Modified ML-\(k\)NN and Rank SVM for Multi-label Pattern Classification

Thanseeha Kassim\(^1\), B S Shajee Mohan\(^2\), K V Ahammed Muneer\(^3\)

\(^1\) M. Tech Signal Processing, AE & I Dept., Govt. Engg. College Kozhikode.
\(^2\) Dept of AE & I, Govt. Engg. College Kozhikode.
\(^3\) Dept of AE & I, Govt. Engg. College Kozhikode.

E-mail: thanu3ponnu5@gmail.com, smohan@geckkd.ac.in, ahammedcet@gmail.com

Abstract. To develop an efficient multi-label classifier is the main objective of this paper. In multi-label learning tasks such as classification, each example is associated with a set of labels, and the task is to predict the label set whose size is unknown apriori for each unseen example. In a realistic scenario each object or entity belongs to a multi-label category. Multi-Label \(k\)-Nearest Neighbor (ML-\(k\)NN), Rank-SVM (Ranking Support Vector Machine) are two popular techniques used for multi-label pattern classification. ML-\(k\)NN is a multi-label version of standard \(k\)NN and Rank SVM is a multi-label extension of standard SVM. The main aim of this work is to enhance the performance of these methods. Multi-label classifiers generally consider ranking loss, Hamming loss, one error, average precision and coverage as a performance metrics.

Keywords- Multi-Label Classification (MLC), Multi-Label \(k\)-Nearest Neighbor (ML-\(k\)NN), Ranking Support Vector Machine (Rank SVM), Feature space, Label space, Anticipated test label

1. Introduction

The objective of classification is to assign a set of labels to a test example. The classifier enhance its capability using the training process that make use of labeled examples. As shown in Fig. 1, Single label classification and Multi-label classification (MLC) are the two types of classifications. In single label classification, each test example is associated with only one label whereas in the case of multi-label classification, each example is associated with more than one label. In single label classification problems, each example or input is mapped to a scalar output. But, in multi-label classification each input is mapped onto a vector. From text categorization, the applications of MLC expand into other fields including medical diagnosis, audio applications, bio-informatics etc., In recent years, the popularity of MLC increased due to its ability to solve the real time problems[4].

Different approaches to MLC are:

- Problem transformation methods
- Algorithm adaptation methods
- Ensemble methods

In problem transformation method, MLC problem is transformed into a number of binary classification problems that consists of single label classifiers[4]. Algorithm adaptation method
extends the single label classifiers to deal with multi-label classification. The base classifier used for ensemble method is either from problem transformation method or algorithm adaptation method. In this paper, algorithm adaptation method such as standard ML-kNN and its modified version and Rank SVM is considered. And also compared their performances. ML-kNN (Multi-Label k Nearest Neighbor) is an extension of standard k nearest neighbor algorithm. ML-kNN is used for multi-label classification which is a lazy learning. In this method the k nearest neighbors of the test example is identified and by using MAP (Maximum A Posteriori) rule the label set for test example is predicted. Rank SVM is the extended version of standard SVM that is based on the principle of large margin that is necessary to reduce the ranking loss.

The remainder of this paper is arranged as follows: Section 2 gives some of the related works based on multi-label learning. Next Section briefly describes the main methodology of this paper. Section 4 presents the existing ML-kNN and its proposed variations. Section 5 summarizes the multi-label classification using Rank SVM. Section 6 explains the experimental results and the final Section gives modifications required for the proposed system in the future.

2. Existing Methods

2.1. Multi-label classification using ML-kNN

For multi-label classification, there are several algorithms available in literature. Min-Lin Zhang et al.,[2] developed one such algorithm for multi-label classification, which is termed to be a lazy learning algorithm known as Multi-label k Nearest Neighbor (ML-kNN). This method is derived from the kNN algorithm. In this method, the k nearest neighbors of the test instance is determined initially. Then by using the MAP (Maximum A Posteriori) rule and the nearest neighbors, the set of labels of the test example is determined.

