Learning Multi-Label Classification from Data Annotated with Unique Labels

Gargi Roy, Lipika Dey

Tata Consultancy Services Limited, Research and Innovation, Kolkata, India
roy.gargi@tcs.com, lipika.dey@tcs.com

Abstract. Classifying consumer-generated text like feedbacks, surveys or support emails has gained tremendous popularity in recent years as these contain important information about customer problems, opinions or processes. Given the volumes and velocity at which such text is generated, manual analysis is impossible. The most common analytical technique applied is classification of these texts into known categories, determined based on domain expertise. However, each piece of text may contain many different issues which even human agents fail to detect correctly. They usually classify the text based on the most important issue or the one occurring first while ignoring the others. The most suitable method for automatically analyzing these texts is to use multi-label classification. However, the training data available usually has single labels associated with them. This work proposes a classifier that learns from data annotated with single labels but predicts multi-label outputs along with confidence values. The proposed method is evaluated on different types of data sets, including those with noisy texts. Multi-labeled output provides richer insights.

Keywords: Classification · Multi-label · Unique annotation · Supervised term weighting.

1 Introduction

As digitization grows rapidly, there is an increasing demand for intelligent text classification to automatically handle large volumes of consumer-generated text in the form of emails, customer complaint logs, call-logs, product reviews on social media and so on. Despite the existence of a multitude of techniques, classification of these kinds of noisy texts remains a challenging task. Real world documents like these need to be assigned multiple labels corresponding to the multiple categories of content contained in them, which most of the existing techniques cannot do. A major challenge is due to the lack of properly annotated multi-label training data. Class boundaries are also ill-defined and overlapping in real data. Manual annotation is often noisy and incomplete in a sense that a document which actually belongs to multiple classes usually gets single class annotation which corresponds to either the most obvious issue or the first issue that is mentioned in the text [6].

This is elaborated with the following customer complaint log from a mobile operator. *"Dear Sir I m buy a new (mobile phone name) on (date). on the box (service provider (SP) name) free data offer and i used already a (SP name) GSM sim (sim number) and i use this sim in the (phone name) but data offer 500mb/month for 6 month not activate on my (SP name) no. and i call (SP name) customer care they don't answer my problem. and i go to nearest (SP name) store they not listen my problem properly. so i m kindly request you plz solved my problem soon..."* [9]. This document is categorized by human agents as belonging to category "Internet Access Issue" and hence assigned that label. However, ignoring the noise, it can be concluded that though this complaint primarily talks about Internet access issue related to the sim card, it also contains customer service related issues. If this document, and many similar to this one, are all assigned only to the single class of "Internet Access Issue", then the issue of bad customer service will never be highlighted. In fact, content pertaining to "customer service" may figure only in conjunction with a main functional business class, rather than by itself. Thus the problem we are trying to address can be defined as one which will learn from training instances uniquely classified as "Internet access issue", "sim related issue" or "customer service issue" etc. in the past - but assign more than one label correctly to future documents. It may be noted that future documents may contain unique and varying combinations of class labels which have not been seen before. None of the existing multi-class or multi-label classifiers can handle learning from these types of incomplete labeling [3], [30], [33].

In this paper, we propose a classification method that learns to assign multiple class labels to unseen new documents even though it is trained with documents that had been assigned a single label. The proposed method thus corresponds to a scenario of learning from noisy or incompletely labeled training data. The algorithm uses supervised term weights learned from single-labeled (fixed set of labels) training samples. The weights capture both the class-discriminating and class-representative powers of words. During classification, the relative distribution of different class labels within a given text is computed based on the strengths of each label assessed independently. The distribution is further analyzed to assign confidence measures to each label. The uniqueness of the proposed method lies in the fact that it can address the difficult task of learning fine-grained classification of content into multiple buckets from noisy training instances that are incompletely labeled.

Our method is also explainable so that we can locate portions/words of the text corresponding to the different class labels along with their corresponding importance. This is an important activity to handle consumer-generated text within a support environment, as described earlier in the example. Though deep learning based text classification techniques have gained wide popularity, the lack of explainability for these methods often make them unsuitable for adoption in such cases. For business applications traceability of a decision to its origin is mandatory in order to resolve possible future disputes.

We have conducted extensive experiments and show that the proposed method outperforms several other classification algorithms for noisy data. The techniques can also handle data with rare classes and can be easily extended to handle numeric or mixed data. We have compared our results with that of [20] which learns a fuzzy classifier for multi-label classification from single labeled mixed and numeric data. This paper is closest to our work but does not report any result for classifying unstructured data.

The rest of the paper is organized as follows. Section 2 presents a review of related work. Section 3 presents the proposed term weighting mechanism. Section 4 presents the proposed classification technique. Section 5 presents results of experimentation with different data sets.

