Proceedings of Machine Learning Research 154:35–48, 2021 LIDTA 2021

ML-NCA: Multi-label Neighbourhood Component Analysis

Arjun Pakrashi ARJUN.PAKRASHI@UCD.IE
School of Computer Science
University College Dublin
Dublin, Ireland

Payel Sadhukhan PAYEL0410@GMAIL.COM
TCG CREST,
Kolkata, India

Brian Mac Namee BRIAN.MACNAMEE@UCD.IE
School of Computer Science
University College Dublin
Dublin, Ireland

Editors: Nuno Moniz, Paula Branco, Luís Torgo, Nathalie Japkowicz, Michał Woźniak and Shuo Wang.

Abstract

In multi-label classification, a datapoint can be assigned to more than one class simultaneously. Input space transformation methods can be used to transform the input space so that classification algorithms can perform better. Although existing algorithms used in binary or multi-class classifications can be used with multi-label datasets, this leads to one transformation per label and hence is very costly. Also, considering each label independently ignores consideration of any label associations in the transformation process which is a missed opportunity. In this work, a new input space transformation algorithm, Multi-label Neighbourhood Component Analysis (ML-NCA), is proposed. ML-NCA performs one single linear transformation of the input space in a supervised fashion, that transforms to a space in which k nearest-neighbour based algorithms are expected to perform well. ML-NCA considers all the labels together while finding the single transformation of the input space, therefore omitting the need for per-label transformations. This also implicitly takes advantage of label associations. An extensive set of experiments and detailed analysis demonstrate that the transformation found by ML-NCA is able to significantly improve the performance of multi-label-specific k nearest neighbour algorithms.

Keywords: multi-label, knn, input space transformation

1. Introduction

Multi-class classification models assign at most one class to a datapoint. However, real-world problems exist where a datapoint should be assigned to more than one class at the same time. In other words, a datapoint can be *labelled* with multiple classes. For example, an image can contain *beach*, *mountains* and *sea* at the same time (Boutell et al., 2004; Luo et al., 2019; Chen et al., 2019). Such problems are known as multi-label classification problems (Tsoumakas and Katakis, 2007).

Multi-label classification problems can be formally defined as follows. Let $\mathbf{x}_i$ be a datapoint from a d-dimensional input space $\mathcal{X}$ of real and/or categorical attributes. Also, let the set of all possible labels for a specific multi-label classification problem be $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$, from which a subset of labels, $\mathcal{L}_i \subseteq \mathcal{L}$, is relevant to the datapoint $\mathbf{x}_i$. Here labels in $\mathcal{L}_i$ are called the *relevant* labels, and labels in $(\mathcal{L} - \mathcal{L}_i)$ are called the *irrelevant* labels for $\mathbf{x}_i$. Then a typical multi-label dataset is defined as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | 1 \leq i \leq n\}$, where n is the number of datapoints in the dataset, $\mathbf{x}_i = \{x_{i1,}x_{i2,} \ldots, x_{id}\}$ is a vector indicating the i^{th} datapoint. The vector $\mathbf{y}_i = \{y_{i1,}y_{i2,} \ldots, y_{iq}\}$ is a binary vector indicating the label assignments $\mathcal{L}_i$ for the i^{th} datapoint. Here $y_{ij} = 1$ if $\lambda_j \in \mathcal{L}_i$, that is, the j^{th} label is applicable to the i^{th} datapoint, and $y_{ij} = 0$ if $\lambda_j \notin \mathcal{L}_i$. The objective of multi-label classification is to learn a model $\mathbf{h}$, that predicts the relevance of each label to a new datapoint, $\mathbf{t}$, i.e. $\mathbf{h}(\mathbf{t})$.

In the field of multi-label learning, input space transformation is challenging, as the labels share the input space. Label-specific positive and negative class partitions being a key aspect of multi-label datasets, has been thoroughly explored for transforming features in the input space. In most supervised transformation cases, a point in the feature space is given more than one transformation under the Binary Relevance (BR) (Boutell et al., 2004) framework. In BR, a single datapoint is given $\mathcal{L}$ transformations for $\mathcal{L}$ labels — one for each label. Sometimes, pairwise label discrimination has been the focus of the transformation (Weng et al., 2018; Huang et al., 2017). In such cases, the transformation depends on the class memberships with respect to label pairs. Overall, the research on feature transformation is skewed towards obtaining a number of discriminative sets of features.

Nearest-neighbour based methods for multi-label classification are particularly attractive because they are "lazy" methods—that is, they need minimal or no computation in the training phase. As a result as new datapoints become available, models can be easily updated. Also, they allow easy interpretation of predictions as explanations can be easily generated based on comparison to nearest neighbours. Although there are a number of nearest-neighbour methods proposed in the multi-label literature (Spyromitros et al., 2008; Zhang and Zhou, 2007; Younes et al., 2008; Cheng and Hüllermeier, 2009), they typically perform relatively poor compared to approaches based on other methods (SVM, Decision trees, etc.) (Pakrashi et al., 2016).

