Applied Soft Computing 110 (2021) 107585

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Duo-LDL method for Label Distribution Learning based on pairwise class dependencies

Adam Żychowski, Jacek Mańdziuk *

Faculty of Mathematics and Information Science, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland

ARTICLE INFO

Article history:
Received 24 March 2020
Received in revised form 18 May 2021
Accepted 31 May 2021
Available online 8 June 2021

Keywords:
Label distribution learning
Neural networks
Duo-LDL

ABSTRACT

Label Distribution Learning (LDL) is a new learning paradigm with numerous applications in various domains. It is a generalization of both standard multiclass classification and multilabel classification. Instead of a binary value, in LDL, each label is assigned a real number which corresponds to a degree of membership of the object being classified to a given class. In this paper a new neural network approach to Label Distribution Learning (Duo-LDL), which considers pairwise class dependencies, is introduced. The method is extensively tested on 15 well-established benchmark sets, against 6 evaluation measures, proving its competitiveness to state-of-the-art non-neural LDL approaches. Additional experimental results on artificially generated data demonstrate that Duo-LDL is especially effective in the case of most challenging benchmarks, with extensive input feature representations and numerous output classes.

1. Introduction

Classification is one of the most popular tasks in machine learning, however, its standard formulation (binary or multiclass) does not cover all real-life cases. Multilabel Classification (MC) [1] extends a baseline classification task by assuming that each object can be assigned to a subset of all available classes. MC formulation has become quite popular in the literature due to many practical applications [2]. Nonetheless, many real-life settings cannot be modeled within the above-mentioned frameworks, e.g. an object may belong to two classes, but not to the same degree (a cross-breed dog may have more traits from one breed than from the other one, or a seaplane in standard vehicle classification should most probably be classified as an aircraft, not as a boat, though only assignment to both of these classes, with different degrees, can fully represent its construction. In order to address the above needs a novel machine learning paradigm — Label Distribution Learning (LDL) was formally proposed in 2016 [3] (before the name LDL was coined in [3] several other works also considered this problem, e.g. [4–6]).

In LDL, instead of assigning a binary value to a class a real number is assigned to each label, whose value indicates a *degree of membership* of the object to the respective class. Similarly to the case of probability distribution the sum of all assigned values equals 1 for each sample.

Despite being established only a few years ago, LDL has already gained visible attention and various methods have been proposed in the literature. In most of the cases, they adopt one of the well-known Machine Learning (ML) algorithms to deal with LDL (e.g. k-Nearest Neighbors or Multilayer Perceptron) or transform LDL to another well-researched problem, e.g. MC. There are only a few methods designed specifically with the aim of solving LDL.

The vast majority of existing LDL solutions represent a supervised approach as the most common LDL formulation is the one considered in this paper, i.e. with an assumption that information about the set of labels and label distribution over training instances is available to the solution process (see Section 3.1 for a formal problem definition).

2. Motivation

The majority of hitherto proposed algorithms focus on finding a solution separately for each label, thus ignore correlations between labels. However, please observe that in MC domain solutions which tackle a task by considering pairwise relations between labels (e.g. by turning the task into *label ranking* problem that is solved using techniques of pairwise comparison [7] or by direct learning of correlations between any two labels [8]) clearly outperform methods that decompose the MC problem into independent binary classification tasks [9]. Since LDL is an extension of MC it is legitimate to expect that a similar idea will be valid for LDL problem. Consequently, our main research claim is that considering pairwise label interdependencies has potential to improve the results for LDL task.

* Corresponding author.
E-mail address: j.mandziuk@mini.pw.edu.pl (J. Mańdziuk).

https://doi.org/10.1016/j.asoc.2021.107585

In the paper we experimentally verify the above claim. To this end we propose a new neural network approach to LDL (called Duo-LDL) which considers pairwise relations between classes in the form of an extended output layer and specifically designed error function that incorporates pairwise correlations between classes. Duo-LDL is extensively evaluated on a set of 15 well-established benchmarks and 6 different evaluation measures and demonstrates its on par performance to state-of-the-art LDL approaches.

To our knowledge, among LDL approaches, there is only one method that employs neural networks [3] which, however, does not take into account inter-label dependencies in the training data and employs a feedforward network with a direct class encoding in the output layer.

3. Label distribution learning

3.1. Problem formulation

Label Distribution Learning problem can be formulated as follows. Let $X = \{x_1, \ldots, x_N : x_i \in \mathbb{R}^n\}$ denotes an n-dimensional instance space and let $Y = \{y_1, y_2, \ldots, y_q\}$ be a finite set of q predefined classes. LDL consists in learning function $p : X \times Y \rightarrow [0, 1]$, where $p(x_i, y_j)$ (in short p_i^j) is a real number from unit interval which denotes a *degree of membership* of instance x_i to class y_j. Furthermore,

$$\forall x_i \in X \quad \sum_{j=1}^{q} p_i^j = 1 \tag{1}$$

Alternatively, in some domains, p_i^j can be interpreted as the probability that instance x_i belongs to class y_j. In this context p_i^j represents a degree to which label y_j describes instance x_i.

It is generally assumed that for some training subset $X^t \subset X$ of data instances the probability distributions of $p_i^j(x_i \in X^t, y_j \in Y)$ is provided. Hence, in ML approaches function p is usually learnt in a supervised training manner.

LDL formulation extends several popular classification tasks. Traditional (single label) classification is a special case of LDL in which for each instance $x_i \in X, p_i^j = 1$ for exactly one class y_j and $p_i^j = 0$ for all the other classes. In terms of LDL formulation MC can be expressed in the following way. For a given instance $x_i \in X$, for each label y_j from the true labels set $y_j \in Y_{true}(x_i)$ assign $p_i^j = \frac{1}{|Y_{true}(x_i)|}$ and $p_i^j = 0$, otherwise.

In some sense, LDL links two of the most popular machine learning tasks: classification and regression. On the one hand, there is a certain finite set of classes to which an instance may be assigned (classification), but, on the other hand the objective is to find a real-valued degree of membership for each class (regression). Deeper theoretical analysis of LDL problem can be found in [10].

3.2. State-of-the-art approaches

This section presents a brief overview of state-of-the-art LDL solution methods which will be used in Section 6 for evaluation of the Duo-LDL approach.

Observe that in the recent literature some new variants of the LDL problem were introduced, for instance, LDL with noisy labels [11] or with partially labeled data [12]. However, as mentioned above, our focus is on the most popular perfect-information formulation of LDL which assumes that distributions of labels are known for a (training) subset of the data samples. Consequently, all state-of-the-art methods we compare with are supervised ones. These approaches and can be roughly divided into 3 groups: problem transformation methods, algorithm adaptation approaches, and specialized (dedicated) algorithms.

Problem transformation methods reformulate the LDL problem to the form of another well-established learning scenario — the single-label learning. For each example x_i, instead of learning a distribution of classes over a vector of q elements, separate learning tasks are performed for each of the q classes with the overall number of training examples increasing from n to qn. This transformation is straightforward, easy to apply and efficient in certain cases, but since each label is considered independently, it does not take into account relations between classes. The above way of approaching LDL problem was proposed in the original Zhang's paper [3] in the form of PT-Bayes or PT-SVM algorithms. The former employs standard Bayesian classifier (the posterior probability computed by the Bayes rule), the latter adopts the (binary) Support Vector Machine (SVM). In either case the final result is obtained using a pairwise coupling multi-class method [13].

The second strategy is *algorithm adaptation* which consists in an adjustment of an existing solution method to make it capable of dealing with LDL. One example is the AA-kNN [3], which adopts the k-Nearest Neighbors method (kNN) [14]. Classification result for a given instance x_i is a normalized mean of label distributions of k instances from the training set that are closest to x_i (in terms of Euclidean distance in a vector feature space). One of the recently introduced adaptation methods is LALOT [15] which adopts Optimal Transport (OT) theory [16] to simultaneously learn label distribution and exploit label correlations. Another approach, AA-BP [3], employs a Multilayer Perceptron (MLP) with one hidden layer trained with backpropagation algorithm. The sizes of the input and output layers are equal to the numbers of input features and the number of classes, respectively. Condition (1) is ascertained by using a softmax activation function in the output layer. The algorithm serves as the basis for the method proposed in this paper. Our method, CARTesian-based Label Distribution Learning (Duo-LDL) which is described in the following section, significantly extends and enhances AA-BP, which leads to a qualitatively new approach with qualitatively stronger experimental results.

