**Multi-Label Topic Classification for Hadith of Bukhari in Indonesian Translation using Information Gain and Backpropagation Neural Network**

Muhammad Yuslan Abu Bakar, Adiwijaya, Said Al Faraby

Dept. of Informatics, Telkom University, Indonesia

**ABSTRACT**

Hadith is the second source of law and guidance for Moslems after the Qur'an and there are so many hadith that have been narrated by the expert of hadith so far. This research builds a system that can classify shahih hadith of Bukhari in Indonesian language. This topic is raised to meet the needs of importance for Moslems so they know about the suggestions and restrictions contained in a hadith. Backpropagation neural network is chosen because this method can do classification with a huge number of varied features. Assisted with information gain as feature selection in order to select influential features for each class label on multi-label data and single label which has never been done before. The results show that 92.61% of multi-label hadith data can be correctly classified. Meanwhile, 65.71% of the single label hadith data can be classified correctly using information gain as feature selection.

**INTRODUCTION**

Hadith is the second source of law after the Qur'an as a guide for the life of Moslems[1]. There are so many hadiths narrated by experts of hadith and have been grouped into several categories to make it easier for others to understand. One of the hadith experts who has narrated many hadiths of the Prophet Muhammad is Bukhari.

Hadith is used to help Muslims especially in understanding the sciences of Islam. There are so many hadiths that encourage us to do good. There is also a prohibition in the matter of sinning to Allah SWT. In order to know that, a system is needed to identify whether the hadith is recommended, prohibited or informed. From that point, the question arises as to how a translated hadith in Indonesian can be classified into a more specific form of hadith of suggestion, hadith of prohibition and hadith of information. A hadith may be one of the three types of classes but it does not rule out that it is a combination of all three.

But there is currently no research that has classified the hadith into several classification groups or commonly referred to as multi-label classification. This is necessary given that a hadith contains not only information but a special message to be conveyed as a form of advice and prohibitions on a particular scientific topic.

On that basis the authors will focus the research in making the classification model based on hadith of Bukhari as much as 1064 data hadith into the form of multi-label classification. These hadiths have been labeled previously for the purpose of training the system. Information Gain is selected as a feature selection because it works by assigning value to each feature and only selected features correlated to the appropriate class. The method that will be used is Backpropagation Neural Network to solve the problems that have been described previously. The Backpropagation Neural Network method was chosen based on research previously conducted by Min-Ling Zhang and Zhi-Hua Zhou on multi-label Neural Networks with Applications to Functional Genomics and Text Categorization [12]. The results obtained from this research indicate that Backpropagation Neural Network method can do better multi-label text classification than some other method. Features that have been obtained in the selection process and feature extraction will be used as input on the Backpropagation Neural Network classification system. Meanwhile, the results obtained from the classification process is the class label of each data.

Some of the problems restricted to this research are the use of a dataset of 1064 Indonesian translations of hadiths with three labels of advice, prohibitions and information. The proportion of data for each class is also uneven, as evidenced by the presence of a class that has a total of about 75% of the data. Difficult to find data with 3 types of labels is also one of the obstacles in this research.

An evaluation method is needed to measure how well the system has been built and applied to this research. Hamming Loss is chosen as an evaluation metric to measure the performance of the multi-label classification results that have been obtained. Hamming Loss is one of the commonly used evaluation metrics for multi-label data testing by calculating the error rate of the classification results.

The next sections in this journal are as follows. In the second part, there are related studies that describe previous research relating to current research and some literature related research. In the third section, will be explained about the system to be built on this research is the classification of Bukhari hadith translation in Indonesian language using the method of Information Gain and Backpropagation Neural Network. Then in the fourth section, will be presented test results and analysis of results that have been obtained. Furthermore, in the fifth section, will be explained about the conclusions obtained from this research.

**RELATED TASKS**

Hadith can be classified into many special groups. As explained in [1], hadith can be categorized as saheeh, maudo ', dhaif and hasan. Hadiths can also be categorized as hadith of suggestions, hadith of restrictions and hadith of information as described by Eliza [3] and Andina [4]. The classification of the hadith into several groups is necessary to obtain more specific information from the hadith. By knowing the group of a hadith, it is hoped that the search for information related to the hadith can be easily done.

