Sādhanā (2019) 44:53
https://doi.org/10.1007/s12046-018-0996-6

ALDL: a novel method for label distribution learning

MAINAK BISWAS*, VENKATANARESHBABU KUPPILI and DAMODAR REDDY EDLA

Department of Computer Science and Engineering, National Institute of Technology Goa, Ponda 403401, India
e-mail: mainakmani@gmail.com; vnareshiitd@gmail.com; dr.reddy@nitgoa.ac.in

MS received 30 March 2017; accepted 7 July 2018; published online 8 February 2019

Abstract. Data complexity has increased manifold in the age of data-driven societies. The data has become huge and inherently complex. The single-label classification algorithms that were discrete in their operation are losing prominence since the nature of data is not monolithic anymore. There are now cases in machine learning where data may belong to more than one class or multiple classes. This nature of data has created the need for new algorithms or methods that are multi-label in nature. Label distribution learning (LDL) is a new way to view multi-labelled algorithms. It tries to quantify the degree to which a label defines an instance. Therefore, for every instance there is a label distribution. In this paper, we introduce a new learning method, namely, angular label distribution learning (ALDL). It is based on the angular distribution function, which is derived from the computation of the length of the arc connecting two points in a circle. Comparative performance evaluation in terms of mean-square error (MSE) of the proposed ALDL has been made with algorithm adaptation of k-NN (AA-kNN), multilayer perceptron, Levenberg–Marquardt neural network and layer-recurrent neural network LDL datasets. MSE is observed to decrease for the proposed ALDL. ALDL is also highly statistically significant for the real world datasets when compared with the standard algorithms for LDL.

Keywords. Machine learning; multi-label classification; multi-label learning; label distribution learning.

1. Introduction

With the advent of multi-label learning (MLL), where the instances can be associated with more than one class, traditional mining algorithms ought to be seen in new light [1, 2]. In biology, protein function for multi-labelling has been developed [3]. A multi-label classification for music categorization [4] and multi-categorical algorithm for semantic scene classification have also been developed [5]. In semantic scene classification, a photograph can belong to more than one conceptual class, i.e., sunset and beach at the same time. Random k-label (RAkEL) sets used random subset of labels called label power set (LP) to deal with application domains having a large number of labels [6, 7]. Maximum entropy for MLL has also been discussed [8]. A method that combines the concepts of random subspace [9], bagging [10] and RAkEL set [7] together to form ensemble learning methods [11] as an approach to classify multi-label data has been developed. At the end of iteration, optimized parameters are selected and the ensemble MLL classifiers are constructed. Pruned sets have been used to perform multi-labelling [12]. Classifier chains (CC) on binary relevance (BR) methods for MLL have been developed [13]. In this method, the input domain is defined as $X^d \in R$ for all possible attribute values. An instance is defined by a vector of d-attribute values $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_d}\}$. The set $\mathbf{L} = \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{l}\}$ is the output domain of all possible labels. Each instance of $\mathbf{x}$ is associated with a subset of these labels. It is represented as L-vector $\mathbf{y}$, where $\mathbf{y} = \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_j}, \ldots, \mathbf{y_L}\}$. If $y_j = 1$ then x_i instance belongs to label y_j. All other values in $\mathbf{y}$ except y_j are zero otherwise. A relationship by appending, $\mathbf{y_i}$ label set, with the instance set $\mathbf{x_i}$, for all instances is formed. Finally, a group of classifiers (called CC) is used to predict the $\mathbf{y_{i+1}} \leftarrow \mathbf{x_{1,2,\ldots,d}} \cdot \mathbf{y_{1,2,\ldots,L}}$ for an unknown instance $\mathbf{x_{i+1}}$.

Label distribution learning (LDL) is a way to label an instance x by assigning a degree d_x^y to each possible label y, representing the degree to which y describes x [14, 15]. For example, if x represents a protein and y represents a cancer, then d_x^y should be the expression level of the protein x in the cancer y. Further, suppose that the label set is complete, i.e., using all the labels in the set can always fully describe the instance. Then, $\sum_y d_x^y = 1$ where d_x^y is called the description degree of y to x. For a particular instance, the degrees of belongingness of all the labels can be seen as confidence level or probability distribution [16]. Data mining algorithms can be adapted to LDL or new mining algorithms can be developed with respect to LDL. There are three ways in which LDL algorithms can be developed, which are as follows:

*For correspondence

(i) **Problem transformation:** Single-label training examples are converted into weighted single-label examples, i.e., each of the n single-label instance is transformed to c single-label examples such that it forms a $c \times n$ matrix, where each weight value represents the degree d_x^y. A machine learning (ML) algorithm must be able to predict confidence/probability or degree of belongingness $d_x^{y_j}$ for each label y_j.

(ii) **Algorithm adaptation:** Some of the prevalent algorithms can be naturally extended to deal with label distributions. The k-NN algorithm [17] is one such algorithm that can be adapted for LDL. Given a new instance x, its nearest neighbours are first found in the training set. Then, the mean of the label distributions of all the nearest neighbours is calculated as the label distribution of x. This adapted algorithm is denoted by AA-kNN, where AA stands for algorithm adaptation.

(iii) **Specialized algorithm:** Some algorithms meet the criteria of LDL explicitly. Two algorithms were proposed, i.e., CPNN and IIS-LLD [18, 19], for facial age estimation.

In view of algorithm adaptation, AA-kNN has been developed [14]. The k-nearest neighbours have been used extensively in ML operation such as classification for both single label [20] and multi-label [21]. In LDL, the mean of label distributions of k nearest neighbours is taken for an unseen instance. In this paper we introduce a new specialized algorithm, i.e., angular label distribution learning (ALDL), which is based on angular distribution function (ADF). ADF takes inspiration from the geometric computation of length of arc connecting two points in a circle. It takes into consideration the alignment that each unseen instance makes with instances that have the highest degree for each individual label. This orientation forms the basis of ADF. Each orientation can be converted into degree of belongingness for LDL. It is tested on real world datasets. It was observed that prediction accuracy for the proposed ALDL method is higher than those of all previous algorithms. Section 2 explains the derivation of ADF. Section 3 provides an algorithm for implementing ALDL. Section 4 presents comparative performance evaluation of ALDL with other well-known methods for LDL on real datasets and finally concluding remarks are given in section 5.

