Abstract

Noisy labeled data represent a rich source of information that often are easily accessible and cheap to obtain, but label noise might also have many negative consequences if not accounted for. How to fully utilize noisy labels has been studied extensively within the framework of standard supervised machine learning over a period of several decades. However, very little research has been conducted on solving the challenge posed by noisy labels in non-standard settings. This includes situations where only a fraction of the samples are labeled (semi-supervised) and each high-dimensional sample is associated with multiple labels. In this work, we present a novel semi-supervised and multi-label dimensionality reduction method that effectively utilizes information from both noisy multi-labels and unlabeled data. With the proposed Noisy multi-label semi-supervised dimensionality reduction (NMLS) algorithm, the noisy multi-labels are denoised and unlabeled data are labeled simultaneously via a specially designed label propagation algorithm. NMLS then learns a projection matrix for reducing the dimensionality by maximizing the dependence between the enlarged and denoised multi-label space and the features in the projected space. Extensive experiments on synthetic data, as well as benchmark datasets, demonstrate the effectiveness of the proposed algorithm and show that it outperforms state-of-the-art multi-label feature extraction algorithms. Finally, we illustrate the benefits of the proposed method in a realistic healthcare case study, achieving statistically significant gains compared to the previous state-of-the-art on the problem of identifying patients suffering from multiple chronic diseases.

Keywords: Noisy labels, Multi-label learning, Semi-supervised learning, Dimensionality reduction, Healthcare case study

1. Introduction

Supervised machine learning crucially relies on the accuracy of the observed labels associated with the training samples [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Observed labels may be corrupted and, therefore, they do not necessarily coincide with the true class of the samples. Such inaccurate labels are also referred to as noisy [2, 11, 4]. Label noise can occur because of imperfect evidence or fatigue on the part of the labeler, e.g. in healthcare where a medical doctor may be annotating or labeling thousands of patients manually, potentially making mistakes in the process [12, 5]. In other cases, noisy labels may result from the use of frameworks such as anchor learning [13, 14] or silver standard learning [15], which have received interest for instance in healthcare analytics [16, 17]. A review of various sources of label noise can be found in [2].

In standard supervised machine learning settings, the challenge posed by noisy labels has been studied extensively. For example, many noise-tolerant versions of well-known classifiers have been proposed, including discriminant analysis [8, 18], logistic regression [9], the k-nearest neighbor classifier [19], boosting algorithms [20, 21], perceptrons [22, 23], support vector machines [24], deep neural networks [7, 25, 26]. Others have proposed more general classification frameworks that are not restricted to particular classifiers [4, 11].

However, very little research has been conducted on solving the challenge posed by noisy labels in non-standard settings, where the magnitude of the noisy label problem is increased considerably. One good example (among many) of such a non-standard setting occurs for instance within the healthcare domain, used here as an illustrative case-study. Non-standard settings include (i) Semi-supervised learning [27], referring to a situation where only a few (noisy) labeled data points are available, making the impact of noise in those few labels more prevalent, and where information must also jointly be inferred from unlabeled data points. In healthcare, it may be realistic to obtain some labels through a (imperfect) manual labeling process, but the vast amount of data remains unlabeled; (ii) Multi-label learning, where objects may not belong exclusively to one category. This situation occurs frequently in a number of domains, including healthcare, where for instance a patient could suffer from multiple chronic diseases; (iii) High-dimensional data, where the abundance of features and the limited (noisy) labeled data, lead to a curse of dimensionality problem. In such situations, dimensionality reduction (DR) [28] is useful, either as a pre-processing step, or as an integral part of the learning procedure. This is a well-known challenge in health, where the
number of patients in the populations under study frequently is small, but heterogeneous potential sources of data features from electronic health records for each patient may be enormous [29, 30, 31, 32]. In this paper, and to the best of our knowledge, we propose the first noisy label, semi-supervised and multi-label DR machine learning method, which we call the Noisy multi-label semi-supervised dimensionality reduction (NMLSDR) method. Towards that end, we propose a label propagation method that can deal with noisy multi-label data. Label propagation [33, 34, 35, 36, 37, 38, 39], wherein one propagates the labels to the unlabeled data in order to obtain a fully labeled dataset, is one of the most successful and fundamental frameworks within semi-supervised learning. However, in contrast to many of these methods that clamp the labeled data, in our multi-label propagation method we allow the labeled part of the data to change labels during the propagation to account for noisy labels. In the second part of our algorithm we aim at learning a lower dimensional representation of the data by maximizing the feature-label dependence. Towards that end, similarly to other DR methods [40, 41], we employ the Hilbert-Schmidt independence criterion (HSIC) [42], which is a non-parametric measure of dependence.

The NMLSDR method is a DR method, which is general and can be used in many different settings, e.g. for visualization or as a pre-processing step before doing classification. However, in order to test the quality of the NMLSDR embeddings, we (preferably) have to use some quantitative measures. For this purpose, a common baseline classifier such as the multi-label k-nearest neighbor (ML-kNN) classifier [43] has been applied to the low-dimensional representations of the data [44, 45]. Even though this is a valid way to measure the quality of the embeddings, to apply a supervised classifier in a semi-supervised learning setting is not a realistic setup since one suddenly assumes that all labels are known (and correct). Therefore, as an additional contribution, we introduce a novel framework for semi-supervised classification of noisy multi-label data. In our experiments, we compare NMLSDR to baseline methods on synthetic data, benchmark datasets, as well as a real-world case study where the proposed framework is used to identify the health status of patients with multiple chronic diseases.

The remainder of the paper is organized as follows. Related work is reviewed in Sec. 2. In Sec. 3, we describe our proposed NMLSDR method and the novel framework for semi-supervised classification of noisy multi-label data. Sec. 4 describes experiments on synthetic and benchmark datasets, whereas Sec. 5 is devoted to the case study where we study chronically ill patients. We conclude the paper in Sec. 6.

2. Related work

In this section we review related unsupervised, semi-supervised and supervised DR methods.1

Unsupervised DR methods do not exploit label information and can therefore straightforwardly be applied to multi-label data by simply ignoring the labels. For example, principal component analysis (PCA) aims to find the projection such that the variance of the input space is maximally preserved [47]. Other methods aim to find a lower dimensional embedding that preserves the manifold structure of the data, and examples of these include Locally linear embedding [48], Laplacian eigenmaps [49] and ISOMAP [50].

One of the most well-known supervised DR methods is linear discriminative analysis (LDA) [51], which aims at finding the linear projection that maximizes the within-class similarity and at the same time minimizes the between-class similarity. LDA has been extended to multi-label LDA (MLDA) in several different ways [52, 53, 54, 55, 56]. The difference between these methods basically consists in the way the labels are weighted in the algorithm. Following the notation in [56], wMLDAb [52] uses binary weights, wMLDAe [53] uses entropy-based weights, wMLDAC [54] uses correlation-based weights, wMLDAf [55] uses fuzzy-based weights, whereas wMLDA [56] uses dependence-based weights.