2 Literature Survey

Though many multi-label classification techniques have been proposed in the past, very few have considered the task of learning to assign multiple labels to unseen elements while training with uniquely labeled instances. None of the earlier work have reported results for text documents. A review of multi-label classification algorithms is presents in [33] which includes techniques that divides the problem into a set of binary classification problems [32] and multi-label ranking approach [11] considering pairwise relations between labels. A fuzzy multi-label classification technique is presented in [10] for noisy data. In [20], a method is proposed to turn discriminative single-task classification into generative multi-task classification by adopting a fuzzy classification approach for numeric and mixed data. Authors in [29] also propose methods to provide crisp decisions for single labels and soft decisions for multi-labels while learning from single labeled numeric data. Methods for estimating confidence of binary classification output for a Case-Based Spam Filter is presented in [12].

3 Text Pre-Processing and Term Weighting

Text Preprocessing: Since consumer-generated digital text is fraught with noise, text cleaning is necessary. An array of noise removal modules proposed in [13] are implemented to improve the quality of text for classification purposes. These include handling text import errors such as presence of html code fragments, unicode characters, unnecessary duplication of characters or text elements that are found in social media. A separate module is implemented to remove mandatory yet unwanted pieces text like signature or disclaimers within emails, advertisement or anchor text in web pages etc. The preprocessing module also removes unconventional use of punctuation marks, symbols and emoticons. Finally, a domain-dependent spell corrector, also explained in [13], is employed to automatically correct misspelt words. This module first learns the frequently occurring non-dictionary words of a domain from the training set using the statistical distribution of the words and then corrects the misspelt words based on

their similarity to the extended dictionary. This is an important step since mistakes or variations in words can drastically affect the word weights and in turn affect the performance of the classifier. Stop-words are removed and all words are stemmed using [25] before computing the weights.

Term Weighting: In this work we have used a term weighting scheme that considers two aspects of a word - (i). its class-discriminating power and (ii). its class-representative power. The class-discriminating power of a word is measured using Inverse Gravity Moment (IGM) introduced in [7]. Equation 1 presents the computation of IGM, $f_{k1} \geq f_{k2} \geq ... \geq f_{kp}$ denote the total number of occurrence of the term t_k in each of the classes sorted in descending order with p number of class labels, λ is an adjustable coefficient.

$$IGM = 1 + \lambda \cdot (\frac{f_{k1}}{\sum_{r=1}^{p} f_{kr} \cdot r}) \tag{1}$$

It can be seen that the higher the non-uniformity of distribution of a term across classes in the whole corpus, the greater is its class distinguishing power. The class-representative power is captured as a function of intra-class frequency. We have worked with following three different weighing mechanisms (denoted by $w^d_{(L/N/R)TF-IGM}(t_k)$) for term t_k, where t^d_k denote the number of occurrence of term t_k in document d. *i*) NTF: which uses the term frequency i.e. the raw term count normalized by document length.

$$w^d_{NTF-IGM}(t_k) = (\frac{t^d_k}{\#terms\ in\ d}) \cdot (1 + \lambda \cdot (\frac{f_{k1}}{\sum_{r=1}^{p} f_{kr} \cdot r})) \tag{2}$$

ii) LTF: uses the logarithm of raw term count, thus introducing a damping factor that does not allow the weight of a word to grow linearly with its frequency in a class.

$$w^d_{LTF-IGM}(t_k) = log(t^d_k + 1) \cdot (1 + \lambda \cdot (\frac{f_{k1}}{\sum_{r=1}^{p} f_{kr} \cdot r})) \tag{3}$$

iii) RTF: uses the square root of raw term count which is another damping function.

$$w^d_{RTF-IGM}(t_k) = \sqrt{t^d_k} \cdot (1 + \lambda \cdot (\frac{f_{k1}}{\sum_{r=1}^{p} f_{kr} \cdot r})) \tag{4}$$

IGM is calculated once during parsing the training corpus and saved.

4 Proposed Classification Method

Given a set of documents D where each document is associated with a unique label from a set of class labels C, a term vocabulary T of size n and a term-document weight computation mechanism as stated earlier, we now present how multi-label class associations are generated for a new document d.

