Horizon 2020

fashion BRAIN project

Understanding Europe's Fashion Data Universe

A Set of Aggregation Algorithms and Their Experimental Evaluation

Deliverable number: D3.2

Version 2.0

Funded by the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 732328

Project Acronym: FashionBrain
Project Full Title: Understanding Europe's Fashion Data Universe
Call: H2020-ICT-2016-1
Topic: ICT-14-2016-2017, Big Data PPP: Cross-sectorial and cross-lingual data integration and experimentation
Project URL: https://fashionbrain-project.eu

Deliverable type	Report (R)
Dissemination level	Public (PU)
Contractual Delivery Date	31 December 2018
Resubmission Delivery Date	4 February 2019
Number of pages	26, the last one being no. 20
Authors	Ines Arous, Mourad Khayati, Zakhar Tymchenko - UNIFR
Peer review	Alessandro Checco, Jennifer Dick - USFD

Change Log

Version	Date	Status	Partner	Remarks
0.1	05/12/2018	Draft	UNIFR	
0.2	12/12/2018	Full Draft	UNIFR	
1.0	20/12/2018	Final	UNIFR, USFD	Rejected 30/01/2019
2.0	04/02/2019	Resubmitted Final	UNIFR, USFD	

Deliverable Description

As a result of task 3.2, this deliverable will consist of implemented algorithms that will be integrated within the data integration infrastructure developed within WP2 and will feed into WP5, 6 and 7.

Abstract

Crowdsourcing allows to build hybrid online platforms that combine scalable information systems with the power of human intelligence to complete tasks that are difficult to tackle for current algorithms. Examples include hybrid systems that use the crowd to perform real-world tasks such as entity resolution (Task T2.2), review classification (T3.4), relation extraction (T4.3), image annotation (T5.1) and image enrichment (T6.3). However, workers often yield low-quality answers, and truth inference (aggregation) algorithms are used to infer the correct answer from workers' answers. In this deliverable, we evaluate different truth inference algorithms on fashion dataset. More specifically, we perform a comprehensive evaluation of the state of the art aggregation algorithms to select the one that fits the needs of our project. We measure the performance on a real-world crowd fashion dataset and we analyze the factors influencing truth inference such as the task redundancy (i.e. the number of workers to whom we show the task) and the task complexity (i.e. the number of workers agreeing on one answer). We find that ZenCrowd is the most accurate method to infer the truth from the FashionBrain dataset except for some cases where worker's agreement is enough to infer the truth and hence Majority Voting is more accurate.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AMT	Amazon Mechanical Turk (www.mturk.com), micro-task crowdsourcing platform
CF	Crowdflower (www.crowdflower.com), micro-task crowdsourcing platform
F8	Figure Eight (www.figure-eight.com), micro-task crowdsourcing platform
MV	Majority Voting
ZC	ZenCrowd
ZC$_s$	ZenCrowd supervised
RY	Raykar
RY$_s$	Raykar supervised

1 Introduction

1.1 Motivation of the Deliverable

Crowdsourcing has been introduced to complete tasks that are difficult to tackle for current algorithms, e.g., entity resolution or classification. Due to the emergence of public crowdsourcing platforms, e.g., Amazon Mechanical Turk (www.mturk.com) (AMT), Crowdflower (www.crowdflower.com) (CF) (recently renamed Figure Eight (www.figure-eight.com) (F8)), running a crowdrourcing task has become an easy task. However, due to the variety of the workers' background, the crowd often yield incorrect or noisy answers. Truth inference algorithms [2, 3, 4, 5, 7] have been proposed to address this problem and to infer the correct answer (called truth). Each task is assigned to multiple workers and the truth inference algorithm aggregates the answers given by different workers. As Zheng et al. show in [8], existing aggregation algorithms are not stable across different datasets and there is no algorithm that outperforms others consistently. In this deliverable we empirically evaluate existing truth inference algorithms to find out which algorithm fits best with our fashion dataset. The remainder of this report is as follows. In Section 1 we describe the connection between this deliverable and the rest of FashionBrain deliverables. In Section 2 we describe the aggregation algorithms we use in our benchmark. In Section 3 we describe the empirical evaluation. In Section 4 we summarize the results.

1.2 Scope of the Deliverable

This deliverable (D3.2) is part of WP3 in which we apply human computation techniques to complement scalable data integration approaches designed in WP2. D3.2 extends the work in D3.1 where we create crowdsourcing interfaces to improve fashion item annotation in social media posts. In this deliverable, we focus on how the collected data can be processed in order to infer the truth. D3.2 identifies the most appropriate aggregation algorithm for different crowdsourcing tasks from the FashionBrain project. This deliverable will contribute to the following tasks:

- T4.3: Interactive stacked deep learning for crowdsourcing,
- T5.1: Scalable Crowdsourced Social Media Annotation,
- T6.3: Enrich Image Data with Human Generated Descriptions and
- T7.3: Human Computation Spin-off Opportunities

Also, D3.2 will contribute to the research challenges about Analytics for Business

Intelligence (i.e., Challenge 5: Linking Entities to Product Catalogue) and to the Core Technologies about Data Curation & Integration (i.e., CT3: Crowdsourcing interfaces and quality metrics).