The last group of methods includes algorithms which are *designed specifically for the purpose of solving LDL problems*. IIS-LLD [3] treats a given LDL task as the maximum entropy model [17] and maximizes its likelihood based on the improved iterative scaling (IIS) strategy [18]. Another approach, BFGS-LLD, relies on Broyden–Fletcher–Goldfarb–Shanno (BFGS) [19] quasi-Newton optimization method. A recent promising method is GD-LDL-SCL [20] which exploits local correlations by means of predicting their distribution/structure based on the original features and the local correlation vector which is created for each instance. A similar approach (named Adam-LDL-SCL) is presented in [21] which improves the above-mentioned GD-LDL-SCL by using Adam optimizer [22]. Another recent method LDL-LCLR [23] utilizes label correlation matrix to capture global correlations among classes. Class correlations among local samples are detected and used to modify the correlation matrix.

All 10 above-mentioned methods serve as baseline approaches for experimental performance assessment of our Duo-LDL algorithm.

4. Proposed approach

Feedforward neural networks are one of the first choices when solving LDL tasks, as they are naturally capable of returning a vector of real numbers normalized to sum up to 1 by means of the softmax activation function in the output layer. Despite this natural fitness, their straightforward application to LDL does not

lead to strong results [3]. The MLP architecture with one hidden layer (1hl) and q output neurons (one per class) utilized in AA-BP method [3], mentioned in the previous section, yielded results inferior to alternative approaches, e.g. AA-kNN, IIS-LLD or BFGS-LLD. A similar situation was observed in the case of MC problem, where initial, direct application of MLPs [24] was not competitive to other established algorithms in that domain. After some adaptation and refinement, in particular suitable reformulation of the error function proposed in [25], which directly incorporates information about correlations and dependencies between labels, the results have improved visibly. Subsequent modifications proposed in [26,27] made the MLP approach a viable alternative to existing state-of-the-art MC methods.

The main goal of this paper is to propose a novel LDL method which takes into account not only degrees of membership of a given sample to target classes, but also considers nonlinear pairwise dependencies between classes. The underlying idea of this method was inspired by the relevance of considering pairwise relations between labels in the learning process in MC domain, observed in several recent papers [28–30,25–27]. Due to a certain degree of similarity between MC and LDL tasks, taking into account the inter-label dependencies may potentially improve LDL solution methods as well.

The proposed method (abbreviated as Duo-LDL) is implemented in the form of a one-hidden-layer MLP with the input layer size equal to the number of features in the considered data set and the output layer consisting of $q(q-1)$ neurons (see Fig. 1). Every $q-1$ consecutive outputs represent differences between degrees of membership of a given input sample x to a certain class and all the other $q-1$ classes, respectively. Formally, let $c_{ij}(x)$ be an output value of $((i-1)(q-1)+j)$th output neuron, for $i,j \in \{1,\ldots,q\}, i \neq j$, in response to the input sample x. Then, the training objective of this neuron is to learn the following difference:

$$c_{ij}(x) = p^i(x) - p^j(x) \tag{2}$$

where $p^k(x), k = 1,\ldots,q$ denotes a degree of membership of sample x to the kth class and is calculated based on the network's outputs $c_{ij}(x)$. First, let us write all Eqs. (2) for a given label k and an additional identity term as the last equation.

$$\begin{aligned}
\hat{p}^k(x) &= c_{k1}(x) + \hat{p}^1(x)\\
&\vdots\\
\hat{p}^k(x) &= c_{k(k-1)}(x) + \hat{p}^{k-1}(x)\\
\hat{p}^k(x) &= c_{k(k+1)}(x) + \hat{p}^{k+1}(x)\\
&\vdots\\
\hat{p}^k(x) &= c_{kq}(x) + \hat{p}^q(x)\\
\hat{p}^k(x) &= \hat{p}^k(x)
\end{aligned} \tag{3}$$

Summing the above equations leads to

$$q \cdot \hat{p}^k(x) = \sum_{\substack{j=1\\ j\neq k}}^{q} c_{kj}(x) + \sum_{j=1}^{q} \hat{p}^j(x) \tag{4}$$

From LDL definition, $\sum_{j=1}^{q} \hat{p}^j(x) = 1$, so the above formula can be rewritten as

$$\hat{p}^k(x) = \frac{1}{q}\left(\sum_{\substack{j=1\\ j\neq k}}^{q} c_{kj}(x) + 1\right) \tag{5}$$

In effect, the error function in the training process is of the following form:

$$E = \sum_{\substack{i,j=1\\ i\neq j}}^{q} \left(\hat{p}^i(x) - \hat{p}^j(x) - c_{ij}(x)\right)^2 \tag{6}$$

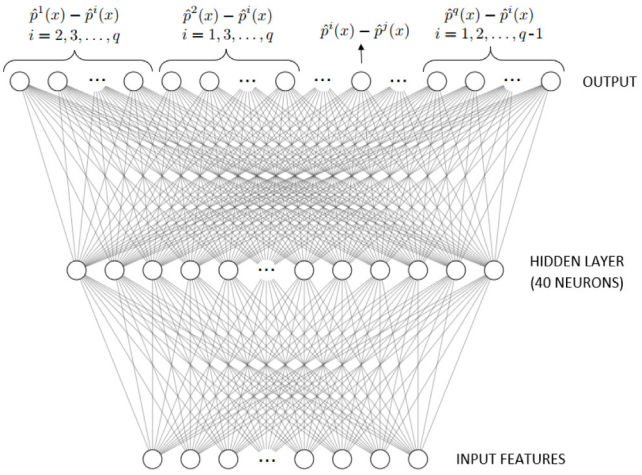

Fig. 1. Proposed network architecture.

In the tests phase, a degree of membership of sample x to a given class k ($\hat{p}^k(x)$) is computed from the network outputs according to (5).

A pseudocode of the proposed method is presented as Algorithm 1. The proposed approach differs from the baseline neural network method (AA-BP) [3] in several key aspects. The main difference is the learning goal — in the AA-BP method labels distribution is learnt directly, whereas our method additionally incorporates information about label inter-dependencies, as well as the LDL probability distribution constraint (1). Furthermore, the methods differ in terms of network architectures and error functions. Finally, on a more technical side, training of Duo-LDL includes several recent learning techniques (e.g. weight initialization [31], weights decay, or mini-batch gradient descent) which overall improve the learning process. In order to make comparison fair and focus exclusively on the impact of added *label correlations part*, the same training techniques were also applied to AA-BP. This updated version of AA-BP algorithm is denoted AA-BP-T in the following results section.

Algorithm 1: Neural network training

$T_{set} = \{(x_1, P_1), (x_2, P_2), \ldots, (x_n, P_n)\}$ - training set
$x_i \in \mathbf{R}^n$ - *training input samples*
$P_i = \{p_i^1, p_i^2, \ldots, p_i^q\}$ - *expected outputs: degrees of membership of sample x_i to the classes* $1, 2, \ldots, q$
Function *TrainNetwork(T_{set})*
  Create mulitlayer perceptron with input layer: n neurons, hidden layer: 40 neurons, output layer: $q(q-1)$ neurons
  Use Xavier Weight Initialization [31] to randomly initialize weights
  for *fixed number of epochs* **do**
    Create random mini batch $B \subset T_{set}$
    forall the *input sample* $b_k = (x_k, P_k) \in B$ **do**
      Feedforward network with x_k and get outputs $c_{ij}(x_k)$, $i,j \in \{1,\ldots,q\}, i \neq j$, where c_{ij} is the value of $((i-1)(q-1)+j)$th output neuron
      Compute error $E_{ij} = p_k^i - p_k^j - c_{ij}(x_k)^2$
    end
    Backpropagate aggregated error from all samples in B and update network weights.
  end

5. Experimental setup

This section presents error measures and benchmark problems used in the experimental evaluation of the proposed method.

5.1. Benchmark problems

Following Geng's introductory paper to LDL [3], in the experimental assessment of Duo-LDL efficacy, the same as therein set of 15 popular and well-established benchmark problems were used. All these data sets are derived from real-life domains and possess various characteristics in terms of the number of instances, input features and labels, as presented in Table 1.

