Semi-automated classification for multi-label open-ended questions

Hyukjun Gweon[*] Matthias Schonlau[†] Marika Wenemark[‡]

Abstract

In surveys, text answers from open-ended questions are important because they allow respondents to provide more information without constraints. When classifying open-ended questions automatically using supervised learning, often the accuracy is not high enough. Alternatively, a semi-automated classification strategy can be considered: answers in the easy-to-classify group are classified automatically, answers in the hard-to-classify group are classified manually. This paper presents a semi-automated classification method for multi-label open-ended questions where text answers may be associated with multiple classes simultaneously. The proposed method effectively combines multiple probabilistic classifier chains while avoiding prohibitive computational costs. The performance evaluation on three different data sets demonstrates the effectiveness of the proposed method.

1 Introduction

Open-ended questions in surveys are often manually classified into different class or categories. When data are large, manual classification is time consuming and expensive in the sense that it requires professional human coders with sufficient knowledge. At the same time, analyzing the text answers from open-ended questions is important because

[*]Department of Statistical and Actuarial Sciences, Western University, London, Ontario, N6A 5B7, Canada

[†]Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

[‡]Centre for Organisational Support and Development, Linkping University, Region stergtland, Sweden

they do not constrain respondents' answers and thus may give more accurate information than closed-ended questions (Schonlau and Couper, 2016).

The advance of statistical learning techniques can be used for automatic classification for text data from open-ended questions. A statistical learning model such as Support Vector Machines (SVM) (Vapnik, 2000) and Random Forests (Breiman, 2001) may be trained based on training data and used to predict new data. Analyzing text data from open-ended questions with statistical learning methods has received increasing attention in social sciences (Matthews et al., 2018; Ye et al., 2018).

While the use of statistical learning methods reduces the total cost for the coding task, fully automated classification for open-ended questions remains challenging. It is often difficult to achieve an overall classification accuracy as high as the accuracy that can be achieved by human coders and with a classification accuracy which is acceptable to use for research purposes. Semi-automated classification uses statistical approaches to partially automated classification in that easy-to-classify answers are categorized automatically and hard-to-classify answers are categorized manually. (Gweon et al., 2017; Schonlau and Couper, 2016).

Answers to open-ended questions are often associated with multiple categories simultaneously. In the community of machine learning, this type of data is referred to as multi-label data. This is different from the traditional multi-class data where a text answer can only belong to a single class or label. Recently, Schonlau et al. (to appear) evaluated the use of existing machine learning algorithms for fully automated coding of multi-label open-ended questions.

This paper focuses on semi-automated classification for multi-labelled text data from open-ended questions. As far as we are aware, there is no published work on semi-automated classification for multi-label data. Most of the previous work on semi-automated classification deal with multi-class data. Also most research in machine learning that analyzes multi-label data assumes fully automated classification. In this paper we consider existing algorithms for multi-label data that may be suitable for semi-automatic classification. We also propose a new method to improve the classification performance of existing methods in the specific context of multi-label semi-automatic classification. This is illustrated with three examples of multi-labelled text data from open-ended questions. We show that the proposed method can achieve a higher accuracy than Binary Relevance, Label Powerset, and Probabilistic Classifier Chains (Dembczyński et al., 2010) for semi-automated classification.

The rest of this paper is organized as follows: In Section 2, we review elements of semi-automated classification for open-ended questions. In Section 3, we review approaches to multi-label classification. In Section 4, we present the details of the proposed approach. In Section 5, we evaluate the proposed method as well as other commonly used algorithms based on multi-label text data from open-ended questions. In Section 6, we conclude with a discussion.

2 Semi-automated classification for text data

This section describes how text answers to open ended-questions are converted into ngram variables and how a learning algorithm is evaluated in semi-automated classification.

2.1 Converting text answers into ngram variables

To use text answers as the input features for a learning algorithm, we may transform the original texts into a different representation using text mining approaches. A common transformation approach is to create indicator variables, each of which indicates the presence or absence of a certain word (unigram) or a short word sequence (bigram, or more generally, ngram variables) (Sebastiani, 2002; Schonlau et al., 2017). Applying this technique, we may convert any text answer into a vector in which each element is binary and corresponds to a word (or a word sequence). Instead of indicator variables, variables containing word frequency can also be used (Manning et al., 2008; Guenther and Schonlau, 2016).

Typically, there are several thousands of ngram variables including redundant words. We may reduce the number of ngram variables by applying some preprocessing techniques such as stemming (i.e. reducing words to their grammatical root) and thresholding (i.e. removing words occurred less than a certain time) and removing very common words (stopwords) (Manning et al., 2008; Guenther and Schonlau, 2016).

2.2 Production rate

Semi-automated classification requires a score or a probability that shows a level of confidence about the prediction. A threshold on that score or probability divides the text answers into easy-to-classify and hard-to-classify texts. All new text answers with high scores above a threshold may be categorized automatically and all others are categorized manually. The threshold is a user-specified value and can be set depending on the combination of desired prediction accuracy in the easy-to-classify group and the acceptable number of difficult-to-classify answers that need manual coding. The production rate refers to as the fraction of text answers that belong to the easy-to-classify group. That is, the production rate is the proportion of observations that can be categorized automatically. In general, production rate and accuracy are inversely related. If we chose a low production rate, only the easiest answers will be in the easy-to-classify group and the accuracy of the automatic classification will be high. If we increase the production rate, more complicated answers will be automatically classified and accuracy will tend to decrease.

For multi-label data, the definition of accuracy is no longer obvious. Evaluation measures for multi-label data are discussed in Section 3.1.

3 Multi-label classification

Consider a set of possible output labels $\mathcal{L} = \{1, 2, ..., L\}$. In multi-label classification, each instance with a feature vector $\mathbf{x} \in \mathbb{R}^d$ is associated with a subset of these labels. Equivalently, the subset can be described as $\mathbf{Y} = (y_1, y_2, ..., y_L)$, where $y_i = 1$ if label i is associated with the instance, and $y_i = 0$ otherwise. A multi-label classifier $\mathbf{h}$ learns from training data to predict $\mathbf{h}(\mathbf{x}) = \hat{\mathbf{Y}} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_L)$ for a given $\mathbf{x}$.