This deliverable uses the fashion reviews dataset described in D2.2. This dataset contains fashion item reviews with the corresponding images of the fashion item. Each review consists of a title, a text and a language code. In this task, workers are asked to classify the reviews in three main categories. The review could be ambiguous (belong to multiple classes) and the goal is to infer the correct class for each review from workers answers.

2 Truth Inference Algorithms

2.1 Majority Voting

Majority Voting (MV) is a simple yet effective method for aggregation. This method infers the truth from the majority as it takes the answer given by the majority of workers. The main limitation of this technique is that it does not take into account worker behavior as all workers are treated equally. In real-world scenarios, workers have different levels of quality. A high quality worker answer questions carefully while a low quality worker might give random answers.

Take the following example where in a binary classification task, five workers independently label five items as follows:

	Item 0	Item 1	Item 2	Item 3	Item 4
Worker 0	0	1	1	0	1
Worker 1	1	0	0	1	1
Worker 2	0	0	1	0	1
Worker 3	1	0	1	0	0
Worker 4	0	0	0	1	1

Table 2.1: Example of a binary classification.

For the first item, two workers classified it as 1 and three workers classified it as 0. With MV, the true label for this item is then 0. Similarly, we apply the same principle on the other items and we obtain the following result:

	Item 0	Item 1	Item 2	Item 3	Item 4
True label (MV)	0	0	1	0	1

Table 2.2: True labels with MV.

2.2 Zen Crowd

ZenCrowd (ZC) [3] is an extension of the MV method, where workers are weighted according to their reliability. This approach mainly relies on the Expectation Maximization (EM) algorithm to simultaneously estimate labels and worker reliability. In particular, suppose the worker's answer is binary so the labels are noted $v_i^* \in \{0,1\}$ and the worker quality is modeled as worker probability q^w, then the conditional probability of a worker's answer v_i^w given the worker quality q^w and true labels v_i^* is:

$$Pr(v_i^w \mid q^w, v_i^*) = (q^w)^{(1-|v_i^w - v_i^*|)} \cdot (1-q^w)^{|v_i^w - v_i^*|} \tag{2.1}$$

By maximizing $Pr(V \mid q^w)$ as a function of $Pr(v_i^w \mid q^w, v_i^*)$ using the EM algorithm, ZC maximizes the likelihood of occurrence of workers answers.

To illustrate the application of the ZC method, we take the example proposed in Table 2.1. We first initialize the probability to assign class 0 or class 1 to each item as shown in Table 2.3.

	Class 0	Class 1
Item x	0.5	0.5

Table 2.3: Classes probabilities.

We then initialize the prior probability of the workers to 0.8 where we assume that all workers are reliable. In the example, we have 5 workers with probability initialized to 0.8 as illustrated in Table 2.4

	Worker 0	Worker 1	Worker 2	Worker 3	Worker 4
Init. of prior probability	0.8	0.8	0.8	0.8	0.8

Table 2.4: Workers prior probabilities.

Now, we alternate between the E-step and the M-step to update priors of the workers by taking into account the evidences accumulated. In the E-step, we compute the new probability to assign class 0 or 1 to each item and get:

	Item 0		Item 1		Item 2		Item 3		Item 4	
class	0	1	0	1	0	1	0	1	0	1
probability	0.8	0.2	0.98	0.02	0.2	0.8	0.8	0.2	0.02	0.098

Table 2.5: Item's labels with ZC (First iteration).

In the M-step, we compute the prior probability for each worker where for each worker, the probability is given by Equation 2.2 and the result is shown in Table 2.6.

$$w_m = \sum_{item=0}^{4} \sum_{label=0}^{1} \frac{P(w|item, label)}{\#items} \tag{2.2}$$

	Worker 0	Worker 1	Worker 2	Worker 3	Worker 4
Init. of prior probability	0.68	0.51	0.56	0.87	0.63

Table 2.6: Worker Prior Probabilities with ZC (First iteration).

These two steps are repeated until the maximum number of iterations is reached. In our example, after 20 iterations we obtain the probabilities for each item classification in Table 2.7:

	Item 0		Item 1		Item 2		Item 3		Item 4	
class	0	1	0	1	0	1	0	1	0	1
probability	0.999	0.001	0.001	0.999	0.001	0.999	0.999	0.001	0.001	0.999

Table 2.7: Item's labels with ZC (Last iteration).