2. Proposed method

The new specialized algorithm ALDL is discussed here. It is based on ADF. ADF is derived from geometric computation of length of arc connecting two points in a circle. Each alignment is recorded and converted into degree of belongingness. It is seen that existing ML algorithms are discrete in approach; that is, they allocate instances to single label or multiple labels in a mutually exclusive way. These algorithms give no information on how much or to what extent these allocations must be done. In the LDL method stated by Geng, i.e., AA-kNN, for computation of LDL, is a simple averaging of label distributions of neighbourhood instances, which gives far lesser accuracy. The process of the model is given in section 2.1.

2.1 *Derivation for ADF*

The proposed methodology is based on geometry. Suppose there are n training instances with L labels. Given a test instance t, its K nearest neighbours from training instances based on Euclidean distance are found out. In addition to the K nearest neighbours, L instances in training dataset with highest individual degree for each label are also found. Based on given data, distance matrix is formed between given test case, K-nearest neighbours and L instances, which forms the distance matrix (say $\mathbf{D_{ij}}$) as shown in table 1. Table 1 presents adjacency matrix for distances between test instance t, k-nearest neighbours and L maximum label instances, where t denotes test instance, $\{l_1, l_2, \ldots, l_L\}$ denotes L instances with highest individual degrees for each label, $\{k_1, k_2, \ldots, k_K\}$ denotes K-nearest neighbours of test instance among n training instances and $\mathbf{D_{ij}}$ (distance matrix) denotes adjacency matrix for distance between i^{th} and j^{th} instance.

In addition to distance matrix there would be a label distribution matrix signifying degrees of each labels for each given instance, d_i^j , where $i \in k$-nearest neighbour and j represents labels as shown in table 2 (say $\mathbf{P_{ij}}$), where $\mathbf{P_{ij}}$ (label matrix) denotes degree of confidence/confidence of i^{th} instance for j^{th} label.

Example 1 Let the test instance be described as T, highest label instance as L and the nearest neighbour in consideration as K. The calculation of an arc joining two points in a circle is considered here. Since the distances between given test instance T and given label L (say TL) and test instance

Table 1. Distance matrix($\mathbf{D}_{ij}$) representing distances between test instance t, k-nearest neighbours and L maximum label instances.

$\mathbf{D_{ij}}$	t	l_1	l_2	$\cdots$	l_L	k_1	k_2	$\cdots$	k_K
t	0	D_{l_1t}	D_{l_2t}	$\cdots$	D_{l_Lt}	D_{tk_1}	D_{tk_2}	$\cdots$	D_{tk_K}
l_1	D_{l_1t}	0	$D_{l_1l_2}$	$\cdots$	$D_{l_1l_2}$	$D_{l_1k_1}$	$D_{1_1k_2}$	$\cdots$	$D_{l_1k_K}$
l_2	D_{l_2t}	$D_{l_1l_2}$	0	$\cdots$	$D_{l_2l_2}$	$D_{l_2k_1}$	$D_{l_2k_2}$	$\cdots$	$D_{1_2k_K}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
l_L	D_{l_Lt}	$D_{l_Ll_2}$	$D_{l_Ll_2}$	$\cdots$	0	$D_{l_Lk_1}$	$D_{l_Lk_2}$	$\cdots$	$D_{l_Lk_K}$
k_1	D_{tk_1}	$D_{l_1k_1}$	$D_{l_2k_1}$	$\cdots$	$D_{l_Lk_1}$	0	$D_{k_1k_2}$	$\cdots$	$D_{k_1k_K}$
k_2	D_{tk_2}	$D_{1_1k_2}$	$D_{l_2k_2}$	$\cdots$	$D_{l_Lk_2}$	$D_{k_1k_2}$	0	$\cdots$	$D_{k_2k_K}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
k_K	D_{tk_K}	$D_{l_1k_K}$	$D_{1_2k_K}$	$\cdots$	$D_{l_Lk_K}$	$D_{k_1k_K}$	$D_{k_2k_K}$	$\cdots$	0

Table 2. Label distribution matrix ($\mathbf{P_{ij}}$) representing degree of belongingness of all k-nearest neighbours.

$\mathbf{P_{ij}}$	$d_k^{l=1}$	$d_k^{l=2}$	$\cdots$	$d_k^{l=L}$
k_1	d_1^1	d_1^2	$\cdots$	d_1^L
k_2	d_2^1	d_2^2	$\cdots$	d_2^L
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
k_K	d_K^1	d_K^2	$\cdots$	d_K^L

Table 3. Angle matrix($\mathbf{A_{ij}}$) representing angle between test instance t, k-nearest neighbours and L maximum label instances.

t	l_1	l_2	$\cdots$	l_L
k_1	θ_{11}	θ_{12}	$\cdots$	θ_{1L}
k_2	θ_{21}	θ_{22}	$\cdots$	θ_{2L}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
k_K	θ_{K1}	θ_{K2}	$\cdots$	θ_{KL}

T and given k-neighbour K (say TK), may not be equal, we will consider radius r as average of the two distances given by Eq. (2). The distance between k-nearest neighbour K and highest degree label instance L (say KL) is known from table 1.

$$r = \frac{TK + TL}{2}. \tag{1}$$

Now, the angle in radians between TK and TL are found out using Eq. (1) as shown in figure 1. All corresponding angles are given in matrix (say $\mathbf{A_{ij}}$) shown in table 3, where $\mathbf{A_{ij}}$ (angle matrix) denotes angle between i^{th} and j^{th} instances. Based on the given distance matrix $\mathbf{D_{ij}}$, the angle is found out using distance between highest label point and k-nearest point as an arc. The formula for the normal arc length in a circle is given by Eq. (2):

$$\theta = \frac{arclength}{radius} = \frac{l}{r} = \frac{2KL}{TK + TL}. \tag{2}$$