This indicates a significant opportunity for improvement in nearest-neighbour based methods for multi-label classification. This work addresses this opportunity using supervised input space transformation. This preserves the benefits of nearest-neighbour methods while improving the classification performance. In this work, a single transformation of the input space is obtained in a supervised way considering all the labels together. This offers massive savings in computation compared to one-transformation-per-label approaches.

The key contributions of this paper are as follows

- A new linear supervised input space transformation method, Multi-label Neighbourhood Component Analysis (ML-NCA), for multi-label datasets. ML-NCA obtains just one input space transformation by considering all the labels together, thereby saving huge computation time compared to approaches that consider each label individually. This single feature transformation is used by all labels of a multi-label dataset.

- A detailed experimental study demonstrating the effectiveness of ML-NCA, indicating that further research in this direction will be fruitful.

The rest of this article is structured as follows. Relevant related work is discussed in Section 2. The proposed ML-NCA method is described in Section 3. The experiment setup is described in Section 4 and the results of this experiment is discussed in Section 5. Finally, Section 6 concludes the paper and discusses ongoing and future work.

2. Related Work

In recent years, feature transformation, either in terms of a reduced feature subset or extraction of a set of new attributes, has been a popular approach in multi-label classification. The goal is to map the features into one or more subspaces where the positive and negative classes of labels are more separable. Additionally, some methods also aim to reduce the cost of computation by removing redundant features using techniques including PCA (Abdi and Williams, 2010), random subspace for decision trees (Ho, 1998), and locality preserving projections (He and Niyogi, 2004). Unsupervised feature selection is one of the early choices in machine learning and it has been widely explored in the domain of single label learning. A variety of schemes including feature-similarity measures (Mitra et al., 2002) and Multi-cluster feature selection (MCFS) (Cai et al., 2010) exist in literature.

In the multi-label context, a few semi-supervised feature transformation techniques are found. One of the most important is by Qian and Davidson (2010)–where reconstruction error is used to determine how well an instance is represented using its k nearest neighbours, and the features are selected corresponding to that information. In (Xu et al., 2018) probabilistic neighbourhood similarities are used to learn correlations in the feature space, and correlation information in the label space is optimized by preserving the feature-label space consistency. This mechanism is used to extract label information in a semi-supervised multi-label learning scenario and to select a set of discriminative features. (Jiang et al., 2018) employs the Hilbert-Schmidt Independence Criterion (HSIC) to determine the feature-label interaction, and obtain a regression coefficient sparse matrix.

Wrapper methods are also used for multi-label feature selection (Yin et al., 2015; Pereira et al., 2018). Label specific feature extraction is an effective tool for learning multi-label datasets but it has a disadvantage of decomposing a multi-label problem into $\mathcal{L}$ subproblems, one for each label. The most notable among them is LIFT (Zhang and Wu, 2015), which uses label-specific positive and negative class clustering to note the key datapoints, followed by a distance based feature extraction.

Several multi-label-specific nearest-neighbour methods are proposed in the literature. *Multi-label k-nearest-neighbours* (MLkNN) (Zhang and Zhou, 2007) was the first lazy approach in multi-label classification which follows a binary relevance approach. Instead of a standard k-NN method, however, MLkNN uses the maximum a-posteriori (MAP) (Kelleher et al., 2020) approach combined with k-NN. BRkNN (Spyromitros et al., 2008), is an efficient implementation of a direct binary relevance based extension of standard k-NN. *Dependent multi-label k-nearest-neighbours* (DMLkNN) (Younes et al., 2008) extends MLkNN to explicitly take label associations into account. *Instance based learning by logistic regression for multi-label learning* (IBLR-ML) (Cheng and Hüllermeier, 2009) considers the labels of the neighbourhood of a query instance as features. Then it derives the relevance of a label based on the influences of all the labels in this feature set using a logistic regression model. The method *IBLR-ML+* (Cheng and Hüllermeier, 2009) is an extension of IBLR-ML that includes the original input space features along with the label predictions as inputs to the regression models. A stacking based approach, Stacked-MLkNN (Pakrashi and Mac Namee, 2017), combining two levels of MLkNN also resulted in improved results.

Deep learning approaches have also been used to improve multi-label learning. Chen and Lin (2012) use *Partially disentangling latent relations for multi-label deep learning* (PDLRMDL) to learn the correlation of labels. Several feature learning classifiers for each label that also preserve some overlapping feature representations through self attention are incorporated in (Lian et al., 2021). *Learning Discriminative Features using Multi-label Dual Space* (LDFM) (Braytee and Liu, 2021) learns a projection using an autoencoder which maps the original feature space to a semantic space and vice-versa. Label correlation and class-imbalance focused binary tree classifiers are employed in (Law and Ghosh, 2021). (Huang et al., 2015) learns label-dedicated features to address multi-label classification. (Mishra and Singh, 2021) integrates feature construction and SMOTE-based transformation to tackle the class-imbalance of multi-label datasets. (Xu et al., 2016) improves over the computation of label-specific features for each label by considering a fuzzy-rough set based scheme. (Mishra and Singh, 2020) uses feature-similarity to find redundant features, thereby reducing complexity.