2.2. Multi-label classification using Rank SVM

Mainly, SVM is used for binary classification problems. In multi-label classification, there exists the correlation between labels, hence the standard SVM is not applicable. Hence, another approach is introduced by Andre Elisseeff et al.,[11] named the ranking approach. Using this approach which is termed as Ranking Support Vector Machine (Rank SVM), class imbalance issue and the ranking loss can be reduced. It is similar to SVM that is based on linear models, defining a cost function and also it depends on the margin. By using the large margin concept, the ranking loss can be reduced in this method.
3. Proposed Method

For multi-label classification, initially the dataset is chosen for the experiments and the features are extracted. Training the model and testing the unknown examples are conducted for different models. Multi-label learning algorithms include ML-kNN and Rank SVM. Using these algorithms the test data are classified. The label vectors are estimated and performance evaluation metrics such as ranking loss, Hamming loss, average precision, coverage, one-error are used to compare the performances. By comparing these metrics the performance of different multi-label learning algorithms can be analyzed.

4. Multi-label Classification using ML-kNN and its Variations

![Figure 2. Block Schematic of multi-label classification using ML-kNN](image)

The training dataset with known label set is used for ML-kNN. The label set of a test example can be found out using this training dataset in which the k nearest neighbor of each test instance is considered. Similar to kNN in single label classification, it is termed to be the lazy learning algorithm in multi-label classification. The label set of test example can be determined using statistical method Maximum A Posteriori (MAP) rule. In this paper, the standard ML-kNN is modified as given in Fig. 2, in order to improve the classification accuracy. Certain improvements in the conventional ML-kNN in the training and testing phase is proposed. ML-kNN mainly based on the distance measurement between training and testing instances, different distance metrics plays an important role in ML-kNN algorithm. Different strategies to measure the distances and analyze the similarities between examples is also proposed. In this paper, modified distance measurement both in the feature space as well as in the label space are considered.

4.1. Distances in the feature space

4.1.1. Euclidean distance: In order to measure the distance between the training example and test example in the feature space, various distance measures can be used. Most widely used distance metric is the Euclidean distance. In the standard ML-kNN, Euclidean distance is used to ascertain the class label of the test example by using its k nearest neighbors. Let \( \mathbf{u} \) and \( \mathbf{v} \) are represented by feature vectors \( \mathbf{u} = (u_1, u_2, ..., u_m) \) and \( \mathbf{v} = (v_1, v_2, ..., v_m) \) where \( m \) is the dimension of feature space. The Euclidean distance between two vectors \( \mathbf{u} \) and \( \mathbf{v} \) is given by,

\[
\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{m}(u_i - v_i)^2}
\]  

(1)
4.1.2. Mahalanobis distance: Another distance metric used for measuring the distance between test and training example is Mahalanobis distance. In multivariate space, the distance between an example $z$ and the distribution $D$ can be computed using Mahalanobis distance, hence for ML-$k$NN this distance measure is expected to provide a good result. The main advantage of the Mahalanobis distance is that, it consider the covariance that better helps to measure the similarity between two different examples. The Mahalanobis distance is given by:

$$Mahalanobis \ distance = \sqrt{(z - \mu_z)^T \Sigma^{-1} (z - \mu_z)} \quad (2)$$

Here, $\Sigma^{-1}$ is the inverse of covariance matrix, and $\mu_z$ is the mean of data vectors. For training phase, the covariance, mean of training data is considered. The $k$ nearest neighbors of the test example can be determined by using the Mahalanobis distance. The test example is labeled with the label of majority of its $k$ nearest examples obtained as above.