Next, we review some common multi-label algorithms and their relationship to an evaluation criterion, subset accuracy.

3.1 Evaluating multi-label algorithms in semi-automated classification

Evaluating the classification of a text answer into a single label is straightforward: the label is either correct or not and accuracy refers to the percentage of correctly classified answers; equivalently, error refers to the percentage of misclassified answers. For answers that are classified into multiple labels, there are several ways to combine the accuracy of each single label to an overall evaluation measure for the set of multiple labels. These evaluation measures include subset accuracy, Hamming loss, F-measure and log loss. For a predicted set of multiple labels, subset accuracy is 1 if all of the L labels are correctly predicted and 0 otherwise. Hamming loss evaluates the fraction of misclassified labels. F-measure is the harmonic mean of precision and recall and log loss evaluates the uncertainty of the prediction averaged over the labels when a probability score for each label is given.

In this paper we develop a methodology for subset accuracy (equivalently, in terms of loss, 0/1 loss). This is a strict metric because a zero score is given even if all labels are correctly classified except one. However, subset accuracy is appropriate for semi-automated classification because if an algorithm has difficulty classifying even a single label, the entire observation needs to be manually classified. That is, automated classification shall be conducted only if the model is highly confident in the entire predicted label set.

Because subset accuracy requires that all labels are simultaneously correctly classified, we are interested in finding the label set Y^* that maximizes the joint probability conditional on a text answer $\mathbf{x}$:

$$Y^* = \underset{\mathbf{Y}}{\operatorname{argmax}}\, P(\mathbf{Y}|\mathbf{x}) = \underset{\mathbf{Y}}{\operatorname{argmax}}\, P(y_1, ..., y_L|\mathbf{x}).$$

In the next section we discuss common approaches to estimating the joint probability proposed in the machine learning community.

3.2 Multi-label approaches that optimize subset accuracy

Various approaches have been proposed for predicting multi-label outcomes. Since we use subset accuracy as the evaluation measure, we focus on methods that aim to maximize the joint conditional distribution.

The simplest approach, called Binary Relevance (BR), transforms a multi-label problem into separate binary problems. That is, BR constructs a binary classification model for each label independently. For an unseen observation, the prediction set of labels is obtained simply by combining the individual binary results. In other words, the predicted label set is the union of the results predicted from the L binary models. If each of the binary models produces probability outcomes, BR can produce an estimate for $P(y_1|\mathbf{x})P(y_2|\mathbf{x})...P(y_L|\mathbf{x})$. Note that this coincides with the joint probability $P(y_1, ..., y_L|\mathbf{x})$ if the labels are independent (conditional on $\mathbf{x}$). This implies that the product of the probabilities obtained by BR will estimate $P(y_1, ..., y_L|\mathbf{x})$ accurately only if the labels are conditionally independent. The joint probability may be inaccurate if the labels are substantially correlated given $\mathbf{x}$.

Another approach tailored for subset accuracy is Label Powerset learning (LP). This approach transforms a multi-label classification into a multi-class (i.e. multinomial) problem by treating each unique label set $\mathbf{Y}$ that exists in the training data as a single class. For example, when L=3 there could be up to 2^3 classes c_i, $(i = 1, ..., 8)$ observed in the training data. Then any algorithm for multi-class problems can be applied using the transformed c_i classes. Training a multi-class classifier takes into consideration dependencies between labels. For a new observation, LP predicts the most probable class (i.e. the most probable label set). If an algorithm for multi-class data gives probabilistic outputs (some algorithms classify without computing probabilities), *LP* directly estimates the class probabilities (i.e. the joint probability $P(\mathbf{Y}|\mathbf{x})$). However, this approach cannot estimate the joint probability for any label set unseen in the training data. As a consequence, if the true label set of the new observation is an *unseen* observation the prediction cannot be correct. Another drawback of LP is that the number of classes in the transformed problem can increase exponentially (up to 2^L number of classes). This can be problematic when L is large since each combination of labels may be present in just one or a few observations in the training data which makes the learning process difficult.

A third approach to multi-label learning is Classifier Chains (CC) (Read et al., 2009, 2011). As in binary relevance, in CC also a binary model is fit for each label. However, CC fits the binary models sequentially and uses the binary label results obtained from previous models as additional predictors in subsequent models. That is, the model for the i^{th} label y_i uses $\mathbf{x}$ and $y_1, ..., y_{i-1}$ as features. (For example, the model for y_1 uses $\mathbf{x}$ as features, the model for y_2 uses $\mathbf{x}$ and y_1 as features and so on.) Passing label information between binary classifiers allows CC to take label dependencies into account. In the prediction stage, CC successively predicts the labels one at a time. The prediction results of the previous labels are used for predicting the next label in the chain.

This idea is extended to Probabilistic Classifier Chains (PCC) (Dembczyński et al., 2010). PCC explains CC using a probabilistic model. Specifically, the conditional joint distribution can be described as

$$P(y_1, ..., y_L|\mathbf{x}) = P(y_1|\mathbf{x}) \prod_{j=2}^{L} P(y_j|y_1, ..., y_{j-1}, \mathbf{x}). \tag{1}$$

and PCC estimates the probabilities $P(y_1|\mathbf{x})$, $P(y_2|\mathbf{x}, y_1)$, ..., $P(y_L|\mathbf{x}, y_1, y_2, ..., y_{L-1})$.

PCC finds the label set that maximizes the right hand side of equation (1). However, there is no closed-form solution for finding the label set. A few different solutions have been suggested. Dembczyński et al. (2010) used an exhaustive search (ES) that considers all possible combinations. However, an exhaustive search may not be practical when L is large, because the number of possible combinations (2^L) increases exponentially. To overcome this problem, optimization strategies based on the uniform cost search (UCS) (Dembczyński et al., 2012) and the A^* algorithm (Mena et al., 2015) have been proposed. First, the estimated joint conditional probability may be represented by a probability binary tree. Then a search algorithm finds the optimal path (in our case, the path that gives the highest joint probability) from the root and the terminal node. Compared with ES, UCS substantially reduces the computational cost for PCC to reach the label set with the highest joint probability (Dembczyński et al., 2012).