In this work, a single linear supervised transformation of the input space is proposed. After the transformation, nearest-neighbour methods are expected to perform well. Generally, feature extraction or transformation methods tend to be used in a per-label binary relevance fashion. There have been very few attempts to learn a single supervised feature transformation which can serve all the labels. In ML-NCA, this aspect is addressed.

3. ML-NCA

In this work, a new method—Multi-label Neighbourhood Component Analysis (ML-NCA)—is proposed. ML-NCA performs a supervised linear transformation of the input space for multi-label datasets. The expectation is – after the transformation of the input space, k nearest neighbour algorithms will perform well with respect to Euclidean distance.

This work starts from Neighbourhood Component Analysis (NCA) (Goldberger et al., 2004), and makes it appropriate for a multi-label context. ML-NCA is specifically designed for multi-label datasets. It considers all target labels together in the cost function that it optimises and produces a single transformation. As one transformation matrix is learned, computation time is reduced compared to approaches that learn one transformation matrix per label. Also, as all the labels are considered together while transforming the input space, label associations are considered, albeit indirectly.

The goal of NCA is to learn a Mahalanobis distance measure that maximises nearest-neighbour classification performance at test time. As the test data is not known at the time of learning the distance measure, leave-one-out cross-validation performance on training data is used as an approximation of performance on test data. As the leave-one-out classification error as a cost function is discontinuous with respect to the transformation matrix, A, (Goldberger et al., 2004) uses a differentiable alternative based on stochastic neighbour selection. The idea is to find a transformation, A, in a way that the probability of a datapoint $\mathbf{x}_i$ being correctly classified under a stochastic nearest-neighbour rule is maximised.

In the case of ML-NCA, the focus is on finding a transformation, A, which maximises the probability of a datapoint receiving correct label assignments under the stochastic nearest-neighbour rule. The ML-NCA method is explained below.

The probability of datapoint $\mathbf{x}_j$ being selected as a nearest neighbour for datapoint $\mathbf{x}_i$ with a stochastic nearest-neighbour rule, p_{ij}, is defined as

$$p_{ij} = \frac{\exp(-||Ax_i - Ax_j||^2)}{\sum_{k \neq i} \exp(||Ax_i - Ax_k||^2)} \quad , \quad p_{ii} = 0 \tag{1}$$

For ML-NCA, the probability, p_i, that $\mathbf{x}_i$ will be classified with the highest degree of correct label assignment using the stochastic neighbour selection rule is

$$p_i = \sum_j s_{ij} \times p_{ij} \tag{2}$$

$$s_{ij} = 1 - Jaccard(\mathbf{y}_i, \mathbf{y}_j) \tag{3}$$

here s_{ij} is the degree of similarity between the label assignments $\mathbf{y}_i$ and $\mathbf{y}_j$. *Jaccard* computes the Jaccard distance between the two binary vectors (label assignments). Therefore if two label assignments are identical, the Jaccard distance will be 0 and the similarity will be 1—and vice-versa.

The target of ML-NCA is to maximise the number of datapoints that have maximal correct label assignments, under the stochastic neighbourhood selection rule. So, the objective function to be maximised can be defined as:

$$f(A) = \sum_i \sum_j s_{ij} \times p_{ij} = \sum_i p_i \tag{4}$$

For multi-class datasets, as in NCA (Goldberger et al., 2004), s_{ij} is either 0 or 1 as the class assignments of datapoints are exclusive. In ML-NCA, s_{ij} can be a real value in the range $[0, 1]$. s_{ij} is 0 when there is no similarity between $\mathbf{y}_i$ and $\mathbf{y}_j$, and 1 indicates identical label assignments in $\mathbf{y}_i$ and $\mathbf{y}_j$. This is computed once at the beginning of training for all training datapoints.

The gradient of $f(A)$ can be computed by differentiating Eq. (4) with respect to A as

$$\frac{\delta f}{\delta A} = -2A \sum_i \sum_j s_{ij} \times p_{ij} \left((\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T - \sum_k p_{ik} (\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_k)^T \right) \tag{5}$$

During training, like in NCA, a stochastic gradient can be used, and an exact gradient computation can be avoided to save on computation time, and therefore the sums in Eq. (5) can be approximated. Using this gradient, Eq. (4) is maximised. This can be done using any gradient based optimisation method, but for this work gradient descent was used.

Table 1: Multi-label datasets used in this work

				Total	Single			
Dataset	Instances	Inputs	Labels	Labelsets	Labelsets	Cardinality	Density	MeanIR
yeast	2417	103	14	198	77	4.237	0.303	7.197
birds	322	260	20	89	55	1.503	0.075	13.004
emotions	593	72	6	27	4	1.869	0.311	1.478
cal500	502	68	174	502	502	26.044	0.150	20.578
foodtruck	407	21	12	116	74	2.290	0.191	7.094
medical	978	1449	45	94	33	1.245	0.028	89.501
PlantPseAAC	978	440	12	32	8	1.079	0.090	6.690
enron	1702	1001	53	753	573	3.378	0.064	73.953
	32	10	1.252	0.046	37.315			

4. Experiment Design

A detailed experiment has been designed to test the effectiveness of ML-NCA, The objective of this experiment is to understand if the ML-NCA method can generate a single effective transformation with respect to all labels for multi-label nearest neighbour algorithms. To evaluate the effectiveness of ML-NCA a comparison of the performance of nearest-neighbour-based multi-label classification algorithms with and without the ML-NCA transformation is performed. The nearest-neighbour-based multi-label classification algorithms used are BRkNN and MLkNN.