The research related to the classification of hadiths in earlier Indonesian translations is to classify whether they are strong (saheeh) or weak hadiths. So far the classification of hadiths with the class of hadith of information, hadith of restrictions and hadith of suggestions have been made on research [3] and [4]. The study classified only into one class of three existing classes or in other words the classification model that is built is a multiclass classification. As for most similar studies that have previously been done are still in Arabic form.

Kawther A. Aldhlan [1], has undertaken a mechanism to improve the performance of the classification of hadiths. The method used is the Decision Tree because according to him is the right approach for the classification of hadith due to the ease of induction rules and interpretation results. This method is also used because it has the ability to overcome the lost value but the effort required to achieve that is considered a weakness. Ignoring this lost value can lead to critical decisions such as the misclassification of hadith. To overcome this, they use a mechanism called Missing Data Detector (MDD).

While Al-Kabi [6], using the TF-IDF method with the help of Microsoft Visual Basic Programming Language. The reason for using this programming language is because it supports Arabic writing / text and data in the form of documents / files. The classification is done into 8 classes form "Knowledge", "Praying", "Eclipse", "Call to Prayer", "Faith", "Good Manners", "Fasting" and "Almsgiving". The hadith used in this classification is Bukhari hadith with an average accuracy of 83.2%. The result of this accuracy depends entirely on the process of stopwords and stemming.

Eliza J [3] classified the hadith of suggestion, hadith prohibition and hadith information on Hadiths shahih Bukhari based on Unigram model using Artificial Neural Network (ANN). It also uses the help of the TF-IDF feature extraction method to get values on every word. Performance results obtained with F1-Score of 85%. Similar research has also been done by Andina K [4] in 2017 using the help of Support Vector Machine method. K-Fold is used to divide data into training data and test data. The average performance obtained using this method is 88%.

For multi-label cases in text, there have been several studies related to it. Min-Ling Zhang and Zhi-Hua Zhou [12] in 2006 have conducted multi-label related research on text categorization. The method used is a neural network called BP-MLL (Backpropagation for Multi-Label Learning). This method derives from popular backpropagation algorithms using error functions that capture multi-label learning characteristics, i.e. the label of a feature must be ranked higher than that not included in the instance. They use the method because the result of accuracy is higher than other multi-label learning methods such as BoosTexter, ADTBoost.MH, Rank-SVM and BasicBP.

**MULTI-LABEL CLASSIFICATION METHODS**

We can group the existing methods for multi-label classification into two main categories: a) *problem transformation methods*, and b) *algorithm adaptation methods*. We call problem transformation methods, those methods that transform the multi-label classification problem either into one or more single-label classification or regression problems, for both of which there exists a huge bibliography

of learning algorithms. We call algorithm adaptation methods, those methods that extend specific learning algorithms in order to handle multi-label data directly.

**Problem transformation methods**

To exemplify these methods we will use the data set of Table 1. It consists of four examples (documents in this case) that belong to one or more of four classes: *Sports*, *Religion*, *Science*, and *Politics*.

**Table 1: Example of a multi-label data set**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ex. | Sports | Religion | Science | Politics |
| 1 | X |  |  | X |
| 2 |  |  | X | X |
| 3 | X |  |  |  |
| 4 |  | X | X |  |

There exist two straightforward problem transformation methods that force the learning problem into traditional single-label classification (Boutell et al., 2004). The first one (dubbed PT1) subjectively or randomly selects one of the multiple labels of each multi-label instance and discards the rest, while

the second one (dubbed PT2) simply discards every multi-label instance from the multi-label data set. Table 2 and Table 3 show the transformed data set using methods PT1 and PT2 respectively. These two problem transformation methods discard a lot of the information content of the original multi- label data set and are therefore not considered further in this work.