Total dimension of circle is 2π radians; here θ varies from $-\pi$ to $+\pi$, completing a circle. Therefore, maximum angular difference between TL and TK will occur at $\theta = +\pi$ or $-\pi$ and minimum at $\theta = 0$. This shows that orientation of test case K with given maximum label instance is in alignment when $\theta = 0$ and is out of alignment at $\theta = +\pi$ or $-\pi$. Now using cosine approximation to normal distribution [22], which is given by Eq. (3):

$$f(\theta) = \frac{1}{2\pi}(1 + \cos\theta) \tag{3}$$

gives us quantification of all points between $-\pi$ and $+\pi$, with the maximum value at $\theta = 0$. The corresponding plot of $f(\theta)$ is given in figure 2.

To raise values to above unity, Eq. (3) is multiplied with 0.1 and its exponential is taken, which is given by Eq. (4):

$$F(\theta) = \exp\left(\left(0.1\left(\frac{1}{2\pi}(1 + \cos\theta)\right)\right)\right). \tag{4}$$

Its corresponding plot is given by figure 3.

Cosine approximation of each angle between instances and instances with highest degree $\mathbf{A_{ij}}$ (shown in table 3) is taken and multiplied with square root of corresponding degree P_{ij} (shown in table 2) of the instance of the same label, to get another matrix $\mathbf{T_{ij}}$, which is given by Eq. (5), which forms the basic ADF:

$$\mathbf{T_{ij}} = F(\mathbf{A_{ij}})\mathbf{P_{ij}}^{\frac{1}{2}}. \tag{5}$$

Next we add all the values for each particular label of matrix $\mathbf{T_{ij}}$, to get matrix $\mathbf{X_j}$ of dimension $1 \times L$, for all labels as shown in Eq. (6), where K denotes the total number of k-nearest neighbours:

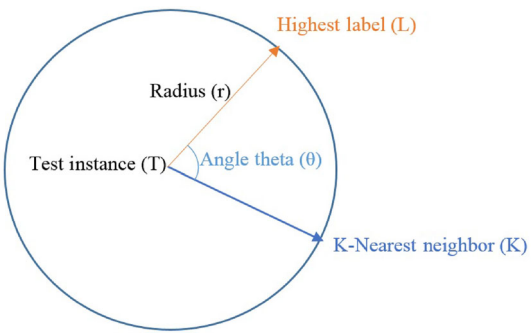

Figure 1. Angle (θ) between test point T, highest label point L and nearest neighbour K.

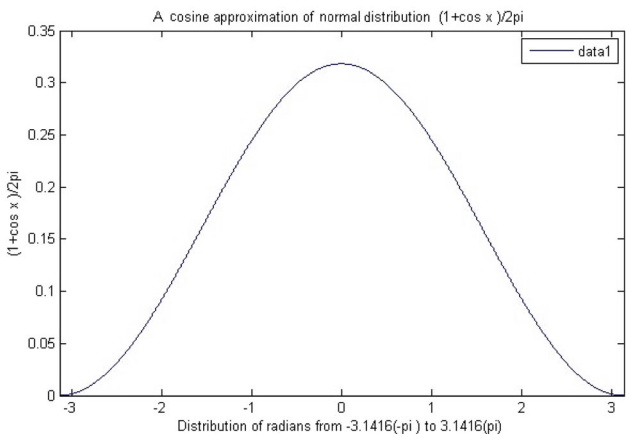

Figure 2. Cosine approximation of normal distribution from $+\pi$ to $-\pi$ (highest value at 0 and lowest at $+\pi$ and π).

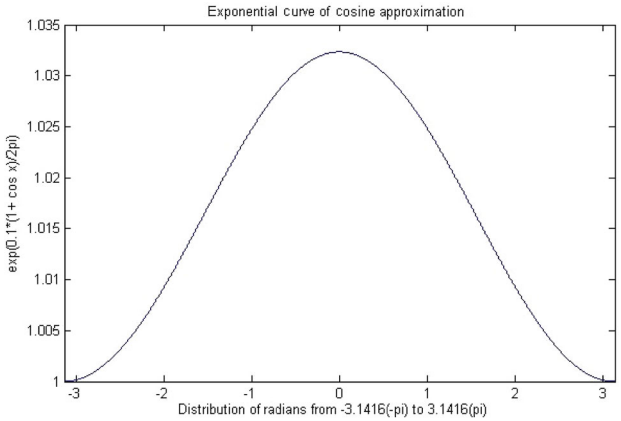

Figure 3. Exponential curve of cosine approximation of normal distribution from $+\pi$ to $-\pi$ (highest value at 0 and lowest at $+\pi$ and π).

$$\mathbf{X_{j=1..L}} = \sum_{i=1}^{K} \mathbf{T_{ij}}. \tag{6}$$

After calculating estimate of all $\mathbf{T_{ij}}$ in $\mathbf{X_j}$ and adding all of them label by label, we obtain $Sum(X_j)$, which is given by

$$Sum(\mathbf{X}_j) = \sum_{l=1}^{L} \mathbf{X}_j. \tag{7}$$

The degree of belongingness is obtained by dividing each estimation by the total sum. The degree of belongingness of test instance for label j is given by Eq. (8):

$$d_{test}^{l} = \frac{\mathbf{X}_j}{Sum(\mathbf{X}_j)} = \frac{\mathbf{X}_j}{\sum_{l=1}^{L} \mathbf{X}_j}. \tag{8}$$

Equation (8) satisfies condition Eq. (3); for example, if there are only two labels A and B, the summation would result in unity, i.e., $d_{test}^{A} + d_{test}^{B} = \frac{\mathbf{X}_A}{\mathbf{X}_A+\mathbf{X}_B} + \frac{\mathbf{X}_B}{\mathbf{X}_A+\mathbf{X}_B} = 1$.