In theory, when applying the product rule, the order of the categories $y_1, ..., y_L$ does not matter. For example, both $P(y_1|\mathbf{x})P(y_2|y_1, \mathbf{x})$ and $P(y_2|\mathbf{x})P(y_1|y_2, x)$ equal to

$P(y_1, y_2|\mathbf{x})$. In practice, the two chains may lead to different estimates. This means the performance of PCC may be affected by the order of the labels in the chain.

To alleviate the influence of the category order, an ensembling approach (EPCC) (Dembczyński et al., 2010) that combines multiple probabilistic chains has been proposed. First m PCC models are trained where each PCC model is based on a randomized order of the labels. In the prediction stage, the average conditional joint probability over the m PCC models is computed for each possible label set. Then the predicted label set is the label set with the highest average predicted probability. Let $\hat{P}_j(\mathbf{Y}|\mathbf{x})$ be the conditional joint probability estimated by the j^{th} PCC model. The ensemble strategy predicts the label set $\hat{\mathbf{Y}}$ such that

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmax}} \frac{\sum_{j=1}^{m} \hat{P}_j(\mathbf{Y}|\mathbf{x})}{m}.$$

Note that EPCC does not combine the predicted label sets but conditional joint probabilities. To find the highest average probability from m PCC models, all individual probabilities are required and this forces us to use ES to compute the conditional joint probability for all 2^L label combinations from all m PCC models. Hence, although EPCC reduces the problem of influence of label order, the method will not be useful if the problem deals with a large number of labels or when m is large. To reduce the computational cost for combining multiple PCC models, we propose a new approach to ensembling the PCC models in the next section.

4 The majority-voted-based ensemble of PCC for semi-automated classification

The proposed method aims to ensemble multiple PCC models at much less computational cost. As mentioned in Section 3.2, the best label set (with the highest joint probability) for a single PCC can be found by a fast search strategy. In this paper, we use UCS, since the implementation is simple and the algorithm always finds the optimal solution. Using UCS, the proposed method obtains $\hat{\mathbf{Y}}_j$ $(j = 1, ..., m)$, the label set predicted by the j^{th} PCC model and $\hat{P}_j$, the estimated probability that $\hat{\mathbf{Y}}_j$ is the true label set. Among the m predicted label sets, the proposed method chooses the most frequent label set for the final prediction. That is, $\hat{\mathbf{Y}} = \text{mode}(\{\hat{\mathbf{Y}}_1, ..., \hat{\mathbf{Y}}_m\})$. In case there are ties in the mode, we choose the label set whose averaged probability estimate is the highest.

Semi-automatic classification requires a score that measures how easy/hard the prediction is. Whether a text answer is classified automatically or manually is determined based on this score. Next, a score is proposed: Let J be the set that contains all indices j $(1 \le j \le m)$ for which $\hat{\mathbf{Y}}_j$ is the most frequent one (i.e. $J = \{j : \hat{\mathbf{Y}}_j = \hat{\mathbf{Y}}\}$). The proposed score for the prediction is

$$\theta = \left(\frac{\sum_{i \in J} \hat{P}_j}{|J|}\right)\left(\frac{|J|}{m}\right) \tag{2}$$

$$= \frac{\sum_{i \in J} \hat{P}_j}{m}. \tag{3}$$

The first factor of equation (2) is the average joint probability of the predicted label set. The second factor of equation (2) is the fraction of the PCC models that predict the predicted label set among the m models. Multiplying the two components makes sense: a prediction may be more accurate if the (average) probability related to the chosen label set is high (the first factor) *and* more individual chain models vote for the same label set (the second component). We call this approach Majority-vote-based Ensemble of Probabilistic Classifier Chains (MEPCC). We later show empirically that combining the two factors indeed improves performance over just using a single factor. Table 1 illustrates an example for 5 labels (L=5) and 7 PCC models (m=7). The MEPCC ap-

Table 1: An example of the MEPCC classification of a single observation with $L = 5$ and $m = 7$.

PCC model	**Prediction**	y_1	y_2	y_3	y_4	y_5	$P(y_1, ..., y_5 \vert \mathbf{x})$
1	$\hat{\mathbf{Y}}_1$	1	1	0	0	1	0.875
2	$\hat{\mathbf{Y}}_2$	1	1	0	0	1	0.921
3	$\hat{\mathbf{Y}}_3$	0	0	1	1	0	0.743
4	$\hat{\mathbf{Y}}_4$	0	0	0	1	0	0.882
5	$\hat{\mathbf{Y}}_5$	0	0	0	1	0	0.643
6	$\hat{\mathbf{Y}}_6$	0	1	0	1	0	0.739
7	$\hat{\mathbf{Y}}_7$	1	1	0	0	1	0.824
final prediction	$\hat{\mathbf{Y}}$	1	1	0	0	1	$\theta = \frac{0.875+0.921+0.824}{7} = 0.374$

proach stores the probability of one label set from each PCC model. Because MEPCC combines over the probabilities corresponding to the best label set from different PCC models, MEPCC can take advantage of the UCS (or any other) strategy. Note that a search strategy like UCS cannot be used for EPCC where all individual probabilities for all label combinations are required. More succinctly, MEPCC combines over the maximal probabilities of each PCC, whereas EPCC maximizes over the average probabilities, requiring evaluation of all individual probabilities. We summarize the procedure of MEPCC in Algorithm 1.