Data set *Movie* [5] contains Netflix ratings of more than 7000 movies. For each movie the feature vector includes information about actors, a director, country of production, genre, etc. in the form of numeric values (one hot encoding). The output is a distribution over 5-level rating (the percentage of each rating level) based on over 54 million spectators assessments.

The next data set *Human Gene* is the largest among considered benchmarks in terms of the number of examples and labels. Instances are human genes represented in the form of 36 real number descriptors [32]. The goal is to assign the gene expression (degrees of membership) for 68 diseases (classes).

The instances of *Natural Scene* data set [33] are images of nature, each described by a 294-dimensional feature vector — a compressed image representation. First, for each image, 10 people sorted the following 8 labels: (plant, sky, cloud, snow, building, desert, mountain, water) by their relevance to the image. Next, based on these 10 rankings, label distributions were created for the images, which present the learning goal in this benchmark.

SJAFFE is a data set derived from the JAFFE database [34] that contains grayscale photos of human faces represented in the form of 243-dimensional feature vectors. For each face, 60 people scored each of the 6 following emotions (happiness, sadness, surprise, fear, anger, and disgust) in a 5-degree scale. For each instance a normalized average score for each emotion is used as a label distribution to be learnt in this problem.

The next benchmark – *SBU_3DFE* – is similar to *SJAFFE* but based on a larger database composed of 2500 instances of 3D facial expression images [35], scored in the way similar to the *SJAFFE* set (23 people scored each of 6 emotions). The goal is to assign a degree of relevance of the above-listed emotions for a given sample.

The final 10 benchmark sets presented in Table 1 contain the data related to gene expressions of yeasts [36]. Each input instance represents one gene and is described by 24 numbers forming the so-called phylogenetic profile. Each output label corresponds to a result of the gene expression process in a certain, pre-defined time point. The task is to learn normalized distributions of these expression levels.

All the above-described benchmark data sets can be downloaded from [37].

5.2. Evaluation measures

Quality verification of an LDL algorithm is a rather ambiguous task due to the lack of comprehensive and well-established evaluation measures. The main reason for that is the fact that in LDL, instead of comparison of two binary values, vectors of real numbers (label distributions) must be confronted. Therefore, the number of possible assessments is much higher and, what is more, each of the existing evaluations measures addresses only specific aspects of the tested algorithm. For this reason we take into consideration 6 distinct evaluation measures which are derived from different metrics. All of them were previously used for making comparisons between LDL algorithms in the related literature [3]. The metrics are briefly introduced in the remainder of this subsection.

Let us denote by $P_x = \{p^1(x), \ldots, p^q(x)\}$ a vector of target (true) values of label distribution ($p(x)$) for sample x and by $\hat{P}_x = \{\hat{p}^1(x), \ldots, \hat{p}^q(x)\}$ a vector of model output values (predictions) for this sample. Then the considered error measures are defined as follows.

Chebyshev distance (Cheb):

$$D_1(P_x, \hat{P}_x) = \max_{j\in\{1,\ldots,q\}} |p^j(x) - \hat{p}^j(x)|$$

Clark distance (Clark):

$$D_2(P_x, \hat{P}_x) = \sqrt{\sum_{j=1}^{q} \frac{(p^j - \hat{p}^j)^2}{(p^j + \hat{p}^j)^2}}$$

Canberra metric (Canber):

$$D_3(P_x, \hat{P}_x) = \sum_{j=1}^{q} \frac{|p^j(x) - \hat{p}^j(x)|}{p^j(x) + \hat{p}^j(x)}$$

Kullback–Leibler divergence (KL-div):

$$D_4(P_x, \hat{P}_x) = \sum_{j=1}^{q} p^j(x) \ln \frac{p^j(x)}{\hat{p}^j(x)}$$

Cosine coefficient (Cosine):

$$D_5(P_x, \hat{P}_x) = \frac{\sum_{j=1}^{q} p^j(x)\hat{p}^j(x)}{\sqrt{\sum_{j=1}^{q}(p^j(x))^2}\sqrt{\sum_{j=1}^{q}(\hat{p}^j(x))^2}}$$

Intersection similarity (Intersec):

$$D_6(P_x, \hat{P}_x) = \sum_{j=1}^{q} \min(p^j(x), \hat{p}^j(x))$$

For each of the above-mentioned evaluation metrics, values obtained for all test samples are averaged, so as to provide the final assessment:

$$D_i = \frac{1}{|X_{test}|} \sum_{x\in X_{test}} D_i(P_x, \hat{P}_x) \tag{7}$$

where X_{test} denotes the test set.

For all measures, their values fit the interval [0, 1]. The first four of them are distance measures and the smaller their values, the higher the quality of the assessed method. The remaining two measures are based on similarity, hence the greater their values, the higher the estimated performance.

According to a measure-related survey paper [38] each of the above-listed measures belongs to a different family of measures and covers different aspects of LDL results. For example *Chebyshev distance* considers only the worst match over the whole label distribution while *Clark* or *Canberra metric* average the errors, each of them in a distinct manner.

5.3. Duo-LDL training procedure

For each data set a randomly selected subset of 90% of all data samples were used for training and the remaining 10% for testing. Network weights update was performed with backpropagation algorithm with the learning rate set to 0.05, and with the weight decay regularization having weight decay cost equal to 0.5. Expected outputs were computed according to Eq. (2) based on true label distribution. Learning process was performed in mini-batch mode with batch size equal to 50 (weight update was calculated based on mean value of the error function (Eq. (6)) for 50 training examples). Training was stopped after 100 epochs. The order of presented examples was randomized in each epoch. A hyperbolic tangent activation function was used to fit the output range ([−1, 1]). Implementation of the method with the

Table 1
Basic parameters of 15 benchmark data sets used in the experimental evaluation of the Duo-LDL approach.

Data set	# Instances	# Features	# Labels	Reference
Movie	7755	1869	5	[5]
Human Gene	30542	36	68	[32]
Natural Scene	2000	294	9	[33]
SJAFFE	213	243	6	[34]
SBU_3DFE	2500	243	6	[35]
Yeast-alpha	2465	24	18	[36]
Yeast-cdc	2465	24	15	[36]
Yeast-elu	2465	24	14	[36]
Yeast-diau	2465	24	7	[36]
Yeast-heat	2465	24	6	[36]
Yeast-spo	2465	24	6	[36]
Yeast-cold	2465	24	4	[36]
Yeast-dtt	2465	24	4	[36]
Yeast-spo5	2465	24	3	[36]
Yeast-spoem	2465	24	2	[36]

above-described setup can be downloaded from a public code repository [39].

Please note that the constraint concerning the probability distribution summing up to 1 (Eq. (1)) is already incorporated into the proposed method by means of the transformation formula (4), and therefore, there is no need to use any normalization techniques, e.g. the softmax function, in the output layer. The initial experiments fully supported this claim.

6. Experimental results

The results presented in this section are derived from 30 independent runs, each with 10-fold cross-validation, for each of 15 benchmark problems. For each benchmark, the final evaluation measure (7) is calculated as the average result of 300 outcomes (30 runs with 10 folds).

The results for 10 methods used for comparison with Duo-LDL were independently reproduced using codes published on the website [37] or original codes provided courtesy of competitive methods' authors.

For the sake of space savings, the detailed results are presented only for the two most challenging benchmarks (*Movie* – with the greatest number of input features, and *Human Gene* – with the greatest number of labels) in Tables 2 and 3, respectively. The detailed results for all 15 data sets can be downloaded from our webpage [40].

In the case of *Movie* set, Duo-LDL yielded the best results for 4 evaluation measures with statistical significance for most of competitive methods (c.f. Table 4). Also for the other benchmark, *Human Gene*, with the greatest number of labels, also for 4 error metrics our method gained the leading position, however in this case differences are smaller (especially to recently introduced methods — LDL-LCLR, GD/Adam-LDL-SCL, LALOT and BFGS-LLD) and not statistically significant. For both benchmarks the worst results were obtained by PT-Bayes. The previous neural network approach (AA-BP) was ranked in the middle of the pack. The remaining detailed results for the other benchmarks (not presented in the paper) fit the following pattern: generally Duo-LDL, BFGS-LLD and LDL-LCLR occupy the two leading positions and PT-Bayes, PT-SVM and AA-BP are recognized as the weakest approaches.