Canonical correlation analysis (CCA) [57] is a method that maximizes the linear correlation between two sets of variables, which in the case of DR are the set of labels and the set of features derived from the projected space. CCA can be directly applied also for multi-labels without any modifications. Multi-label informed latent semantic indexing (MLSI) [58] is a DR method that aims at both preserving the information of inputs and capturing the correlations between the labels. In the Multi-label least square (ML-LS) method one extracts a common subspace that is assumed to be shared among multiple labels by solving a generalized eigenvalue decomposition problem [59].

In [40], a supervised method for doing DR based on dependence maximization [42] called Multi-label dimensionality reduction via dependence maximization (MDDM) was introduced. MDDM attempts to maximize the feature-label dependence using the Hilbert-Schmidt independence criterion and was originally formulated in two different ways. MDDM is

---

1DR may be obtained both by feature extraction, i.e. by a data transformation, and by feature selection [46]. Here, we refer to DR in the sense of feature extraction.
based on orthonormal projection directions, whereas MDDMF makes the projected features orthonormal. Yu et al. showed that MDDMP can be formulated using least squares and added a PCA term to the cost function in a new method called Multi-label feature extraction via maximizing feature variance and feature-label dependence simultaneously (MVMD) [41].

The most closely related existing DR methods to NMLSDR are the semi-supervised multi-label methods. The Semi-supervised dimension reduction for multi-label classification method (SSD-MC) [60], Coupled dimensionality reduction and classification for supervised and semi-supervised multi-label learning [61], and Semisupervised multilabel learning with joint dimensionality reduction [62] are semi-supervised multi-label methods that simultaneously learn a classifier and a low dimensional embedding.

Other semi-supervised multi-label DR methods are semi-supervised formulations of MLDA have also been proposed. Multi-label dimensionality reduction based on semi-supervised discriminant analysis (MSDA) [65], which later was modified and renamed Semi-supervised multi-label discriminant analysis (SMMLDA) [65], which later was modified and renamed Semi-supervised multi-label dimensionality reduction based on dependence maximization (SMMDm) [45]. In SMDRm the scatter matrices are computed based on only labeled data. However, a HSIC term is also added to the familiar Rayleigh quotient containing the two scatter matrices, which is computed based on soft labels for both labeled and unlabeled data obtained in a similar way as in SMMLDR.

Common to all these methods is that none of them explicitly assume that the labels can be noisy. In SSMLDR and SMMDm, the labeled data are clamped during the label propagation and hence cannot change. Moreover, these two methods are both based on LDA, which is known heavily affected by outliers, and consequently also wrongly labeled data [66, 67, 68].

3. The NMLSDR method

We start this section by introducing notation and the setting for noisy multi-label semi-supervised linear feature extraction, and thereafter elaborate on our proposed NMLSDR method.

3.1. Problem statement

Let \( \{x_i\}_{i=1}^{n} \) be a set of \( n \) \( D \)-dimensional data points, \( x_i \in \mathbb{R}^D \). Assume that the data are ordered such that the first of the data points are labeled and \( u \) are unlabeled, \( l + u = n \). Let \( X \) be an \( n \times d \) matrix with the data points as row vectors. Assume that the number of classes is \( C \) and let \( Y^l_i \in \{ 0,1 \}^C \) be the label-vector of data point \( x_i \), \( i = 1, \ldots, l \). The elements are given by \( Y^l_i = 1 \), \( c = 1, \ldots, C \) if data point \( x_i \) belongs to the \( c \)-th class and \( Y^l_i = 0 \) otherwise. Define the label matrix \( Y^L \in \{ 0,1 \}^{l \times C} \) as the matrix with the known label-vectors \( Y^l_i \), \( i = 1, \ldots, l \) as row vectors and let \( Y^U \in \{ 0,1 \}^{u \times C} \) be the corresponding label matrix of the unknown labels.

The objective of linear feature extraction is to learn a projection matrix \( P \in \mathbb{R}^{D \times d} \) that maps a data point in the original feature space \( x \in \mathbb{R}^D \) to a lower dimensional representation \( z \in \mathbb{R}^d \):

\[
z = P^T x,
\]

where \( d < D \) and \( P^T \) denotes the transpose of the matrix \( P \).

In our setting, we assume that the label matrix \( Y^L \) is potentially noisy and that \( Y^U \) is unknown. The first part of our proposed NMLSDR method consists of doing label propagation in order to learn the labels \( Y^U \) and update the estimate of \( Y^L \). We do this by introducing soft labels \( F \in \mathbb{R}^{u \times C} \) for the label matrix \( Y = \begin{pmatrix} Y^L \\ Y^U \end{pmatrix} \), where \( F_{ic} \) represents the probability that data point \( x_i \) belong to the \( c \)-th class. We obtain \( F \) with label propagation and thereafter use \( F \) to learn the projection matrix \( P \). However, we start by explaining our label propagation method.

3.2. Label propagation using a neighborhood graph

The underlying idea of label propagation is that similar data points should have similar labels. Typically, the labels are propagated using a neighborhood graph [33]. Here, inspired by [69], we formulate a label propagation method for multi-labels that is robust to noise. The method is as follows.

Step 1. First, a neighborhood graph is constructed. The graph is described by its adjacency matrix \( W \), which can be designed e.g. by setting the entries to

\[
W_{ij} = \exp(-\sigma^{-2}||x_i - x_j||^2),
\]

where \( ||x_i - x_j|| \) is the Euclidean distance between the datapoints \( x_i \) and \( x_j \), and \( \sigma \) is a hyperparameter. Alternatively, one can use the Euclidean distance to compute a k-nearest neighbors (kNN) graph where the entries of \( W \) are given by

\[
W_{ij} = \begin{cases} 1, & \text{if } x_i \text{ among } x_j \text{'s } k \text{NN or } x_j \text{ among } x_i \text{'s } k \text{NN} \\ 0, & \text{otherwise}. \end{cases}
\]

Step 2. Symmetrically normalize the adjacency matrix \( W \) by letting

\[
\tilde{W} = D^{-1/2}WD^{-1/2},
\]

where \( D \) is a diagonal matrix with entries given by \( d_{ii} = \sum_{j=1}^{n} W_{ij} \).