Algorithm 1 The MEPCC algorithm

```
**Input:** Number of models $m$, an instance vector $\mathbf{x}$, corresponding PCC models $h_j$,
the uniform cost search algorithm $U$
for $j = 1$ to $m$ do
    (a) Using $h_j$ and $U$, obtain $\hat{\mathbf{Y}}_j = \text{argmax}_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{x})$
    (b) Store $\hat{P}_j = P(\hat{\mathbf{Y}}_j|\mathbf{x})$
end for
Obtain the label set $\hat{\mathbf{Y}} = \text{mode}(\{\hat{\mathbf{Y}}_1, ..., \hat{\mathbf{Y}}_m\})$
Obtain $J = \{j : \hat{\mathbf{Y}}_j = \hat{\mathbf{Y}}\}$
Obtain the score $\theta = \frac{\sum_{i \in J} \hat{P}_j}{m}$
Return $\hat{\mathbf{Y}}$ and $\theta$
```

5 Experiments

5.1 Data

We evaluated the performance of the MEPCC algorithm on three different data sets: Civil disobedience, Immigrant and Happy data[1]. For each data set, an open-ended question was asked to the respondents and their answers have been coded manually with possibly multiple labels.

The Civil data set was collected to study cross-cultural equivalence about Civil disobedience. Behr et al. (2014) first asked respondents a closed-ended question from the ISSP (ISSP Research Group, 2012) How important is it that citizens may engage in acts of Civil disobedience when they oppose government actions? (Not at all important 1 — Very important 7). The respondents were then asked: What ideas do you associate with the phrase ‘Civil disobedience? Please give examples. Answers were classified into

[1]The Happy data are available upon request by contacting Marika Wenemark marika.wenemark@liu.se. The Immigrant and Civil Disobedience data are available from the GESIS Datorium `http://dx.doi.org/10.7802/1795`

12 labels: non-productive, violence, disturbances, peaceful, listing activities, breadth of actions, breaking law, breaking rules, government:dissatisfaction, government:deep rift, copy/paste from the Internet, other. The survey data were collected in different languages and we use a merged data set (Spanish, German and Danish) that contains 1029 observations.

The Immigrant data set was collected to study cross-national equivalence of measures of xenophobia. In the 2003 International Social Survey Program (ISSP) on National Identity, the questionnaire contained four statements regarding beliefs on Immigrants such as Immigrants take jobs from people who were born in Germany. After rating each statement, respondents were asked to answer to an open-ended question: Which type of Immigrants were you thinking of when you answered the question? The previous statement was: [text of the corresponding item]. Braun et al. (2013) classified answers into 14 labels: non-productive, positive, negative, neutral/work, general, Muslim countries, eastern European, Asia, ex-Yugoslavia, EU15, sub Sahara, Sinti/Roma, legal/illegal, other. In this article, we use 1006 observations from the German survey.

The Happy data set was collected to study the relationship between positive factors and mental health and care needs. Wenemark et al. (2018) asked respondents "Name some positive things in your life, that are uplifting or make you Happy: (you may write several things)". Answers were classified into 13 labels: nothing, relationships (family or romantic), working/studying, health, self-esteem, joy/happiness, well-being: drinking/eating/drugs/sex, spirituality, money, nature, hobbies, culture, and exercise. The data set contains 2350 observations.

Table 2 contains summary statistics about the three data sets.

Table 2: Summary statistics of data sets: number of total observations, features and labels and average number of relevant labels, and percentage of observations that are associated with more than one label ($P_{|L|>1}$).

Data	**# observations**	**# features**	**L**	**av. # of labels**	$\mathbf{P}_{\|L\|>1}$
Civil	1029	305	12	1.15	13.80%
Immigrant	1006	273	14	1.19	13.72%
Happy	2350	492	13	2.77	87.40%

5.2 Experimental setup

We compared the proposed MEPCC method against BR and LP and PCC. For PCC, we used the uniform search to reach a predicted label set and the estimated probability of equation (1) for the confidence score of the prediction. EPCC was not included in the comparison because its computational cost makes it infeasible for prediction for our data sets[2]. Support vector machines (SVM) (Vapnik, 2000) were used as the base classifier on unscaled variables with a linear kernel and tuning parameter $C = 1$. For probabilistic output, the SVM scores were converted into probabilities using Platt's method (Platt, 2000). The analysis was conducted in R (R Core Team, 2014) using the $e1071$ package (Meyer et al., 2014) for SVM.

For each data set, 5-fold cross validation (CV) was performed. That is, we randomly divided the data into five equal-sized parts and used the first four parts as the training data and the last part as the test data. Performance evaluation is only made on the test data. Each of the five parts were used as test data and the results were averaged.

5.3 Performance of the MEPCC approach

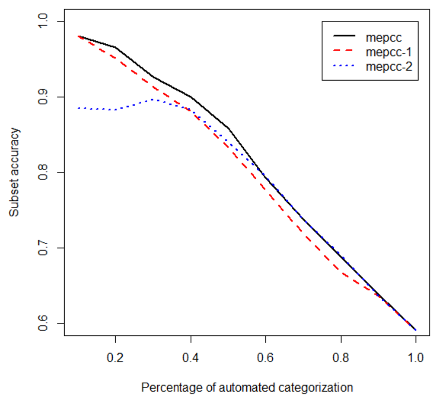
We first investigated the performance of the MEPCC. The score in equation (2) has two components. To demonstrate that both components are helpful, we evaluate the proposed score as well as two different scores where one of the components is missing. That is, we compared the MEPCC with three different scores θ, θ_1 and θ_2:

$$\begin{aligned}
\text{(MEPCC)} \quad \theta &= \left(\frac{\sum_{i\in J}\mathbf{P}_j}{|J|}\right)\left(\frac{|J|}{m}\right) \\
\text{(MEPCC-1)} \quad \theta_1 &= \left(\frac{\sum_{i\in J}\mathbf{P}_j}{|J|}\right) \\
\text{(MEPCC-2)} \quad \theta_2 &= \left(\frac{|J|}{m}\right).
\end{aligned}$$

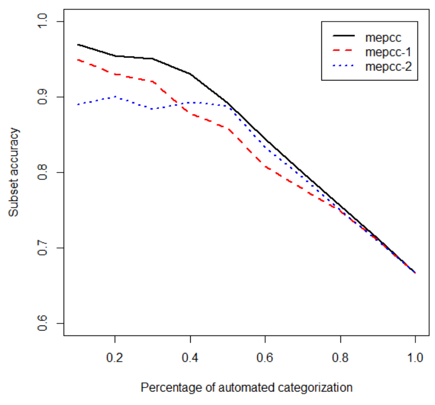
Prioritizing the text answers based on θ_2 results in many ties. The tied answers were randomly reordered to be able to calculate subset accuracy at each production rate. Figure 1 shows the subset accuracy of each approach as a function of the production rate.