2.2 *Performance criteria*

The squared difference, mean-square error (MSE), is obtained from squaring the difference between label distribution of predicted and actual test instance, which is given by Eq. (9):

$$MSE = \frac{1}{L}\sqrt{\sum_{l=1}^{L}(d_{testprediction}^{label} - d_{testactual}^{label})}. \tag{9}$$

3. Algorithm for ALDL

In this section the pseudocode to implement ALDL will be presented.

The pseudocode for ALDL is given in figure 4. ADF has been applied in AA-kNN [14]. Initially, training dataset is fed into the algorithm. A random test case is selected among training datasets. An array **DIFFX** is formed from the values of label distributions of test case. Based on label distributions, L instances with highest individual degrees for each label is found out. A distance matrix $\mathbf{DIS_{ij}}$ is formed from the euclidean distances among K nearest neighbours, L instances and the test instance. A label distribution matrix, $\mathbf{P_{ij}}$ is formed for K nearest neighbours and angle matrix $\mathbf{A_{ij}}$ is formed between K nearest neighbours and L instances with highest L individual degrees. Cosine approximation of each angle between instances and instances with highest degree $\mathbf{A_{ij}}$ is taken and multiplied with square root of corresponding degree $\mathbf{P_{ij}}$ in another matrix $\mathbf{T_{ij}}$. The values in $\mathbf{T_{ij}}$ are added to obtain SumArray for each label l. Values in SumArray are added to get *Sum*. All the *SumArray* values when divided by *Sum* give degree of belongingness/label distributions for test instance and they are stored in $\mathbf{DIFF}^{p}X$. Finally, MSE is computed between $DIFF_x$ and $DIFF^{p}X$ to give the difference between predicted and actual output.

4. Results

The real world datasets (*http://cse.seu.edu.cn /people/ xgeng/LDL/ index.htm#data*) are given in table 4. Each dataset has passed through 10-fold cross-validation.

4.1 *Yeast dataset and its varients*

Datasets mentioned from row #1 to row #10. (from Yeast-alpha to Yeast-spoem) in table 4 [23] are collected from biological experiments on the Saccharomyces Cerevisiae yeast. In each dataset, the labels correspond to the discrete times during one biological experiment. The gene expression level at each time point is recorded. The data is normalized and provides a natural measure of the description degree/degree of belongingness of the corresponding label. The description degrees (normalized gene expression levels) of all the labels (discrete times) constitute the label distribution for a particular yeast gene.

4.2 *SJAFFE dataset*

The dataset in row #11 shown in table 4 is an extension of facial expression image database, i.e., [24]. The JAFFE dataset contains 213 greyscale expression images from 10 Japanese female models. A 243-dimension feature vector is extracted from each image by the method of Local Binary Patterns (LBP) [25]. Each of the images is given a score judged by 60 persons on six basic emotions, (i.e., happiness, sadness, surprise, fear, anger and disgust) with a

Input: the training set tr $= \{(X_i, D_i)\}_{i=1}^{n}$

Output: diff ←squared difference between label distribution of predicted and actual test instance

1. Randomly find test case 'X' within training set tr
2. $DIFF_X$← Store all $d_{test\,actual}^{label}$ of X in $DIFF_X$
3. Find K-Nearest Neighbor of 'X' , L instances with highest individual degrees based on Euclidean Distance
4. Initialize
   a. Sum ← zeros(1,1)
   b. SumArray=zeros(1,L)
   c. DIFF'$_X$← zeros(1,L)
   d. diff←zeros(1,1)
   e. DIS_{ij}← Distance matrix between L instances, K nearest neighbors and X test instance
   f. P_{ij}← label distribution of K nearest neighbors and K- test instances
   g. A_{ij}← Angle between K nearest neighbors and K- test instances
   h. T_{ij}← zeros(K,L)
5. for each k=1:K
   a. for each label l=1:L
      i. $\theta = A(k, l)$
      ii. $F(\theta) = \exp\left(0.1 * \left(\frac{1}{2\pi}(1 + \cos\theta)\right)\right)$
      iii. $A(k, l) = F(\theta)$
      iv. $T(k, l) = \{P(k, l)\}^{\frac{1}{2}} * A(k, l)$
   b. end for

   end for
6. for each label l=1:L
   a. for each k=1:k-NN
      i. SumArray(1,l)← SumArray(1,l)+ T(k, l)
   b. end for
   c. Sum← Sum+SumArray(1,l)

   end for
7. for each label l=1:L
   a. $DIFF_X$ (1,l)← SumArray(1,l)/ Sum
   b. diff ← diff+(DIFF'$_X$ (1,l)- $DIFF_X$ (1,l))* (DIFF'$_X$ (1,l)- $DIFF_X$ (1,l))^T

   end for
8. diff ← sqrt(diff)
9. END Program

Figure 4. Algorithm showing working of angular label distribution learning.

5-point scale. The average score of each emotion is used to represent the emotion intensity. Instead of considering only the emotion with the highest score as most work on JAFFE does, the dataset SJAFFE (scored JAFFE) keeps all the scores and normalizes them into a label distribution over all the six emotion labels.

4.3 *Natural scene dataset*

The row # 12 dataset in table 4 is the Natural scene dataset [15], which is developed from the inconsistent multi-label rankings from 2,000 Natural scene images.

4.4 *Experimental evaluation*

The proposed methodology has been compared to prevalent methods such AA-kNN[14], multilayer perceptron (MP) [26], Levenberg–Marquardt (LM) [27] and layer-recurrent neural network (LRN) [28]. AA-kNN takes a general mean of degrees of all *K*-nearest neighbours. MP is the simplest type of artificial neural network (ANN) devised. In this ANN, the information flows in only forward direction, from the input to the output nodes, through the hidden nodes if there are any. The LM network is another type of ANN, where training function is defined. An LRN is a class of ANN where connections form directed cycles, which create

Table 4. Real world datsets used for LDL.