Let us assume that there are p number of class labels, $C = \{c_1, c_2, ..., c_p\}$. Let D_i denote the subset of documents in D that have the single class-label c_i associated to them. Let $\{o_1(d), o_2(d), ...o_p(d)\}$ denote the multi-class membership values that are initially computed for d for all classes, based on the words contained in it. Computation of $o_i(d)$ is computed as a function of the term weights for each class label i, which in turn is computed as an aggregate of the term-document weights for all training instances labeled with i.

$$o_i(d) = \sum_{j=1}^{m} \widehat{z}^{D_i}(t_j) w^d_{(N/L/R)TF-IGM}(t_j) \tag{5}$$

where m denotes the number of terms from T contained in d, $IGM^D(t_j)$ is the IGM value computed for term t_j in the document set D, $w^d_{(N/L/R)TF-IGM}(t_j)$ is computed as follows:

$$w^d_{(N/L/R)TF-IGM}(t_j) = w^d_{(N/L/R)TF}(t_j) * IGM^D(t_j) \tag{6}$$

$\widehat{z}^{D_i}(t_j)$ denotes the relative significance of term t_j for class c_i against all classes, computed from class-association of documents in D_i

$$\widehat{z}^{D_i}(t_j) = \widehat{y}^{D_i}(t_j) / \sum_{r=1}^{(p)} \widehat{y}^{D_r}(t_j) \tag{7}$$

where, $\widehat{y}^D_i(t_j)$ is computed as a normalized weight with respect to the vocabulary T as shown below where, $v^{D_i}(t_j)$ is the weight computed for a term t_j using one of the term weighting schemes presented in the section 3 :

$$\widehat{y}^{D_i}(t_j) = v^{D_i}(t_j) / \sum_{m=1}^{n} v^{D_i}(t_m) \tag{8}$$

For an imbalanced data set, where the class distribution for the instances is very skewed, an additional normalization is done as follows.

$$\widehat{y}^{D_i}(t_j) = \widehat{y}^{D_i}(t_j) / max(\widehat{y}^{D_i}(t_{j=1\ to\ n})) \tag{9}$$

Algorithm 1 presents the computational steps. The class membership values obtained for d, are further normalized as shown in Step 7 of the algorithm and the class membership distribution is stored as a vector $\mathbf{X}$. Algorithm 1 has complexity $O(pm^2)$.

ALGORITHM 1: ComputeClassMembershipDistribution(D, T, d)

Input : D, T, d
Output: $\{o_1, o_2, ...o_p\}$ for d

1 Compute $\mathbf{Y_{p\times n}}$ from D, $\widehat{y}_{ij} \leftarrow v_{ij} / \sum_{j=1}^{n} v_{ij}$ where $v_{ij} \leftarrow \Sigma_{document\ d' \in D\ has\ label\ i}(w(t_j^{d'}))$;
2 **if** *if imbalanced data* **then**
3 $\quad \widehat{y}_{ij} = \widehat{y}_{ij} / max(\widehat{y}_{ij=1\ to\ n})$
4 Compute $\mathbf{Z_{p\times n}}$ from D where $\widehat{z}_{ij} = \widehat{y}_{ij} / \sum_{i=1}^{p} \widehat{y}_{ij}$;
5 Calculate term weight vector $\mathbf{N_{1\times m}}$ from d;
6 for d compute membership value for each class in matrix $\mathbf{O_{p\times 1}}$ where $\mathbf{O}_{p\times 1} = \mathbf{Z}_{p\times m}\mathbf{N^T}_{m\times 1}$;
7 Normalize class membership values, $o_i(d)^{updated} = o_i(d) / \sum_{i=1}^{p} o_i(d)$

Final Prediction with Confidence Category and Score: The distribution vector $\mathbf{X}$ which contains the strength of each class in the document is analyzed to generate the final predicted label set along with a confidence category and score associated with the predicted label set. We now define few terms. Given p number of class labels and the class membership distribution for a new instance as $\mathbf{X} = \{x_1, x_2, ..., x_p\}$, let $x_\mu, \bar{x}, \sigma^2, \gamma, \kappa$ denote the maximum value, mean, variance, skewness and kurtosis of the distribution $\mathbf{X}$ respectively. Let $\dot{x_i}$ denote the percentage deviation of $x_i \in \mathbf{X}$ from mean of $\mathbf{X}$, which is,

$$\dot{x_i} = \frac{x_i - \bar{x}}{x_i} \cdot 100 \tag{10}$$

We then compute a score $\psi(x_i)$ for each $x_i \in \mathbf{X}$ as follows which takes into account the relative label strengths of the predicted labels with respect to the mean along with the number of labels, variability and degree of symmetry of the distribution and combined weight of the tails relative to the rest of the dis-

tribution to give insight about the significant peak in a distribution considering several structural characteristics of the distribution.

$$\psi(x_i) = \frac{\dot{x_i}}{100} \cdot p \cdot \sigma^2 \cdot |\gamma + \kappa| \tag{11}$$

Let $\mathbf{X}_i^{\eta}$ denote the set of values within $\eta\%$ boundary of x_i in the distribution $\mathbf{X}$ derived as follows

$$\mathbf{X}_i^{\eta} = \{y_i\} such\ that\ y_i \in \mathbf{X}\ and\ y_i \in (x_i, \frac{x_i \cdot (100 - \eta)}{100}) \tag{12}$$