4.2. Distances in the label space

4.2.1. Jaccard distance: The $k$NN algorithm is a distance based algorithm, different distance metrics between the feature vectors can be considered. The Jaccard distance provides the similarity between a sequence of inputs like $1, 0, 0,...$. Here Jaccard distance is considered to measure the similarity in the label space. To measure the dissimilarity between two sets can be determined by Jaccard distance, which is the complement of the Jaccard index. Jaccard similarity is good for cases where duplication does not matter. Let $S_1$ and $S_2$ are two sets and the Jaccard distance is given by:

$$Jaccard \ distance = 1 - J(S_1, S_2) = \frac{|S_1 \cup S_2| - |S_1 \cap S_2|}{|S_1 \cup S_2|} \quad (3)$$

4.2.2. Anticipated test label: Anticipated test label can be determined from information available from the feature space. In order to compute the distance between test and the training examples in the label space, some intuition about the label of a test example is needed. And denoted this intuition as "anticipated test label". The steps involved in the process of determining the anticipated label of a test example are as follows:

(i) Model the training instances of all classes as separate Gaussian distributions.
(ii) Compute the Mahalanobis distance between the distributions and the test instance.
(iii) Normalize the distance by dividing all values by the maximum of the distances.
(iv) Assign the label field of each test instance to 1 for normalized distance is greater than 0.5.
(v) Otherwise assign the label field to 0, repeat for all the label fields the anticipated test label is obtained.
(vi) Compute the Jaccard distance between anticipated test label and train label in the label space.
(vii) Take the $k$ nearest examples that are having smaller distance values.
(viii) Depending upon the majority, assign the final label for a test instance.

After labeling all the test instances, the performance analysis of this method is conducted using the performance metrics discussed in Section 6.2.
5. Multi-label Classification using Rank SVM

Rank SVM is a variant of SVM in which techniques are applied to minimize the ranking loss by making use of the large margin principle. In order to perform the feature selection on ranking problems Rank SVM method can be used which is demonstrated as in Fig. 3. Therefore, this method is very much applicable for the real time data. It also helps to handle nonlinear cases with kernel tricks. It has some disadvantages of high computational complexity and error in prediction.

6. Experimentation

6.1. Dataset description

Mulan dataset is used for the experiments. Mulan dataset is mainly used for multi-label classification, recently it is also extended for Multi-Target Regression. It is utilized from open source library. Most of the algorithms for multi-label classification and label ranking is supported by Mulan dataset. It also provides dimensionality reduction by suitable feature selection method. By cross-validation and hold-out evaluation techniques, Mulan dataset provides approaches to calculate different evaluation metrics for multi-label classification problems[2].

| Name     | Domain | Instances | Labels |
|----------|--------|-----------|--------|
| bibtex   | text   | 7395      | 159    |
| birds    | audio  | 645       | 19     |
| bookmarks| text   | 87856     | 208    |
| flags    | images | 194       | 7      |
| emotions | music  | 593       | 6      |
| delicious| text   | 16105     | 983    |
| medical  | text   | 978       | 45     |
| yahoo    | text   | 5423 ±1259| 31±6   |
| yeast    | biology| 2417      | 14     |
| scene    | image  | 2407      | 6      |
6.2. Performance metrics

The performance of multi-label learning algorithms can be evaluated using different performance metrics. Performance evaluation is different for multi-label classification algorithms as compared to single-label classification algorithms. The following are the performance metrics for multi-label learning:

(i) Hamming loss: This metric determines the misclassification between the example-label pair. When $hloss_s(h) = 0$, the performance is perfect and for better performance, the value of $hloss_s(h)$ should be small.

(ii) One-error: It provides the number of times a top-ranked label is not in the set of proper label of the example. The smaller value of $one-error_s(f)$ provides better performance, hence the performance is perfect when $one-error_s(f)$ is perfect.

(iii) Coverage: This metric provides an average that accounts how far to go in the list of labels in order to cover all the proper labels of an example. For better performance the value of $coverage_s(f)$ should be small or for perfect result it must be zero.

(iv) Ranking loss: It gives an average of reversely ordered label pairs of an instance. When $rloss_s(f) = 0$, the algorithm is ideal, which provides better performance. An increase in the value of $rloss_s(f)$, the performance become worse.

(v) Average precision: This metric provides the average fraction of labels that are ranked above a particular label. For large values of $avgprec_s(f)$ the performance of the algorithm is good. That is, it provides better performance when $avgprec_s(f) = 1$, otherwise a decrease in the value, the performance become worse.