Thus, the classification of these item with the ZC method is given by Table 2.8:

	Item 0	Item 1	Item 2	Item 3	Item 4
True label (ZC)	0	1	1	0	1

Table 2.8: True labels with ZC.

2.3 Dawid Skene and Naive Bayes

The Dawid Skene model [2] is a classical yet robust approach. Each worker w is associated with a confusion matrix. The reliability degree of a worker is the diagonal elements of the confusion matrix. The higher the degree is, the higher the probability a worker w provides a true label. The worker's answer follows the probability $Pr(v_i^w \mid q^w, vi^*) = q^w_{v_i^*, v_i^w}$. Similar to ZC, the Dawid Skene model maximizes the probability $Pr(V \mid q^w)$ using the EM algorithm. This model captures worker behavior as a function of each example's true class. The Naive Bayes model [6] is an extension of the Dawid Skene model where they consider the fully-supervised case of maximum-likelihood estimation with Laplacian (add-one) smoothing.

In what follows, we illustrate the application of the Dawid Skene model on our running example from Table 2.1. We start by initializing the item labels using MV as shown in Table 2.2 and proceed to the M-step which consists of estimating the class prior and confusion matrices based on the current beliefs about the true labels of each item. The class prior in our example is calculated as follows:

$$p_0 = \frac{3}{5} = 0.6; p_1 = \frac{2}{5} = 0.4 \tag{2.3}$$

Now, we need to compute the confusion matrices for each worker with the following formula:

$$\pi^w_{i,j} = \frac{\sum_{i=0}^{5} T_{n,i} t^w_{n,j}}{\sum_{j=0}^{5} \sum_{i=0}^{5} T_{n,i} t^w_{n,j}} \tag{2.4}$$

Where $T_{n,i}$ represents the true labels given by the E-step and $t^w_{n,j}$ is the class labels for worker w. For example for worker 0, the true label $T_{n,i}$ is shown in Table 2.9 and his classification of items is given in Table 2.10:

1	0
1	0
0	1
1	0
0	1

Table 2.9: True labels given by the E-step.

1	0
0	1
0	1
1	0
0	1

Table 2.10: Class labels of a Worker.

By applying Equation 2.4, we obtain the confusion matrix illustrated in Table 2.11 for worker 0:

0.67	0.33
1	0

Table 2.11: Confusion Matrix of a Worker.

The E-step and M-step are repeated until convergence. For this example, we obtain the results shown in Table 2.12:

	Item 0	Item 1	Item 2	Item 3	Item 4
True label (Dawid Skene)	0	1	1	0	1

Table 2.12: True labels with DW

2.4 Raykar

Raykar [5] proposes a Bayesian approach to add worker specific priors to each class. Similarly to ZC and Dawid-Skene, the EM algorithm is applied to simultaneously estimate labels and model parameters. The Raykar model's novelty consists on using a feature representation of each example and infer the labels with automatic classifier. This model applies only when the task consists of a binary classification. We start by randomly initializing the confusion matrix for each worker as follows: We associate each worker with the confusion matrix given in Table 2.13:

0.7	0.3
0.3	0.7

Table 2.13: Confusion matrix of a worker.

We associate for each class the same probability as in equation 2.5:

$$p_0 = 0.5; p_1 = 0.5. \tag{2.5}$$

We randomly initialize the probability of classification of each item as show in Table 2.14:

	Item 0		Item 1		Item 2		Item 3		Item 4	
class	0	1	0	1	0	1	0	1	0	1
probability	0.2	0.8	0.4	0.6	0.4	0.6	0.6	0.4	0.8	0.2

Table 2.14: Item's labels with Raykar (First iteration).

We start by updating the class probabilities according to the Beta distribution, the new probabilities for each class are given in Equation 2.6

$$p_0 = 0.48; p_1 = 0.52. \tag{2.6}$$

Next, we apply the same E-step and M-step described in the Dawid Skene model. Finally, we obtain the true labels shown in Table 2.15.

	Item 0	Item 1	Item 2	Item 3	Item 4
True label (Raykar)	0	1	1	0	1

Table 2.15: True labels with Raykar

3 Experiments

3.1 Experimental Setup

In D4.2 of the FashionBrain project, we launched a crowdsourcing experiment on F8 to classify fashion items reviews provided by Zalando (this dataset is described in detail in D2.2) into three main classes: Size issue (class 0), Fit issue (class 1) and other (class 2). This task was designed to capture the language expressing a sentiment about a “Size” or a “Fit” problem. The review should be classified to “Size issue” if the review contains a negative feedback about the item’s size, which means the item’s size is either too large or too small compared to the regular one. (e.g. “this is not XL”, “bigger than expected”). The review should be classified to “Fit issue” if the review contains a negative feedback about how the item fits to the customer (e.g. “The shoes really rub on your ankle”, “The sleeves are too long”). Otherwise, the item should be classified to class 2 “Other”.