**Table 2: Transformed data set using PT1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ex. | Sports | Religion | Science | Politics |
| 1 | X |  |  |  |
| 2 |  |  |  | X |
| 3 | X |  |  |  |
| 4 |  | X |  |  |

**Table 3: Transformed data set using PT2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ex. | Sports | Religion | Science | Politics |
| 3 | X |  |  |  |

The third problem transformation method that we will mention (dubbed PT3), considers each different set of labels that exist in the multi-label data set as a single label. It so learns one single-label

classifier *H*: *X* → P(*L*) , where P(*L*) is the power set of *L*. Table 4 shows the result of transforming the data set of Table 1 using this method. One of the negative aspects of PT3 is that it may lead to data sets with a large number of classes and few examples per class. PT3 has been used in the past in (Boutell et al., 2004; Diplaris et al., 2005).

**Table 4: Transformed data set using PT3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ex. | Sports | (Sports ∧ Politics) | (Science ∧ Politics) | (Science ∧ Religion) |
| 1 |  | X |  |  |
| 2 |  |  | X |  |
| 3 | X |  |  |  |
| 4 |  |  |  | X |

The most common problem transformation method (dubbed PT4) learns |*L*| binary classifiers *Hl*: *X* →

{*l*, ¬*l*} , one for each different label *l* in *L*. It transforms the original data set into |*L*| data sets *Dl* that contain all examples of the original data set, labelled as *l* if the labels of the original example contained *l* and as ¬*l* otherwise. It is the same solution used in order to deal with a single-label multi- class problem using a binary classifier.

For the classification of a new instance *x* this method outputs as a set of labels the union of the labels

that are output by the |*L*| classifiers:

*H PT* 4 (*x*) = U {*l*} : *H l* (*x*) = *l*

*l*∈*L*

Figure 1 shows the four data sets that are constructed by PT4 when applied to the data set of Table 1. PT4 has been used in the past in (Boutell et al., 2004; Goncalves & Quaresma, 2003; Lauser & Hotho,

2003; Li & Ogihara, 2003).

**Figure 1: The four data sets that are constructed by PT4**

|  |  |  |
| --- | --- | --- |
| Ex. | Politics | ¬Politics |
| 1 | X |  |
| 2 | X |  |
| 3 |  | X |
| 4 |  | X |

|  |  |  |
| --- | --- | --- |
| Ex. | Sports | ¬Sports |
| 1 | X |  |
| 2 |  | X |
| 3 | X |  |
| 4 |  | X |

(a)

|  |  |  |
| --- | --- | --- |
| Ex. | Science | ¬Science |
| 1 |  | X |
| 2 | X |  |
| 3 |  | X |
| 4 | X |  |

(b)

|  |  |  |
| --- | --- | --- |
| Ex. | Religion | ¬Religion |
| 1 | X |  |
| 2 | X |  |
| 3 |  | X |
| 4 |  | X |

(c)

(d)

A straightforward, yet undocumented, problem transformation method is the following (dubbed PT5): Firstly, it decomposes each example (*x*, *Y*) into |*Y*| examples (*x*, *l*) for all *l* ∈ *Y*. Then it learns one single-label *coverage-based* classifier from the transformed data set. Distribution classifiers are those classifiers that can output a distribution of certainty degrees (or probabilities) for all labels in L. Finally it post-processes this distribution to output a set of labels. One simple way to achieve this is to output those labels for which the certainty degree is greater than a specific threshold (e.g. 0.5). A

more complex way is to output those labels for which the certainty degree is greater than a percentage

(e.g. 70%) of the highest certainty degree. Table 5 shows the result of transforming the data set of

Table 1 using this method.

**Table 5: Transformed data set using PT5**

|  |  |
| --- | --- |
| Ex. | Class |
| 1 | Sports |
| 1 | Politics |
| 2 | Science |
| 2 | Politics |
| 3 | Sports |
| 4 | Religion |
| 4 | Science |

**Algorithm adaptation methods**

Clare and King (2001) adapted the C4.5 algorithm for multi-label data. They modified the formula of entropy calculation as follows:

*N*

*entropy*(*S* ) = −∑ ( *p*(*ci* )log *p*(*ci* ) + *q*(*ci* )log *q*(*ci* ))

*i* =1

where *p*(*ci*) = relative frequency of class *ci* and *q*(*ci*) = 1−*p*(*ci*). They also allowed multiple labels in the leaves of the tree.