A cumulative comparison of the methods is presented in Table 4. Each value in the table belongs to the interval $[1.00, 12.00]$ are represents the average ranking score of the respective (*method*, *benchmark*) pair across all 6 error measures. Two algorithms (BFGS-LLD and Duo-LDL) are distinctly better than all the remaining ones. Duo-LDL gained the first place in the case of 5 benchmarks. Additionally, in 2 out of these 7 its average ranking position was equal to 1.00, which means receiving the best score for all evaluation measures. BFGS-LLD was the winning algorithm in 6 benchmarks, out of which 4 were scored with 1.00.

In a head-to-head comparison the BFGS-LLD method was slightly more effective than our approach achieving the overall average score (across all benchmarks) equal to 2.11 compared to 2.20 of Duo-LDL. The main reason for that was visibly worse performance of Duo-LDL on *SJAFFE*, for which it gained around 6 place. This weaker performance of Duo-LDL stems from a very small number of training examples in the data set which is insufficient to make the MLP training process effective. Generally speaking, nonlinear methods (such as neural networks) need more input data to learn/solve LDL tasks than linear methods [41] (such IIS-LLD or BFGS-LLD). Please observe that AA-BP algorithm, which is also neural network based, obtained even worse results than Duo-LDL for this benchmark. Another methods like GD-LDL-SCL and Adam-LDL-SCL also obtained significantly worse results for *SJAFFE* benchmark and suffer from lack of training examples.

Statistical relevance of the differences between our method and competitive approaches was tested according to 1-tailed t-test with significance level equal to 0.05 and with normal distribution of data checked by Shapiro–Wilk test. As can be observed in Table 4, majority of Duo-LDL results are statistically significantly better than those of PT-Bayes, PT-SVM, AA-kNN, AA-BP, and AA-BP-T algorithms. A comparison of Duo-LDL with the strongest among tested methods (BFGS-LLD) shows statistically significant advantage only in two cases. For *Movie* benchmark, Duo-LDL outperforms BFGS-LLD and for *SJAFFE* data set BFGS-LLD dominates (for the reasons explained above). For all other benchmarks no statistically significant differences were detected. Generally, there is not much statistically significant differences recognized between 7 rightmost methods and based on detailed results all of them yield very similar results.

Another comparison of the methods, form a different perspective, i.e. considering the average ranking positions for each evaluation measure across all benchmark sets, is presented in Table 5. Again BFGS-LLD and Duo-LDL superiority over competitive approaches can be easily noticed, and again BFGS-LLD outperforms Duo-LDL — in this case in 4 out of 6 evaluation metrics, albeit the differences are not statistically significant. No significant differences in results can be observed among the measures.

One of the disadvantages of presentation of results by means of ranking positions is blurring the detailed differences between methods under comparison. In order to address this problem cumulative normalized scores over all benchmarks, per each evaluation measure were calculated for each method and presented in Table 6. Observe that evaluation measures return values from different intervals and for some of them, the greater the

Table 2
The average results for the ***Movie*** benchmark set. Best results for each evaluation measure are bolded.

	PT-Bayes	PT-SVM	AA-kNN	AA-BP	AA-BP-T	IIS-LLD	BFGS-LLD	LALOT	GD-LDL-SCL	Adam-LDL-SCL	LDL-LCLR	Duo-LDL
Chebyshev	0.199	0.213	0.154	0.157	0.154	0.150	0.136	0.271	0.134	**0.124**	0.128	**0.124**
Clark	0.799	0.797	0.652	0.675	0.633	0.591	0.589	1.439	0.571	**0.543**	0.564	0.565
Canberra	1.547	1.537	1.276	1.269	1.206	1.137	1.138	2.232	1.123	**1.041**	1.085	1.077
Kullback–Leibler	0.953	0.268	0.201	0.179	0.160	0.137	0.140	0.469	0.233	0.215	0.125	**0.113**
Cosine	0.850	0.806	0.880	0.895	0.900	0.905	0.912	0.739	0.913	0.924	0.921	**0.926**
Intersection	0.725	0.711	0.780	0.788	0.793	0.800	0.809	0.647	0.811	0.820	0.819	**0.821**

Table 3
The average results for the ***Human Gene*** benchmark set. Best results for each evaluation measure are bolded.

	PT-Bayes	PT-SVM	AA-kNN	AA-BP	AA-BP-T	IIS-LLD	BFGS-LLD	LALOT	GD-LDL-SCL	Adam-LDL-SCL	LDL-LCLR	Duo-LDL
Chebyshev	0.195	0.054	0.065	0.059	0.057	**0.053**	**0.053**	**0.053**	**0.053**	**0.053**	**0.053**	**0.053**
Clark	4.674	2.139	2.388	3.344	2.736	2.123	2.111	2.115	2.112	2.112	**2.108**	2.110
Canberra	34.238	14.631	16.283	22.788	19.321	14.541	14.453	14.487	14.451	**14.433**	14.443	14.442
Kullback–Leibler	1.887	0.240	0.301	0.500	0.366	0.238	**0.236**	0.237	**0.236**	0.237	**0.236**	**0.236**
Cosine	0.456	0.832	0.766	0.726	0.773	0.833	0.834	0.834	0.834	**0.835**	0.834	**0.835**
Intersection	0.470	0.781	0.742	0.671	0.717	0.783	0.784	0.784	**0.785**	**0.785**	**0.785**	**0.785**

Table 4
Average ranking positions of tested methods for each benchmark data set, across all 6 evaluation measures. Best values for each benchmark are **bolded**, gray background denotes statistically significant difference in results between Duo-LDL and the respective method.

	PT-Bayes	PT-SVM	AA-kNN	AA-BP	AA-BP-T	IIS-LLD	BFGS-LLD	LALOT	GD-LDL-SCL	Adam-LDL-SCL	LDL-LCLR	Duo-LDL
Movie	10.67	10.50	8.17	8.00	6.67	5.33	5.00	11.83	4.83	2.50	2.67	**1.50**
Human Gene	12.00	8.00	9.50	10.83	9.67	6.00	3.00	4.33	2.33	2.17	1.67	**1.33**
Natural Scene	11.33	11.33	6.17	6.50	6.67	6.67	3.67	9.83	3.67	8.00	2.50	**1.50**
SJAFFE	5.17	6.33	2.67	8.50	7.67	3.83	**1.17**	9.33	11.00	11.00	2.83	6.50
s-BU 3DFE	9.50	11.00	3.67	8.83	7.33	5.83	**1.67**	11.33	2.00	5.67	7.83	2.83
Yeast-alpha	12.00	9.00	6.50	11.00	10.00	7.50	1.50	4.67	3.50	3.17	1.33	**1.00**
Yeast-cdc	12.00	9.00	7.00	11.00	10.00	6.67	1.67	6.00	2.50	**1.00**	1.50	1.17
Yeast-cold	12.00	9.00	7.00	11.00	10.00	6.67	**1.00**	6.00	3.00	1.67	**1.00**	1.83
Yeast-diau	12.00	9.83	6.83	11.00	9.00	6.00	2.83	8.00	1.50	**1.00**	3.00	3.50
Yeast-dtt	12.00	8.67	8.00	11.00	10.00	5.00	**1.00**	6.67	3.00	1.50	1.17	1.50
Yeast-elu	12.00	9.50	8.00	11.00	8.67	5.50	2.67	6.83	**1.00**	1.33	2.67	3.00
Yeast-heat	12.00	10.00	7.17	10.17	7.17	4.17	**1.00**	6.67	5.50	4.17	1.17	1.50
Yeast-spo	12.00	7.83	6.83	11.00	10.00	6.50	**1.00**	4.00	4.17	3.67	**1.00**	**1.00**
Yeast-spo5	12.00	6.83	11.00	9.33	8.00	6.83	2.00	2.50	5.17	**1.00**	2.00	1.83
Yeast-spoem	12.00	10.00	11.00	6.67	6.67	6.17	2.50	6.00	2.50	**1.00**	3.00	3.00
all	11.24	9.12	7.30	9.72	8.50	5.91	**2.11**	6.93	3.71	3.26	2.36	2.20

Table 5
Average ranking positions of the tested methods for each evaluation measure, across all 15 benchmarks. Best values for each measure are **bolded**, gray background denotes statistically significant differences in results between Duo-LDL and the respective method.