We have 1480 reviews, each one of them is shown at least to 5 workers who need to classify it into one of the three above-mentioned classes. Among the 1480 reviews, we use 300 reviews as “test” questions which means we label these questions in advance and if a worker misses more than 30% of these questions, s/he is considered as “low quality” worker and s/he is automatically eliminated from the task.

In the F8 platform, we specify that only workers from Europe can participate to the task. We gathered 6028 judgments from 252 workers. We use the open-source aggregation algorithms provided in the “Square” Benchmark [1].

3.2 Performance Evaluation on a Multi-Class Task

ZC was originally proposed as an unsupervised method but it can easily be used as a supervised method by maximum-likelihood called “ZenCrowd supervised (ZC$_s$)”. We evaluate the performance of the unsupervised version of ZC with MV and two supervised methods: Bayes and ZC$_s$. The performance is evaluated in terms of both efficiency by measuring the run time and in terms of accuracy by measuring the precision, the recall and the F1-score. All results are presented in Table 3.1.

The results show that ZC$_s$ achieves the best results in terms of Precision while MV has the highest recall while being the most efficient technique. ZC is the least efficient technique in both unsupervised (100x less efficient than MV) and supervised methods (94x less efficient than Bayes). For the case of unsupervised methods, ZC computes the worker quality modeled as the worker probability and maximizes the likelihood of workers answers. Majority voting instead simply infers the truth from

	Precision	Recall	F1-score	Run Time (ms)
MV	0.70	0.79	0.920	12.08
ZC	0.72	0.73	0.924	1239.61
ZC_s	0.73	0.71	0.924	4755.79
Bayes	0.68	0.78	0.917	50.45

Table 3.1: Performance evaluation for multi-class task.

the majority of workers which explains why MV is much more efficient than ZC. For the case of supervised methods, the Bayes method uses a similar approach to ZC except that it applies the Laplacian smoothing to reduce the sparsity of the matrix which makes the Bayes method much more efficient than ZC. As a result, MV is the most efficient technique while ZC is the most accurate technique for the fashion reviews classification. However, workers do not have the same quality of answers. Moreover, this result is only valid in the case we have at least five answers per review. In the case of smaller number of answers, there is no guarantee that ZC would still give the most accurate result. In the following sections, we study these factors for truth inference.

3.2.1 Worker Quality

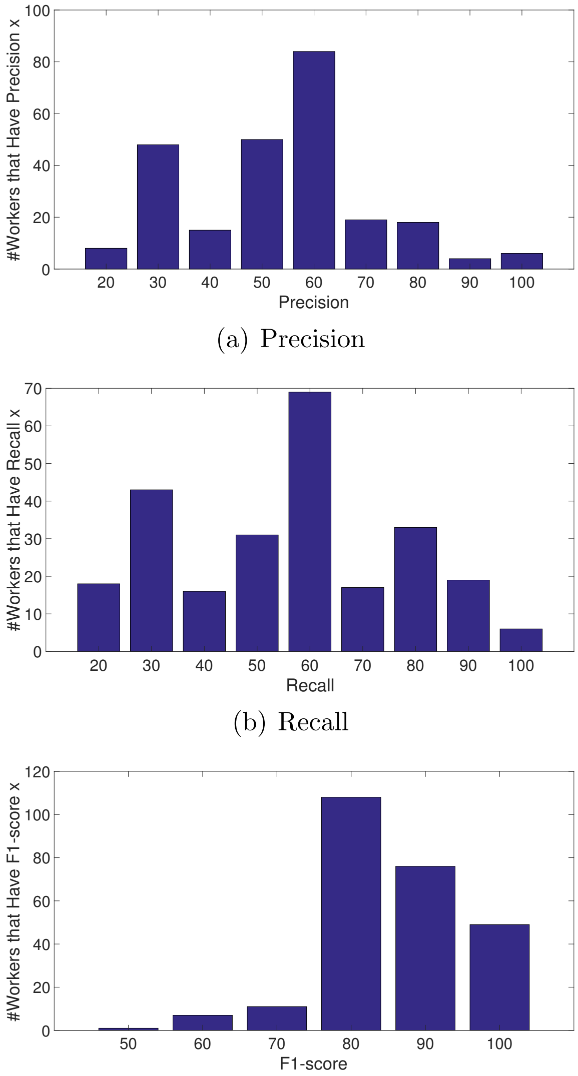
We compute each worker's precision, recall and F1-score. The higher these values are, the better the quality of the worker is. For each of these metrics, we compute the corresponding metric for each worker and draw the histograms of the results.