Eight multi-label datasets[1]are used in our experiments, as listed in Table 1. In Table 1 *Instances*, *Inputs* and *Labels* indicate the total number of datapoints, the number of predictor variables, and the number of potential labels, respectively. *Total Labelsets* indicates the number of unique combinations of relevant labels in the dataset, where each such unique label combination is a *labelset*. *Single Labelsets* indicates the number of datapoints having a unique combination of relevant labels. *Cardinality* indicates the average number of labels assigned per datapoint. *Density* is a normalised, dimensionless indicator of cardinality computed by dividing the value of cardinality by the number of labels. *MeanIR* (Herrera et al., 2016) indicates the average degree of label imbalance in a multi-label dataset—a higher value indicates more imbalance. These label parameters together describe the properties of the datasets which may influence the performance of the algorithms.

For this experiment BRkNN and MLkNN were used as ML-NCA is specifically targeted to be used with nearest-neighbour algorithms. BRkNN was used because it is a direct extension of k nearest neighbours to the multi-label context, and MLkNN is used as it is a well-known multi-label specific nearest-neighbour algorithm. The target of the experiment is to understand how much improvement the ML-NCA transformation leads to when combined with BRkNN and MLkNN. The ML-NCA transformation followed by BRkNN and MLkNN are will be indicated as ML-NCA-BRkNN and ML-NCA-MLkNN respectively.

For each dataset, a 5×2 folds stratified cross-validation experiment was performed. For both BRkNN and MLkNN a total of 12 values of k were explored in the range of $k \in \{2, 4, 6, \dots, 24\}$. The smoothing parameter for MLkNN is constant at 1. All datasets

[1] Datasets sourced from: http://mulan.sourceforge.net/datasets.html

were scaled in the range of $[0, 1]$ before the experiments. For each value of k, the results of cross-validated label-based macro-averaged F-Scores were then compared. When ML-NCA is used, for each fold the training folds was first scaled, then a transformation, A, was found through ML-NCA. Next, the input space of the testing fold was transformed using A.

Hamming loss is a popular evaluation metric to measure multi-label algorithm performance. However, Hamming loss suffers from the same problems as simple classification accuracy when used in multi-class problems with imbalanced classes—the majority class performance overwhelms the minority classes (Kelleher et al., 2020). In this study macro-averaged F-score was chosen over Hamming loss for evaluation as it gives a better indication of how well the algorithms perform over the different labels on average.

The ML-NCA implementation uses gradient descent to learn the input transformation matrix A. The resultant transformed space has the same dimensionality as original dataset for this experiment. The learning rate and momentum was set to 0.1 and 0.9, respectively, based on initial exploration. A total of 100 epochs were performed for all datasets except *medical*, for which 10 epochs were performed due to excessive execution time. For all datasets, the transformation matrix A is initialised randomly from a normal distribution with mean 0 and standard deviation of 0.01.

The implementations of ML-NCA and BRkNN were performed by the authors in R[2], and the implementation of MLkNN was from the utiml library (Rivolli and de Carvalho, 2018), also in R.

5. Results

The objective of this analysis is to understand if the ML-NCA stage is effective in obtaining a single transformation with respect to all the labels that can improve upon the nearest neighbour based algorithms in a multi-label classification context. The label-based macro-average F-Score results (for the best k value found through the cross-validation experiments) for each approach on each dataset is shown in Table 2 (this shows the mean and standard deviation over all folds). The upper triangle of Table 3 shows simple win/lose/tie counts of the approach in the row with respect to the approach in the column. ML-NCA-MLkNN attained better results than MLkNN for all the datasets, and ML-NCA-BRkNN attained better results than BRkNN for seven of the datasets, but did not lead to an improvement for the *yeast* dataset.

From Tables 2 and 3, the effectiveness of ML-NCA is clearly evident. To further understand if ML-NCA makes a significant performance difference, Wilcoxon's signed rank test (Demšar, 2006) was performed between each pair. The resulting p-values are shown in the lower triangle of Table 3. Note that, this test is performed to understand if the performance of a pair of algorithms were significantly different or not when the pair is considered in isolation. Table 3 indicates that ML-NCA-MLkNN was significantly better than MLkNN at significance level $\alpha = 0.01$, and ML-NCA-BRkNN was significantly better than BRkNN at significance level $\alpha = 0.05$. This indicates that the transformation found by ML-NCA was able to help BRkNN and MLkNN significantly improve their performances. The effectiveness of ML-NCA at learning this transfromation is illustrated by the convergence of each of

[2] https://github.com/phoxis/mlnca

Table 2: Label-based macro-averaged F-Scores results. Each values is the mean label-based macro-averaged F-Score (± standard deviation) for the best cross-validated k values.