Adaboost.MH and Adaboost.MR (Schapire & Singer, 2000) are two extensions of AdaBoost (Freund

& Schapire, 1997) for multi-label classification. They both apply AdaBoost on weak classifiers of the form *H*:*X* ×*L* → *R*. In AdaBoost.MH if the sign of the output of the weak classifiers is positive for a new example *x* and a label *l* then we consider that this example can be labelled with *l*, while if it's negative then this example is not labelled with *l*. In AdaBoost.MR the output of the weak classifiers is considered for ranking each of the labels in *L*.

Although these two algorithms are adaptations of a specific learning approach, we notice that at their core, they actually use a problem transformation (dubbed PT6): Each example (*x*, *Y*) is decomposed into |*L*| examples (*x*, *l*, *Y*[*l*]), for all *l* ∈ *L*, where *Y*[*l*] = 1 if *l* ∈ *Y*, and [*l*] = −1 otherwise. Table 6 shows the result of transforming the data set of Table 1 using this method.

ML-*k*NN (Zhang & Zhou, 2005) is an adaptation of the *k*NN lazy learning algorithm for multi-label data. Actually this method follows the paradigm of PT4. In essence, ML-*k*NN uses the *k*NN algorithm independently for each label *l*: It finds the *k* nearest examples to the test instance and considers those that are labelled at least with *l* as positive and the rest as negative. What mainly differentiates this method from the application of the original *k*NN algorithm to the transformed problem using PT4 is the use of prior probabilities. ML-*k*NN has also the capability of producing a ranking of the labels as an output.

**Table 6: Transformed data set using PT6**

|  |  |  |
| --- | --- | --- |
| Ex. | *l* | *Y*[*l*] |
| 1 | Sports | 1 |
| 1 | Religion | -1 |
| 1 | Science | -1 |
| 1 | Politics | 1 |
| 2 | Sports | -1 |
| 2 | Religion | -1 |
| 2 | Science | 1 |
| 2 | Politics | 1 |
| 3 | Sports | 1 |
| 3 | Religion | -1 |
| 3 | Science | -1 |
| 3 | Politics | -1 |
| 4 | Sports | -1 |
| 4 | Religion | 1 |
| 4 | Science | 1 |
| 4 | Politics | -1 |

Luo and Zincir-Heywood (2005) present two systems for multi-label document classification, which are also based on the *k*NN classifier. The main contribution of their work is on the pre-processing stage for the effective representation of documents. For the classification of a new instance, the systems initially find the *k* nearest examples. Then for every appearance of each label in each of these examples, they increase a corresponding counter for that label. Finally they output the *N* labels with the largest counts. *N* is chosen based on the number of labels of the instance. This is an inappropriate strategy for real-world use, where the number of labels of a new instance is unknown.

McCallum (1999) defines a probabilistic generative model according to which, each label generates different words. Based on this model a multi-label document is produced by a mixture of the word distributions of its labels. The parameters of the model are learned by maximum a posteriori estimation from labelled training documents, using Expectation Maximization to calculate which

labels were both the mixture weights and the word distributions for each label. Given a new document the label set that is most likely is selected with Bayes rule. This approach for the classification of a

new document actually follows the paradigm of PT3, where each different set of labels is considered independently as a new class.

Elisseeff and Weston (2002) present a ranking algorithm for multi-label classification. Their algorithm follows the philosophy of SVMs: it is a linear model that tries to minimize a cost function while maintaining a large margin. The cost function they use is ranking loss, which is defined as the average fraction of pairs of labels that are ordered incorrectly. However, as stated earlier, the disadvantage of a ranking algorithm is that it does not output a set of labels.