Step 3. Calculate the stochastic matrix

\[
T = D^{-1/2} \tilde{W} D^{-1/2},
\]

Step 4. Compute soft labels \( F \in \mathbb{R}^{u \times C} \) by iteratively using the following update rule

\[
F(i+1) = I_yTF(i) + (I - I_y)Y,
\]
where $L$ is a $n \times n$ diagonal matrix with the hyperparameters $\alpha_i$, $0 \leq \alpha_i < 1$, on the diagonal. To initialize $F$, we let $F(0) = Y$, where the unlabeled data are set to $Y_{ic} = 0$, $c = 1, \ldots, C$.

### 3.2.1. Discussion

Setting $\alpha_i = 0$ for the labeled part of the data corresponds to clamping of the labels. However, this is not what we aim for in the presence of noisy labels. Therefore, a crucial property of the proposed framework is to set $\alpha_i > 0$ such that the labeled data can change labels during the propagation.

Moreover, we note that our extension of label propagation to multi-labels is very similar to the single-label variant introduced in [69], with the exception that we do not add the outlier class, which is not needed in our case. In other extensions to the multi-label label propagation [44, 45], the label matrix $Y$ is normalized such that the rows sum to 1, which ensures that the output of the algorithm $F$ also has rows that sum to 1. In the single-label case this makes sense in order to maintain the interpretability of probabilities. However, in the multi-label case the data points do not necessarily exclusively belong to a single class. Hence, the requirement $\sum_i F_{ic} = 1$ does not make sense since then $x_i$ can maximally belong to one class if one think of $F$ as a probability and require the probability to be 0.5 or higher in order to belong to a class.

On the other hand, in our case, a simple calculation shows that $0 \leq F_{ic}(t + 1) \leq 1$:

\[
F_{ic}(t + 1) = \alpha_i \sum_{m=1}^{n} T_{im} F_{mc}(t) + (1 - \alpha_i) Y_{ic} \\
\leq \alpha_i \sum_{m=1}^{n} T_{im} + (1 - \alpha_i) = \alpha_i + (1 - \alpha_i) = 1, \tag{7}
\]

since $F_{ic}(t) \leq 1$ and $Y_{ic} \leq 1$. However, we do not necessarily have that $\sum_i F_{ic} = 1$.

From matrix theory it is known that, given that $I - L_T T$ is nonsingular, the solution of the linear iterative process (6) converges to the solution of

\[
(I - L_T T) F = (I - L_T) Y, \tag{8}
\]

for any initialization $F(0)$ if and only if $L_T T$ is a convergent matrix [70] (spectral radius $\rho(I_T T) < 1$). $I_T T$ is obviously convergent if $0 \leq \alpha_i < 1 \forall i$. Hence, we can find the soft labels $F$ by solving the linear system given by Eq. (8).

Moreover, $F_{ic}$ can be interpreted as the probability that data-point $x_i$ belongs to class $c$, and therefore, if one is interested in hard label assignments, $\hat{Y}$, these can be found by letting $\hat{Y}_{ic} = 1$ if $F_{ic} > 0.5$ and $\hat{Y}_{ic} = 0$ otherwise.

### 3.3. Dimensionality reduction via dependence maximization

In this section we explain how we use the labels obtained using label propagation to learn the projection matrix $P$.

The motivation behind dependence maximization is that there should be a relation between the features and the label of an object. This should be the case also in the projected space. Hence, one should try to maximize the dependence between the feature similarity in the projected space and the label similarity. A common measure of such dependence is the Hilbert-Schmidt independence criterion (HSIC) [42], defined by

\[
HSIC(X, Y) = \frac{1}{(n - 1)^2} \text{tr}(KHLH), \tag{9}
\]

where $tr$ denotes the trace of a matrix. $H \in \mathbb{R}^{n \times n}$ is given by $H_{ij} = \delta_{ij} - n^{-1}$, where $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. $K$ is a kernel matrix over the feature space, whereas $L$ is a kernel computed over the label space.

Let the projection of $x$ be given by the projection matrix $P \in \mathbb{R}^{D \times d}$ and function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $\Phi(x) = P^T x$. We select a linear kernel over the feature space, and therefore the kernel function is given by

\[
\mathcal{K}(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = \langle P^T x_i, P^T x_j \rangle = P^T x_i P^T x_j \tag{10}
\]

Hence, given data $\{x_i\}_{i=1}^n$, the kernel matrix can be approximated by $K = X P^T X P^T$.

The kernel over the label space, $L$, is given via the labels $y_{i} \in \{0, 1\}^C$. One possible such kernel is the linear kernel

\[
L(y_i, y_j) = (y_i, y_j). \tag{11}
\]

However, in our semi-supervised setting, some of the labels are unknown and some are noisy. Hence, the kernel $L$ cannot be computed. In order to enable DR in our non-standard problem, we propose to estimate the kernel using the labels obtained via our label propagation method. For the part of the data that was labeled from the beginning we use the hard labels, $\hat{Y}$, obtained from the label propagation, whereas for the unlabeled part we use the soft labels, $\hat{F}$. Hence, the kernel is approximated via $L = \hat{F} \hat{F}^T$, where $\hat{F} = \left[ \hat{F}_{U} \hat{F}_{L} \right]$. The reason for using the hard labels obtained from label propagation for the labeled part is that we want some degree of certainty for those labels that change during the propagation (if the soft label $\hat{F}_{L}$ changes with less than 0.5 from its initial value 0 or 1 during the propagation, the hard label $Y_{iU}$ does not change).

The constant term, $(n-1)^{-2}$, in Eq. (9) is irrelevant in an optimization setting. Hence, by inserting the estimates of the kernels into Eq. (9), the following objective function is obtained,

\[
\Psi(P) = \text{tr}(H XP^T P^T H \hat{F} \hat{F}^T) = \text{tr}(P^T X^T H \hat{F} \hat{F}^T HXP). \tag{12}
\]

Note that the matrix $X^T H \hat{F} \hat{F}^T HX$ is symmetric. Hence, by requiring that the projection directions are orthogonal and that the new dimensionality is $d$, the following optimization problem is obtained

\[
\arg \max_P \Psi(P) = \max_P \text{tr}(P^T (X^T H \hat{F} \hat{F}^T HX) P), \tag{13}
\]

s.t. $P \in \mathbb{R}^{D \times d}$, $P^T P = I$.

As a consequence of the Courant-Fisher characterization [71], it follows that the maximum is achieved when $P$ is an orthonormal basis corresponding to the $d$ largest eigenvalues. Hence, $P$ can be found by solving the eigenvalue problem

\[
X^T H \hat{F} \hat{F}^T HXP = \lambda P. \tag{14}
\]
The dimensionality of the projected space, \( d \), is upper bounded by the rank of \( FF^T \), which in turn is upper bounded by the number of classes \( C \). Hence, \( d \) cannot be set larger than \( C \).