[2]In our experiment on the Immigrant data with 14 labels, running the exhaustive search for PCC (m=1) for a single prediction took a single computer (Intel Core i7 CPU with 8GB RAM) over 30 minutes. This implies that predicting 200 observations using EPCC (m=10) would take more than 1000 hours.

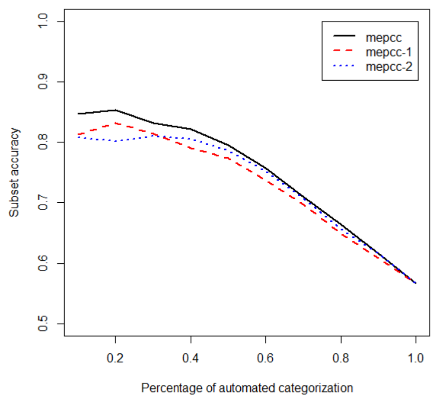
The text answers with higher scores were classified first. For example, production rate 0.2 means only 20% of the test data with the highest scores were classified automatically by the models. When the production rate equals 1, there was no difference between the MEPCC models because the predicted label sets are always the same. The difference is how they prioritize the text answers from the easiest-to-classify to the hardest-to-classify answers. When the production rate was less than 1, MEPCC outperformed MEPCC-1 and MEPCC-2 for all three data. The results show that both components in equation (2) were helpful for prioritizing the observations.



(a) Civil



(b) Immigrant

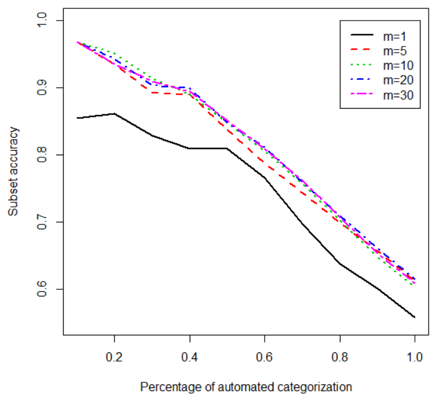


(c) Happy

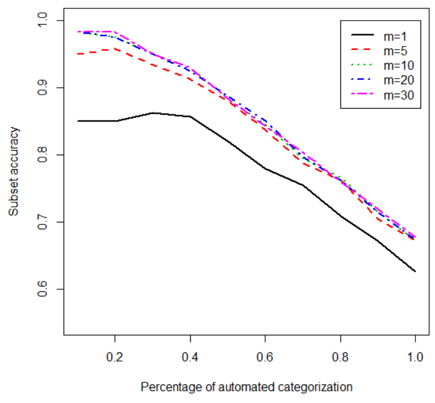
Figure 1: Subset accuracy of three variations on MEPCC as a function of production rate.

5.4 Effect of the number of PCC models

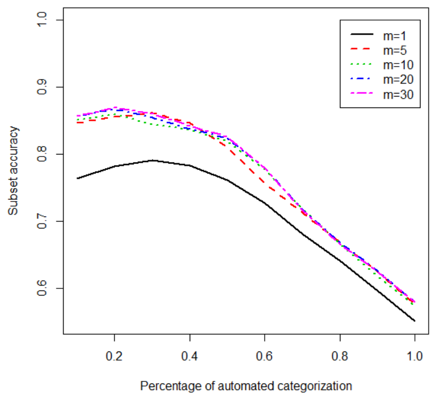
We then investigated to what extent the number of PCC models affects the predictive performance of MEPCC. Figure 2 shows the performance of MEPCC for different number of PCC models (m). When m was low, increasing m led to huge improvement of the subset accuracy of MEPCC. However, once there were enough PCC models (e.g. m=10), adding more PCC models did not improve the subset accuracy. The empirical results show that MEPCC does not require many PCC models for performing well.



(a) Civil



(b) Immigrant



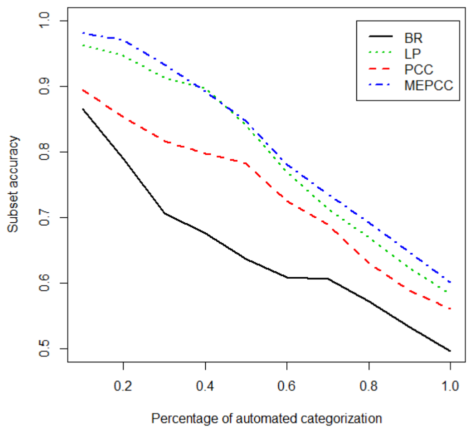
(c) Happy

Figure 2: The effect of the number of PCC models (m) used for MEPCC

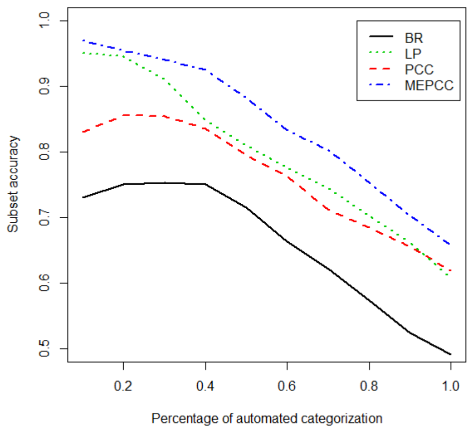
5.5 Comparison with other methods

At last we investigated the performance of MEPCC ($m = 10$) compared to the established methods (BR, LP and PCC). For all methods, a production rate of x% refers to the x% of the data that have the highest score. MEPCC used θ as a score, while each of the other approaches used the probability of the predicted label set estimated by that method. Note when m=1, MEPCC and PCC are identical; the score θ coincides with the probability of the label set predicted by PCC.