Sl. no.	Dataset	Examples	Features	Labels
1	Yeast-alpha	2,465	24	18
2	Yeast-cdc	2,465	24	15
3	Yeast-elu	2,465	24	14
4	Yeast-spo5	2,465	24	4
5	Yeast-heat	2,465	24	6
6	Yeast-spo	2,465	24	6
7	Yeast-dtt	2,465	24	4
8	Yeast-spoem	2,465	24	2
9	Yeast-cold	2,465	24	4
10	Yeast-diau	2,465	24	7
11	SJAFFE	213	243	6
12	Natural Scene	2,000	294	9

Table 5. MSE obtained using ALDL and other standard algorithms for LDL on Yeast alpha dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.019316**	0.023236	0.023075	0.023090	0.023016
2	**0.021000**	0.022567	0.022610	0.022580	0.022652
3	**0.018801**	0.023457	0.023247	0.023295	0.023287
4	**0.015828**	0.022962	0.022776	0.022767	0.022790
5	**0.017065**	0.022759	0.022632	0.022623	0.022561
6	**0.016793**	0.022829	0.022592	0.022593	0.022455
7	**0.017623**	0.023114	0.023158	0.023318	0.023035
8	**0.018721**	0.023477	0.023351	0.023335	0.023511
9	**0.017152**	0.022968	0.022912	0.023157	0.023092
10	**0.016799**	0.024196	0.023985	0.023997	0.024069

Table 6. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-cdc dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.023802**	0.028496	0.028368	0.028305	0.028291
2	**0.022162**	0.027250	0.027233	0.026922	0.026992
3	**0.019348**	0.028510	0.028245	0.028339	0.028329
4	**0.025610**	0.028392	0.028009	0.028226	0.028182
5	**0.019698**	0.027837	0.027628	0.027718	0.027735
6	**0.022534**	0.027731	0.027524	0.027683	0.027463
7	**0.026695**	0.028125	0.027987	0.028158	0.027984
8	**0.018659**	0.025541	0.025526	0.025431	0.025386
9	**0.018949**	0.027849	0.02776	0.027897	0.02768
10	**0.020860**	0.028545	0.028238	0.028374	0.028356

an internal state of the network, allowing it to exhibit dynamic temporal behaviour.

The results are given with their corresponding datasets in tables 5–16. The best results are shown in bold. The real world datasets are given in table 4.

Table 7. MSE obtained using ALDL and other standard algorithms for LDL on Yeast elu dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.022196**	0.027859	0.027953	0.028005	0.028063
2	**0.023493**	0.027046	0.026787	0.026916	0.026750
3	**0.019520**	0.028737	0.028423	0.028432	0.028386
4	**0.019172**	0.027791	0.027807	0.028035	0.027701
5	**0.028096**	0.028507	0.028535	0.028448	0.028316
6	**0.019473**	0.027320	0.027410	0.027096	0.027467
7	**0.024241**	0.027520	0.027309	0.027223	0.027403
8	**0.024809**	0.027575	0.027577	0.027503	0.027513
9	**0.028107**	0.028609	0.028726	0.028599	0.028589
10	**0.023266**	0.028674	0.028643	0.028806	0.028749

Table 8. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-spo5 dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.078682**	0.114433	0.115290	0.113419	0.113118
2	**0.079748**	0.113101	0.120042	0.120080	0.117338
3	**0.108920**	0.119179	0.120167	0.121138	0.123791
4	**0.078084**	0.108972	0.113647	0.113698	0.113827
5	**0.081088**	0.117045	0.117893	0.116661	0.117103
6	**0.088224**	0.117446	0.116778	0.117060	0.117297
7	**0.101615**	0.119418	0.122073	0.125987	0.121947
8	**0.097013**	0.119404	0.123468	0.119564	0.120893
9	**0.109107**	0.119566	0.121267	0.122642	0.121079
10	**0.094325**	0.112370	0.113755	0.114874	0.115024

Table 9. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-heat dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.054379**	0.059914	0.059240	0.059079	0.059770
2	**0.040228**	0.058415	0.057851	0.058482	0.058793
3	**0.048607**	0.059160	0.059495	0.059558	0.059996
4	**0.040309**	0.058489	0.059270	0.058848	0.058712
5	**0.052089**	0.058868	0.060181	0.059224	0.059139
6	**0.051470**	0.058854	0.058894	0.058122	0.058421
7	**0.040282**	0.058489	0.059270	0.058848	0.058712
8	**0.042566**	0.058868	0.060181	0.059224	0.059139
9	**0.040982**	0.057507	0.058306	0.058337	0.057997
10	**0.045686**	0.056918	0.058217	0.057373	0.057577

Each dataset passes through 10-fold cross-validation test using nine parts for training and one part for testing. This process of transforming each dataset and simulating it is repeated 10 times. Table 5 shows MSE obtained using ALDL and other standard algorithms for LDL on Yeast alpha. It is observed that ALDL gives the least MSE when

Table 10. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-spo dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.081713**	0.117492	0.116848	0.120386	0.119480
2	**0.088341**	0.113515	0.114743	0.115743	0.115482
3	**0.103450**	0.109587	0.115912	0.110385	0.113206
4	**0.063981**	0.077110	0.078238	0.078983	0.078399
5	**0.074923**	0.083858	0.083724	0.084327	0.087167
6	**0.077512**	0.083953	0.082308	0.083050	0.083021
7	**0.063728**	0.078941	0.079236	0.080577	0.078280
8	**0.058554**	0.082964	0.082601	0.082413	0.082843
9	**0.061976**	0.084420	0.085810	0.085032	0.085208
10	**0.064652**	0.082812	0.083161	0.083208	0.083428