### 3.4. Semi-supervised classification for noisy multi-label data

The multi-label k-nearest neighbor (ML-kNN) classifier [43] is a widely adopted classifier for multi-label classification. However, similarly to many other classifiers, its performance can be hampered if the dimensionality of the data is too high. Moreover, the ML-kNN classifier only works in a completely supervised setting. To resolve these problems, as an additional contribution of this work, we introduce a novel framework for semi-supervised classification of noisy multi-label data, consisting of two steps. In the first step, we compute a low dimensional embedding using NMLSDR. The second step consists of applying a semi-supervised ML-kNN classifier. For this classifier we use our label propagation method on the learned embedding to obtain a fully labeled dataset, and thereafter apply the ML-kNN classifier.

### 4. Experiments

In this paper, we have proposed a method for computing a low-dimensional embedding of noisy, partially labeled multi-label data. However, it is not a straightforward task to measure how well the method works. Even though the method is definitely relevant to real-world problems (illustrated in the case study in Sec. 5), the framework cannot be directly applied to most multi-label benchmark datasets since most of them are completely labeled, and the labels are assumed to be clean. Moreover, the NMLSDR provides a low dimensional embedding of the data, and we need a way to measure how good the embedding is. If the dimensionality is 2 or 3, this can to some degree be done visually by plotting the embedding. However, in order to quantitatively measure the quality and simultaneously maintain a realistic setup, we will apply our proposed end-to-end framework for semi-supervised classification and dimensionality reduction. In our experiments, this realistic semi-supervised setup will be applied in an illustrative example on synthetic data and in the case study.

A potential disadvantage of using a semi-supervised classifier, is that it does not necessarily isolate effect of the DR method that is used to compute the embedding. For this reason, we will also test our method on some benchmark datasets, but in order to keep everything coherent, except for the method used to compute the embedding, we compute the embedding using NMLSDR and baseline DR methods based on only the noisy and partially labeled multi-label training data. Thereafter, we assume that the true multi-labels are available when we train the ML-kNN classifier on the embeddings.

The remainder of this section is organized as follows. First we describe the performance measures we employed, baseline DR methods, and how we select hyper-parameters. Thereafter we provide an illustrative example on synthetic data, and secondly experiments on the benchmark data. The case study is described in the next section.

#### 4.1. Evaluation metrics

Evaluation of performance is more complicated in a multi-label setting than for traditional single-labels. In this work, we decide use the seven different evaluation criteria that were employed in [40], namely Hamming loss (HL), Macro F1-score (MaF1), Micro F1 (MiF1), Ranking loss (RL), Average precision (AP), One-error (OE) and Coverage (Cov).

HL simply evaluates the number of times there is a mismatch between the predicted label and the true label, i.e.

\[
HL = \frac{1}{nC} \sum_{i=1}^{n} \left\| \hat{y}_i \oplus y_i \right\|_1,
\]

where \( \hat{y}_i \) denotes the predicted label vector of data point \( x_i \) and \( \oplus \) is the XOR-operator. MaF1 is obtained by first computing the F1-score for each label, and then averaging over all labels.

\[
MaF1 = \frac{1}{C} \sum_{c=1}^{C} \frac{2 \sum_{i=1}^{n} \hat{y}_{ic} y_{ic}}{\sum_{i=1}^{n} \hat{y}_{ic} + \sum_{i=1}^{n} y_{ic}}.
\]

MiF1 calculates the F1 score on the predictions of different labels as a whole,

\[
MiF1 = \frac{2 \sum_{i=1}^{n} \sum_{c=1}^{C} \hat{y}_{ic} y_{ic}}{\sum_{i=1}^{n} \sum_{c=1}^{C} \hat{y}_{ic} + \sum_{i=1}^{n} \sum_{c=1}^{C} y_{ic}}.
\]

We note that HL, MiF1 and MaF1 are computed based on hard labels assignments, whereas the four other measures are computed based on soft labels. In all of our experiments, we obtain the hard labels by putting a threshold at 0.5.

RL computes the average ratio of reversely ordered label pairs of each data point. AP evaluates the average fraction of relevant labels ranked higher than a particular relevant label. OE gives the ratio of data points where the most confident predicted label is wrong. Cov gives an average of how far one relevant label ranked higher than a particular relevant label. For a more detailed description of these measures, we point the interested reader to [72].

In this work, we modify four of the evaluation metrics such that all of them take values in the interval \([0, 1]\) and "higher always is better". Hence, we define

\[
HL' = 1 - HL,
\]

\[
RL' = 1 - RL,
\]

\[
OE' = 1 - OE,
\]

and normalized coverage (Cov') by

\[
Cov' = 1 - Cov/(C - 1).
\]

#### 4.2. Baseline dimensionality reduction methods

In this work, we consider the following other DR methods: CCA, MVMD, MDDM, MDDM and four variants of MLDA, namely wMLDAh, wMLDAe, wMLDAc and wMLDAd. These methods are supervised and require labeled data, and are therefore trained only on the labeled part of the training data. In addition, we compare to a semi-supervised method, SSMLDR, which we adapt to noisy multi-labels by using the label propagation algorithm we propose in this paper instead of the label propagation method that was originally proposed in SSMLDR.
4.3. Hyper-parameter selection

For the ML-kNN classifier we set $k = 10$. The effect of varying the number of neighbors will be left for further work. In order to learn the NMLSDR embedding we use a $k$NN-graph with $k = 10$ and binary weights. Moreover, we set $\alpha_i = 0.6$ for labeled data and $\alpha_i = 0.999$ for unlabeled data. By doing so, one ensures that an unlabeled datapoint is not affected by its initial value, but gets all contribution from the neighbors during the propagation.

4.4. Illustrative example on synthetic toy data

Dataset description. To test the framework in a controlled experiment, a synthetic dataset is created as follows.

A dataset of size 8000 samples is created, where each of the data points has dimensionality 320. The number of classes is set to 4, and we generate 2000 samples from each class. 30% from class 1 also belong to class 2, and vice versa. 20% from class 2 also belong to class 3 and vice versa, whereas 25% from class 3 also belong to class 4 and vice versa.

A sample from class $i$ is generated by randomly letting 10% of the features in the interval $[20(i-1) + 1, \ldots, 20i]$ take a random integer value between 1 and 10. Since there are 4 classes, this means that the first 80 features are directly dependent on the class-membership.

For the remaining 240 features we consider 20 of them at the time. We randomly select 50% of the 8000 samples and randomly let 20% of the 20 features take a random integer value between 1 and 10. We repeat this procedure for the 12 different sets of 20 features $[20(i-1) + 1, \ldots, 20i]$, $i = 5, 6, \ldots, 16$.