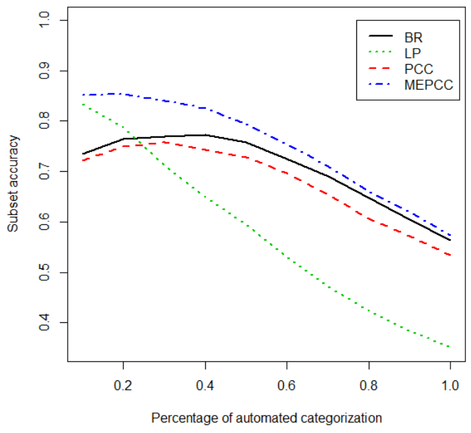
Figures 3 and 4 illustrate the respective subset accuracy and Hamming loss for the different methods as a function of the production rate on the Happy, Immigrant and Civil data. For the Immigrant and Happy data, the highest subset accuracy at most production rates was obtained by MEPCC. For the Civil data, MEPCC and LP performed the best. In terms of Hamming loss, MEPCC achieved the lowest error at most production rates for all data.



(a) Civil

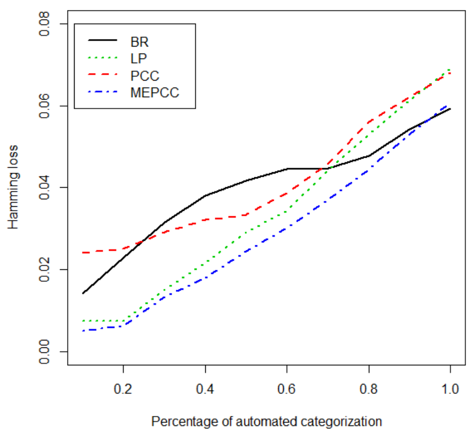


(b) Immigrant

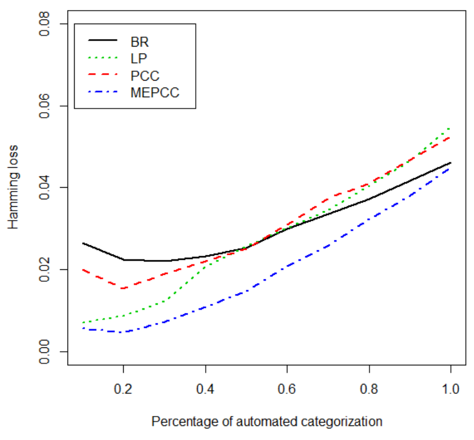


(c) Happy

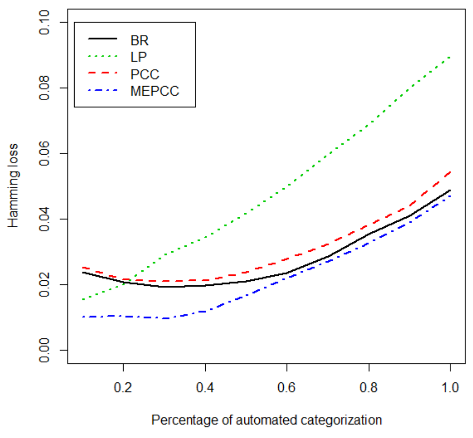
Figure 3: Semi-automated result (subset accuracy) for the three data from the 5-fold cross validation.



(a) Civil



(b) Immigrant



(c) Happy

Figure 4: Semi-automated result (Hamming loss) for the three data from the 5-fold cross validation.

Next, we consider the performance of each method given target predicted accuracy values. To decide the fraction of automatic categorization, a practitioner will typically set a threshold probability above which texts are coded automatically. For MEPCC, the relationship between true accuracy and the confidence score (θ) were estimated via cross-validation on the training data. We used Platt's scaling to convert the confidence scores into probability outputs. Since Platt's scaling could improve the level of calibration (Niculescu-Mizil and Caruana, 2005), the same technique was also applied to BR, LP and PCC.

Table 3 illustrates the tradeoff between the percentages of automated prediction and the corresponding subset accuracy of each method as a function of different thresholds. The threshold refers to the minimum predicted subset accuracy required for automated prediction. The minimum predicted subset accuracy helps us decide which text answers should be classified automatically and which should be classified manually. For example, if the client decides that at least 80% accuracy is required for automated classification, then approximately 39.3% of the Civil data, 42.5% of the Immigrant data, and 27.6% of the Happy data can be classified automatically by MEPCC with subset accuracy 0.891, 0.916 and 0.857, respectively. Note that this is a huge improvement compared to applying BR that could only automatically classify 9.3% of the Civil data, 12.8% of the Immigrant data, and 8.7% of the Happy data with lower subset accuracies. Table 4 shows the relationship between predicted and actual accuracy by aggregating to ranges of predictions for each method and data set. For MEPCC the actual accuracy is within the range of the predicted accuracy in most cases, much better than for the other methods.

Table 3: Semi-automated result for the three data at different decision thresholds. P represents the percentage of automated predictions and SA represents the subset accuracy for the automated prediction results.

Data	Threshold	BR		LP		PCC		MEPCC	
		P	SA	P	SA	P	SA	P	SA
Civil	0.9	0.7%	0.667	16.5%	0.967	0.0%	NA	13.0%	0.978
	0.8	9.3%	0.893	34.3%	0.898	15.1%	0.787	39.3%	0.891
	0.7	18.4%	0.846	46.6%	0.852	36.4%	0.817	45.8%	0.860
	0.6	25.4%	0.768	50.6%	0.831	52.1%	0.771	52.9%	0.820
Immigrant	0.9	3.7%	0.858	11.1%	0.959	1.3%	0.558	31.5%	0.947
	0.8	12.8%	0.779	30.4%	0.890	27.7%	0.859	42.5%	0.916
	0.7	26.6%	0.743	38.6%	0.863	42.4%	0.829	55.1%	0.862
	0.6	41.7%	0.715	53.6%	0.806	50.5%	0.795	62.7%	0.839
Happy	0.9	1.3%	0.592	8.9%	0.850	0.1%	0.750	1.0%	0.830
	0.8	8.7%	0.734	14.3%	0.802	7.2%	0.726	27.6%	0.857
	0.7	32.8%	0.776	17.7%	0.793	29.9%	0.767	43.7%	0.817
	0.6	53.2%	0.745	22.2%	0.761	49.2%	0.744	52.0%	0.790

Table 5 shows the runtime of each method for training the model and predicting all instances in test data (Intel Core i7 CPU with 8GB RAM). Unsurprisingly, the runtime of MEPCC at $m = 10$ is roughly 10 times of that of PCC in both of the training and prediction stages.