Godbole and Sarawagi (2004) present two improvements for the Support Vector Machine (SVM) classifier in conjunction with the PT4 method for multi-label classification. The first improvement could easily be abstracted in order to be used with any classification algorithm and could thus be considered an extension to PT4. The main idea is to extend the original data set with |*L*| extra features containing the predictions of each binary classifier. Then a second round of training |*L*| new binary classifiers takes place, this time using the extended data sets. For the classification of a new example, the binary classifiers of the first round are initially used and their output is appended to the features of the example to form a meta-example. This meta-example is then classified by the binary classifiers of the second round. Through this extension the approach takes into consideration the potential dependencies among the different labels. Note here that this improvement is actually a specialized case of applying Stacking (Wolpert, 1992) (a method for the combination of multiple classifiers) on top of PT4.

The second improvement of (Godbole & Sarawagi, 2004) is SVM-specific and concerns the margin of SVMs in multi-label classification problems. They improve the margin by a) removing very similar negative training instances which are within a threshold distance from the learnt hyperplane, and b) removing negative training instances of a complete class if it is very similar to the positive class,

based on a confusion matrix that is estimated using any fast and moderately accurate classifier on a

held out validation set. Note here that the second approach for margin improvement is actually SVM

independent. Therefore, it could also be used as an extension to PT4.

MMAC (Thabtah, Cowling & Peng, 2004) is an algorithm that follows the paradigm of *associative classification*, which deals with the construction of classification rule sets using association rule mining. MMAC learns an initial set of classification rules through association rule mining, removes the examples associated with this rule set and recursively learns a new rule set from the remaining examples until no further frequent items are left. These multiple rule sets might contain rules with similar preconditions but different labels on the right hand side. Such rules are merged into a single multi-label rule. The labels are ranked according to the support of the corresponding individual rules.

**ISSUES**

**How much multi-label is a data set?**

Not all data sets are equally multi-label. In some applications the number of labels of each example is small compared to |*L*|, while in others it is large. This could be a parameter that influences the performance of the different multi-label methods. We here introduce the concepts of *label cardinality* and *label density* of a data set. Let *D* be a multi-label data set consisting of |*D*| multi-label examples (*xi*, *Yi*), *i* = 1..|*D*|.

*Definition 1*: Label cardinality of *D* is the average number of labels of the examples in *D*:

1 *D*

LC(*D*)= ∑ *Yi*

*D i* =1

*Definition 2*: Label density of *D* is the average number of labels of the examples in *D* divided by |*L*|:

1 *D Y*

LD(*D*) =

∑  *i*

*D i* =1 *L*

Label cardinality is independent of the number of labels |*L*| in the classification problem, and is used to quantify the number of alternative labels that characterize the examples of a multi-label training data set. Label density takes into consideration the number of labels in the classification problem. Two data sets with the same label cardinality but with a great difference in the number of labels (different label density) might not exhibit the same properties and cause different behaviour to the multi-label classification methods. The two metrics are related to each other: LC(*D*) = |*L*| LD(*D*).

**Evaluation metrics**

Multi-label classification requires different metrics than those used in traditional single-label classification. This section presents the various metrics that have been proposed in the literature. Let *D* be a multi-label evaluation data set, consisting of |*D*| multi-label examples (*xi*, *Yi*), *i* = 1..|*D*|, *Yi* ⊆ *L*. Let *H* be a multi-label classifier and *Zi* = *H(xi)* be the set of labels predicted by *H* for example *xi*.

Schapire and Singer (2000) consider the *Hamming Loss*, which is defined as:

1 *D Y* Δ*Z*

HammingLoss(*H*, *D*) =

∑  *i i*

*D i* =1 *L*

Where Δ stands for the symmetric difference of two sets and corresponds to the XOR operation in

Boolean logic.