Let α and β be two parameters which denote the upper and lowers boundaries in determining different confidence categories. α, β and η are parameters that control the degree of flexibility or rigidity allowed while interpreting the results. Now we present the determination of the final output label set along with the confidence categories using the above mentioned terms. The first four categorization is done for $|\sum_{i=1}^{p} \psi(x_i)| > (0 + \rho), x_i \in \mathbf{X}$, where ρ is an adjustable parameter.

i) Single Label with Very High confidence (SLVH): When the class-membership distribution has a distinct peak which is significantly higher than the rest satisfying the following conditions, then the class-label corresponding to the peak is inferred with very high confidence.

$$((\dot{x}_\mu > \alpha) \wedge (\psi(\dot{x}_\mu) > 0)) \wedge ((|\mathbf{X}_\mu^{\eta}| = 0) \vee ((|\mathbf{X}_\mu^{\eta}| > 1) \wedge ((\nexists x_i \in \mathbf{X}_\mu^{\eta}) \wedge (\dot{x}_i > \alpha)))) \tag{13}$$

ii) Multi-Label with High confidence (MLH): The class distribution has multiple high peaks satisfying the following conditions, those high-valued labels are predicted with high confidence.

$$((\dot{x}_\mu > \alpha) \wedge (\psi(\dot{x}_\mu) > 0) \wedge (|\mathbf{X}_\mu^{\eta}| > 1)) \wedge (\exists x_i \in \mathbf{X}_\mu^{\eta} \wedge (\dot{x}_i > \alpha) \wedge (\psi(\dot{x_i}) > 0)) \tag{14}$$

iii) Single Label with Medium confidence (SLM): The distribution has a maximum peak (satisfying the following conditions), which is not significantly higher than the mean, then the peak-valued label is predicted with medium confidence.

$$((\alpha \geq \dot{x}_\mu > \beta) \wedge (\psi(\dot{x}_\mu) > 0)) \wedge ((|\mathbf{X}_\mu^{\eta}| = 0) \vee ((|\mathbf{X}_\mu^{\eta}| > 1) \wedge (\nexists x_i \in \mathbf{X}_\mu^{\eta} \wedge (\alpha \geq \dot{x}_i > \beta)))) \tag{15}$$

iv) Multi-Label with Medium confidence (MLM): The distribution contains more than one peak, satisfying the following conditions, though not one with very high amplitude, multiple labels corresponding to the peaks are output, with medium confidence.

$$((\alpha \geq \dot{x}_\mu > \beta) \wedge (\psi(\dot{x}_\mu) > 0) \wedge (|\mathbf{X}_\mu^{\eta}| > 1)) \wedge (\exists x_i \in \mathbf{X}_\mu^{\eta} \wedge (\alpha \geq \dot{x}_i > \beta) \wedge (\psi(\dot{x_i}) > 0)) \tag{16}$$

v) Reject Classification for LOW confidence (RCLC): The distribution is flat type with almost uniform low values for each label, the prediction result is not accepted due to low confidence. The condition is as follows.

$$(\beta \geq \dot{x}_\mu) \wedge (|\sum_{i=1}^{p} \psi(x_i)| \approx 0, \forall x_i \in \mathbf{X}) \tag{17}$$

The final label set contains the predicted labels in order such that the first label is the most significant one and so on. After the final label set determination, the confidence score is computed in the following manner.

$$s = (Avg(\psi(x_i), \forall x_i \in output\ label\ set) * (|\sum_{i=1}^{p} \psi(x_i)|, \forall x_i \in \mathbf{X}) \tag{18}$$

Finally, the scores are normalized.

5 Results and Comparative study

This section presents the dataset description, classifier results along with comparative results with another relevant work.

Evaluation scheme: The evaluation of this work can not be done in the straight forward manner. We have evaluated our results from different aspects. As no baseline is available for this kind of work for unstructured data, we first compute the standard performance measures such as precision, recall, macro F1 score, accuracy using the first label which is also the most significant label in the output label set in ten fold cross validation setup. Then using the multi-label output set, the overall prediction performance is updated by updating the confusion matrix as when given annotation does not match with the first label with highest membership value but matches with the second or third highest in the multi-label output. We have compared our results with the other reported works including deep learning techniques for 20 Newsgroup data. We also have analysed the labels from multi-label output set to see if they have some similarities among them which depicts the relevance of the output with respect to the given text. Then we analyse the interpretation of the predicted multi-labels by inspecting which words or portion of text have contributed to which predicted label with how much weight to verify the results. Our method is extended for structured data and results are compared with an existing baseline for structured data.