7. Experimental Results

The experiments for multi-label classification are performed using Matlab software. By using the Mulan dataset, the performances of different methods like ML-kNN and Rank SVM are evaluated. The performances of different modifications of ML-kNN are analyzed by using different distances, like standard ML-kNN using Euclidean distance, Mahalanobis distance and Jaccard distance given in Table 2, 3 and 4. Jaccard distance is computed in label space using anticipated label set which is given in Table 5. Thus the different methods are compared using different performance metrics discussed above.

| Dataset          | $k$ | Hamming loss | Ranking loss | Coverage | Average precision | One-error |
|------------------|-----|--------------|--------------|----------|-------------------|-----------|
| Flag (Image domain) | 3   | 0.7099       | 0.2441       | 3.9385   | 0.791             | 0.2462    |
|                  | 5   | 0.6857       | 0.2292       | 3.8015   | 0.793             | 0.2615    |
|                  | 7   | 0.7033       | 0.2313       | 3.9077   | 0.798             | 0.2308    |
|                  | 9   | 0.6791       | 0.2528       | 4.0460   | 0.7736            | 0.2615    |
|                  | 11  | 0.6835       | 0.2562       | 4.1438   | 0.7781            | 0.2769    |
| Emotion (Music domain) | 3   | 0.8787       | 0.2741       | 2.4307   | 0.7095            | 0.3911    |
|                  | 5   | 0.8845       | 0.2795       | 2.5149   | 0.7088            | 0.3614    |
|                  | 7   | 0.8765       | 0.2733       | 2.4257   | 0.7084            | 0.3812    |
|                  | 9   | 0.896        | 0.2792       | 2.5198   | 0.7076            | 0.3515    |
|                  | 11  | 0.8828       | 0.3001       | 2.5743   | 0.6848            | 0.4307    |
Table 3. Performance analysis of multi-label classification using Mahalanobis distance

| Dataset          | k  | Hamming loss | Ranking loss | Coverage | Average precision | One-error |
|------------------|----|--------------|--------------|----------|-------------------|-----------|
| Flag (Image domain) | 3  | 0.6703       | 0.2087       | 3.6154   | 0.8203            | 0.238     |
|                  | 5  | 0.7106       | 0.2005       | 3.6462   | 0.8353            | 0.138     |
|                  | 7  | 0.6967       | 0.2187       | 3.7846   | 0.8092            | 0.238     |
|                  | 9  | 0.6813       | 0.2344       | 3.8923   | 0.8022            | 0.2       |
|                  | 11 | 0.6967       | 0.221        | 3.8154   | 0.8015            | 0.2615    |
| Emotion (Music domain) | 3  | 0.9282       | 0.3802       | 2.8661   | 0.6381            | 0.495     |
|                  | 5  | 0.9599       | 0.3961       | 2.9901   | 0.6177            | 0.529     |
|                  | 7  | 0.9747       | 0.3681       | 2.8465   | 0.6287            | 0.529     |
|                  | 9  | 0.9332       | 0.3744       | 2.8762   | 0.6257            | 0.549     |
|                  | 11 | 0.9299       | 0.3815       | 2.896    | 0.6072            | 0.584     |

Table 4. Performance analysis of multi-label classification using Jaccard distance

| Dataset          | k  | Hamming loss | Ranking loss | Coverage | Average precision | One-error |
|------------------|----|--------------|--------------|----------|-------------------|-----------|
| Flag (Image domain) | 3  | 0.7077       | 0.2233       | 3.6923   | 0.8011            | 0.2462    |
|                  | 7  | 0.6484       | 0.2233       | 3.6923   | 0.8011            | 0.2462    |
|                  | 11 | 0.7269       | 0.2318       | 3.7538   | 0.794             | 0.2462    |
|                  | 15 | 0.7143       | 0.2246       | 3.8923   | 0.794             | 0.2462    |
| Emotion (Music domain) | 3  | 0.9999       | 0.4334       | 3.1337   | 0.5877            | 0.5248    |
|                  | 7  | 0.999        | 0.4334       | 3.1337   | 0.5877            | 0.5248    |
|                  | 11 | 0.9999       | 0.4334       | 3.1337   | 0.5877            | 0.5248    |
|                  | 15 | 0.9999       | 0.4334       | 3.1337   | 0.5877            | 0.5248    |
|                  | 20 | 0.9999       | 0.4334       | 3.1337   | 0.5877            | 0.5248    |