	MLkNN	ML-NCA-MLkNN	BRkNN	ML-NCA-BRkNN
yeast	0.3674 ± 0.01	**0.3682 ± 0.01**	**0.3757 ± 0.01**	0.3622 ± 0.01
birds	0.2670 ± 0.03	**0.3050 ± 0.02**	0.2901 ± 0.03	**0.3174 ± 0.02**
emotions	0.6179 ± 0.02	**0.6394 ± 0.02**	0.6159 ± 0.02	**0.6438 ± 0.02**
CAL500	0.0604 ± 0.00	**0.0793 ± 0.00**	0.0831 ± 0.01	**0.0943 ± 0.00**
foodtruck	0.1114 ± 0.02	**0.1394 ± 0.01**	0.1373 ± 0.01	**0.1407 ± 0.01**
medical	0.2351 ± 0.01	**0.2865 ± 0.01**	0.1493 ± 0.02	**0.2583 ± 0.01**
PlantPseAAC	0.1077 ± 0.01	**0.2196 ± 0.02**	0.0701 ± 0.02	**0.1576 ± 0.02**
enron	0.0861 ± 0.01	**0.1106 ± 0.01**	0.0939 ± 0.01	**0.1309 ± 0.01**

Table 3: Significance test. Upper diagonal: win/lose/tie. Lower diagonal: Wilcoxon's Signed Rank Test p-values. Significance levels: ***: $\alpha = 0.01$, **: $\alpha = 0.05$, *: $\alpha = 0.1$.

	MLkNN	ML-NCA-MLkNN	BRkNN	ML-NCA-BRkNN
MLkNN		0/8/0	3/5/0	1/7/0
ML-NCA-MLkNN	0.0059 ***		6/2/0	3/5/0
BRkNN	0.3897	0.0344 **		1/7/0
ML-NCA-BRkNN	0.0086 ***	0.5000	0.0178 **	

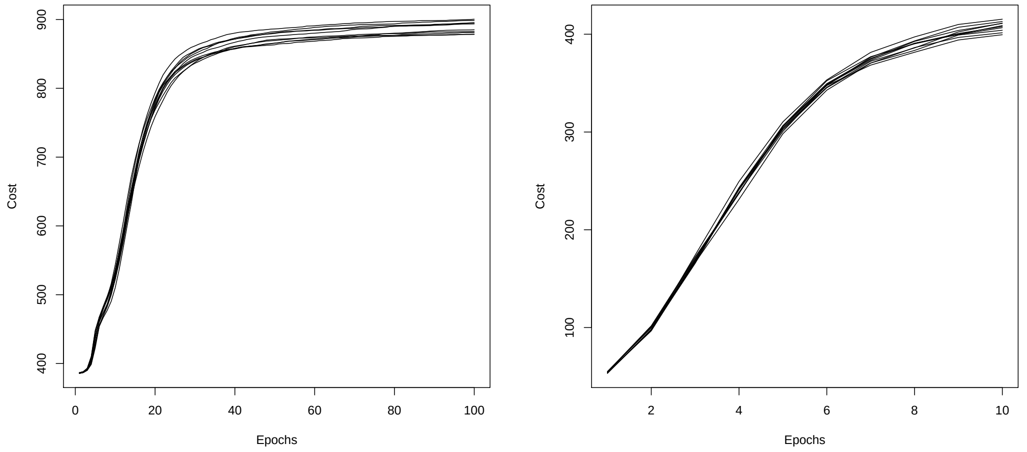
the folds from the cross-validation experiment for the *yeast* and *medical* datasets that are shown in Figure 1. Note that, the objective function in Eq. (4) is being optimised.

It is also interesting to note that ML-NCA-BRkNN was able to perform better compared to ML-NCA-MLkNN in five of the datasets, just as BRkNN was able to perform better on the same number of datasets compared to MLkNN. Therefore, it looks like ML-NCA method was able to transform the input space effectively, and the final performance of the classifier will depend on the nearest-neighbour classifier used, as expected.

To investigate the importance of the value of k used, the mean (over folds) label-based macro-averaged F-Score and the related standard errors for each value of k are plotted in Figure 3 and 2 for BRkNN and MLkNN, respectively. Figure 3 compares the performance of ML-NCA-BRkNN vs BRkNN, and Figure 2 compares the performance of ML-NCA-MLkNN vs MLkNN, for different values of k.

In Figure 3, ML-NCA-BRkNN is shown to consistently outperform BRkNN, except for the *yeast* and *foodtruck* datasets. Generally a similar performance trend is seen in the case of ML-NCA-MLkNN in Figure 2, which was consistently able to outperform MLkNN except for the *yeast* and *birds* datasets where some amount of overlap of the plots can be observed. Overall, Figures 3 and 2 show the effectiveness of the ML-NCA transformation.

The best k values found through cross-validation for each dataset are shown in Table 4. These values of k were used to generate the scores in Table 2. The k values are almost



(a) *yeast* (b) *medical*

Figure 1: ML-NCA convergence for each of the 5 times 2 folds from the cross-validation experiment. *yeast* and *medical* datasets were run using gradient descent with 100 and 10 epochs respectively (see Section 4).

Table 4: Best k-values for nearest-neighbour methods found through cross-validation.