All features that are not given a value using the procedure described above are set to 0. Noise is injected into the labels by randomly flipping a fraction $p = 0.1$ of the labels and we make the data partially labeled by removing 50% of the labels. 2000 of the samples are kept aside as an independent test set. We note that noisy labels are often easier and cheaper to obtain than true labels and it is therefore not unreasonable that the fraction of labeled examples is larger than what it commonly is in traditional semi-supervised learning settings.

Results. We apply the NMLSDR method in combination with the semi-supervised ML-kNN classifier as explained above and compare to SSMLDR. We create two baselines by, for both of these methods, using a different value for the hyperparameter $\alpha_i$ for the labeled part of the data, namely 0, which corresponds to clamping. We denote these two baselines by SSMLDR* and NMLSDR*. In addition, we compare to baselines that only utilize the labeled part of the data, namely the supervised DR methods explained above in combination with a ML-kNN classifier. The data is standardized to 0 mean and 1 in standard
NMLSDR with methods gives the best performance for all metrics. Moreover, SSDs on the synthetic dataset. As we can see, our proposed places where they naturally belong, namely between the axes NMLSDR embedding than the SSMLDR embedding. Fig. 1c that the classes are better separated and more compact in the NMLSDR embedding. For comparison, we also see, in the PCA embedding the classes are not separated from each other, whereas in the NMLSDR and SSMLDR embeddings the classes are aligned along different axes. It can be seen that the classes are better separated and more compact in the NMLSDR embedding than the SSMLDR embedding. Fig. 1c shows that the data points that belong to multiple classes are placed where they naturally belong, namely between the axes responding to both of the classes they are member of. Tab. 1 shows the results obtained using the different methods on the synthetic dataset. As we can see, our proposed method gives the best performance for all metrics. Moreover, NMLSDR with \( \alpha^2 = 0 \), which corresponds to clamping of the labeled data during label propagation gives the second best results but cannot compete with our proposed method, in which the labels are allowed to change during the propagation to account for noisy labels. We also note that, even though the SSMLDR improves the MLDA approaches that are based on only the labeled part of the data, it gives results that are considerably worse than NMLSDR.

4.5. Benchmark datasets

**Experimental setup.** We consider the following benchmark datasets: Birds, Corel, Emotions, Enron, Genbase, Medical, Scene, Tmc2007 and Yeast. We also add our synthetic toy dataset as a one of our benchmark datasets (described in Sec. 4.4). These datasets are shown in Tab. 2, along with some useful characteristics. In order to be able to apply our framework to the benchmark datasets, we randomly flip 10% of the labels to generate noisy labels and let 30% of the data points training sets be labeled. All datasets are standardized to zero mean and standard deviation one.

We apply the DR methods to the partially and noisy labeled multi-label training sets in order to learn the projection matrix.

### Table 1: Performance of different embeddings on the synthetic dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>HL</th>
<th>RL</th>
<th>AP</th>
<th>OE</th>
<th>Conv</th>
<th>MaF1</th>
<th>MiF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCA</td>
<td>0.863</td>
<td>0.884</td>
<td>0.896</td>
<td>0.852</td>
<td>0.816</td>
<td>0.787</td>
<td>0.785</td>
</tr>
<tr>
<td>MVMD</td>
<td>0.906</td>
<td>0.912</td>
<td>0.924</td>
<td>0.897</td>
<td>0.836</td>
<td>0.850</td>
<td>0.849</td>
</tr>
<tr>
<td>MDDM</td>
<td>0.906</td>
<td>0.911</td>
<td>0.924</td>
<td>0.897</td>
<td>0.836</td>
<td>0.851</td>
<td>0.850</td>
</tr>
<tr>
<td>MDDMf</td>
<td>0.859</td>
<td>0.888</td>
<td>0.900</td>
<td>0.855</td>
<td>0.819</td>
<td>0.785</td>
<td>0.783</td>
</tr>
<tr>
<td>wMLDAb</td>
<td>0.844</td>
<td>0.871</td>
<td>0.885</td>
<td>0.831</td>
<td>0.807</td>
<td>0.754</td>
<td>0.750</td>
</tr>
<tr>
<td>wMLDAd</td>
<td>0.864</td>
<td>0.885</td>
<td>0.899</td>
<td>0.855</td>
<td>0.818</td>
<td>0.790</td>
<td>0.788</td>
</tr>
<tr>
<td>wMLDAc</td>
<td>0.865</td>
<td>0.887</td>
<td>0.900</td>
<td>0.857</td>
<td>0.818</td>
<td>0.787</td>
<td>0.785</td>
</tr>
<tr>
<td>wMLDAd</td>
<td>0.869</td>
<td>0.891</td>
<td>0.907</td>
<td>0.869</td>
<td>0.822</td>
<td>0.788</td>
<td>0.786</td>
</tr>
<tr>
<td>SSMLDR</td>
<td>0.863</td>
<td>0.883</td>
<td>0.899</td>
<td>0.859</td>
<td>0.814</td>
<td>0.796</td>
<td>0.793</td>
</tr>
<tr>
<td>SSMLDR*</td>
<td>0.879</td>
<td>0.898</td>
<td>0.910</td>
<td>0.871</td>
<td>0.827</td>
<td>0.817</td>
<td>0.814</td>
</tr>
<tr>
<td>NMLSDR</td>
<td>0.913</td>
<td>0.925</td>
<td>0.935</td>
<td>0.912</td>
<td>0.846</td>
<td>0.868</td>
<td>0.866</td>
</tr>
</tbody>
</table>

**Figure 2:** Mean of the Wilcoxon score obtained over the 7 different metrics.

P, which in turn is used to map the D-dimensional training and test sets to a \( d \)-dimensional representation. \( d \) is set as large as possible, i.e. to \( C - 1 \) for the MLDA-based methods and \( C \) for the other methods. Then we train a ML-kNN classifier using the low-dimensional training sets, assuming that the true multi-labels are known and validate the performance on the low-dimensional test sets.

In total we are evaluating the performance over 10 different datasets and across 7 different performance measures for all the feature extraction methods we use. Hence, to investigate which method performs better according to the different metrics, we also report the number of times each method gets the highest value of each metric. In addition, we compare all pairs of methods by using a Wilcoxon signed rank test with 5% significance level [73]. Similarly to [56], if method A performs better than B according to the test, A is assigned the score 1 and B the score 0. If the null hypothesis (method A and B perform equally) is not rejected, both A and B are assigned an equal score of 0.5.

**Results.** Tab. 3 shows results in terms of HL'. NMLSDR gets best HL'-score for eight of the datasets and achieves a maximal Wilcoxon score, i.e performs statistically better than all nine other methods according to the test at a 5% significance level. The second best method MDDMp gets the highest HL' score for three datasets and Wilcoxon score of 7.5. From Tab. 4 we see that NMLSDR achieves the highest RL'-score seven times and a Wilcoxon score of 8.5. The second best method is MVMD, which obtains three of the highest RL' values and a Wilcoxon score of 8.0.