The results of the experiment in Figure 3.1 show show that the majority of workers have a precision higher than 40% and 57% of the workers have a recall higher than 50%. The overall result is reflected in the F1-score where 92% of the workers have an F1-score higher than 70%.

3.2.2 Varying Data Redundancy

We define Data redundancy as the number of answers collected for each task. In our fashion dataset, we have at least 5 answers for each task. We evaluate the quality of each one of the four algorithms with varying data redundancy. We vary the data redundancy $r \in [3, 5]$ where we select randomly r answers and construct a dataset with the selected answers. We also take the entire set of available answers ($r = 6+$) and we consider the case where we have initially 3 randomly selected answers and in case workers have a strong disagreement, (i.e each worker classify the review to a different class), we add 2 more answers. Then, we run each method on the constructed dataset and measure Precision, Recall and F1-score.

The results of the experiment in Figure 3.2 show that in terms of precision, ZC performs best even for only 3 randomly selected answers. This result is expected as



(a) Precision

(b) Recall

(c) F1-score

Figure 3.1: The Statistics of Worker Quality.

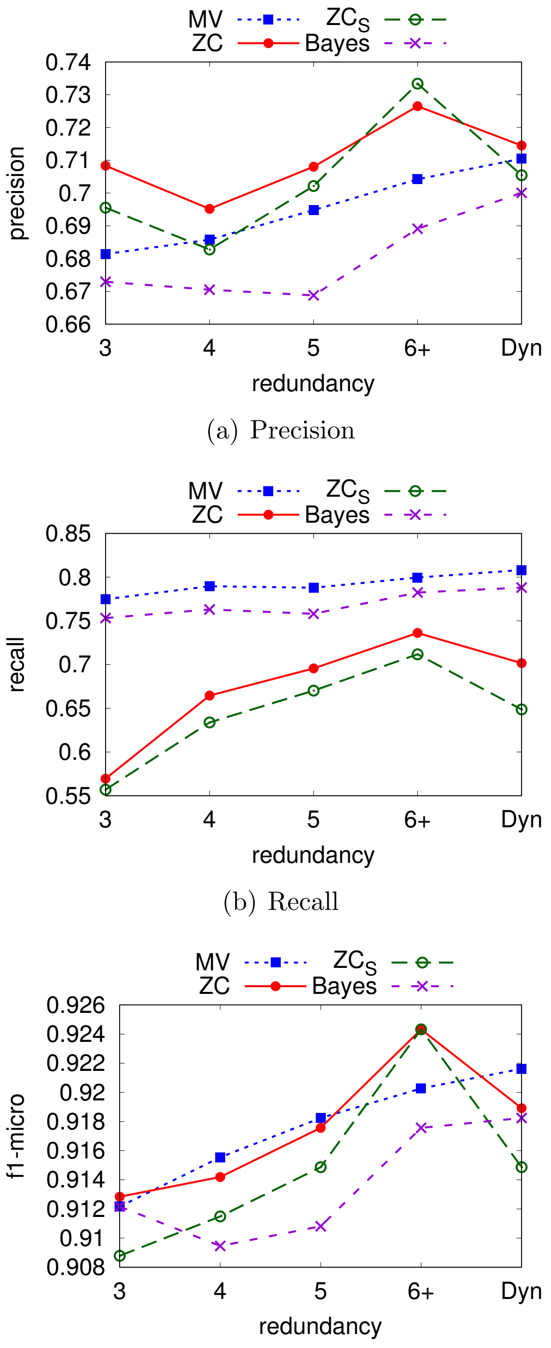
ZC takes into account the worker reliability to infer the truth so it is more robust to smaller number of answers unlike MV for example that relies only on worker's agreement. In terms of recall, MV performs best which means workers agreement reduces the number of false negatives and relying on workers agreement is more beneficial than modeling workers reliability. In terms of F1-score, the best result is achieved with both supervised and unsupervised versions of ZC for more than 6 answers per task where F1 is 0.924. It is worth noting that with the dynamic method, we obtain an F1-score of 0.921 with MV. With this method where we start with 3 workers and require additional 2 workers only in case of strong disagreement, we achieve a close to optimal F1-score with MV while asking a total of five workers only when needed which is around 20% of the dataset.

3.2.3 Varying Task Complexity

We define the level of complexity of a crowdsourcing task by the number of workers who agree on one answer. For the FashionBrain task, 5 workers answer each task and we distinguish three levels for the task complexity:

- Trivial: If all five workers give the same answer.
- Easy: If four out of the five workers give the same answer.
- Normal: If three out the five workers give the same answer.