Dataset Description: Ten uniquely labeled datasets are considered for the experimentation. Among them eight are text datasets covering noisy, class overlapping, class imbalanced, short text, long text datasets and two are numeric and mixed datasets, Autos and Zoo, retrieved from UCI repository [19].

Results reported by	Classifier	Macro F1	Accuracy
[17]	Naive Bayes	83	
	Rocchio	78.6	
	K-NN	81.2	
	SVM	86	
[2]	SVM	78.19	
	L Square	83.05	
[24]	SVM (CS&T)	82.4	
	LR (CS&T)	81.5	
[16]	RSV-NN	83	
[15]	GE1-MNB	63	
	MaxEnt	79	
[8]	LSTM		82
	LM-LSTM		84.7
	SA-LSTM		84.4
[31]	SC-LSTM-P		82.98
[4]	CNN2		80.19
This paper	-	**84.7**	**84.87**

Table 1. Performance measures of 20 Newsgroup data from other reported works and this paper.

Text datasets are as follows. *1*) full 20 Newsgroups corpus data [18] having around 20,000 instances with 20 classes. *2*) 'DisjointNews': 5000 instances from five top level disjoint classes within 20 Newsgroups corpus. *3*) 'ScienceNews': 4000 instances within the science class of the 20 Newsgroups corpus. *4*) 'CompScNews': 5000 instances within the computer class of the 20 Newsgroups corpus having class overlaps. *5*) 'HR Internal': internal enterprise email communications corresponding to several enterprise activities (16,356 instances with seven classes, due to confidentiality issue class names are made anonymous). *6*) 'Telecom': telecom customer complaint data from a publicly accessible consumer complaint management website [14] used in [10] (500 instances and six classes). *7*) 'IMDB': IMDB movie review sentiment dataset,

proposed by [21] [1] (25,000 instances with two classes). *8*) 'RT': Rotten Tomatoes dataset [23][2] (10.662 instances, short text like one line, with two classes). 5 and 6 have non-uniform class distribution with rare classes and multi-label aspect in significant amount.

Prediction: Ten fold cross validation is done on the datasets to generate average standard performance measures. Results are generated using four different term weighing schemes such as considering only local factor NTF and then considering both local and global factor LTF-IGM, NTF-IGM, RTF-IGM using $\lambda = 7$ (as empirically found and used in [7]).

Table 1 presents some state-of-the-art classification performance including deep learning techniques published for the full 20 Newsgroups dataset along with the proposed classifier, which shows that our result is comparable with the existing methods. As no baseline is available for our work, we initially present the performance measures using the first label in the predicted label set which is presented in Table 2 for all the text datasets.

Dataset	Performance measures	Term weighting scheme			
		NTF	LTF-IGM	NTF-IGM	RTF-IGM
20 News (Full)	Macro F1	82.99	84.1	84.1	84.2
	Accuracy	83.53	84.49	84.49	**84.56**
ScienceNews	Macro F1	95.59	96.82	96.67	96.79
	Accuracy	95.6	**96.82**	96.67	96.8
DisjointNews	Macro F1	97.35	98.29	98.32	98.28
	Accuracy	97.35	98.3	**98.32**	98.28
CompScNews	Macro F1	83.46	85.73	86.27	85.68
	Accuracy	83.58	85.8	**86.32**	85.74
HR (internal)	Macro F1	80.37	85.15	84.47	85.08
	Accuracy	82.21	84.86	**84.89**	84.56
Telecom	Macro F1	60.35	61.02	63.1	60.4
	Accuracy	69.6	64	**70.4**	69
IMDB	Macro F1	86.06	87.63	87.89	87.62
	Accuracy	86.07	87.64	**87.9**	87.62
RT	Macro F1	75.34	79.03	78.85	79.05
	Accuracy	75.35	79.04	78.87	**79.06**

Table 2. Prediction performances using first label for text datasets. Model training for imbalanced data is used for HR (internal) data.