Tab. 5 shows performance in terms of AP. The highest AP score is achieved for NMLSDR for eight datasets and it gets a maximal Wilcoxon score of 9.0. According to the Wilcoxon score second place is tied between MVMD and MDDMp. However, MVMD gets the highest AP score for two datasets, whereas MDDMp does not get the highest score for any of them. OE' is presented in Tab. 6. We can see that NMLSDR gets a maximal Wilcoxon score and the highest OE' score for seven datasets. MVMD is number two with a Wilcoxon score 2Downloaded from mulan.sourceforge.net
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Train instances</th>
<th>Test instances</th>
<th>Attributes</th>
<th>Labels</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birds</td>
<td>audio</td>
<td>322</td>
<td>323</td>
<td>260</td>
<td>19</td>
<td>1.06</td>
</tr>
<tr>
<td>Corel</td>
<td>scene</td>
<td>5188</td>
<td>1744</td>
<td>500</td>
<td>153</td>
<td>2.87</td>
</tr>
<tr>
<td>Emotions</td>
<td>music</td>
<td>391</td>
<td>202</td>
<td>72</td>
<td>6</td>
<td>1.81</td>
</tr>
<tr>
<td>Enron</td>
<td>text</td>
<td>1123</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Genbase</td>
<td>biology</td>
<td>463</td>
<td>199</td>
<td>199</td>
<td>29</td>
<td>1.26</td>
</tr>
<tr>
<td>Medical</td>
<td>text</td>
<td>645</td>
<td>333</td>
<td>1161</td>
<td>39</td>
<td>1.24</td>
</tr>
<tr>
<td>Scene</td>
<td>scene</td>
<td>1211</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Tmc2007</td>
<td>text</td>
<td>3000</td>
<td>7077</td>
<td>493</td>
<td>22</td>
<td>2.25</td>
</tr>
<tr>
<td>Toy</td>
<td>synthetic</td>
<td>6000</td>
<td>2000</td>
<td>320</td>
<td>4</td>
<td>1.38</td>
</tr>
<tr>
<td>Yeast</td>
<td>biology</td>
<td>1500</td>
<td>917</td>
<td>103</td>
<td>14</td>
<td>4.23</td>
</tr>
</tbody>
</table>

**Table 2: Description of benchmark datasets considered in our experiments.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Train instances</th>
<th>Test instances</th>
<th>Attributes</th>
<th>Labels</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birds</td>
<td>audio</td>
<td>322</td>
<td>323</td>
<td>260</td>
<td>19</td>
<td>1.06</td>
</tr>
<tr>
<td>Corel</td>
<td>scene</td>
<td>5188</td>
<td>1744</td>
<td>500</td>
<td>153</td>
<td>2.87</td>
</tr>
<tr>
<td>Emotions</td>
<td>music</td>
<td>391</td>
<td>202</td>
<td>72</td>
<td>6</td>
<td>1.81</td>
</tr>
<tr>
<td>Enron</td>
<td>text</td>
<td>1123</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Genbase</td>
<td>biology</td>
<td>463</td>
<td>199</td>
<td>199</td>
<td>29</td>
<td>1.26</td>
</tr>
<tr>
<td>Medical</td>
<td>text</td>
<td>645</td>
<td>333</td>
<td>1161</td>
<td>39</td>
<td>1.24</td>
</tr>
<tr>
<td>Scene</td>
<td>scene</td>
<td>1211</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Tmc2007</td>
<td>text</td>
<td>3000</td>
<td>7077</td>
<td>493</td>
<td>22</td>
<td>2.25</td>
</tr>
<tr>
<td>Toy</td>
<td>synthetic</td>
<td>6000</td>
<td>2000</td>
<td>320</td>
<td>4</td>
<td>1.38</td>
</tr>
<tr>
<td>Yeast</td>
<td>biology</td>
<td>1500</td>
<td>917</td>
<td>103</td>
<td>14</td>
<td>4.23</td>
</tr>
</tbody>
</table>

**Table 3: Performance in terms of 1 - Hamming loss (HL') across 10 different benchmark datasets.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Train instances</th>
<th>Test instances</th>
<th>Attributes</th>
<th>Labels</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birds</td>
<td>audio</td>
<td>322</td>
<td>323</td>
<td>260</td>
<td>19</td>
<td>1.06</td>
</tr>
<tr>
<td>Corel</td>
<td>scene</td>
<td>5188</td>
<td>1744</td>
<td>500</td>
<td>153</td>
<td>2.87</td>
</tr>
<tr>
<td>Emotions</td>
<td>music</td>
<td>391</td>
<td>202</td>
<td>72</td>
<td>6</td>
<td>1.81</td>
</tr>
<tr>
<td>Enron</td>
<td>text</td>
<td>1123</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Genbase</td>
<td>biology</td>
<td>463</td>
<td>199</td>
<td>199</td>
<td>29</td>
<td>1.26</td>
</tr>
<tr>
<td>Medical</td>
<td>text</td>
<td>645</td>
<td>333</td>
<td>1161</td>
<td>39</td>
<td>1.24</td>
</tr>
<tr>
<td>Scene</td>
<td>scene</td>
<td>1211</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Tmc2007</td>
<td>text</td>
<td>3000</td>
<td>7077</td>
<td>493</td>
<td>22</td>
<td>2.25</td>
</tr>
<tr>
<td>Toy</td>
<td>synthetic</td>
<td>6000</td>
<td>2000</td>
<td>320</td>
<td>4</td>
<td>1.38</td>
</tr>
<tr>
<td>Yeast</td>
<td>biology</td>
<td>1500</td>
<td>917</td>
<td>103</td>
<td>14</td>
<td>4.23</td>
</tr>
</tbody>
</table>

