.118	37.104	100.000
laplace	84.311	16.683	89.758	53.238	99.926
mean_shifted_laplace	84.456	16.593	89.773	51.299	99.926
poisson	81.178	17.727	84.393	42.857	100.000
poisson2	73.291	19.881	75.394	32.416	100.000
poissonbalanced	81.179	17.414	84.211	46.723	100.000
poissonmbo	84.956	15.797	89.665	55.783	100.000
poissonmbo_old	84.467	16.200	88.604	54.597	100.000
poissonmbobalanced	84.688	15.780	89.474	56.558	100.000
poissonvolume	81.388	17.381	83.650	45.636	100.000
randomwalk	84.635	13.928	87.719	60.396	100.000
sparselabelpropagation	85.103	16.022	90.075	55.278	100.000
wnll	84.520	16.209	89.415	53.808	99.853
Text					
centeredkernel	73.157	17.774	72.498	31.188	96.663
laplace	58.561	30.067	66.536	0.000	94.309
mean_shifted_laplace	58.843	30.432	67.284	0.000	95.074
poisson	69.131	20.892	68.391	30.096	95.866
poisson2	70.744	19.008	69.847	33.277	96.063
poissonbalanced	69.159	21.006	68.902	33.105	96.270
poissonmbo	76.726	17.141	80.788	36.056	97.550
poissonmbo_old	76.546	17.608	80.704	35.820	97.509
poissonmbobalanced	76.746	17.231	80.642	35.110	97.511
poissonvolume	68.899	21.034	68.750	33.124	95.973
randomwalk	76.105	17.761	78.205	36.279	97.548
sparselabelpropagation	77.978	16.492	81.762	56.187	96.450
wnll	58.543	30.084	66.896	0.000	94.892

Table A.V
RANKING OF THE ALGORITHMS

Models	Median Rank	Variance of the Rank
sparselabelpropagation	1	0.276
poissonmbo	2	0.297
poissonmbo_old	3	0.293
poissonmbobalanced	4	0.269
randomwalk	5	0.000
centeredkernel	6	0.000
poissonbalanced	7	0.185
poisson	8	0.341
poissonvolume	9	0.212
laplace	11	0.595
mean_shifted_laplace	11	0.668
wnll	11	0.653
poisson2	13	0.000

Table A.VI
RANKING OF THE ALGORITHMS OF IMAGE DATATYPE

Models	Median Rank	Variance of the Rank
centeredkernel	1	0.000
poissonmbo_old	2	0.443
laplace	4	2.012
sparselabelpropagation	4	1.576
poissonmbobalanced	5	1.504
wnll	6	1.907
poissonmbo	7	1.214
mean_shifted_laplace	8	0.773
randomwalk	9	0.076
poisson	10	0.610
poissonvolume	11	0.644
poissonbalanced	12	0.445
poisson2	13	0.000

Table A.VII
RANKING OF THE ALGORITHMS OF TEXT DATATYPE

Models	Median Rank	Variance of the Rank
randomwalk	1	0.297
poissonmbobalanced	2	0.448
poissonmbo	3	0.755
poissonmbo_old	4	0.678
sparselabelpropagation	5	0.608
poissonbalanced	6	0.124
poisson	7	0.310
poissonvolume	8	0.193
centeredkernel	9	0.533
poisson2	10	0.714
laplace	11	0.735
mean_shifted_laplace	12	0.233
wnll	13	0.021

Table A.VIII
RANKING OF THE ALGORITHMS OF NUMERIC DATATYPE

Models	Median Rank	Variance of the Rank
poissonmbo	1	0.133
sparselabelpropagation	2	0.133
poissonmbobalanced	3	0.221
poissonmbo_old	4	0.357
wnll	5	0.660
laplace	6	0.747
mean_shifted_laplace	7	0.923
randomwalk	8	0.945
centeredkernel	9	0.529
poisson	11	0.656
poissonbalanced	11	0.583
poissonvolume	11	0.638
poisson2	13	0.000