The results of the experiment in Figure 3.3 show that the performance of all algorithms decreases as the complexity of the task increases. For the trivial and easy classes, the precision, recall and the F1-score are almost the same for all methods which means that all methods have the same performance when there is an agreement between workers. The difference between these methods is more visible when the task is classified as normal. In this case, ZC has the highest precision and MV has the highest recall. The supervised and unsupervised ZC has the highest F1-score value which supports the choice of ZC as an aggregation methods when there's a disagreement between workers.

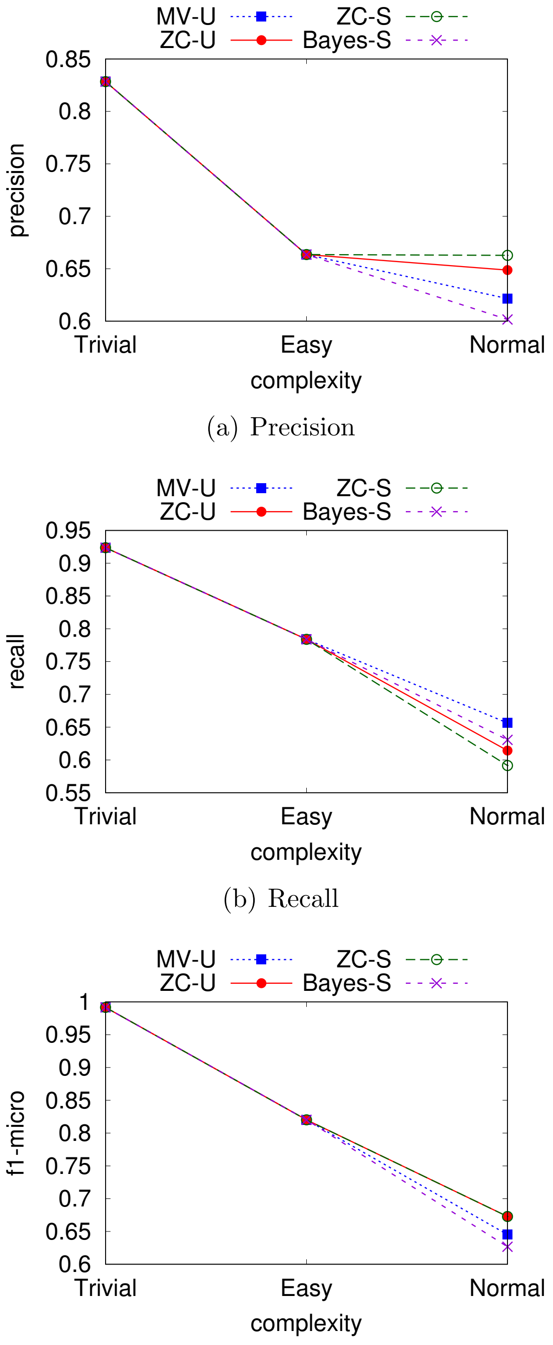


(a) Precision

(b) Recall

(c) F1-score

Figure 3.2: Quality Comparisons on the review classification task.



(a) Precision

(b) Recall

(c) F1-score

Figure 3.3: Complexity of the review classification task.

3.3 Performance Evaluation on a Binary Classification Task

In this Section, we map the previous setup to a binary case where given a review, workers have to say if the review contains a “Size/Fit issue” or not. In addition to the methods used in the previous section, we also use Raykar (RY) for binary classification. Similarly to ZC, RY was proposed as an unsupervised method but can be easily extended to be a supervised method Raykar supervised (RY$_s$). We evaluate the performance of MV, ZC, RY and Bayes in terms of both efficiency and accuracy. All results are presented in Table 3.2

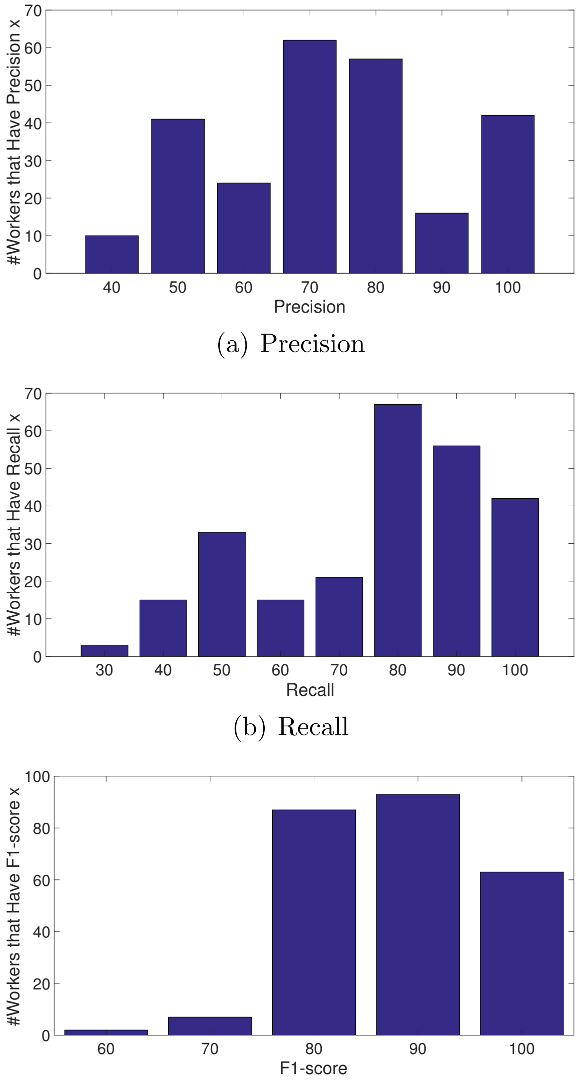
	Precision	Recall	F1-score	Run Time (ms)
MV	0.806	0.902	0.926	9.1
ZC	0.811	0.885	0.927	707.96
ZC$_s$	0.809	0.863	0.925	1226.29
RY	0.804	0.891	0.925	523.29
RY$_s$	0.805	0.907	0.925	726.72
Bayes	0.804	0.904	0.925	14.51