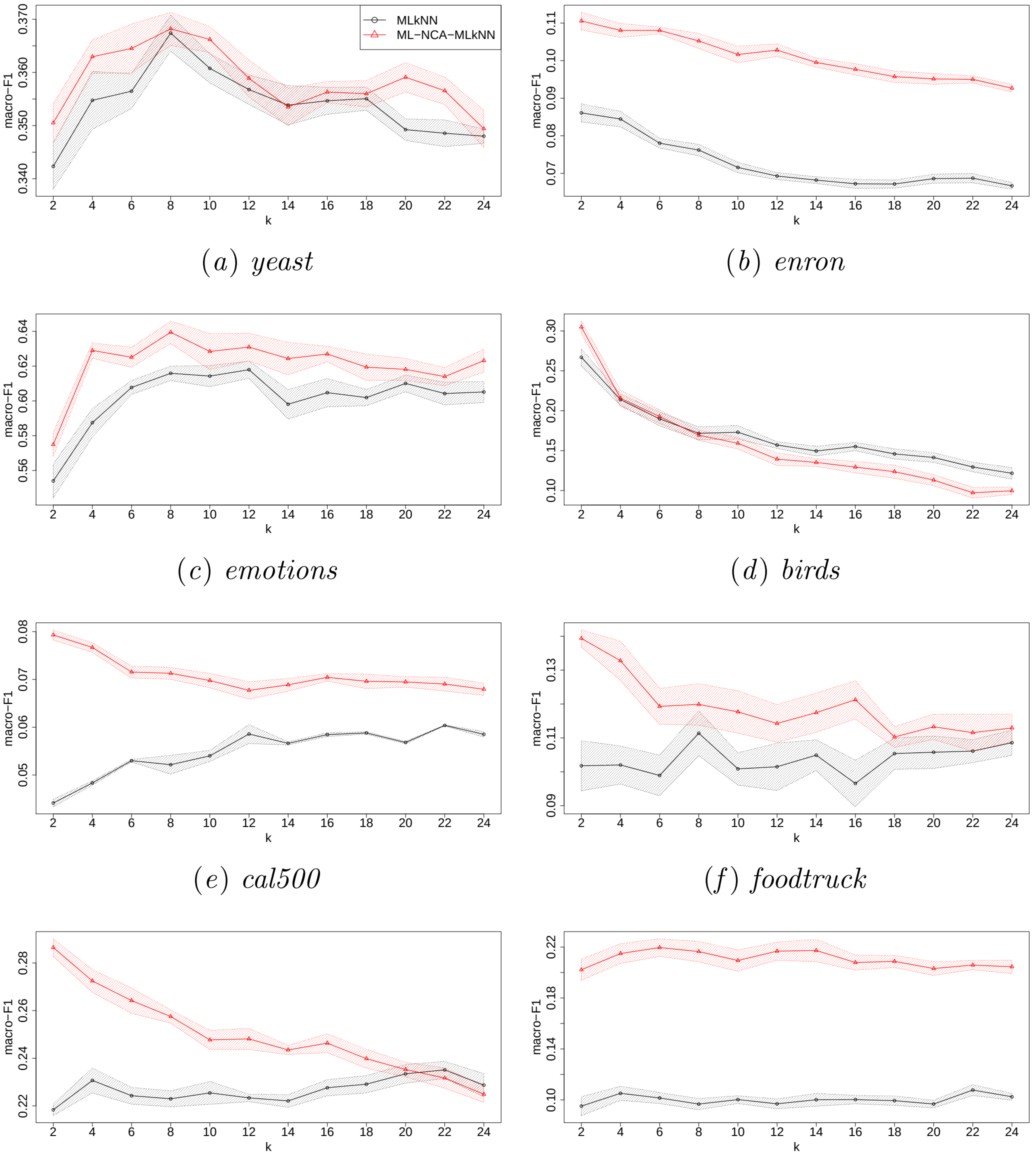
	MLkNN	ML-NCA-MLkNN	BRkNN	ML-NCA-BRkNN
yeast	8	8	6	8
birds	2	2	2	2
emotions	12	8	14	22
CAL500	22	2	4	4
foodtruck	8	2	2	2
medical	22	2	2	2
PlantPseAAC	22	6	2	2
enron	2	2	2	2

the same for BRkNN and ML-NCA-BRkNN. But almost always have a much higher value for ML-NCA-MLkNN than for MLkNN. Looking at Figure 2, for the datasets for which the k values are higher in MLkNN, it starts with a low performance and slightly improves as k is increased; while ML-NCA-MLkNN starts with a much higher value and then the performance starts to drop as k is increased.

The experiments and the detailed analysis indicate that ML-NCA did improve the performance of BRkNN and MLkNN consistently, and therefore is an effective transformation method to be used with nearest neighbour based multi-label methods.