	PT-Bayes	PT-SVM	AA-kNN	AA-BP	AA-BP-T	IIS-LLD	BFGS-LLD	LALOT	GD-LDL-SCL	Adam-LDL-SCL	LDL-LCLR	Duo-LDL
Chebyshev	11.07	9.20	7.47	9.53	8.20	5.67	**1.87**	6.60	3.67	3.40	2.47	2.07
Clark	11.20	9.27	7.20	9.93	8.80	6.47	**2.20**	7.47	4.40	3.53	2.33	2.60
Canberra	11.40	9.40	7.40	10.07	8.80	6.33	2.87	7.20	4.47	3.33	2.73	**2.53**
Kullback–Leibler	11.33	8.53	7.33	9.80	8.53	4.73	**1.73**	6.53	3.33	3.60	2.13	2.00
Cosine	11.20	9.13	7.27	9.40	8.20	5.93	1.80	6.60	2.93	2.67	2.07	**1.73**
Intersection	11.27	9.20	7.13	9.60	8.47	6.33	**2.20**	7.20	3.47	3.00	2.40	2.27
all	11.24	9.12	7.30	9.72	8.50	5.91	**2.11**	6.93	3.71	3.26	2.36	2.20

value the better while for the others, conversely, smaller values are preferred. Combining them into a common cumulative assessment relied on normalization of their values to the unit interval. For each evaluation measure and each benchmark set the best value among all tested methods was mapped to 1 and the worst one to 0. All the others were mapped linearly between 0 and 1. This presentation perspective reveals that in most of the cases (especially for all *Yeast* data sets) differences among the seven rightmost algorithms are minor. This observation was also confirmed in already discussed tables, in which only a few

Table 6
Normalized results averaged over all evaluation measures. Each value belongs to the interval [0, 1]. The higher the value the better the assessment of the respective method. Best values for each data set are bolded.

	PT-Bayes	PT-SVM	AA-kNN	AA-BP	AA-BP-T	IIS-LLD	BFGS-LLD	LALOT	GD-LDL-SCL	Adam-LDL-SCL	LDL-LCLR	Duo-LDL
Movie	0.4702	0.5394	0.8151	0.8338	0.8668	0.9046	0.9349	0.0960	0.9270	0.9770	0.9766	**0.9909**
Human Gene	0.0000	0.9913	0.8925	0.7075	0.8370	0.9960	0.9987	0.9980	0.9991	0.9996	0.9995	**0.9998**
Natural Scene	0.0598	0.1229	0.6392	0.5321	0.5236	0.5156	0.6091	0.2634	0.6194	0.4822	0.6870	**0.8314**
SJAFFE	0.9849	0.9756	0.9935	0.9593	0.9621	0.9901	**0.9998**	0.8655	0.0000	0.0000	0.9924	0.9672
s-BU 3DFE	0.3464	0.2204	0.7520	0.3914	0.4993	0.6268	**0.9646**	0.0579	0.8626	0.6798	0.5400	0.8454
Yeast-alpha	0.0000	0.9707	0.9899	0.7773	0.8739	0.9890	0.9997	0.9931	0.9975	0.9978	0.9997	**1.0000**
Yeast-cdc	0.0000	0.9763	0.9884	0.8110	0.8912	0.9889	0.9993	0.9917	0.9988	**1.0000**	0.9994	0.9994
Yeast-cold	0.0000	0.9799	0.9889	0.8216	0.9018	0.9894	**1.0000**	0.9923	0.9986	0.9998	**1.0000**	0.9998
Yeast-diau	0.0000	0.9309	0.9763	0.9080	0.9426	0.9807	0.9918	0.9585	0.9981	**1.0000**	0.9915	0.9910
Yeast-dtt	0.0000	0.9754	0.9793	0.9267	0.9560	0.9913	**1.0000**	0.9884	0.9969	0.9995	0.9997	0.9995
Yeast-elu	0.0000	0.9461	0.9528	0.9205	0.9495	0.9860	0.9949	0.9799	**1.0000**	0.9993	0.9949	0.9947
Yeast-heat	0.0000	0.9605	0.9725	0.9602	0.9725	0.9862	**1.0000**	0.9744	0.9808	0.9869	0.9995	0.9992
Yeast-spo	0.0000	0.9799	0.9840	0.9582	0.9723	0.9847	**1.0000**	0.9917	0.9905	0.9925	**1.0000**	**1.0000**
Yeast-spo5	0.0000	0.9802	0.9551	0.9736	0.9778	0.9802	0.9923	0.9916	0.9875	**1.0000**	0.9923	0.9927
Yeast-spoem	0.0000	0.9405	0.9255	0.9575	0.9575	0.9585	0.9742	0.9628	0.9779	**1.0000**	0.9725	0.9725
Average	0.1241	0.8327	0.9203	0.8292	0.8723	0.9245	0.9639	0.8070	0.8890	0.8743	0.9430	**0.9722**

differences between Duo-LDL and these 6 methods were marked as statistically significant. The leading position in Table 6 was gained by Duo-LDL followed by BFGS-LLD and LDL-LCLR.

As mentioned in Section 4, apart from a novel idea of incorporating information related to pairwise inter-dependencies between the classes, Duo-LDL also benefits from application of certain state-of-the-art training techniques (weights decay, mini-batch gradient descent) which were not applied during the training process of the previous neural network based method — AA-BP. In order to assess the influence of these new training techniques on the Duo-LDL efficacy, the AA-BP-T method, i.e. AA-BP extended by the training techniques used in Duo-LDL training, was additionally evaluated. The results presented in Tables 4 and 6 show that these training enhancements definitely improved the AA-BP performance, however, the gap between AA-BP-T and the best approaches is still significant. Consequently, it can be concluded that application of the up-to-date training techniques is not the key aspect of Duo-LDL overall efficiency. The main strength of the proposed algorithm lays in adequately defined error function and design of the output layer, which effectively handles pairwise inter-dependencies between classes.

6.1. Detailed comparison of Duo-LDL and BFGS-LLD on artificially generated data sets

The results presented so-far show dominance of BFGS-LLD and Duo-LDL methods. In 8 cases BFGS-LLD obtained better results and for 7 remaining ones, Duo-LDL was superior. Since the methods employ very different solution techniques — quasi-Newton function optimization versus neural network training, further analysis of their operational differences, in particular their strengths and weaknesses seems to be worthwhile. In order to accomplish this task, a set of artificial benchmarks inspired by a design method proposed in [3] and [42] were created. In our experiments the baseline idea from the above-cited papers was extended to creation of various data sets with different numbers of labels, features and training examples, to allow for detailed evaluation of method-specific aspects. Artificial data sets were created based on the following set of equations.

$$\begin{aligned} t_i &= f_i + 0.5f_i^2 + 0.2f_i^3 + 1, i = 1, \ldots, n \\ \psi_1 &= (\mathbf{w}_1^T \mathbf{t})^2 \\ \psi_2 &= (\mathbf{w}_2^T \mathbf{t} + \lambda\phi_1)^2 \\ &\vdots \\ \psi_q &= (\mathbf{w}_q^T \mathbf{t} + \lambda\phi_{q-1})^2 \\ p_x^j &= \frac{\psi_j}{\sum_{k=1}^{q}\psi_k} j = 1, \ldots, q \end{aligned} \tag{8}$$

where $\mathbf{P_x} = \{p_x^1, \ldots, p_x^q\}$ is distribution of labels assigned to instance $\mathbf{x} = [f_1, \ldots, f_n]$, $\mathbf{t} = [t_1, \ldots, t_n]^T$, $\lambda = 0.001$ and $\mathbf{w}_j = [j \bmod q, (j+1) \bmod q, \ldots, (j+q-1) \bmod q]^T, j = 1, \ldots, q$. Each component of x was uniformly sampled within the range $[-1, 1]$. A total of 392 data sets and corresponding label distributions were generated according to (8), with the following parameters:

- $q \in \{2, 3, 5, 10, 15, 20, 30\}$,
- $n \in \{2, 3, 5, 10, 15, 20, 30\}$,
- $d \in \{50, 100, 200, 500, 1000, 2000, 5000, 10000\}$.