Table 11. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-dtt dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.045947**	0.046979	0.046975	0.047239	0.046770
2	**0.045115**	0.045705	0.046016	0.045592	0.046041
3	**0.044623**	0.045321	0.045287	0.045151	0.045505
4	**0.045681**	0.046398	0.047854	0.047348	0.046421
5	**0.047121**	0.047974	0.048943	0.047827	0.048875
6	**0.047188**	0.047769	0.049230	0.048267	0.048607
7	**0.048064**	0.048950	0.050767	0.050046	0.050137
8	**0.049900**	0.051158	0.050623	0.051252	0.050897
9	**0.047488**	0.048044	0.048236	0.049603	0.048214
10	**0.046344**	0.047089	0.048112	0.048779	0.047106

Table 12. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-spoem dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.076090**	0.086370	0.085758	0.085805	0.086838
2	**0.067410**	0.079480	0.079884	0.079779	0.081122
3	**0.070490**	0.087385	0.087991	0.085576	0.085369
4	**0.079815**	0.108239	0.110450	0.111228	0.111231
5	**0.118873**	0.122859	0.124795	0.124647	0.126046
6	**0.113397**	0.125204	0.131128	0.125449	0.126082
7	**0.105319**	0.127953	0.133426	0.131262	0.135882
8	**0.092333**	0.115785	0.117353	0.118727	0.120796
9	**0.106611**	0.11945	0.123848	0.120426	0.121971
10	**0.089494**	0.109483	0.11348	0.112659	0.109892

Table 13. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-cold dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.068480**	0.071155	0.073091	0.071092	0.071347
2	**0.044694**	0.066345	0.066956	0.066442	0.066973
3	**0.052530**	0.070208	0.071445	0.072399	0.071163
4	**0.054040**	0.065321	0.066277	0.066483	0.067855
5	**0.064508**	0.069895	0.070941	0.070247	0.070827
6	**0.058481**	0.065751	0.06605	0.065462	0.066116
7	**0.049029**	0.068484	0.069836	0.069648	0.070647
8	**0.056625**	0.068489	0.069551	0.07126	0.068772
9	**0.050072**	0.066072	0.066543	0.068361	0.068076
10	**0.064013**	0.068504	0.069223	0.069333	0.067833

Table 14. MSE obtained using ALDL and other standard algorithms for LDL on Yeast-diau dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.044447**	0.055022	0.054967	0.055531	0.055007
2	**0.039880**	0.053230	0.053540	0.053867	0.053986
3	**0.039504**	0.055547	0.056113	0.055734	0.055712
4	**0.041782**	0.052945	0.053172	0.053155	0.053365
5	**0.046378**	0.052302	0.052666	0.052300	0.052348
6	**0.050551**	0.051420	0.051934	0.051345	0.051993
7	**0.039668**	0.055566	0.055645	0.057216	0.056417
8	**0.046728**	0.056441	0.056532	0.056341	0.056754
9	**0.037768**	0.054921	0.057247	0.054959	0.054993
10	**0.037502**	0.051959	0.052802	0.052505	0.052493

Table 15. MSE obtained using ALDL and other standard algorithms for LDL on SJAFFE dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.119692**	0.150587	0.155682	0.147966	0.141498
2	**0.104827**	0.171959	0.168850	0.165407	0.144417
3	**0.095072**	0.177095	0.177812	0.176340	0.137702
4	**0.112572**	0.138654	0.142028	0.151086	0.125493
5	**0.116501**	0.157462	0.166281	0.177213	0.147262
6	**0.090087**	0.138226	0.139727	0.140227	0.136918
7	**0.098532**	0.163615	0.169007	0.170427	0.135172
8	**0.111785**	0.184684	0.179317	0.149981	0.145006
9	**0.109084**	0.168684	0.153977	0.148237	0.127942
10	**0.106839**	0.151174	0.159345	0.154712	0.142873

compared with all other methods, while AA-kNN gives the maximum MSE. Table 6 shows simulation results on Yeast cdc dataset. It has been observed from all simulations that ALDL prevails over other methods in terms of MSE. Table 7 shows MSE obtained for ALDL and other standard LDL on Yeast elu dataset. It is observed that ALDL is better than all other existing methods while AA-kNN gives the maximum error in terms of MSE. The 10 rows of table 8 are obtained during 10-fold cross-validation on Yeast spo5 dataset. It has also been observed that ALDL outperforms other standard methods in terms of MSE, while LM shows maximum error in terms of MSE. Table 9 shows MSE obtained for ALDL and other standard LDL on Yeast heat dataset. It has been observed from all simulations that ALDL prevails over other methods in terms of MSE while MP gives maximum loss. The 10 rows of table 10 are obtained during 10-fold cross-validation for Yeast spo dataset. ALDL betters all existing methods while LRN

Table 16. MSE obtained using ALDL and other standard algorithms for LDL on Natural scene dataset.

Sl. no.	ALDL	AA-kNN	MP	LM	LRN
1	**0.438931**	0.444873	0.492555	0.454138	0.504717
2	**0.370946**	0.404144	0.427019	0.429468	0.470456
3	**0.310397**	0.441782	0.462840	0.466122	0.496508
4	**0.345258**	0.412978	0.449441	0.438886	0.471064
5	**0.307045**	0.420276	0.446425	0.464567	0.465071
6	**0.345499**	0.427857	0.442321	0.435305	0.482579
7	**0.320116**	0.410636	0.495222	0.436106	0.478701
8	**0.315386**	0.430547	0.431071	0.453555	0.476084
9	**0.385992**	0.420600	0.419565	0.429188	0.474481
10	**0.306641**	0.417943	0.433793	0.426828	0.489627

gives the least accuracy in terms of MSE. Table 11 shows MSE obtained for ALDL and other standard LDL on Yeast dtt dataset. It has also been observed that ALDL outperforms other standard methods in terms of MSE, while MP shows the least accuracy in terms of MSE. Ten rows of table 12 are obtained during 10-fold cross-validation for Yeast spoem. It has been observed that ALDL outperforms other standard methods while MP shows maximum loss in terms of MSE. Table 13 shows MSE obtained for ALDL and other standard LDL on Yeast cold dataset. It is observed that ALDL gives the least MSE when compared with all other methods, while MP shows the maximum MSE. The 10 rows of table 14 are obtained during 10-fold cross-validation for dataset Yeast diau. It has also been observed that ALDL outperforms other standard methods in

Table 17. Average of MSE obtained using ALDL and other standard algorithms for LDL all datasets.