6. Conclusions and Future Work

This paper proposes Multi-Label Neighbourhood Component Analysis (ML-NCA), a new supervised linear input space transformation method for multi-label datasets. ML-NCA



(a) *yeast*

(b) *enron*

(c) *emotions*

(d) *birds*

(e) *cal500*

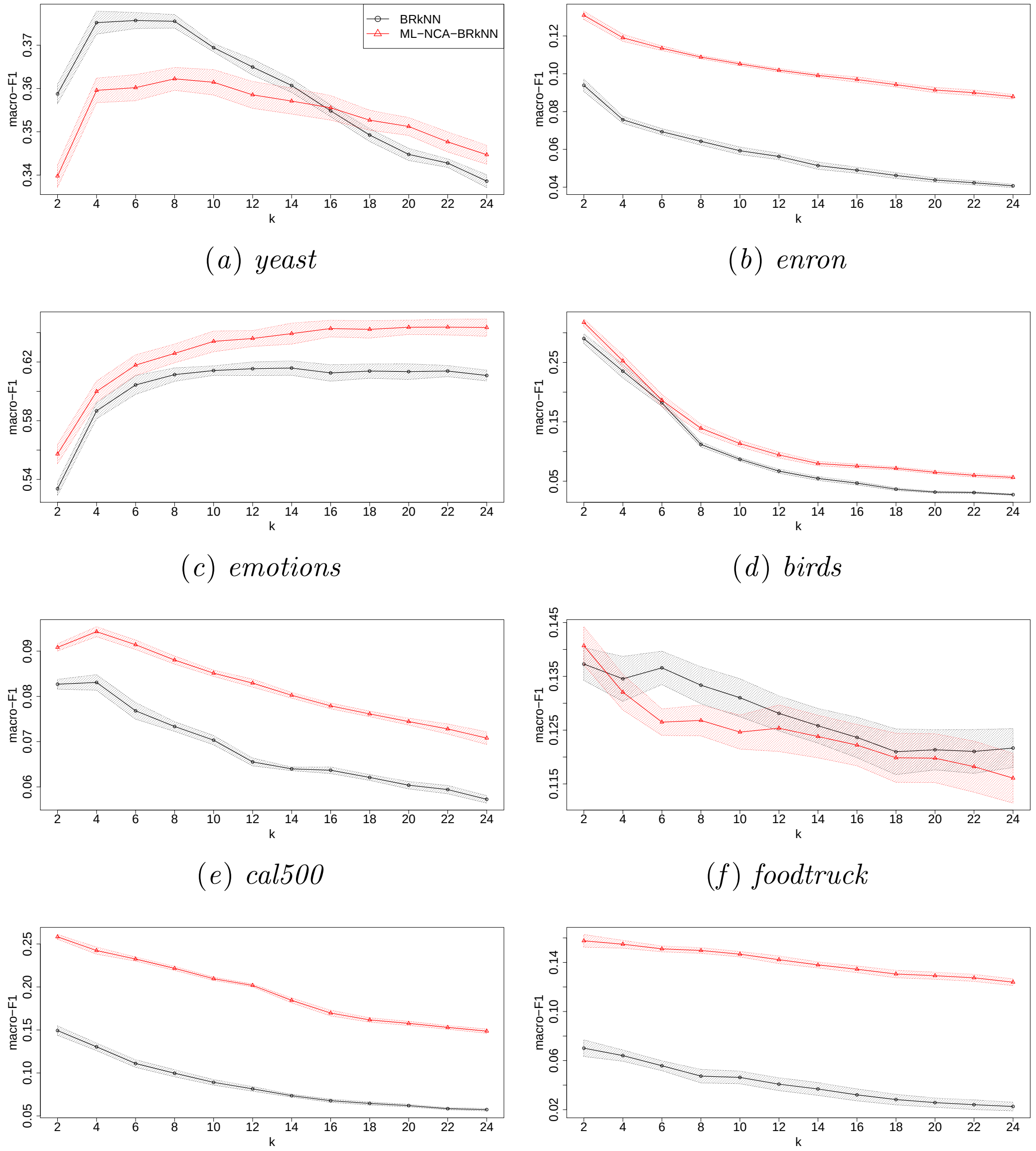
(f) *foodtruck*

(g) *medical*

(h) *PlantPseAAC*

Figure 2: Change of label-based macro-averaged F-score with standard errors with respect to the number of nearest-neighbours k for MLkNN vs ML-NCA-MLkNN

builds upon NCA (Goldberger et al., 2004) to work with multi-label datasets. The novelty of this work is that ML-NCA finds a single supervised linear transformation of the features of a multi-label dataset (over all labels), in which k nearest neighbour methods are expected to perform well. The linear transformation is computed such that in the transformed space the number of datapoints having a high degree of correct assignments is maximimsed under the stochastic nearest-neighbour rule. Therefore, after the transformation, a nearest-neighbour



(*a*) *yeast* (*b*) *enron*

(*c*) *emotions* (*d*) *birds*

(*e*) *cal500* (*f*) *foodtruck*

(*g*) *medical* (*h*) *PlantPseAAC*

Figure 3: Change of label-based macro-averaged F-score with standard errors with respect to the number of nearest-neighbours k for BRkNN vs ML-NCA-BRkNN

based algorithm is expected to perform better than in the original space. Extensive experiments show that the transformation found using ML-NCA is consistently effective, and leads to significantly better results compared to when no such transformation is performed. The results are encouraging, which motivates further investigation. A detailed experimental study in terms of datasets and competing methods would be an initial plan for exploration, including methods with unsupervised input space transformations or non-linear supervised input space transformations. The dimensionality reduction property of NCA in the input space transformation, is also currently being investigated in the multi-label context. It would also be interesting to also address the label frequency imbalance common in multi-label datasets in this transformation approach.