The following metrics are used in (Godbole & Sarawagi, 2004) for the evaluation of *H* on *D*:

1

*D Y* I *Z*

Accuracy(*H*, *D*) = ∑ *i i*

*D i* =1

*Yi* U *Z i*

1

*D Y* I *Z*

Precision(*H*, *D*) =

∑ *i i*

*i* =1 *Z i*

*D*

1

*D Y* I *Z*

Recall(*H*, *D*) =

∑ *i i*

*i* =1 *Yi*

*D*

Boutell et al. (2004) give a more generalized version of the above accuracy using a parameter α ≥ 0,

called forgiveness rate:

1

α

*D* ⎛  *Y* I *Z* ⎞

∑

Accuracy(*H*, *D*) =

*D*

⎜ *i*

⎜

*Y*

*i* =1 ⎝ *i*

*i* ⎟

U *Z i* ⎟

⎠

This parameter is used in order to control the forgiveness of errors that are made in predicting labels. They also give an even more generalized version of the accuracy by introducing two additional parameters in order to allow different costs for false positives and true negatives. These two general measures of accuracy are too complex, due to the additional parameters, but could be useful in certain applications.

**EXPERIMENTAL COMPARISON OF PT METHODS**

We implemented the PT3, PT4 and PT6 methods in Java, within the framework of the WEKA (Witten

& Frank, 1998) library of machine learning algorithms, and made the software publicly available at the following URL (mlkd.csd.auth.gr/multilabel.html). We experimented with the three PT methods in conjunction with the following classifier learning algorithms: kNN (Aha, Kibler & Albert), C4.5 (Quinlan, 1993), Naive Bayes (John & Langley, 1995) and SMO (Platt, 1998). For performance

evaluation, we used the HammingLoss, Accuracy, Precision and Recall metrics that were presented in the previous section.

We experimented on the following multi-label data sets: *genbase* (Diplaris et al., 2005) and *yeast* (Elisseeff & Weston, 2002) are biological data sets that are concerned with protein function classification and gene function classification respectively. The *scene* data set (Boutell et al., 2004) contains data related to a scene classification problem. These data sets were retrieved from the site of the Support Vector Classification library LIBSVM (Chang & Lin, 2001), and transformed to a specific format that is suitable for our software, based on the ARFF file format of the WEKA library. The transformed data sets are also available at the aforementioned URL.

The details of the data sets, such as the number of examples, the number of numeric and discrete attributes the number of classes and their label density are given in Table 7. We notice that *genbase* (LD=0.05) and *scene* (LD=0.18) are quite sparse multi-label data sets with less than 1.5 labels per example on average. The *yeast* dataset on the other hand is denser (LD=0.30) with more than 4 labels per example on average.

**Table 7: Examples, numeric and discrete attributes, labels and LD of datasets**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Data set** | **Examples** | | **Attributes** | | **Labels** | **Label**  **Density** | **Label**  **Cardinality** |
| **Train** | **Test** | **Numeric** | **Discrete** |
| genbase | 463 | 199 | 0 | 1185 | 27 | 0.05 | 1.35 |
| yeast | 1500 | 917 | 103 | 0 | 14 | 0.30 | 4.25 |
| scene | 1211 | 1196 | 294 | 0 | 6 | 0.18 | 1.08 |

Table 8 presents analytical results on the three data sets. We will first discuss the results in terms of accuracy. The combination of the PT3 method together with the SMO learning algorithm gives the best results in each of the three data sets. In addition the PT3 method has the highest mean accuracy for all learning algorithms in each of the three data sets, followed by PT4 and then by PT6. This means that it is the best method independently of learning algorithm in each of the three data sets. This is an interesting result, given that the PT3 method is not so popular in the literature compared to PT4.

We will now discuss the results in terms of Hamming loss. In *genbase* the best results are obtained with PT4 in combination with either *k*NN or SMO. In *yeast* the best results are obtained again with PT4 in combination with SMO, while in *scene* the best results are obtained with PT3 in conjunction with SMO. Independently of the algorithm used, PT3 is the best method in *scene*, PT4 in *genbase* and PT6 in *yeast*.

One noteworthy result is that PT6 does not perform well in combination with SMO for the *scene* and *genbase* data sets. Note that these two data sets are quite sparse as LD(*scene*)=0.18 and LD(*genbase*)=0.05. This means that after the transformation, the class attribute will have a large

number of examples with a value of -1. It seems that in these cases SMO learns to predict always -1. This leads to zero accuracy, precision and recall, while Hamming loss becomes equal to the label density of the data set.