**Table 4: Performance in terms of 1 - Ranking loss (RL') across 10 different benchmark datasets.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Train instances</th>
<th>Test instances</th>
<th>Attributes</th>
<th>Labels</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birds</td>
<td>audio</td>
<td>322</td>
<td>323</td>
<td>260</td>
<td>19</td>
<td>1.06</td>
</tr>
<tr>
<td>Corel</td>
<td>scene</td>
<td>5188</td>
<td>1744</td>
<td>500</td>
<td>153</td>
<td>2.87</td>
</tr>
<tr>
<td>Emotions</td>
<td>music</td>
<td>391</td>
<td>202</td>
<td>72</td>
<td>6</td>
<td>1.81</td>
</tr>
<tr>
<td>Enron</td>
<td>text</td>
<td>1123</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Genbase</td>
<td>biology</td>
<td>463</td>
<td>199</td>
<td>199</td>
<td>29</td>
<td>1.26</td>
</tr>
<tr>
<td>Medical</td>
<td>text</td>
<td>645</td>
<td>333</td>
<td>1161</td>
<td>39</td>
<td>1.24</td>
</tr>
<tr>
<td>Scene</td>
<td>scene</td>
<td>1211</td>
<td>1196</td>
<td>294</td>
<td>6</td>
<td>1.06</td>
</tr>
<tr>
<td>Tmc2007</td>
<td>text</td>
<td>3000</td>
<td>7077</td>
<td>493</td>
<td>22</td>
<td>2.25</td>
</tr>
<tr>
<td>Toy</td>
<td>synthetic</td>
<td>6000</td>
<td>2000</td>
<td>320</td>
<td>4</td>
<td>1.38</td>
</tr>
<tr>
<td>Yeast</td>
<td>biology</td>
<td>1500</td>
<td>917</td>
<td>103</td>
<td>14</td>
<td>4.23</td>
</tr>
</tbody>
</table>

**Table 5: Performance in terms of Average precision (AP) across 10 different benchmark datasets.**

Tab. 7 shows Cov'. NMLSDR gets a maximal Wilcoxon score and the highest Cov' value for seven datasets. Despite that MVMD gets the highest Cov' for three datasets and MDDMf of none of the datasets, the second best Wilcoxon score is 7.5 and tied between MVMD and MDDMp. MaF1 is shown in Tab. 8. The best method, which is our proposed method gets a maximal Wilcoxon score and the highest MaF1 value of 8.0 and two best values.

Tab. 8. The best method, which is our proposed method gets a maximal Wilcoxon score and the highest MaF1 value of 8.0 and two best values.
for six datasets. Tab. 9 shows MiF1. NMLSDR achieves 8.5 in Wilcoxon score and has the highest MiF1 score for seven datasets.

In total, NMLSDR consistently gives the best performance for all seven evaluation metrics. Moreover, in order to summarize our findings, we compute the mean Wilcoxon score across all seven performance metrics and plot the result in Fig. 2. If we sort these results, we get NMLSDR (8.86), MVMD (7.64),...
 Chronic diseases. According to The World Health Organisation, a disease is defined as chronic if one or several of the following criteria are satisfied: the disease is permanent, requires special training of the patient for rehabilitation, is caused by non-reversible pathological alterations, or requires a long period of supervision, observation, or care. The two most prevalent chronic diseases for people over 64 years are those that we study in this paper, namely hypertension and diabetes mellitus [74]. These types of diseases represent an increasing problem in modern societies all over the world, which to a large degree is due to a general increase in life expectancy, along with an increased prevalence of chronic diseases in an aging population [75]. Moreover, the economical burden associated with these chronic conditions is high. For example, in 2017, treatment of diabetic patients accounted for 1 out of 4 healthcare dollars in the United States [76]. Hence, in the future, a significant amount of resources must be devoted to the care of chronic patients and it will be important not only to improve the patient care, but also more efficiently allocate the resources spent on treatment of these diseases.

5.1. Data

In this case study, we study a dataset consisting of patients that potentially have one or more chronic diseases. All of these patients got some type of treatment at University Hospital of Fuenlabrada, Madrid (Spain) in the year 2012. The patients are described by diagnosis codes following the International Classification of Diseases 9th revision, Clinical Modification (ICD9-CM) [77], and pharmacological dispensing codes according to Anatomical Therapeutic Chemical (ATC) classification systems [78]. Some preprocessing steps are considered. Similarly to [79, 80], the ICD9-CM and ATC codes are represented using frequencies, i.e., for each patient, we consider all encounters with the health system in 2012 and we count how many times each ICD9-CM and ATC code appear in the electronic health record. In total there are 1517 ICD9-CM codes and 746 ATC codes. However, all codes that appear for less than 10 patients across the training set are removed. After this feature selection, the dimensionality of the data is 455, of which 267 represent ICD9-CM codes and 188 represent ATC codes.

We do have access to ground truth labels that indicate what type of chronic disease(s) the patients have. These are provided by a patient classification system developed by the company 3M [81]. This classification system stratify patients into so-called Clinical Risk Groups (CRG) that indicate what type(s) of chronic disease the patient has and the severity based on the patient encounters with the health system during a period of time, typically one year. A five-digit classification code is used to assign each patient to a severity risk group. The first digit of the CRG is the core health status group, ranging from healthy (1) to catastrophic (9); the second to fourth digits represents the base 3M CRG; and the fifth digit is used for characterizing the severity-of-illness levels.

For the purpose of this work, the ground truth labels are only used for cohort selection and final evaluation of our models. For the remaining parts they are considered unknown. To select a cohort, we consider the first four digits of the CRGs to analyze the the following chronic conditions: CRG-1000 (healthy), which contains 46835 individuals; CRG-5192 (hypertension) with 12447 patients; CRG-5424 (diabetes), which has 2166 patients; and CRG-6144 (hypertension and diabetes), with a total of 3179 patients. We employ an undersampling strategy and randomly select 2166 patients from each of the four categories, and thereby obtain balanced classes. An independent test set is

MDDMp (7.43), wMLDAd (4.43), MDDMf (4.21), SSMLDR (3.79), CCA (2.79), wMLDAe (2.71) and wMLDAb (1.57). The best method, which is our proposed method, gets a mean value that is 1.22 higher than number two. The second best method is MVMD, slightly better than MDDMp. The best MLDA-based method is wMLDAd, which is ranked 4th, however, with a much lower mean value than the three best methods. The semi-supervised extension of MLDA (SSMLDR) is ranked 6th and is actually performing worse than wMLDAd, which is a bit surprising. However, SSMLDR also uses a binary weighting scheme, and should therefore be considered as a semi-supervised variant of wMLDAb, which it performs considerably better than. wMLDAb and wMLDAc give the worst performance of all the 10 methods.

The main reason why the MLDA-based approaches in general perform worse than the other DR methods is probably related to what we discussed in Sec. 2, namely that LDA-based approaches are heavily affected by outliers and wrongly labeled data. More concretely, the fact that the number of labeled data points are relatively few and that the labels are noisy, leads to errors in the scatter matrices that even might amplify since one has to invert a matrix to solve the generalized eigenvalue problem. The semi-supervised extension of MLDA, SSMLDR, improves quite much compared to wMLDAb, but the starting point is so bad that even though it improves, it cannot compete with the best methods. On the other hand, the MDDM-based methods (MVMD and MDDMp) are not so sensitive to label noise and the fact that there are few labels, and therefore these methods can perform quite well even though they are trained only on the labeled subset. Hence, the reasons to the good performance of NMLSDR are probably that MDDMp is the basis of NMLSDR, and that NMLSDR in addition uses our label propagation method to improve.