For each possible selection of (q, n, d) one benchmark was generated and the results of 10-fold cross validation were averaged. The outcomes for two evaluation measures (Chebyshev loss and Intersection loss) are presented in Figs. 2 and 3, respectively.

Analysis of results with respect to the number of labels (Figs. 2(a) and 3(a)) leads to a conclusion that BFGS-LLD is better suited for the tasks with smaller numbers of labels (below 10–15), while the advantage of Duo-LDL is visible for the sets with greater number of classes. Similar conclusions can be drawn in reference to the number of input features (Figs. 2(b) and 3(b)). Again BFGS-LLD outperforms Duo-LDL for data sets with smaller numbers of features (less than 10). Both algorithms naturally yield better results for greater numbers of features, however, the improvement is faster in the case of Duo-LDL. This observation explains Duo-LDL supremacy in *Movie* benchmark discussed in the previous section. Similarly, poor Duo-LDL performance for *SJAFFE* was presumably caused by small number of input instances (213 only). This hypothesis was confirmed in the experiments with artificially generated data. Figs. 2(c) and 3(c) show fast improvement of Duo-LDL results along with an increase of the number of training samples whereas this parameter (the number of instances) influences BFGS-LLD results to a little extent only. The efficacy of Duo-LDL grows fast with the increasing number of examples and above approximately 1000 of them exceeds BFGS-LLD.

In summary, a comparison of Duo-LDL and BFGS-LLD on artificially generated data sets demonstrates that for benchmarks with smaller number of labels, input features or examples BFGS-LLD is a superior approach whereas for bigger benchmark sets (in terms of the number of classes or features) the prevailing method is Duo-LDL.

6.2. Computation time

Besides efficacy another relevant aspect of designed algorithms is computation time. Duo-LDL being a neural-based method spends most of its computation time on training. Requests to

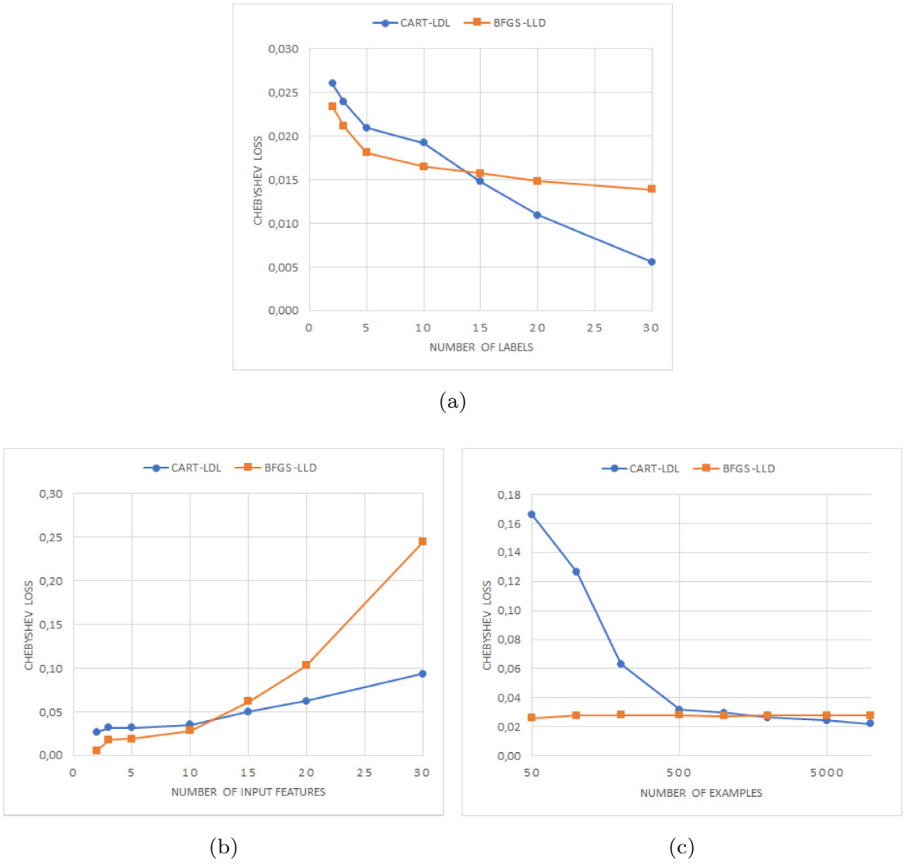

Fig. 2. Scalability results of BFGS-LLD and Duo-LDL results measured by **Chebyshev loss** with respect to the number of (a) labels, (b) features and (c) examples, respectively.

already trained network are responded immediately and testing phase takes less than 0.1% of the whole experimental time. While in many real-life applications the training time is not critical and the model response time is actually the most important, in order to make a fair comparison, for all tested methods computation times of the entire process (training and testing) are presented in Table 7. All experiments were run on Intel Core i7-7500U @ 2.70 GHz with 16 GB RAM. Clearly, all algorithms reported the longest times for *Movie* and *Human Gene* benchmarks which have the greatest numbers of input features and instances, respectively. Generally speaking, PT-SVM and LALOT are the slowest methods, whereas AA-kNN and Adam-LDL-SCL are the fastest ones.

Among the three best-performing methods (cf. Tables 4–6) computations times of BFGS-LLD and Duo-LDL are similar — the latter obtained better time results in 8 out of 15 benchmarks. The third method (LDL-LCLR) which demonstrated the highest results quality suffers from poor scalability. For small data sets its computation time is among the shortest ones, but for benchmarks with higher numbers of more features or classes its time performance significantly worsens.

Fig. 4 compares time scalability of BFGS-LLD and Duo-LDL. BFGS-LLD scales better with respect to the number of labels because in Duo-LDL the size of the output layer is proportional to the squared number of labels, so increasing the number of them has a clear impact on the network complexity (the number of output nodes and number of connections). On the contrary, in BFGS-LLD increasing the number of labels only affects the size of an auxiliary matrix used for computing the search direction and step. On the other hand, with respect to the number of features time scalability of Duo-LDL outperforms BFGS-LLD, since in Duo-LDL adding a new feature results in addition of one input node only (with outgoing connections), whereas in BFGS-LLD it impacts the most time-consuming part of the algorithm — inverse Hessian matrix approximation. Both methods scale approximately linearly concerning the number of examples (please note a logarithmic scale in Fig. 4(c)).

7. Practical relevance and possible applications

LDL problems arise in various domains including biology [3], sociology [43,44], security [6], image processing [33,45,46] or multimedia [5]. It appears, for instance, in gene expression domain [3], e.g. detection to which functions the genes correspond or which disease they may cause. Another application area is recognition of emotions based on face expression images [43,47]. Since human face often reflects a mixture of several emotions to different degrees (e.g. anger and sadness), LDL is perfectly suited to tackle this task. LDL is also well-fitted to the task of recognition of objects in the image. Classical approaches provide binary information about the existence of an object in the picture. However, in certain real-life applications the task is not only to decide whether or not the object appears in the picture, but to assess how important it is from the point of view of the image composition or a topic presented in that image [33], or to which extent an object dominates the picture (e.g. in size or quantity) [46]. Please refer to Fig. 5 as an example.