Table A.IX
RANKING OF THE ALGORITHMS WITH 10% OF AVAILABLE LABELS

Models	Median Rank	Variance of the Rank
poissonmbo	1	0.133
poissonmbobalanced	3	1.354
randomwalk	3	1.252
poissonmbo_old	4	1.302
sparselabelpropagation	4	1.261
poisson	6	0.493
poissonbalanced	7	0.743
centeredkernel	8	0.747
poissonvolume	9	0.578
wnll	10	0.474
laplace	11	0.579
mean_shifted_laplace	12	0.520
poisson2	13	0.000

Table A.X
RANKING OF THE ALGORITHMS WITH 25% OF AVAILABLE LABELS

Models	Median Rank	Variance of the Rank
sparselabelpropagation	1	0.582
poissonmbo	2	0.699
poissonmbobalanced	3	0.741
poissonmbo_old	4	0.490
randomwalk	5	0.258
centeredkernel	6	0.258
laplace	7	0.371
mean_shifted_laplace	8	0.514
wnll	9	0.492
poisson	10	0.269
poissonbalanced	11	0.369
poissonvolume	12	0.270
poisson2	13	0.000

Table A.XI
RANKING OF THE ALGORITHMS WITH 50% OF AVAILABLE LABELS

Models	Median Rank	Variance of the Rank
poissonmbo	2	1.090
centeredkernel	3	1.217
poissonmbobalanced	3	1.201
sparselabelpropagation	3	1.190
poissonmbo_old	5	0.071
randomwalk	6	0.000
poissonvolume	7	0.098
poissonbalanced	8	0.346
poisson	9	0.344
mean_shifted_laplace	10	0.372
laplace	11	0.423
wnll	12	0.215
poisson2	13	0.003

Table A.XII
RANKING OF THE ALGORITHMS WITH 75% OF AVAILABLE LABELS

Models	Median Rank	Variance of the Rank
sparselabelpropagation	1	0.147
poissonmbo	2	0.405
poissonmbo_old	3	0.389
randomwalk	4	0.314
poissonmbobalanced	5	0.277
centeredkernel	6	0.001
poisson	7	0.227
poissonbalanced	8	0.273
poissonvolume	9	0.069
poisson2	10	0.640
laplace	11	0.938
wnll	11	0.840
mean_shifted_laplace	13	0.464

Table A.XIII
RANKING OF THE ALGORITHMS WITH 90% OF AVAILABLE LABELS

Models	Median Rank	Variance of the Rank
poissonmbo	2	0.773
randomwalk	2	0.802
sparselabelpropagation	2	0.761
poissonmbobalanced	4	0.383
poissonmbo_old	5	0.392
centeredkernel	6	0.000
poissonvolume	7	0.069
poisson	8	0.313
poissonbalanced	9	0.324
poisson2	10	0.013
laplace	11	0.458
mean_shifted_laplace	12	0.539
wnll	12	0.548

Table A.XIV
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL FOR AGGREGATED DATA

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	0.323	1.381	704.515	1.008
a_laplace	0.282	1.326	695.539	1.008
a_mean_shifted_laplace	0.254	1.289	709.772	1.008
a_poisson	0.153	1.165	699.738	1.008
a_poisson2	-0.470	0.625	700.028	1.008
a_poissonbalanced	0.103	1.108	696.868	1.008
a_poissonmbo	0.720	2.054	700.926	1.008
a_poissonmbo_old	0.572	1.772	692.109	1.009
a_poissonmbobalanced	0.677	1.968	691.343	1.009
a_poissonvolume	0.141	1.152	703.880	1.008
a_randomwalk	0.638	1.893	694.506	1.008
a_sparselabelpropagation	0.842	2.321	698.446	1.008
a_wnll	0.188	1.207	702.986	1.008
b_centeredkernel	-1.713	0.180	12313.943	1.000
b_laplace	-1.730	0.177	10686.764	1.000
b_mean_shifted_laplace	-1.666	0.189	10204.055	1.000
b_poisson	-1.743	0.175	11601.933	1.000
b_poisson2	-1.591	0.204	11418.623	1.000
b_poissonbalanced	-1.702	0.182	9915.078	1.000
b_poissonmbo	-1.914	0.148	7388.807	1.000
b_poissonmbo_old	-1.928	0.145	11322.511	1.000
b_poissonmbobalanced	-1.892	0.151	9246.354	1.000
b_poissonvolume	-1.721	0.179	10634.814	1.000
b_randomwalk	-2.026	0.132	11804.084	1.000
b_sparselabelpropagation	-1.879	0.153	10378.166	1.000
b_wnll	-1.688	0.185	11673.173	1.000
s	2.590	13.334	831.072	1.003