File	Annotation	atheism	religion .misc	christian	graphics	Windo ws.x	windows .misc	pc.hard Ware	mac. Hard Ware	misc .forsale	autos	Motor Cycles	Base Ball	hockey	crypt	Electro Nics	med	space	politics .guns	politics .mideast	politics .misc
53202	atheism	0.43	0.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53198	atheism	0.24	0	0.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
102886	autos	0	0	0	0	0	0	0	0	0	0.19	0	0	0	0	0.2	0	0	0	0	0
102883	autos	0	0	0	0	0	0	0	0	0	0.7	0.68	0	0	0	0	0	0	0	0	0
102663	baseball	0	0	0	0	0	0	0	0	0	0	0	0.31	0.31	0	0	0	0	0	0	0
20929	christian	0.22	0	0.23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53529	electronics	0	0	0	0	0	0	0	0.32	0	0	0	0	0	0	0.33	0	0	0	0	0
38761	graphics	0	0	0	2.06	2.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39498	graphics	0	0	0	0.76	0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54712	hockey	0	0	0	0	0	0	0	0	0	0	0	0.46	0.47	0	0	0	0	0	0	0
52641	hockey	0	0	0	0	0	0	0	0	0	0	0	1.3	1.3	0	0	0	0	0	0	0
50508	mac.hardware	0	0	0	0	0	0	0.1	0.11	0	0	0	0	0	0	0	0	0	0	0	0
51501	mac.hardware	0	0	0	0	0	0	0.4	0.4	0	0	0	0	0	0	0	0	0	0	0	0
74818	misc.forsale	0	0	0	0	0	0	0	0	0.85	0	0	0	0	0	0.87	0	0	0	0	0
61088	pc.hardware	0	0	0	0	0	0.47	0.48	0	0	0	0	0	0	0	0	0	0	0	0	0
54492	politics.guns	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.26	0	0.25
83627	religion.misc	0	0.43	0.42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84341	religion.misc	0.33	0.32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10008	windows.misc	0	0	0	1.7	0	1.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9142	windows.misc	0	0	0	0	0	1	1.1	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 1. Few instances with multi-label output from 20 Newsgroups dataset.

Prediction Enhancement with Confidence: Table 3 shows the enhanced classification performance (using the multiple labels in the predicted output)

[1] http://ai.stanford.edu/ amaas/data/sentiment/
[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/

with confidence for CompScNews, HR (internal), Telecom and full 20 Newsgroups datasets. The results depict that there are instances which actually belong to multiple labels with varying confidence and for many instances first predicted label does not match with the given annotation, however, the multi-label output contains the true class, this, in essence enhance the classification performance. Also there are many instances for which the module rejects the predicted result due to low confidence and is seen that predicted single label is incorrect for many of those instances.

Category	Statistics	Dataset (% of total)			
		CompScNews	HR (internal)	Telecom	20 News
SLVH	Frequency	1405 (28%)	8372 (51%)	214 (42%)	18876 (94%)
	FirstLabel	1377 (27%)	6087 (37%)	189 (37%)	16432 (82%)
MLH	Frequency	0	17	0	930 (4%)
	FirstLabel	0	5	0	390 (1%)
	MultiLabel	0	5	0	367 (1%)
SLM, MLM	SLM Freq.	3436 (68%)	6842 (41%)	240 (48%)	43
	MLM Freq.	16	757 (4%)	33 (6%)	106
	FirstLabel	2879 (57%)	3989 (24%)	156 (31%)	37
	MultiLabel	9	330 (2%)	15 (3%)	49
RCLC	Frequency	130 (2%)	91	13 (2%)	0
	FirstLabel	49	19	7 (1%)	0
	FirstLabel′	81 (1%)	72	6 (1%)	0
	Macro F1	86.3	85.27	64.1	84.7
	Accuracy	**86.35**	**85.61**	**71.26**	**84.87**

Table 3. Statistics for several confidence categories along with updated performance. Freq.:frequency, FirstLabel: First Label of the predicted output matches with the given annotation, MultiLabel: Given annotation does not match with first Label but matches with the second or third label in the multi-label output set, FirstLabel′: First Label of the predicted output does not matches with the given annotation

Although we found that the 20 Newsgroup data is not fully single-labeled which is also supported by [26], [28], [1], however, we have found that around 4% of the instances show multi-label aspect. It is seen that many classes with some overlaps have appeared in the multi-label output (shown in Figure 1), such as religion.misc and politics.guns (also found out by [27]), religion.misc and atheism, christian and religion.misc, politics.misc and politics.guns, pc.hardware and windows.misc etc. After final prediction and generating confidence scores, interpretability of the multi-label output is done by tracing the contributory words for each class. Figure 2 shows this result interpretability for the customer complaint presented in the section 1 from Telecom dataset.

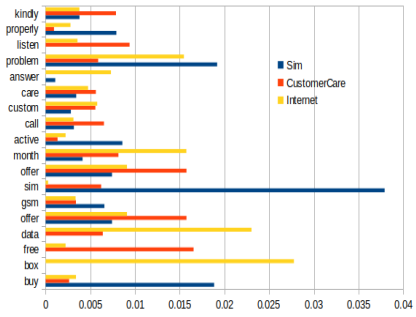

Fig. 2. Interpretability of the predicted label set, {Sim, Internet, CustomerCare} for the customer complaint presented in the section 1 from Telecom dataset.