Sl. no.	Dataset	ALDL	AA-kNN	MP	Le-Ma	LRN
1	Yeast-alpha	**0.017910**	0.023157	0.023034	0.023076	0.023047
2	Yeast-cdc	**0.021832**	0.027828	0.027652	0.027705	0.02764
3	Yeast-elu	**0.023237**	0.027964	0.027917	0.027906	0.027894
4	Yeast-spo5	**0.091681**	0.116093	0.118438	0.118512	0.118142
5	Yeast-heat	**0.045660**	0.058548	0.059091	0.058710	0.058826
6	Yeast-spo	**0.073883**	0.091465	0.092258	0.092410	0.092651
7	Yeast-dtt	**0.039946**	0.047539	0.048204	0.048110	0.047857
8	Yeast-spoem	**0.091983**	0.108221	0.110811	0.109556	0.110523
9	Yeast-cold	**0.042421**	0.053935	0.054462	0.054295	0.054307
10	Yeast-diau	**0.042421**	0.053935	0.054462	0.054295	0.054307
11	SJAFFE	**0.091983**	0.108221	0.110811	0.109556	0.110523
12	Natural scene	**0.344621**	0.423164	0.450025	0.443416	0.480929

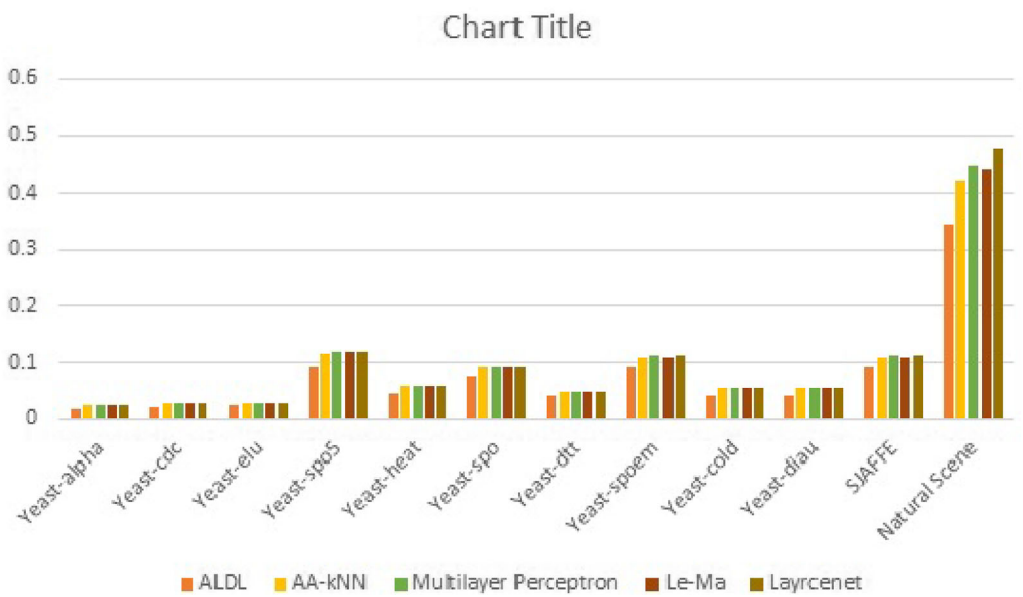

Figure 5. Graphical representation of MSE obtained using ALDL and other standard algorithms for LDL on all datasets.

Table 18. Statistical significance of proposed method over other standard methods for LDL.

Dataset	ALDL & AA-kNN	ALDL & MP	ALDL & LM	ALDL & LRN
Yeast-alpha	3.9e−06(α=3.9e−06)	3.6e−06(α=3.6e−06)	3.8e−06(α=3.8e−06))	3.6e−06(α=3.6e−06)
Yeast-cdc	4.7e−05(α=4.7e−05)	6.0e−05(α=6.0e−05)	5.5e−05(α=5.5e−05)	6.2e−05(α=6.2e−05)
Yeast-elu	0.0011(α=0.00109)	0.0011(α=0.00109)	0.0012(α=0.00121)	0.0012(α=0.00125)
Yeast-spo5	2.8e−05(α=2.8e−05)	2.2e−05(α=2.2e−05)	1.1e−05(α=1.1e−05)	9.3e−06(α=9.3e−06)
Yeast-heat	2.4e−05(α=2.4e−05)	2.3e−05(α=2.3e−05)	3.6e−05(α=3.6e−05)	2.5e−05(α=2.5e−05)
Yeast-spo	2.4e−04(α=2.4e−04)	1.3e−04(α=1.3e−04)	2.8e−04(α=2.8e−04)	2.7e−04(α=2.7e−04)
Yeast-dtt	1.3e−06(α=1.3e−06)	1.4e−04(α=1.4e−04)	1.4e−04(α=1.4e−04)	4.1e−05(α=4.1e−05)
Yeast-spoem	6.6e−05(α=6.6e−05)	4.2e−05(α=4.2e−05)	1.1e−04(α=1.1e−04)	8.6e−05(α=8.6e−05)
Yeast-cold	3.6e−04(α=3.6e−04)	1.9e−04(α=1.9e−04)	3.2e−04(α=3.2e−04)	3.7e−04(α=3.7e−04)
Yeast-diau	5.3e−05(α=5.3e−05)	5.7e−05(α=5.7e−05)	6.2e−05(α=6.2e−05)	4.3e−05(α=4.3e−05)
SJAFFE	7.2e−06(α=7.2e−06)	2.3e−06(α=2.3e−06)	4.4e−06(α=4.4e−06)	6.8e−06(α=6.8e−06)
Natural scene	2.3e−04(α=2.3e−04)	4.9e−05(α=4.9e−05)	1.1e−04(α=1.1e−04)	2.2e−04(α=2.2e−04)