Table A.XV
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL FOR IMAGE DATA

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	8.226	3735.216	3090.514	1.001
a_laplace	7.606	2010.952	4368.767	1.000
a_mean_shifted_laplace	4.691	108.930	3033.036	1.001
a_poisson	8.299	4019.048	3590.157	1.000
a_poisson2	8.226	3738.408	3157.185	1.000
a_poissonbalanced	8.252	3837.098	3639.010	1.001
a_poissonmbo	8.226	3735.736	2348.278	1.001
a_poissonmbo_old	8.158	3489.973	2694.494	1.001
a_poissonmbobalanced	8.259	3862.406	2290.591	1.002
a_poissonvolume	8.265	3887.367	2615.526	1.001
a_randomwalk	8.242	3798.297	3920.952	1.000
a_sparselabelpropagation	8.169	3531.157	3357.880	1.001
a_wnll	5.459	234.830	3428.687	1.000
b_centeredkernel	-3.670	0.025	3050.585	1.001
b_laplace	-3.729	0.024	4310.673	1.000
b_mean_shifted_laplace	-2.574	0.076	3431.378	1.001
b_poisson	-3.853	0.021	3620.375	1.000
b_poisson2	-3.932	0.020	3191.296	1.000
b_poissonbalanced	-3.912	0.020	3632.680	1.001
b_poissonmbo	-3.905	0.020	2336.588	1.001
b_poissonmbo_old	-3.940	0.019	2695.296	1.001
b_poissonmbobalanced	-4.007	0.018	2262.748	1.001
b_poissonvolume	-3.843	0.021	2630.078	1.001
b_randomwalk	-4.004	0.018	3928.982	1.000
b_sparselabelpropagation	-3.790	0.023	3391.451	1.000
b_wnll	-2.908	0.055	3262.676	1.000
s	0.235	1.265	1729.386	1.002

Table A.XVI
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL FOR TEXT DATA

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	-1.454	0.234	388.610	1.008
a_laplace	-1.902	0.149	406.958	1.008
a_mean_shifted_laplace	-1.842	0.159	401.908	1.008
a_poisson	-1.496	0.224	394.433	1.008
a_poisson2	-1.322	0.267	389.700	1.008
a_poissonbalanced	-1.452	0.234	387.553	1.008
a_poissonmbo	-0.866	0.421	387.908	1.008
a_poissonmbo_old	-1.026	0.358	386.568	1.008
a_poissonmbobalanced	-0.981	0.375	384.108	1.008
a_poissonvolume	-1.529	0.217	399.650	1.008
a_randomwalk	-0.803	0.448	392.119	1.008
a_sparselabelpropagation	-0.740	0.477	390.005	1.008
a_wnll	-1.936	0.144	400.359	1.008
b_centeredkernel	-0.851	0.427	10391.902	1.000
b_laplace	-0.709	0.492	9284.027	1.000
b_mean_shifted_laplace	-0.625	0.535	7255.258	1.000
b_poisson	-0.806	0.447	14224.974	1.000
b_poisson2	-1.003	0.367	8942.023	1.000
b_poissonbalanced	-0.763	0.466	9988.819	1.000
b_poissonmbo	-0.915	0.401	7097.817	1.000
b_poissonmbo_old	-0.962	0.382	7292.305	1.000
b_poissonmbobalanced	-0.924	0.397	6117.961	1.001
b_poissonvolume	-0.824	0.439	7936.773	1.000
b_randomwalk	-1.178	0.308	9590.113	1.000
b_sparselabelpropagation	-0.979	0.376	11384.404	1.001
b_wnll	-0.896	0.408	7428.442	1.000
s	1.540	4.664	1024.601	1.000