**Table 8: Resuls on the three data sets**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **genbase** | | | | | | | | | | | | |
|  | **PT3** | | | | **PT4** | | | | **PT6** | | | |
| **Metric** | ***k*NN** | **C4.5** | **NB** | **SMO** | ***k*NN** | **C4.5** | **NB** | **SMO** | ***k*NN** | **C4.5** | **NB** | **SMO** |
| **HamLoss** | 0,004 | 0,046 | 0,057 | 0,001 | 0,000 | 0,001 | 0,035 | 0,000 | 0,025 | 0,002 | 0,103 | 0,046 |
| **Accuracy** | 0,964 | 0,984 | 0,340 | 0,993 | 0,989 | 0,987 | 0,273 | 0,991 | 0,543 | 0,984 | 0,019 | 0,000 |
| **Recall** | 0,990 | 0,995 | 0,347 | 1,000 | 0,997 | 0,995 | 0,276 | 0,997 | 0,548 | 0,994 | 0,020 | 0,000 |
| **Precision** | 0,964 | 0,984 | 0,340 | 0,993 | 0,992 | 0,992 | 0,273 | 0,993 | 0,543 | 0,990 | 0,117 | 0,000 |
| **yeast** | | | | | | | | | | | | |
|  | **PT3** | | | | **PT4** | | | | **PT6** | | | |
| **Metric** | ***k*NN** | **C4.5** | **NB** | **SMO** | ***k*NN** | **C4.5** | **NB** | **SMO** | ***k*NN** | **C4.5** | **NB** | **SMO** |
| **HamLoss** | 0,229 | 0,286 | 0,243 | 0,206 | 0,243 | 0,259 | 0,301 | 0,200 | 0,208 | 0,259 | 0,261 | 0,233 |
| **Accuracy** | 0,495 | 0,399 | 0,464 | 0,530 | 0,479 | 0,423 | 0,421 | 0,502 | 0,514 | 0,423 | 0,329 | 0,337 |
| **Recall** | 0,628 | 0,528 | 0,608 | 0,672 | 0,601 | 0,593 | 0,531 | 0,711 | 0,665 | 0,593 | 0,604 | 0,748 |
| **Precision** | 0,596 | 0,529 | 0,575 | 0,615 | 0,596 | 0,561 | 0,610 | 0,579 | 0,623 | 0,561 | 0,407 | 0,337 |
| **scene** | | | | | | | | | | | | |
|  | **PT3** | | | | **PT4** | | | | **PT6** | | | |
| **Metric** | ***k*NN** | **C4.5** | **NB** | **SMO** | ***k*NN** | **C4.5** | **NB** | **SMO** | ***k*NN** | **C4.5** | **NB** | **SMO** |
| **HamLoss** | 0,113 | 0,148 | 0,139 | 0,100 | 0,125 | 0,139 | 0,247 | 0,114 | 0,147 | 0,139 | 0,357 | 0,181 |
| **Accuracy** | 0,668 | 0,572 | 0,603 | 0,704 | 0,637 | 0,513 | 0,435 | 0,571 | 0,242 | 0,513 | 0,076 | 0,000 |
| **Recall** | 0,703 | 0,598 | 0,631 | 0,737 | 0,669 | 0,534 | 0,443 | 0,596 | 0,253 | 0,534 | 0,078 | 0,000 |
| **Precision** | 0,668 | 0,584 | 0,633 | 0,713 | 0,651 | 0,611 | 0,816 | 0,628 | 0,243 | 0,611 | 0,317 | 0,000 |

**CONCLUSIONS AND FUTURE WORK**

This work was involved with the task of multi-label classification: It introduced the problem, gave an organized presentation of the methods that exist in the literature and provided comparative experimental results for some of these methods. To the best of our knowledge, there is no other review paper on the interesting and upcoming task of multi-label classification.

In the future we intend to perform a finer-grained categorization of the different multi-label classification methods and perform more extensive experiments with more data sets and methods. We also intend to perform a comparative experimental study of problem adaptation methods.

**REFERENCES**

Aha, D.W., Kibler, D., & Albert, M.K. (1991), 'Instance-based learning algorithms', Machine

Learning, vol. 6, no. 1, pp. 37-66 .