Likewise, certain types of multiclass predictions can be modeled as LDL problem and solved by dedicated methods. One example is a prediction of people's opinion about the movie

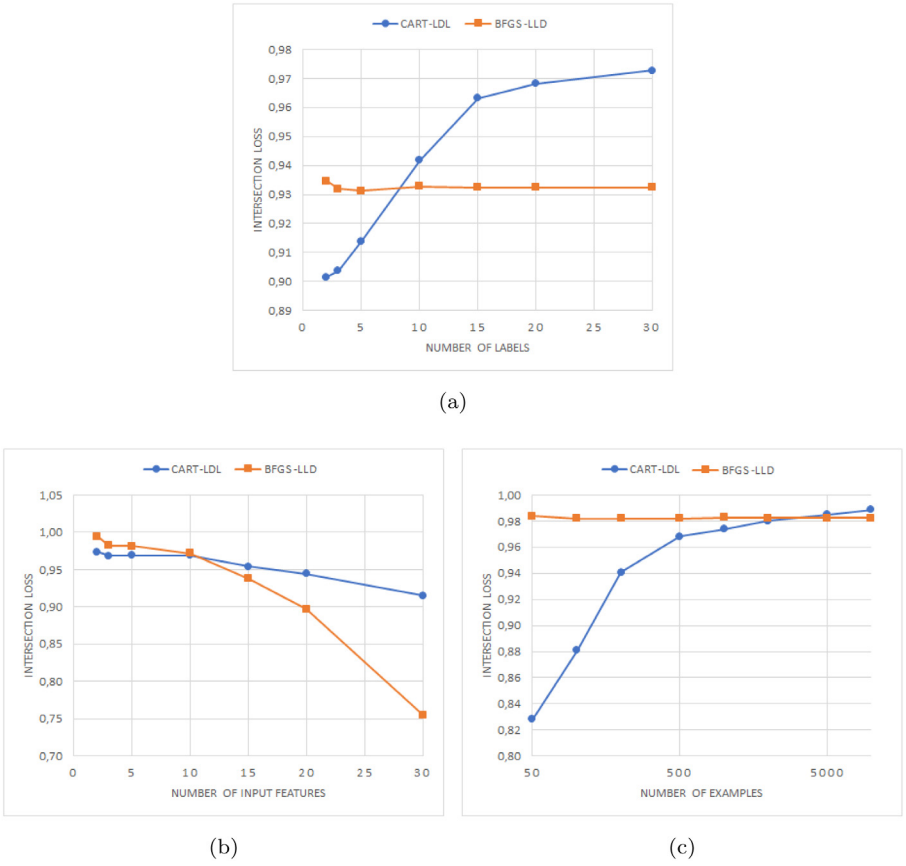

Fig. 3. Scalability results of BFGS-LLD and Duo-LDL measured by **Intersection loss** with respect to the number of (a) labels, (b) features and (c) examples, respectively.

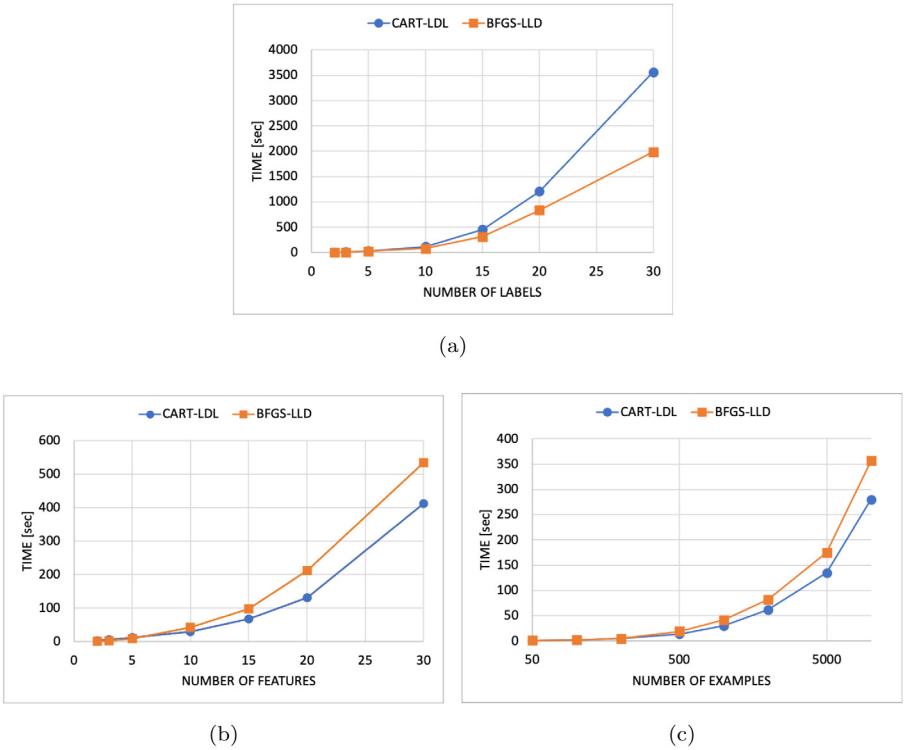

Fig. 4. Time scalability of BFGS-LLD and Duo-LDL with respect to the number of (a) labels, (b) features and (c) examples, respectively.

Table 7
Computation time (in seconds) over all tested benchmarks.

	PT-Bayes	PT-SVM	AA-kNN	AA-BP	AA-BP-T	IIS-LLD	BFGS-LLD	LALOT	GD-LDL-SCL	Adam-LDL-SCL	LDL-LCLR	Duo-LDL
Movie	215	68177	431	727	491	377	1801	33746	3642	973	19539	8752
Human Gene	102	63994	1315	539	352	573	2276	15076	184	51	13529	1429
Natural Scene	24	29856	33	79	43	123	2047	4591	64	17	9415	91
SJAFFE	29	8714	41	89	60	83	370	3487	55	5	534	71
s-BU 3DFE	2	1907	3	20	22	38	85	261	3	1	615	5
Yeast-alpha	28	3425	34	84	59	46	42	1115	153	8	150	177
Yeast-cdc	29	2614	36	68	55	39	37	1101	121	26	3701	116
Yeast-cold	28	163	36	61	31	33	34	533	16	3	14	15
Yeast-diau	29	472	34	63	48	35	40	523	48	4	34	28
Yeast-dtt	28	145	36	61	37	34	32	405	2	2	16	15
Yeast-elu	29	2293	37	67	39	38	35	900	120	11	3353	104
Yeast-heat	28	364	36	62	34	34	34	709	55	13	24	24
Yeast-spo	28	304	36	62	34	33	40	679	54	13	24	23
Yeast-spo5	29	79	36	60	46	32	31	490	35	2	13	13
Yeast-spoem	28	45	36	60	31	31	30	469	24	4	11	11

Fig. 5. In LDL problem formulation, not only information about the existence of objects in the image but also its importance for the entire image content can be considered. For example, in the above picture a dog in foreground is more relevant than people and trees in the background, or a bottle lying in the grass. Classical multi-label approach cannot grade objects and consider this kind of contextual assessment.

before its premiere [5]. The movie can be annotated by its genre, keywords, actors playing the main roles, a director, etc. A typical approach is to predict one particular movie-related aspect, e.g. the average rating. In the case of LDL formulation of this task, a distribution of ratings can be modeled. Other real-life applications of LDL include head pose estimation [45,48], facial landmark detection [49], crowd counting in public video surveillance [6] or facial age estimation [44,50]. Furthermore, multilabel classification tasks can be considered as special cases of LDL, so all kinds of problems defined as multilabel learning tasks can, in principle, be transformed to a more general LDL framework. All the above examples show that the proposed Duo-LDL algorithm can be applied to various real-world scenarios.

For the sake of comparability with other methods, the evaluation of Duo-LDL was performed on standard benchmark sets that are widely used in the LDL domain. However, please note that all of 15 benchmark sets used in the evaluation process are related to practical problems originated from real-world demands, e.g. movies' rating distribution [5], gene expressions for diseases [32], landscape images classification [33], human faces emotion recognition [34,35] (2 data sets), or phylogenetic profile of yeasts genes [36] (10 data sets).

8. Conclusions

Label Distribution Learning is a relatively new type of classification problems with straightforward applications in various real-life domains. To the best of our knowledge, among LDL solution methods proposed in the literature, there has been only one approach (AA-BP [3]) that employs neural networks. In this paper we propose another neural network solution to LDL (Duo-LDL) which extends a straightforward approach presented in [3] by incorporating the information about pairwise inter-class dependencies into the network training process by means of adequate design of the output layer and specific form of the error function used during training.

Duo-LDL is first evaluated on a set of 15 well-established benchmarks and 6 error measures proving its advantage over AA-BP in all cases, in the majority of them the improvement is statistically significant. Furthermore, the results confirm that Duo-LDL performance is comparable to the stat-of-the-art approaches to LDL. The method is superior to all but two competitive approaches, and is on par with the best overall (non-neural) LDL algorithms (BFGS-LLD and LDL-LCLR). Even though in a head-to-head comparison BFGS-LLD achieves in average slightly better results the advantage is not meaningful in statistical sense.

An in-depth comparison of both methods on a wide selection of artificially generated data sets revealed that Duo-LDL is especially strong in the case of the most challenging benchmarks, with extensive feature representation in the input and/or numerous classes in the output. This property gives a promise for successful application of the method to large and demanding data sets frequently appearing in various real-life domains.