Table A.XVII
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL FOR NUMERIC DATA

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	-0.225	0.799	429.567	1.010
a_laplace	0.146	1.157	433.252	1.009
a_mean_shifted_laplace	0.081	1.085	418.344	1.010
a_poisson	-0.598	0.550	415.806	1.010
a_poisson2	-2.396	0.091	430.805	1.009
a_poissonbalanced	-0.772	0.462	436.692	1.010
a_poissonmbo	0.062	1.064	422.014	1.010
a_poissonmbo_old	-0.104	0.901	428.668	1.010
a_poissonmbobalanced	0.106	1.112	413.901	1.010
a_poissonvolume	-0.604	0.547	432.684	1.009
a_randomwalk	-0.223	0.800	416.458	1.010
a_sparselabelpropagation	0.206	1.228	426.388	1.010
a_wnll	0.001	1.001	414.882	1.010
b_centeredkernel	-1.448	0.235	7605.750	1.000
b_laplace	-1.652	0.192	9148.817	1.000
b_mean_shifted_laplace	-1.694	0.184	8834.989	1.000
b_poisson	-1.402	0.246	8196.091	1.000
b_poisson2	-0.749	0.473	11767.954	1.000
b_poissonbalanced	-1.242	0.289	11329.245	1.000
b_poissonmbo	-1.594	0.203	9246.894	1.000
b_poissonmbo_old	-1.567	0.209	5557.617	1.001
b_poissonmbobalanced	-1.472	0.229	5621.187	1.001
b_poissonvolume	-1.320	0.267	8934.852	1.000
b_randomwalk	-1.591	0.204	7665.451	1.000
b_sparselabelpropagation	-1.487	0.226	7179.380	1.001
b_wnll	-1.524	0.218	11205.458	1.000
s	2.757	15.749	327.218	1.016

Table A.XVIII
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL WHEN HAVING 10% OF AVAILABLE LABELS

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	-0.254	0.775	779.584	1.003
a_laplace	-0.354	0.702	873.305	1.003
a_mean_shifted_laplace	-0.672	0.511	844.840	1.003
a_poisson	-0.324	0.723	793.481	1.003
a_poisson2	-1.472	0.229	753.297	1.003
a_poissonbalanced	-0.327	0.721	821.361	1.003
a_poissonmbo	-0.250	0.779	817.052	1.003
a_poissonmbo_old	-0.463	0.630	784.795	1.003
a_poissonmbobalanced	-0.323	0.724	788.945	1.002
a_poissonvolume	-0.323	0.724	796.945	1.003
a_randomwalk	-0.178	0.837	798.842	1.003
a_sparselabelpropagation	-0.182	0.834	787.449	1.002
a_wnll	-0.768	0.464	839.798	1.003
b_centeredkernel	-1.665	0.189	8667.428	1.001
b_laplace	-1.612	0.199	7042.689	1.000
b_mean_shifted_laplace	-1.586	0.205	7007.332	1.000
b_poisson	-1.641	0.194	7441.032	1.000
b_poisson2	-1.437	0.238	6808.986	1.000
b_poissonbalanced	-1.576	0.207	9005.802	1.000
b_poissonmbo	-1.667	0.189	7113.325	1.000
b_poissonmbo_old	-1.673	0.188	7642.807	1.000
b_poissonmbobalanced	-1.720	0.179	7271.886	1.000
b_poissonvolume	-1.643	0.193	6770.011	1.000
b_randomwalk	-1.865	0.155	7823.042	1.000
b_sparselabelpropagation	-1.690	0.185	7518.793	1.000
b_wnll	-1.274	0.280	9654.630	1.000
s	3.109	22.393	1073.593	1.005