Comparative Study with Fuzzy Classifier: [20] proposes an approach by turning discriminative single-task classification into generative multi-task classification by adopting a fuzzy rule induction approach implemented on the KN-

IME platform [5], empirically they have shown that an instance can belong to multiple classes in spite of its single annotation. Both the fuzzy rule induction technique of KNIME and and our classifier are run on the two UCI datasets, autos and zoo, to compare the class membership distribution. Before applying our classifier, different kinds of data preprocessing techniques such as data scaling, data standardization, binarization of the categorical features, missing value handling through imputation are done. Mutual information [22] is used for feature selection. The results for autos dataset (representative instances) is given

Classifier	ID	Anno-tation	Class name (Dataset: Autos)					Out-put	Confidence category	NC
			-1	0	1	2	3			
KNIME	3	2	0.36	1	0.18	0.5	1	0	-	-
OUR			0.237	0.236	0.195	0.199	0.13	-1	SLM	0.21
KNIME	15	0	0.71	1	0	0.33	0.63	0	-	-
OUR			0.24	0.243	0.18	0.19	0.14	0	SLM	0.24
KNIME	66	0	0.25	0.81	0.076	0.61	0	0	-	-
OUR			0.239	0.245	0.187	0.2	0.12	0	SLM	0.09
KNIME	67	-1	0.37	1	0	0	0	0	-	-
OUR			0.244	0.249	0.17	0.2	0.13	0	SLM	0.37
KNIME	123	-1	0.5	1	0	0	0	0	-	-
OUR			0.23	0.26	0.21	0.18	0.11	0	SLM	0.07
KNIME	183	2	0	0	0.54	0.71	0	0	-	-
OUR			0.19	0.19	0.21	0.23	0.17	2	SLM	0.08
KNIME	192	0	0.26	0.45	0	0.54	0	2	-	-
OUR			0.248	0.242	0.18	0.21	0.12	-1	SLM	0.19
KNIME	202	-1	0.46	0.2	0.17	0.5	0	2	-	-
OUR			0.32	0.23	0.16	0.17	0.12	-1	SLM	0.73

Table 4. Few representative sample results are presented which are obtained from running KNIME and our technique on the autos dataset. NC: Normalized confidence

in the Table 4 and for most of the instances it is seen that their are similarities in the two distributions although our distributions range within [0,1] and the other is fuzzy degree. For the zoo dataset, shown in Table 5, the fuzzy classifier,

Classifier	ID	Anno-tation	Class name (Dataset: Zoo)							Out-put	Confidence category	NC
			1	2	3	4	5	6	7			
KNIME	5	1	**1**	0	0	0	0	0	0	1	-	-
OUR			**0.42**	0.1	0.13	0.12	0.12	0.09	0.02	1	SLVH	0.83
KNIME	13	7	0	0	**1**	0	0	0	0	3	-	-
OUR			0.07	0.14	0.17	0.16	0.18	0.1	**0.19**	7	SLM	0.07
KNIME	21	2	0	**1**	0	0	0	0	0	2	-	-
OUR			0.09	**0.33**	0.11	0.13	0.14	0.13	0.07	2	SLM	0.56
KNIME	25	5	0	0	0	0	**1**	0	0	5	-	-
OUR			0.14	0.13	0.16	0.19	**0.21**	0.07	0.11	5	SLM	0.06
KNIME	30	6	0	0	0	0	0	**1**	0	6	-	-
OUR			0.07	0.27	0.1	0.06	0.12	**0.29**	0.08	6	SLVH	0.07
KNIME	34	4	0	0	0	0	0	0	**1**	4	-	-
OUR			0.14	0.14	0.17	**0.23**	**0.21**	0.03	0.08	{4,5}	MLM	0.12
KNIME	60	4	0	0	0	**1**	0	0	0	3	-	-
OUR			0.14	0.13	0.16	**0.23**	**0.21**	0.03	0.1	{4,5}	MLM	0.08
KNIME	73	4	0	0	0	**1**	0	0	0	4	-	-
OUR			0.14	0.14	0.16	**0.24**	**0.21**	0.03	0.08	{4,5}	MLM	0.12

Table 5. Few representative sample results are presented which are obtained from running KNIME and our technique on the zoo dataset. NC: Normalized confidence

apart from correct single class prediction, fails to capture any multi-label aspect, however, interestingly our technique, along with correct prediction, captured a weak label association between two classes 4 and 5. Where class 4 and 5 consist of instances such as {catfish, piranha, seahorse, tuna} and {frog, toad, newt} respectively. Instances of class 4 belongs to aquatic and class 5 belongs to both aquatic and terrestrial and both the classes are predator and tooted. The results indicate that an instance can belong to multiple classes in spite of its single annotation.

6 Conclusion

With the emergence of large volume of consumer generated data, demand for interpretable methods to mine business insights from those data also emerges. In many cases, each piece of text contains multiple issues which requires multi-label classification. However, most of the available data have single label annotation as manual annotation is costly and sometimes incomplete. Hence, this work proposes a classifier that learns from data annotated with single labels but predicts multi-label outputs. Thereafter a class-confidence computation mechanism is also proposed.