Table 4: Semi-automated result for the three data at different ranges of thresholds. P represents the percentage of automated predictions and SA represents the subset accuracy for the automated prediction results.

Data	Predicted accuracy	BR P	BR SA	LP P	LP SA	PCC P	PCC SA	MEPCC P	MEPCC SA
Civil	$[0.9, 1.0]$	0.7%	0.667	16.5%	0.967	0.0%	NA	13.0%	0.978
	$[0.8, 0.9)$	8.7%	0.896	17.8%	0.834	15.1%	0.787	26.2%	0.846
	$[0.7, 0.8)$	9.0%	0.769	12.2%	0.710	21.3%	0.828	6.5%	0.681
	$[0.6, 0.7)$	7.0%	0.566	4.1%	0.584	15.7%	0.655	7.1%	0.563
Immigrant	$[0.9, 1.0]$	3.7%	0.858	11.1%	0.959	1.3%	0.558	31.5%	0.947
	$[0.8, 0.9)$	9.1%	0.750	19.3%	0.843	26.4%	0.869	11.0%	0.829
	$[0.7, 0.8)$	13.8%	0.710	8.2%	0.747	14.7%	0.757	12.5%	0.688
	$[0.6, 0.7)$	15.1%	0.602	15.0%	0.659	8.1%	0.623	7.7%	0.670
Happy	$[0.9, 1.0]$	1.3%	0.592	8.9%	0.850	0.1%	0.750	1.0%	0.830
	$[0.8, 0.9)$	7.4%	0.755	5.4%	0.717	7.1%	0.730	26.5%	0.858
	$[0.7, 0.8)$	24.0%	0.792	3.4%	0.751	22.7%	0.779	16.2%	0.749
	$[0.6, 0.7)$	20.4%	0.693	4.6%	0.615	19.3%	0.703	8.3%	0.647

Table 5: Runtime (in seconds) of each method for the three data

Data	Stage	BR	LP	PCC	MEPCC
Civil	Train	1.688	0.641	1.128	11.787
	Prediction	0.269	0.044	37.142	374.611
Immigrant	Train	1.363	0.510	0.894	8.724
	Prediction	0.200	0.056	35.369	334.075
Happy	Train	11.160	16.164	7.371	78.293
	Prediction	0.567	3.691	177.847	1746.529

6 Discussion

Using three examples, we have investigated several approaches for automated classification for any desired production rate when data are multi-labeled. In terms of subset accuracy and Hamming loss, the proposed method, MEPCC, achieved the best performance at most production rates in all three data sets.

There were trade-offs between the prediction performance and the production rate for all methods. At low production rates, high subset accuracy and low Hamming loss were achieved for a small number of easy-to-classify answers. However, accuracy (loss) tended to decrease (increase) as more difficult answers were included (i.e. production rate increased).

Either subset accuracy or production rate can be set at a target rate which determines the second measure. For example, targeting 80% minimum subset accuracy for an automated prediction, MEPCC categorizes 39.3% of the Civil data, 42.5% of the Immigrant data, and 27.6% of the Happy data automatically. Such a reduction is considerable. In an applied research environment, reducing the need for manual coding in a data set with 5,000 a reduction by 50% may save several weeks of coding time. If production rate is fix at 80%, MEPCC could achieve a subset accuracy of 70% (Civil), 75% (Immigrant), and 68% (Happy).

The Hamming loss represents the fraction of misclassified labels. Figure 4 show s that the improvement of MEPCC over BR was quite noticeable at lower production rates but relatively small at 100% production rate.

MEPCC outperformed PCC at most production rates on all three data. This shows that combining multiple PCC models substantially improves the performance. As can be seen from Figure 2, even combining 5 models resulted in a substantial improvement throughout the whole range of production rate. The difference tended to be greater at lower production rates. This means MEPCC is even more preferred for semi-automated classification, where a high accuracy is required rather than a high production rate. The performance of MEPCC converged as m increased in all three data sets. The difference between the MEPCC models were negligibly small when m was larger than 10. This is a desirable result in practice because employing too many PCC models for an ensemble model is unnecessary.

For all three data we found that the proposed method was not sensitive to the choice of the search algorithm for each PCC model (results and figures not shown). That is, the classification results of MEPCC with the uniform cost search were similar to those with the greedy search. While the proposed method uses the uniform cost search, the greedy approach may also be considered especially when the fast prediction time matters.

Figure 3 shows LP beats BR for the Civil and Immigrant data sets and BR beats LP for the Happy data set with respect to subset accuracy. We see two reasons: 1) LP performed well when the number of unique label sets was relatively small (Civil: 39, Immigrant: 59). However, the performance of LP was not effective but less well for the Happy data where the number of unique label sets was large (346). 2) BR does not take into account correlations among the labels. BR beat LP where bivariate label correlation were low (Happy data) and LP beat BR where bivariate label correlations were larger (Civil and Immigrant Data). Compared to BR and LP, MEPCC seems to be robust to those aspects (the number of unique label sets and the magnitude of label correlations).