Table A.XX
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL WHEN HAVING 50% OF AVAILABLE LABELS

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	1.073	2.925	910.268	1.001
a_laplace	1.371	3.939	987.516	1.001
a_mean_shifted_laplace	1.594	4.924	911.723	1.001
a_poisson	0.810	2.249	910.522	1.001
a_poisson2	-0.091	0.913	882.162	1.002
a_poissonbalanced	0.565	1.759	916.596	1.001
a_poissonmbo	1.844	6.324	921.461	1.001
a_poissonmbo_old	1.626	5.084	965.818	1.001
a_poissonmbobalanced	1.916	6.793	906.694	1.001
a_poissonvolume	0.876	2.401	933.547	1.001
a_randomwalk	1.137	3.118	877.777	1.001
a_sparselabelpropagation	3.078	21.721	933.928	1.001
a_wnll	1.442	4.229	970.725	1.001
b_centeredkernel	-2.088	0.124	9890.437	1.000
b_laplace	-2.196	0.111	9388.486	1.000
b_mean_shifted_laplace	-2.271	0.103	8463.966	1.000
b_poisson	-2.085	0.124	8193.398	1.000
b_poisson2	-2.051	0.129	9221.631	1.000
b_poissonbalanced	-2.055	0.128	8569.824	1.000
b_poissonmbo	-2.431	0.088	8689.602	1.000
b_poissonmbo_old	-2.356	0.095	8873.729	1.000
b_poissonmbobalanced	-2.334	0.097	7135.659	1.000
b_poissonvolume	-2.023	0.132	7552.449	1.000
b_randomwalk	-2.302	0.100	7489.467	1.000
b_sparselabelpropagation	-2.722	0.066	7506.287	1.000
b_wnll	-2.351	0.095	8018.946	1.000
s	3.135	22.998	1148.295	1.003

Table A.XIX
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL WHEN HAVING 25% OF AVAILABLE LABELS

Parameter	Mean	OR Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	-0.270	0.764	653.207	1.002
a_laplace	0.570	1.768	654.864	1.002
a_mean_shifted_laplace	0.311	1.364	672.839	1.001
a_poisson	-0.499	0.607	647.910	1.001
a_poisson2	-1.313	0.269	633.972	1.002
a_poissonbalanced	-0.574	0.563	638.920	1.002
a_poissonmbo	1.367	3.922	680.607	1.002
a_poissonmbo_old	0.846	2.330	647.559	1.002
a_poissonmbobalanced	1.292	3.638	642.056	1.001
a_poissonvolume	-0.647	0.523	658.757	1.002
a_randomwalk	0.926	2.525	638.336	1.002
a_sparselabelpropagation	1.153	3.168	670.767	1.002
a_wnll	0.404	1.498	653.149	1.002
b_centeredkernel	-1.692	0.184	6948.891	1.000
b_laplace	-2.397	0.091	7528.165	1.000
b_mean_shifted_laplace	-2.038	0.130	6412.071	1.000
b_poisson	-1.911	0.148	7068.210	1.000
b_poisson2	-1.542	0.214	8189.207	1.000
b_poissonbalanced	-1.835	0.160	7708.906	1.000
b_poissonmbo	-2.479	0.084	6421.991	1.000
b_poissonmbo_old	-2.484	0.083	6353.535	1.000
b_poissonmbobalanced	-2.334	0.097	6869.441	1.000
b_poissonvolume	-1.765	0.171	6665.552	1.000
b_randomwalk	-2.603	0.074	6291.734	1.000
b_sparselabelpropagation	-2.166	0.115	5727.958	1.000
b_wnll	-2.124	0.120	6731.097	1.001
s	3.728	41.616	1154.709	1.001