In the paper we have focused on the LDL formulation as a supervised classification problem. There are, however, also problem domains in which full label-related information is not available, for instance, either a set of possible classes or per sample class distribution is revealed only partially. For such imperfect-information formulations of LDL problems one can employ semi-supervised learning approaches, e.g Label Propagation [51] or Gaussian Mixture Models [52]. We consider this topic as an interesting future work.

CRediT authorship contribution statement

Adam Żychowski: Conceptualization, Software, Investigation, Writing - original draft. **Jacek Mańdziuk:** Conceptualization, Writing - review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, Int. J. Data Warehous. Min. 3 (3) (2007) 1–13.

[2] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Džeroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit. 45 (9) (2012) 3084–3104.

[3] X. Geng, Label distribution learning, IEEE Trans. Knowl. Data Eng. 28 (7) (2016) 1734–1748.

[4] X. Geng, C. Yin, Z.-H. Zhou, Facial age estimation by learning from label distributions, IEEE Trans. Pattern Anal. Mach. Intell. 35 (10) (2013) 2401–2412.

[5] X. Geng, P. Hou, Pre-release prediction of crowd opinion on movies by label distribution learning, in: IJCAI, 2015, pp. 3511–3517.

[6] Z. Zhang, M. Wang, X. Geng, Crowd counting in public video surveillance by label distribution learning, Neurocomputing 166 (2015) 151–163.

[7] J. Fürnkranz, E. Hüllermeier, E.L. Mencía, K. Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2) (2008) 133–153.

[8] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, H.-J. Zhang, Correlative multi-label video annotation, in: Proceedings of the 15th ACM International Conference on Multimedia, 2007, pp. 17–26.

[9] M.-L. Zhang, K. Zhang, Multi-label learning by exploiting label dependency, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 999–1008.

[10] J. Wang, X. Geng, Theoretical analysis of label distribution learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 5256–5263.

[11] Y.-P. Liu, N. Xu, Y. Zhang, X. Geng, Label distribution for learning with noisy labels, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, 2020.

[12] S. Xu, H. Ju, L. Shang, W. Pedrycz, X. Yang, C. Li, Label distribution learning: A local collaborative mechanism, Internat. J. Approx. Reason. 121 (2020) 59–84.

[13] T.-F. Wu, C.-J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (Aug) (2004) 975–1005.

[14] M.-L. Zhang, Z.-H. Zhou, A k-nearest neighbor based algorithm for multi-label classification, in: Granular Computing, 2005 IEEE International Conference on, vol. 2, IEEE, 2005, pp. 718–721.

[15] P. Zhao, Z.-H. Zhou, Label distribution learning by optimal transport, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[16] C. Villani, Optimal Transport: Old and New, Vol. 338, Springer Science & Business Media, 2008.

[17] A.L. Berger, V.J.D. Pietra, S.A.D. Pietra, A maximum entropy approach to natural language processing, Comput. Linguist. 22 (1) (1996) 39–71.

[18] S. Della Pietra, V. Della Pietra, J. Lafferty, Inducing features of random fields, 1995, arXiv preprint cmp-lg/9506014.

[19] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, 2013.

[20] X. Zheng, X. Jia, W. Li, Label distribution learning by exploiting sample correlations locally, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[21] X. Jia, Z. Li, X. Zheng, W. Li, S.-J. Huang, Label distribution learning with label correlations on local samples, IEEE Trans. Knowl. Data Eng. (2019).

[22] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[23] T. Ren, X. Jia, W. Li, S. Zhao, Label distribution learning with label correlations via low-rank approximation, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 3325–3331, http://dx.doi.org/10.24963/ijcai.2019/461.

[24] M.-L. Zhang, Z.-H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351.

[25] R. Grodzicki, J. Mańdziuk, L. Wang, Improved multilabel classification with neural networks, in: Parallel Problem Solving from Nature, in: Lecture Notes in Computer Science, vol. 5199, Springer Verlag, 2008, pp. 409–416.

[26] J. Mańdziuk, A. Żychowski, L. Wang, A TCART-M—Tuned CARTesian-based error function for multilabel classification with the MLP, in: 2017 International Joint Conference on Neural Networks, IJCNN 2017, IEEE, 2017, pp. 565–572.

[27] J. Mańdziuk, A. Żychowski, Dimensionality reduction in multilabel classification with neural networks, in: 2019 International Joint Conference on Neural Networks, IJCNN 2019, IEEE, 2019, pp. 1–8.

[28] Y. Yu, W. Pedrycz, D. Miao, Multi-label classification by exploiting label correlations, Expert Syst. Appl. 41 (6) (2014) 2989–3004.

[29] S.-J. Huang, Z.-H. Zhou, Multi-label learning by exploiting label correlations locally, in: Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.

[30] E.L. Mencía, S.-H. Park, J. Fürnkranz, Efficient voting prediction for pairwise multilabel classification, Neurocomputing 73 (7–9) (2010) 1164–1176.

[31] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[32] J.-F. Yu, D.-K. Jiang, K. Xiao, Y. Jin, J.-H. Wang, X. Sun, Discriminate the falsely predicted protein-coding genes in Aeropyrum Pernix K1 genome based on graphical representation, MATCH Commun. Math. Comput. Chem. 67 (3) (2012) 845.

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

[34] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, in: Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference, IEEE, 1998, pp. 200–205.

[35] L. Yin, X. Wei, Y. Sun, J. Wang, M.J. Rosato, A 3D facial expression database for facial behavior research, in: Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference, IEEE, 2006, pp. 211–216.

[36] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. 95 (25) (1998) 14863–14868.

[37] X. Geng, Label distribution learning, URL http://cse.seu.edu.cn/PersonalPage/xgeng/LDL.

[38] S.-H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, Int. J. Math. Models Methods Appl. Sci. 1 (4) (2007) 300–307.

[39] A. Żychowski, Duo-LDL Method - source code, 2020, URL https://bitbucket.org/adam24/neuralnetworkldl.

[40] A. Żychowski, J. Mańdziuk, Label Distribution Learning with neural networks. Detailed experimental results of Duo-LDL method, URL http://www.mini.pw.edu.pl/~mandziuk/ldl/detailed_results.pdf.

[41] S.J. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: Recommendations for practitioners, IEEE Trans. Pattern Anal. Mach. Intell. (3) (1991) 252–264.

[42] N. Xu, A. Tao, X. Geng, Label enhancement for label distribution learning, in: IJCAI, 2018, pp. 2926–2932.

[43] Y. Zhou, H. Xue, X. Geng, Emotion distribution recognition from facial expressions, in: Proceedings of the 23rd ACM International Conference on Multimedia, ACM, 2015, pp. 1247–1250.

[44] X. Geng, Q. Wang, Y. Xia, Facial age estimation by adaptive label distribution learning, in: Pattern Recognition, ICPR, 2014 22nd International Conference, IEEE, 2014, pp. 4465–4470.

[45] X. Geng, Y. Xia, Head pose estimation based on multivariate label distribution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1837–1842.

[46] Y.-K. Li, M.-L. Zhang, X. Geng, Leveraging implicit relative labeling-importance information for effective multi-label learning, in: IEEE International Conference on Data Mining, ICDM, IEEE, 2015, pp. 251–260.

[47] X. Haitao, H. Liu, B. Zhong, Y. Fu, Structured and sparse annotations for image emotion distribution learning, in: AAAI, 2019.

[48] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, X. Geng, Deep label distribution learning with label ambiguity, IEEE Trans. Image Process. 26 (6) (2017) 2825–2838.

[49] K. Su, X. Geng, Soft facial landmark detection by label distribution learning, in: AAAI, 2019.

[50] Z. He, X. Li, Z. Zhang, F. Wu, X. Geng, Y. Zhang, M.-H. Yang, Y. Zhuang, Data-dependent label distribution learning for age estimation, IEEE Trans. Image Process. 26 (8) (2017) 3846–3858.

[51] X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, Comput. Sci. (2002).

[52] P. Hou, X. Geng, Z.-W. Huo, J. Lv, Semi-supervised adaptive label distribution learning for facial age estimation, in: AAAI, 2017, pp. 2015–2021.