terms of MSE and MP shows the maximum loss. Table 15 shows MSE obtained for ALDL and other standard LDL on SJAFFE dataset. It is observed that ALDL gives the least MSE when compared with all other methods, while MP shows the maximum MSE. The 10 rows of table 16 are obtained during 10-fold cross-validation for dataset Natural scene. It has been observed that ALDL prevails over other methods in terms of MSE while AA-kNN gives the least accuracy. Table 17 shows the average MSE computed of all the 10 cross-validation trials of each method for each dataset. Figure 5 presents average MSE obtained using ALDL and other standard algorithms for LDL on all datasets. Table 18 shows that the results are statistically significant.

5. Conclusion and future works

In this paper a new method, ALDL, has been proposed for LDL. A new function ADF has been proposed and it has been observed that it is highly capable of predicting label distributions than existing methods such as AA-kNN, MP, LM and LRN. Twelve benchmark datasets including Yeast dataset and its variants, SJAFFE and Natural scene datasets have been considered for comparative evaluation of the existing methods. Comparative performance of ALDL with the existing methods shows superiority of ALDL in terms of accuracy over other methods. The results of experiments suggest further investigation of application of ALDL in non-textual and non-numeric datasets and usage of the same in Big Data applications.

Acknowledgements

We are thankful to the Media Lab Asia, Department of Electronics and Information Technology (DEITY), Ministry of Communications and Information Technology, Government of India, for providing us support for carrying out this work as a part of the sponsored project.

Nomenclature

AA-kNN	algorithm adaptation *k*-nearest neighbour
ADF	angular distribution function
ALDL	angular label distribution learning
ANN	artificial neural network
LDL	label distribution learning
LM	Levenberg–Marquardt
LRN	layer-recurrent network
MP	multilayer perceptron
MSE	mean-square error

References

[1] Zhang M L and Zhou Z H 2007 ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recogn*. 40: 2038–2048

[2] Zhou Z H *et al* 2012 Multi-instance multi-label learning. *Artif. Intell.* 176: 2291–2320

[3] Zhang Y, Zincir-Heywood N and Milios E 2005 Narrative text classification for automatic key phrase extraction in web document corpora. In: *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, pp. 51–58

[4] Li T, Ogihara M and Li Q 2003 A comparative study on content-based music genre classification. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 282–289

[5] Boutell M R *et al* 2004 Learning multi-label scene classification. *Pattern Recogn*. 37: 1757–1771

[6] Tsoumakas G and Ioannis K 2007 Multi-label classification: an overview. *Int. J. Data Ware. Min*. 3: 1–13

[7] Tsoumakas G, Ioannis K and Ioannis V 2011 Random k-labelsets for multilabel classification. *IEEE Trans. Knowl. Data Eng*. 23: 1079–1089

[8] Zhu S, Ji X, Xu W and Gong Y 2005 Multi-labelled classification using maximum entropy method. In: *SIGIR*, pp. 274–281

[9] Ho T K 1998 The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell*. 20: 832–844

[10] Breiman L 1996 Bagging predictors. *Mach. Learn.* 24: 123–140

[11] Nasierding G, Abbas Z K and Grigorios T 2010 A triple-random ensemble classification method for mining multi-label data. In: *IEEE International Conference on Data Mining Workshops*, pp. 49–56

[12] Read J, Bernhard P and Geoff H 2008 Multi-label classification using ensembles of pruned sets. In: *Eighth IEEE International Conference on Data Mining*, pp. 995–1000

[13] Read J *et al* 2011 Classifier chains for multi-label classification. *Mach. Learn*. 85: 333

[14] Geng X 2016 Label distribution learning. *IEEE Trans. Knowl. Data Eng*. 28: 1734–1748

[15] Geng X and Luo L 2014 Multilabel ranking with inconsistent rankers. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014*, pp. 3742–3747

[16] Vapnik V N and Vlamimir V 1998 Statistical Learning Theory, Vol. 1, New York, Wiley

[17] Larose D T and Larose C D 2014 *Discovering knowledge in data: an introduction to data mining.* John Wiley & Sons, New Jersey

[18] Geng X, Smith-Miles K and Zhou Z H 2009 Facial age estimation by multilinear subspace analysis. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 865–868

[19] Geng X, Yin C and Zhou Z H 2013 Facial age estimation by learning from label distributions. *IEEE Trans. Pattern Anal. Mach. Intell*. 35: 2401–2412

[20] Han J, Pei J and Kamber M 2011 *Data mining: concepts and techniques.* Elsevier, MA, USA

[21] Zhang M L and Zhou Z H 2005 A k-nearest neighbor based algorithm for multi-label classification. In: *IEEE International Conference on Granular Computing*, pp. 718–721

[22] Raab D H and Green E H 1961 A cosine approximation to the normal distribution. *Psychometrika*. 26: 447–50

[23] Eisen M B *et al* 1998 Cluster analysis and display of genome-wide expression patterns. In: *Proceedings of the National Academy of Sciences*, pp. 14863–14868

[24] Lyons M *et al* 1998 Coding facial expressions with gabor wavelets. In: *Automatic Face and Gesture Recognition, 1998. Proceedings of Third IEEE International Conference*, pp. 200–205

[25] Ahonen T, Abdenour H and Matti P. 2006 Face description with local binary patterns: application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*. 28: 2037–2041

[26] Sanger, T D 1989 Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Netw.* 2: 459–473

[27] Hagan M T *et al* 1996 *Neural Network Design*, vol. 20, Boston, PWS Publishing Company.

[28] Liu Q and Jun W 2008 A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. *IEEE Trans. Neural Netw*. 19: 558–570