Acknowledgments

This work has emanated from research supported in part by TCG CREST, Kolkata, India and a grant from Science Foundation Ireland under Grant number [16/RC/3835]. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004. ISSN 0031-3203.

Ali Braytee and Wei Liu. Learning discriminative features using multi-label dual space. In *Advances in Knowledge Discovery and Data Mining*, pages 233–245, Cham, 2021. Springer International Publishing. ISBN 978-3-030-75768-7.

Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342, 2010.

Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. *Advances in neural information processing systems*, 25:1529–1537, 2012.

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5172–5181, 2019.

Weiwei Cheng and Eyke Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, page 513–520, Cambridge, MA, USA, 2004. MIT Press.

Xiaofei He and Partha Niyogi. Locality preserving projections. *Advances in neural information processing systems*, 16(16):153–160, 2004.

Francisco Herrera, Francisco Charte, Antonio J. Rivera, and Mara J. del Jesus. *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Springer Publishing Company, Incorporated, 1st edition, 2016. ISBN 3319411101.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu. Learning label specific features for multi-label classification. In *2015 IEEE International Conference on Data Mining*, pages 181–190. IEEE, 2015.

Jun Huang, Guorong Li, Shuhui Wang, Zhe Xue, and Qingming Huang. Multi-label classification by exploiting local positive and negative pairwise label correlation. *Neurocomputing*, 257:164–174, 2017.

Lin Jiang, Jun Wang, and Guoxian Yu. Semi-supervised multi-label feature selection based on sparsity regularization and dependence maximization. In *2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 325–332. IEEE, 2018.

John D Kelleher, Brian Mac Namee, and Aoife D'arcy. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.

Anwesha Law and Ashish Ghosh. Multi-label classification using binary tree of classifiers. *IEEE Transactions on Emerging Topics in Computational Intelligence*, pages 1–13, 2021.

Si-ming Lian, Jian-wei Liu, Run-kun Lu, and Xiong-lin Luo. Partially disentangled latent relations for multi-label deep learning. *Neural Computing and Applications*, 33(11):6039–6064, 2021.

Yan Luo, Ming Jiang, and Qi Zhao. Visual attention in multi-label image classification. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 820–827, 2019.

Nitin Kumar Mishra and Pramod Kumar Singh. FS-MLC: Feature selection for multi-label classification using clustering in feature space. *Information Processing & Management*, 57(4):102240, 2020. ISSN 0306-4573.

Nitin Kumar Mishra and Pramod Kumar Singh. Feature construction and smote-based imbalance handling for multi-label learning. *Information Sciences*, 563:342–357, 2021. ISSN 0020-0255.

Pabitra Mitra, CA Murthy, and Sankar K. Pal. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):301–312, 2002.

Arjun Pakrashi and Brian Mac Namee. Stacked-MLkNN: a stacking based improvement to multi-label k-nearest neighbours. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 51–63. PMLR, 2017.

Arjun Pakrashi, Derek Greene, and Brian Mac Namee. Benchmarking multi-label classification algorithms. In *24th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'16), Dublin, Ireland, 20-21 September 2016*. CEUR Workshop Proceedings, 2016.

Rafael B Pereira, Alexandre Plastino, Bianca Zadrozny, and Luiz HC Merschmann. Categorizing feature selection methods for multi-label classification. *Artificial Intelligence Review*, 49(1):57–78, 2018.

Buyue Qian and Ian Davidson. Semi-supervised dimension reduction for multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

Adriano Rivolli and Andre C. P. L. F. de Carvalho. The utiml Package: Multi-label Classification in R. *The R Journal*, 10(2):24–37, 2018.

Eleftherios Spyromitros, Grigorios Tsoumakas, and Ioannis Vlahavas. An empirical study of lazy multilabel classification algorithms. In *Hellenic conference on artificial intelligence*, pages 401–406. Springer, 2008.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

Wei Weng, Yaojin Lin, Shunxiang Wu, Yuwen Li, and Yun Kang. Multi-label learning based on label-specific features and local pairwise label correlation. *Neurocomputing*, 273:385–394, 2018.

Suping Xu, Xibei Yang, Hualong Yu, Dong-Jun Yu, Jingyu Yang, and Eric C.C. Tsang. Multi-label learning with label-specific feature reduction. *Knowledge-Based Systems*, 104: 52–61, 2016. ISSN 0950-7051.

Yuanyuan Xu, Jun Wang, Shuai An, Jinmao Wei, and Jianhua Ruan. Semi-supervised multi-label feature selection by preserving feature-label space consistency. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 783–792, 2018.

Jing Yin, Tengfei Tao, and Jianhua Xu. A multi-label feature selection algorithm based on multi-objective optimization. In *2015 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2015.

Zoulficar Younes, Fahed Abdallah, and Thierry Denœux. Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies. In *2008 16th European Signal Processing Conference*, pages 1–5. IEEE, 2008.

Min-Ling Zhang and Lei Wu. Lift: Multi-label learning with label-specific features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):107–120, 2015.

Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.