Table 3.2: Performance evaluation for binary classification task.

The results of the experiment in Table 3.2 show that in terms of precision, the unsupervised version of ZC achieves the best result while in terms of recall, the supervised version of Raykar RY$_s$ has the highest values. Using the F1-score, all methods achieve same performance (around 0.925) except for MV with an F1-score of 0.926 and ZC with an F1-score of 0.927. In terms of efficiency, MV is the most efficient method, and it is 77x faster than ZC, 134x faster than ZC$_s$, 57x faster than RY and 80x faster than RY$_s$. The closest one to MV in terms of efficiency is Bayes as it is 1.6x slower than MV. Worker’s agreement seem to be the best approach to use when it comes to binary classification and hence Majority Voting is the best method for this case. To have a better understanding of the factors influencing the truth inference of binary classification tasks, we explore in the next section the effect of worker’s quality and task complexity on the aforementioned methods performance.

3.3.1 Worker Quality

As shown in Figure 3.4, workers achieve a better performance with binary classification than with multi-class classification. 70% of the workers have a precision higher than 60% and more than 60% have higher recall than 70%. The F1-score is strongly improved in the case of binary classification where 96% of the workers have an F1-score above 70%.



(a) Precision

(b) Recall

(c) F1-score

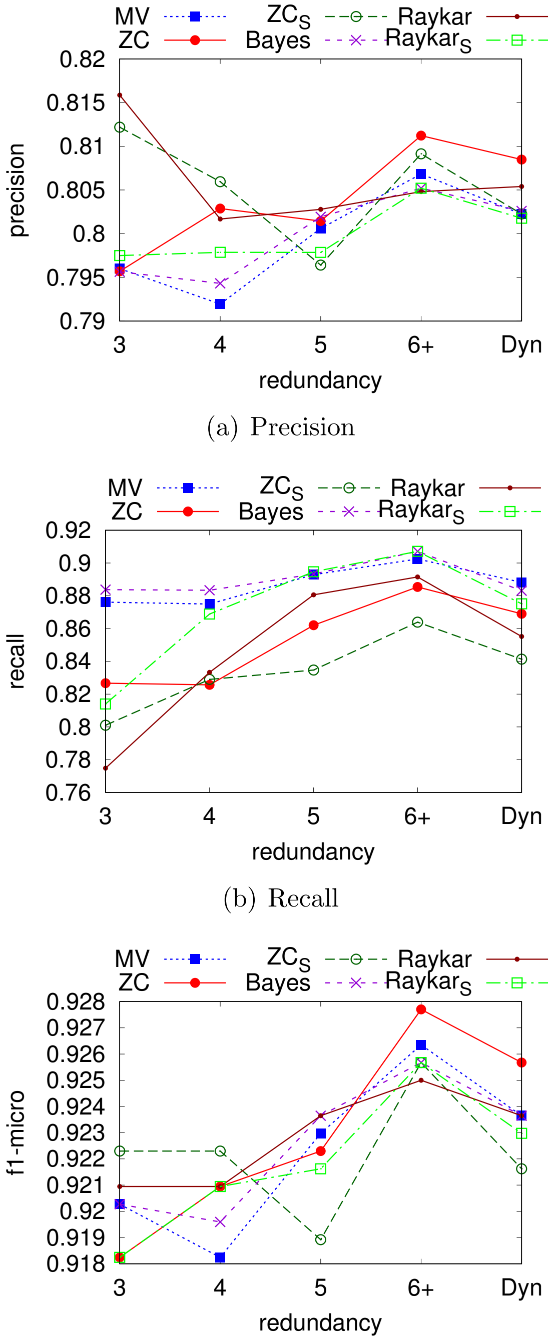
Figure 3.4: The Statistics of Worker Quality.