Table 5. Performance analysis of multi-label classification using Jaccard distance (using anticipated test label)

| Dataset          | k  | Hamming loss | Ranking loss | Coverage | Average precision | One-error |
|------------------|----|--------------|--------------|----------|-------------------|-----------|
| Flag (Image domain) | 3  | 0.8835       | 0.3927       | 3.3077   | 0.573             | 0.6129    |
|                  | 7  | 0.8484       | 0.3723       | 3.7092   | 0.5336            | 0.6129    |
|                  | 11 | 0.833        | 0.2917       | 3.7846   | 0.5209            | 0.6129    |
|                  | 15 | 0.7648       | 0.2917       | 3.3077   | 0.573             | 0.6129    |
| Emotion (Music domain) | 3  | 0.9999       | 0.4279       | 3.3612   | 0.586             | 0.5207    |
|                  | 7  | 0.999        | 0.4279       | 3.3612   | 0.586             | 0.5207    |
|                  | 11 | 0.9999       | 0.4279       | 3.3612   | 0.586             | 0.5207    |
|                  | 15 | 0.9999       | 0.4279       | 3.3612   | 0.586             | 0.5207    |
|                  | 20 | 0.9999       | 0.4279       | 3.3612   | 0.586             | 0.5207    |

7.1. Rank SVM

Table 6. Performance analysis of multi-label classification using Rank SVM

| Dataset          | Kernel | Hamming loss | Ranking loss | Coverage | Average precision | One-error |
|------------------|--------|--------------|--------------|----------|-------------------|-----------|
| Emotion (Music)  | Linear | 0.9926       | 0.4651       | 3.0782   | 0.551             | 0.4782    |
|                  | Gaussian | 0.8845    | 0.4795       | 2.9140   | 0.56              | 0.4614    |
|                  | Polynomial | 0.9018        | 0.4733      | 3.1257   | 0.5484           | 0.4812    |
| Flag (Image)     | Linear | 0.7429       | 0.5121       | 3.6923   | 0.751             | 0.6385    |
|                  | Gaussian | 0.7407       | 0.4177       | 3.8      | 0.7688           | 0.5215    |
|                  | Polynomial | 0.5824        | 0.2233      | 3.6923   | 0.8011           | 0.2462    |

In this paper, standard ML-kNN and Rank SVM with their modified versions are compared. The observations show that the ML-kNN with Euclidean distance provides good performance for large datasets, while for smaller datasets the Mahalanobis distance provides better results. It is also evident that multi-label classification can be performed in the label space. In both data space and label space analysis, ML-kNN outperforms well-established multi-label learning algorithms.
The comparison of standard Rank SVM with the ML-\(k\)NN and its variations reveals that, ML-\(k\)NN performs better than Rank SVM. But the performance varies for different datasets. The comparison of different methods with different performance metrics is provided in Table 7.

### Table 7. Performance comparison of different methods

| Methods                        | Hamming loss | Ranking loss | Coverage | Average precision | One-error |
|-------------------------------|--------------|--------------|----------|-------------------|-----------|
| ML-\(k\)NN (Euclidean)        | 0.75±0.05    | 0.15±0.05    | 3.5±0.1  | 0.75±0.1          | 0.2±0.05  |
| ML-\(k\)NN (Mahalanobis)      | 0.85±0.05    | 0.25±0.05    | 4.5±0.1  | 0.70±0.1          | 0.2±0.05  |
| ML-\(k\)NN (Jaccard distance) | 0.85±0.1     | 0.3±0.01     | 3.3±0.2  | 0.75±0.2          | 0.3±0.1   |
| ML-\(k\)NN (Anticipated test label) | 0.86±0.05   | 0.29±0.2     | 3.3±0.2  | 0.65±0.5          | 0.3±0.1   |
| Rank SVM                       | 0.85±0.05    | 0.15±0.05    | 3.5±0.15 | 0.70±0.1          | 0.2±0.05  |