5. Case study

In this section, we describe a case study where we study patients potentially suffering from multiple chronic diseases. This healthcare case study reflects the need for label noise-tolerant methods in a non-standard situation (semi-supervised learning, multiple labels, high dimensionality). The objective is to identify patients with certain chronic diseases, more specifically hypertension and/or diabetes mellitus. In order to do so, we take an approach where we use clinical expertise to create a partially and noisy labeled dataset, and thereafter apply our proposed end-to-end framework, namely NMLSDR for dimensionality reduction in combination with semi-supervised ML-kNN to classify these patients. An overview of the framework employed in the case study is shown in Fig. 3.

In this section, we describe a case study where we study patients potentially suffering from multiple chronic diseases. This healthcare case study reflects the need for label noise-tolerant methods in a non-standard situation (semi-supervised learning, multiple labels, high dimensionality). The objective is to identify patients with certain chronic diseases, more specifically hypertension and/or diabetes mellitus. In order to do so, we take an approach where we use clinical expertise to create a partially and noisy labeled dataset, and thereafter apply our proposed end-to-end framework, namely NMLSDR for dimensionality reduction in combination with semi-supervised ML-kNN to classify these patients. An overview of the framework employed in the case study is shown in Fig. 3.

In this section, we describe a case study where we study patients potentially suffering from multiple chronic diseases. This healthcare case study reflects the need for label noise-tolerant methods in a non-standard situation (semi-supervised learning, multiple labels, high dimensionality). The objective is to identify patients with certain chronic diseases, more specifically hypertension and/or diabetes mellitus. In order to do so, we take an approach where we use clinical expertise to create a partially and noisy labeled dataset, and thereafter apply our proposed end-to-end framework, namely NMLSDR for dimensionality reduction in combination with semi-supervised ML-kNN to classify these patients. An overview of the framework employed in the case study is shown in Fig. 3.
created by randomly selecting 20% of these patients. Hence, the training set contains 6932 patients and the test set 1732 patients.

5.2. Rule-based creation of noisy labeled training data using clinical knowledge

There are some important ICD9-CM codes and ATC-drugs that are strongly correlated with hypertension and diabetes, respectively. These are verified by our clinical experts and described in Tab. 10. In particular, the ICD9-CM code 250 is important for diabetes because it is the code for diabetes mellitus. Similarly, the ICD9-CM codes 401-405 are important for hypertension because they describe different types of hypertension.

In this case study we are interested in four groups, namely those that have hypertension, those that have diabetes, those that have both, and those that do not have any of these two chronic diseases. Thanks to the clinical expertise and the information that they provided us with, which is summarized in Tab. 10, we can create a partially and noisy labeled dataset using the following set of rules.

1. Those that have the ICD codes 250 and any of the codes 401-405 are assigned to both the hypertension and diabetes class.
2. Those that have the ICD code 250, but none of the 7 ICD9-CM codes and 64 ATC drugs listed by the clinicians as indicators for hypertension, are labeled with diabetes.
3. Those that have any of the ICD9-CM codes 401-405, but none of the 4 ICD9-CM codes for diabetes or 12 ATC drugs for diabetes, are labeled with hypertension.
4. Those that do not have any of the ICD9-CM codes or ATC drugs listed up in Tab. 10 are labeled as healthy.
5. The remaining patients do not get a label.

In total, this leads to 1734 in the healthy class, 2547 in the hypertension class, 1971 in the diabetes class, 1302 of the patients in the hypertension class also belongs to the diabetes class. 1982 of the patients do not get a label using the routine described above. To be able to examine for statistical significance, we randomly select 1000 of the noisy labeled patients and 1000 of the unlabeled patients. By doing so, we can repeat the experiments several times and test for significance using a pairwise t-test. We do the repetition 10 times and let the significance level be 95%.

5.3. Results

Tab. 11 shows the performance of the different DR methods on the task of classifying patients with chronic diseases in terms of seven different evaluation metrics. According to the pairwise t-test, our method achieves the best performance for all metrics. Second place is tied between MDDMp and MVMD. The semi-supervised variant of MLDA, namely SSMLDR, performs better than the supervised counterparts (wMLDAb, wMLDAc, wMLDaAb, wMLDAe) and is consistently ranked 4th according to all metrics. Interestingly, the more advanced weighting schemes in wMLDAc and wMLDAd actually lead to worse results than what the simple weights in wMLDAb and wMLDAe give. CCA gives the worst performance according to 4 of the evaluation measures, for the 3 other measures the difference between CCA and wMLDAd is not significant.

Fig. 4 shows plots of the two-dimensional embeddings of the chronic patients obtained using four different DR methods, namely MDDMp, wMLDAb, NMLSDR and SSMLDR. The different colors and markers represent the true CRG-labels of the patients. As we can see, visually the MDDMp and NMLSDR embeddings look quite similar. The healthy patients are squeezed together in a small area (purple dots), and the yellow dots that represent patients that have both diabetes and hypertension are placed between the blue dots, which are those that have only hypertension, and the red dots, which represent the patient that only have diabetes. Intuitively, this placement makes sense. On the other hand, the embedding obtained using SSMLDR does not look similar to its counterpart obtained using wMLDAb, and it is easy to see why the performance of wMLDAb is worse.

6. Conclusions

In this paper we have introduced the NMLSDR method, a dimensionality reduction method for partially and noisy labeled multi-label data. To our knowledge, NMLSDR is the only
method the method can explicitly deal with this type of data. Key components in the method are a label propagation algorithm that can deal with noisy data and maximization of feature-label dependence using the Hilbert-Schmidt independence criterion. Our extensive experimental sections show that NMLSDR is a good dimensionality reduction method in settings where one has access to partially and noisy labeled multi-label data.

In the future, we will investigate more thoroughly the extent of using different weighting schemes in NMLSDR, similarly to how it is done in MLDA with wMLDa, wMLDa, wMLDa and wMDLAd.

Acknowledgments

This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the Next Generation Learning Machines. Cristina Soguero-Ruiz is partially supported by project TEC2016-75361-R from Spanish Government and by Project DTS17/00158 from Institute of Health Carlos III (Spain). The authors would like to thank clinicians at University Hospital of Fuenlabrada for their helpful advice and comments on various issues examined in the case study.

References


Figure 4: Plot two-dimensional embeddings of the chronic patients obtained using four different DR methods: (a) MDDMp, (b) wMLDAb (c) NMLSDr (d) SSMLDR. The different colors and markers represent the true CRG-labels of the patients.
[74] A. Soni, E. Mitchell, Expenditures for commonly treated conditions among adults age 18 and older in the U.S. civilian noninstitutionalized