3.3.2 Varying Data Redundancy

In Figure 3.5, we use a similar setup to the one described in the multi-class task, where we vary the number of workers to whom we show the reviews. In terms of precision, the unsupervised version of RY achieves the best precision for only 3 workers per task. In terms of recall, the supervised version of Bayes outperforms all other methods and achieves the best result for 6+ answers per worker. When measuring the F1-score, we notice that all methods achieve similar performance where the F1 score ranges between 0.918 and 0.928. The best F1-score value obtained is 0.927 with ZC.

3.3.3 Varying Task Complexity

In this section, we classify reviews by the level of agreement as we did for the multi-class task. As shown in Figure 3.6, we obtain a similar result for both classes Trivial and Easy where all algorithms have the same performance. The difference is more noticeable when the task is classified as normal and when measuring the Precision where RY_s has the best precision 0.71 compared to ZC_s with a precision of 0.67.

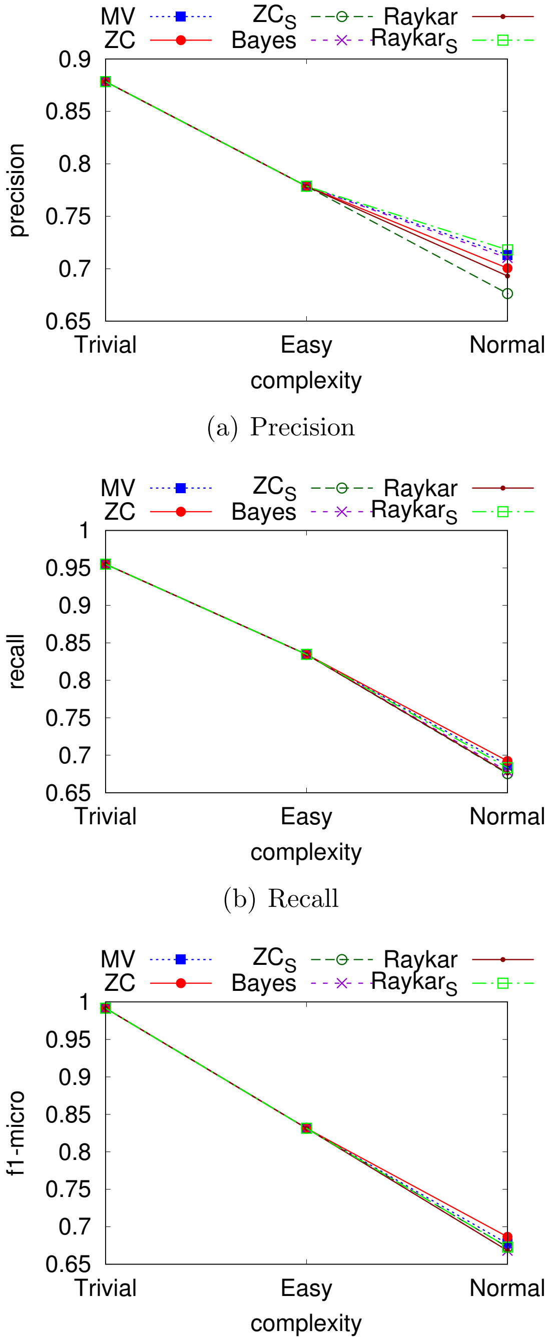


(a) Precision

(b) Recall

(c) F1-score

Figure 3.5: Quality Comparisons on the review classification task.



(a) Precision

(b) Recall

(c) F1-score

Figure 3.6: Complexity of the review classification task.

4 Conclusions

In this deliverable, we have evaluated the state of the art aggregation methods for a review classification task using fashion datasets. The evaluated methods rely either on workers agreement and/or worker's reliability to infer the truth. The results of our empirical evaluation show that i) MV is the most efficient method and ii) ZC is the most robust technique where it achieves high precision even with a small number of asked workers. The outcome of this deliverable will be used to identify which aggregation algorithm(s) should be used for the different FashionBrain tasks that involve crowdsourcing such as entity resolution or sentiment analysis.

Bibliography

[1] Square benchmark v2.0, 2013. Square, homepage: http://ir.ischool.utexas.edu/square/index.html.

[2] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[3] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478. ACM, 2012.

[4] Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. Cdas: a crowdsourcing data analytics system. *Proceedings of the VLDB Endowment*, 5(10):1040–1051, 2012.

[5] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[6] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

[7] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[8] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *PVLDB*, 10(5):541–552, 2017. doi: 10.14778/3055540.3055547. URL http://www.vldb.org/pvldb/vol10/p541-zheng.pdf.