References

1. Ahmed, M.S., Khan, L., Oza, N.C., Rajeswari, M.: Multi-label asrs dataset classification using semi supervised subspace clustering. In: CIDU. pp. 285–299 (2010)
2. Al-Mubaid, H., Umair, S.A.: A new text categorization technique using distributional clustering and learning logic. IEEE Transactions on Knowledge and Data Engineering **18**(9), 1156–1165 (2006)
3. Aly, M.: Survey on multiclass classification methods. Neural Netw **19** (2005)
4. Arras, L., Horn, F., Montavon, G., Müller, K.R., Samek, W.: " what is relevant in a text document?": An interpretable machine learning approach. PloS one **12**(8), e0181142 (2017)
5. Berthold, M.R., Wiswedel, B., Gabriel, T.R.: Fuzzy logic in knime–modules for approximate reasoning–. International Journal of Computational Intelligence Systems **6**(sup1), 34–45 (2013)
6. Bucak, S.S., Jin, R., Jain, A.K.: Multi-label learning with incomplete class assignments. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 2801–2808. IEEE (2011)
7. Chen, K., Zhang, Z., Long, J., Zhang, H.: Turning from tf-idf to tf-igm for term weighting in text classification. Expert Systems with Applications **66**, 245–260 (2016)
8. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in neural information processing systems. pp. 3079–3087 (2015)
9. Dasgupta, T., Dey, L.: Multi-label classification and analysis of customer complaint logs. In: Workshop on Enterprise Intelligence, Knowledge and Data Discovery (KDD). ACM (2016)
10. Dasgupta, T., Dey, L., Verma, I.: Fuzzy multi-label classification of customer complaint logs under noisy environment. In: International Joint Conference on Rough Sets. pp. 376–385. Springer (2016)
11. Dekel, O., Singer, Y., Manning, C.D.: Log-linear models for label ranking. In: Advances in neural information processing systems. pp. 497–504 (2004)
12. Delany, S.J., Cunningham, P., Doyle, D., Zamolotskikh, A.: Generating estimates of classification confidence for a case-based spam filter. In: ICCBR. vol. 3620, pp. 177–190. Springer (2005)
13. Dey, L., Roy, G.: Auto-correction of consumer generated text in semi-formal environment. In: 7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. pp. 203–207. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu (November 2015)

14. consumer complaint forum, I.: Consumer complaints (2016), www.consumercomplaints.in
15. Jin, Y., Wanvarie, D., Le, P.: Combining lightly-supervised text classification models for accurate contextual advertising. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 545–554 (2017)
16. Ko, Y.: How to use negative class information for naive bayes classification. Information Processing & Management **53**(6), 1255–1268 (2017)
17. Ko, Y., Park, J., Seo, J.: Improving text categorization using the importance of sentences. Information processing & management **40**(1), 65–79 (2004)
18. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning. pp. 331–339 (1995)
19. Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
20. Liu, H., Cocea, M., Mohasseb, A., Bader, M.: Transformation of discriminative single-task classification into generative multi-task classification in machine learning context. In: Advanced Computational Intelligence (ICACI), 2017 Ninth International Conference on. pp. 66–73. IEEE (2017)
21. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. pp. 142–150. Association for Computational Linguistics (2011)
22. Murphy, K.P.: Machine learning: a probabilistic perspective. Cambridge, MA (2012)
23. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd annual meeting on association for computational linguistics. pp. 115–124. Association for Computational Linguistics (2005)
24. Parambath, S.P., Usunier, N., Grandvalet, Y.: Optimizing f-measures by cost-sensitive classification. In: Advances in Neural Information Processing Systems. pp. 2123–2131 (2014)
25. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980)
26. Read, J.: Scalable multi-label classification. Ph.D. thesis, University of Waikato (2010)
27. Read, J., Bifet, A., Holmes, G., Pfahringer, B.: Efficient multi-label classification for evolving data streams (2010)
28. Read, J., Perez-Cruz, F., Bifet, A.: Deep learning in partially-labeled data streams. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing. pp. 954–959. ACM (2015)
29. Tang, W., Mao, K., Mak, L.O., Ng, G.W.: Classification for overlapping classes using optimized overlapping region detection and soft decision. In: Information Fusion (FUSION), 2010 13th Conference on. pp. 1–8. IEEE (2010)
30. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining **3**(3) (2006)
31. Wang, Y., Tian, F.: Recurrent residual learning for sequence classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 938–943 (2016)
32. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern recognition **40**(7), 2038–2048 (2007)
33. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering **26**(8), 1819–1837 (2014)