Table A.XXI
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL WHEN HAVING 75% OF AVAILABLE LABELS

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	0.493	1.637	702.683	1.007
a_laplace	0.377	1.458	707.281	1.006
a_mean_shifted_laplace	0.618	1.855	722.773	1.006
a_poisson	-0.058	0.944	725.772	1.007
a_poisson2	-0.962	0.382	701.259	1.007
a_poissonbalanced	-0.059	0.942	716.808	1.008
a_poissonmbo	1.630	5.105	695.013	1.007
a_poissonmbo_old	1.284	3.611	710.041	1.007
a_poissonmbobalanced	1.400	4.055	702.463	1.007
a_poissonvolume	0.009	1.009	718.208	1.008
a_randomwalk	1.574	4.824	705.747	1.006
a_sparselabelpropagation	1.631	5.108	691.306	1.007
a_wnll	0.382	1.465	724.048	1.006
b_centeredkernel	-2.246	0.106	7164.311	1.000
b_laplace	-1.976	0.139	6908.095	1.000
b_mean_shifted_laplace	-1.800	0.165	7670.668	1.000
b_poisson	-2.060	0.128	7361.244	1.000
b_poisson2	-1.539	0.215	6980.412	1.000
b_poissonbalanced	-1.839	0.159	8417.561	1.000
b_poissonmbo	-2.498	0.082	6912.454	1.000
b_poissonmbo_old	-2.248	0.106	7635.792	1.000
b_poissonmbobalanced	-2.423	0.089	7176.228	1.000
b_poissonvolume	-2.084	0.124	7390.189	1.000
b_randomwalk	-2.806	0.060	7301.945	1.000
b_sparselabelpropagation	-2.445	0.087	6907.517	1.000
b_wnll	-2.188	0.112	5711.759	1.000
s	3.276	26.457	968.431	1.005

Table A.XXII
ESTIMATED PARAMETERS OF THE INVERSE LOGIT MODEL WHEN HAVING 90% OF AVAILABLE LABELS

Parameter	Mean	OR.Mean	n_{eff}	$\widehat{R}$
a_centeredkernel	0.343	1.409	802.953	1.007
a_laplace	-0.112	0.894	829.139	1.006
a_mean_shifted_laplace	-0.365	0.694	846.277	1.007
a_poisson	0.001	1.001	813.383	1.008
a_poisson2	-0.811	0.444	812.403	1.009
a_poissonbalanced	-0.062	0.940	814.115	1.007
a_poissonmbo	0.207	1.230	768.141	1.008
a_poissonmbo_old	0.291	1.337	800.959	1.009
a_poissonmbobalanced	0.205	1.228	811.720	1.007
a_poissonvolume	-0.135	0.874	812.687	1.008
a_randomwalk	0.633	1.884	814.736	1.007
a_sparselabelpropagation	0.219	1.245	818.412	1.006
a_wnll	-0.469	0.626	848.209	1.007
b_centeredkernel	-1.869	0.154	7711.254	1.000
b_laplace	-1.614	0.199	9929.694	1.000
b_mean_shifted_laplace	-1.741	0.175	8052.302	1.000
b_poisson	-1.788	0.167	9811.605	1.000
b_poisson2	-1.689	0.185	6459.582	1.000
b_poissonbalanced	-1.886	0.152	9879.303	1.000
b_poissonmbo	-1.893	0.151	8783.242	1.000
b_poissonmbo_old	-2.185	0.112	7012.193	1.000
b_poissonmbobalanced	-2.028	0.132	8507.623	1.000
b_poissonvolume	-1.817	0.163	8659.477	1.000
b_randomwalk	-2.034	0.131	8406.667	1.000
b_sparselabelpropagation	-1.938	0.144	8637.927	1.000
b_wnll	-1.557	0.211	8215.369	1.000
s	3.388	29.607	1035.012	1.008