The semi-automatic procedure introduced here works best in repeated survey questions where results from previous waves have been labeled or for one-off questions where the sample size is large. How large should the training data be? We have used 5-fold cross-validation to evaluate the algorithm, but cross-validation is not appropriate in a production environment. If the question was asked in a previous wave, train the algorithm on all labeled data from all previous waves. If not, set a "sufficiently large" number of texts aside for labeling and training, and use the semi-automatic procedure on the remainder of the data. How large "sufficiently large" is depends on the task at hand. For single labeling tasks we have found that often 500 training samples are sufficient (Schonlau and Couper, 2016). There is a tradeoff: a larger data set predicts more accurately but also reduces the scope for time savings as fewer unlabeled observations remain. Under reasonable assumptions, Schonlau and Couper (2016) suggested human coding time savings for a single-label semi-automatic coding procedure attempting to code 1000 (9500) texts might be 14 (133) hours. 133 hours is equivalent to 16.6 eight-hour working days. Whether those time savings are large enough to warrant implementation of a semi-automatic procedure may be best decided with knowledge of the specific task and in the context of the specific production environment.

If some label combinations cannot occur in individual data sets, such constraints on label combinations may be added. For example, for the Happy data, if the label “nothing is turned on all other labels must be turned off. Knowing that “nothing is incompatible with other labels requires some domain expertise. It would be straightforward to modify the algorithm to accommodate this constraint. Of course, all methods except BR already exploit dependencies between labels; implementing this constraint may not affect performance very much. We did not implement such constraints in this article to avoid the appearance of the algorithms heavily relying on the constraints.

Limitations of this work include that the experimental study was conducted using three text data sets only. While there is no guarantee that performance will be equally good on other data sets, data used in this paper consider different topics in different languages, which increases the appeal of MEPCC. Also, all of the multi-label algorithms in this article used the same base learner (SVM) for classification. While SVM is one of the best performing approaches, other learning methods that produce probability outcomes could be chosen.

In conclusion, we investigated semi-automated classification for open-ended questions when the data are multi-labelled using existing multi-label algorithms. We have proposed a new algorithm for semi-automatic classification that effectively combines multiple PCC models. The experimental results on three different example data show that the proposed approach outperforms BR, LP and PCC in terms of subset accuracy and Hamming loss at most production rates. Although we focused on survey data from open-ended questions, the proposed approach can also be applied to other types of multi-label data when semi-automated classification is desired. A comprehensive analysis encompassing a variety of data in the context of semi-automated classification deserves further investigation.

References

Behr, D., M. Braun, L. Kaczmirek, and W. Bandilla (2014). Item comparability in cross-national surveys: results from asking probing questions in cross-national web surveys about attitudes towards civil disobedience. *Quality & Quantity 48*(1), 127–148.

Braun, M., D. Behr, and L. Kaczmirek (2013). Assessing cross-national equivalence of measures of xenophobia: Evidence from probing in web surveys. *International Journal of Public Opinion Research 25*(3), 383–395.

Breiman, L. (2001). Random forests. *Machine Learning 45*, 5–32.

Dembczyński, K., W. Cheng, and E. Hüllermeier (2010). Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 279–286.

Dembczyński, K., W. Waegeman, and E. Hüllermeier (2012). An analysis of chaining in multi-label classification. In L. De Raedt, C. Bessiere, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. Lucas (Eds.), *Frontiers in Artificial Intelligence and Applications*, Volume 242, pp. 294–299. IOS Press.

Guenther, N. and M. Schonlau (2016). Support vector machines. *The Stata Journal 16*(4), 917–937.

Gweon, H., M. Schonlau, L. Kaczmirek, M. Blohm, and S. Steiner (2017). Three methods for occupation coding based on statistical learning. *Journal of Official Statistics 33*(1), 101–122.

ISSP Research Group (2012). International social survey programme: Citizenship - ISSP 2004. GESIS data archive, Cologne. ZA3950 data file version 1.3.0, `https://doi.org/10.4232/1.11372`.

Manning, C., P. Raghavan, and H. Schütze (2008). *Introduction to Information Retrieval*, Chapter 2.2. Cambridge, England: Cambridge University Press.

Matthews, P., G. Kyriakopoulos, and M. Holcekova (2018). Machine learning and verbatim survey responses: Classification of criminal offences in the crime survey for England and Wales. Paper presented at BigSurv18, Barcelona, Spain.

Mena, D., E. Montañés, J. R. Quevedo, and J. J. Del Coz (2015). Using A* for inference in probabilistic classifier chains. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 3707–3713. AAAI Press.

Meyer, D., E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch (2014). *e1071: Misc Functions of The Department of Statistics, TU Wien.* `http://CRAN.R-project.org/package=e1071`.

Niculescu-Mizil, A. and R. Caruana (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, New York, NY, USA, pp. 625–632. ACM.

Platt, J. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans (Eds.), *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press.

R Core Team (2014). *R: a Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. `http://www.R-project.org/`.

Read, J., B. Pfahringer, G. Holmes, and E. Frank (2009). Classifier chains for multi-label classification. In W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor (Eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 254–269. Springer.

Read, J., B. Pfahringer, G. Holmes, and E. Frank (2011). Classifier chains for multi-label classification. *Machine Learning 85*(3), 333–359.

Schonlau, M. and M. P. Couper (2016). Semi-automated categorization of open-ended questions. *Survey Research Methods 10*(2), 143–152.

Schonlau, M., N. Guenther, and I. Sucholutsky (2017). Text mining using ngram variables. *The Stata Journal 17*(4), 866–881.

Schonlau, M., H. Gweon, and M. Wenemark (to appear). Automatic classification of open-ended questions: Check-all-that-apply questions. *Social Science Computer Review*. First published online August 20, 2019. `https://journals.sagepub.com/doi/full/10.1177/0894439319869210`.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys 34*(1), 1–47.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory.* 2nd edition. Springer.

Wenemark, M., M. Borgstedt-Risberg, P. Garvin, S. Dahlin, J. Jusufbegovic, C. Gamme, V. Johansson, and E. Bjrn (2018). Psykisk hlsa i sydstra sjukvrdsregionen: En kartlggning av sjlvskattad psykisk hlsa i jnkping. Kalmar och stergtlands ln hsten 2015/16. Retrieved from `https://vardgivarwebb.regionostergotland.se/pages/285382/Psykisk_halsa_syostra_sjukvarsregionen.pdf`.

Ye, C., R. Medway, and C. Kelley (2018). Natural language processing for open-ended survey questions. Paper presented at BigSurv18, Barcelona, Spain.