Boutell, M.R., Luo, J., Shen, X. & Brown, C.M. (2004), 'Learning multi-label scene classification', Pattern Recognition, vol. 37, no. 9, pp. 1757-71.

Chang, C.-C., & Lin, C.-J. (2004), 'LIBSVM : a library for support vector machines', Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Clare, A. & King, R.D. (2001), 'Knowledge Discovery in Multi-Label Phenotype Data', paper presented to Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001), Freiburg, Germany.

Diplaris, S., Tsoumakas, G., Mitkas, P. & Vlahavas, I. (2005), 'Protein Classification with Multiple Algorithms', paper presented to Proceedings of the 10th Panhellenic Conference on Informatics (PCI 2005), Volos, Greece, November.

Elisseeff, A. & Weston, J. (2002), 'A kernel method for multi-labelled classification', paper presented to Advances in Neural Information Processing Systems 14.

Freund, Y. & Schapire, R.E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119-39.

Godbole, S. & Sarawagi, S. (2004), 'Discriminative Methods for Multi-labeled Classification', paper presented to Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2004).

Goncalves, T. & Quaresma, P. (2003), 'A Preliminary Approach to the Multilabel Classification Problem of Portuguese Juridical Documents', paper presented to Proceedings of the 11th Portuguese Conference on Artificial Intelligence (EPIA '03).

Jin, R. & Ghahramani, Z. (2002), 'Learning with Multiple Labels', paper presented to Proceedings of

Neural Information Processing Systems 2002 (NIPS 2002), Vancoucer, Canada.

John, G. & Langley, P. (1995), 'Estimating continuous distributions in bayesian classifiers', paper presented to Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Vancoucer, Canada

Lauser, B. & Hotho, A. (2003), 'Automatic multi-label subject indexing in a multilingual environment', paper presented to Proceedings of the 7th European Conference in Research and Advanced Technology for Digital Libraries (ECDL 2003).

Lewis, D.D., Tony, Y. Y., Rose, G. & Li, F. (2004). 'RCV1: A new benchmark collection for text categorization research', Journal of Machine Learning Research, Vol 5, pp 361-397.

Li, T. & Ogihara, M. (2003), 'Detecting emotion in music', paper presented to Proceedings of the

International Symposium on Music Information Retrieval, Washington D.C., USA.

Luo, X. & Zincir-Heywood, A.N. (2005), 'Evaluation of Two Systems on Multi-class Multi-label Document Classification', paper presented to Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems.

Maron, O. & Lozano-Perez, T. (1997), 'A framework for Multiple-Instance learning', paper presented to Proceedings of Neural Information Processing Systems 1997 (NIPS 1997).

McCallum, A. (1999), 'Multi-label text classification with a mixture model trained by EM', paper presented to Proceedings of the AAAI' 99 Workshop on Text Learning.

Platt, J. (1998), 'Fast training of support vector machines using sequential minimal optimization', In B.

Scholkopf, B., Burges, C., & Smola, A., Advances in Kernel Methods - Support Vector

Learning, MIT Press.

Quinlan, J.R. (1993), C4.5: Programs for Machine Learning, Morgan Kaufmann.

Schapire, R.E. & Singer, Y. (2000), 'Boostexter: a boosting-based system for text categorization', Machine Learning, vol. 39, no. 2/3, pp. 135-68.

Thabtah, F.A., Cowling, P. & Peng, Y. (2004), 'MMAC: A New Multi-class, Multi-label Associative Classification Approach', paper presented to Proceedings of the 4th IEEE International Conference on Data Mining, ICDM '04.

Witten, I.H. & Frank, E. (1999), 'Data Mining: Practical machine learning tools with Java implementations', Morgan Kaufmann.

Wolpert, D.H. (1992), 'Stacked Generalization', Neural Networks, vol. 5, pp. 241-59.

Zhang, M.-L. & Zhou, Z.-H. (2005), 'A k-Nearest Neighbor Based Algorithm for Multi-label Classification', paper presented to Proceedings of the 1st IEEE International Conference on Granular Computing.