8. Conclusion

ML-\(k\)NN is one of the multi-label learning algorithm, which is an extension of traditional \(k\)NN algorithm. It is a lazy learning algorithm for multi-label classification. In this method \(k\) nearest neighbor is considered and by using the MAP rule the label set of unknown example is determined. Improvements in ML-\(k\)NN is also proposed in this work with different options for computing distances, an algorithm is also established that computes distances in both the feature space as well as in the label space. Rank SVM is a variation of SVM that is based on the minimization of the ranking loss. The experiments are conducted on real time problems that are essentially a multi-label problems, using Mulan dataset. The performance is analyzed using different performance metrics. It is observed that, ML-\(k\)NN outperforms well-established multi-label learning algorithms. It is also observed that Rank SVM provides good performance for multi-label classification.

References

[1] Zhang M L, Zhou Z H 2006 Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization *IEEE Transactions on Knowledge and Data Engineering* 18(10), pp 1338-1351

[2] Min-Ling Zhang, Zhi-Hua Zhou 2007 ML-\(k\)NN: A Lazy Learning Approach to Multi-Label Learning *National Laboratory for Novel Software Technology* Nanjing University, Nanjing 210093, China

[3] Ruiquan Ge, Renfeng Zhang, et al., 2017 Prediction of Chronic Diseases with Multi-label Neural Network *Digital Object Identifier, IEEE Access*

[4] Seema Sharma, Deepti Mehrotra 2018 Comparative Analysis of Multi-label Classification Algorithms *ICSCCC, 978-1-5386-6373-8/18 IEEE Access*

[5] Elissef A, Weston J 2002 A Kernel Method for Multi-Labelled Classification *Advances in Neural Information Processing Systems* 14, *vol. 1 and 2*, vol 14 pp 681-687

[6] D Zufferey, T Hofer, et al., 2015 Performance comparison of multi-label learning algorithms on clinical data for chronic diseases *Comput Biol Med*, *vol. 65*, pp 34-43

[7] A Maxwell, R Li, et al., 2017 Deep learning architectures for multi-label classification of intelligent health risk prediction *BMC Bioinformatics*, *vol. 18*, pp 523

[8] Rafal Grodzicki, Jacek Mandziuk, et al., 2008 Improved Multilabel Classification with Neural Networks *PPSN X, LNCS* 5199, pp 409-416

[9] Grigoris Tsoumakas, Eleftherios Spyromitros-Xioufis, et al., 2011 MULAN: A Java Library for Multi-Label Learning *Article in Journal of Machine Learning Research*

[10] Raed Alazaidah, Farzana Kabir Ahmad 2016 Trending Challenges in Multi Label Classification *IJACSA International Journal of Advanced Computer Science and Applications, Vol. 7, No. 10*

[11] Andre Elisseeff, Jason Weston 2002 Kernel methods for Multi-labelled classification and Categorical regression problem *Bioeulff Technologies 305 Broadway, New-York, NY 10007.*

[12] M Zhang, Z Zhou, 2014 A review on multi-label learning algorithms *IEEE Trans. Knowl. Data Eng. 26* pp 1819-1837
[13] X Wang, W Zhang, et al., 2015 MultiP-SChlo: multi-label protein subchloroplast localization prediction with Chou’s pseudo amino acid composition and a novel multi-label classifier *Bioinformatics, vol. 31, pp 2639-2645*

[14] X Zhang, H Zhao, et al., 2019 A Novel Deep Neural Network Model for Multi-Label Chronic Disease Prediction *Front Genet, vol. 10, pp 351*

[15] X B Al-Salemi, S A M Noah, et al., 2016 RFBoost: An improved multi-label boosting algorithm and its application to text categorisation *Knowledge-Based Systems, vol. 103, pp 104-117*