Diagnosis Code Assignment Using Sparsity-Based Disease Correlation Embedding

Sen Wang, Xiaojun Chang, Xue Li, *Member, IEEE*, Guodong Long, Lina Yao, and Quan Z. Sheng

Abstract—With the latest developments in database technologies, it becomes easier to store the medical records of hospital patients from their first day of admission than was previously possible. In Intensive Care Units (ICU), modern medical information systems can record patient events in relational databases every second. Knowledge mining from these huge volumes of medical data is beneficial to both caregivers and patients. Given a set of electronic patient records, a system that effectively assigns the disease labels can facilitate medical database management and also benefit other researchers, e.g., pathologists. In this paper, we have proposed a framework to achieve that goal. Medical chart and note data of a patient are used to extract distinctive features. To encode patient features, we apply a Bag-of-Words encoding method for both chart and note data. We also propose a model that takes into account both global information and local correlations between diseases. Correlated diseases are characterized by a graph structure that is embedded in our sparsity-based framework. Our algorithm captures the disease relevance when labeling disease codes rather than making individual decision with respect to a specific disease. At the same time, the global optimal values are guaranteed by our proposed convex objective function. Extensive experiments have been conducted on a real-world large-scale ICU database. The evaluation results demonstrate that our method improves multi-label classification results by successfully incorporating disease correlations.

Index Terms—ICD code labeling, multi-label learning, sparsity-based regularization, disease correlation embedding

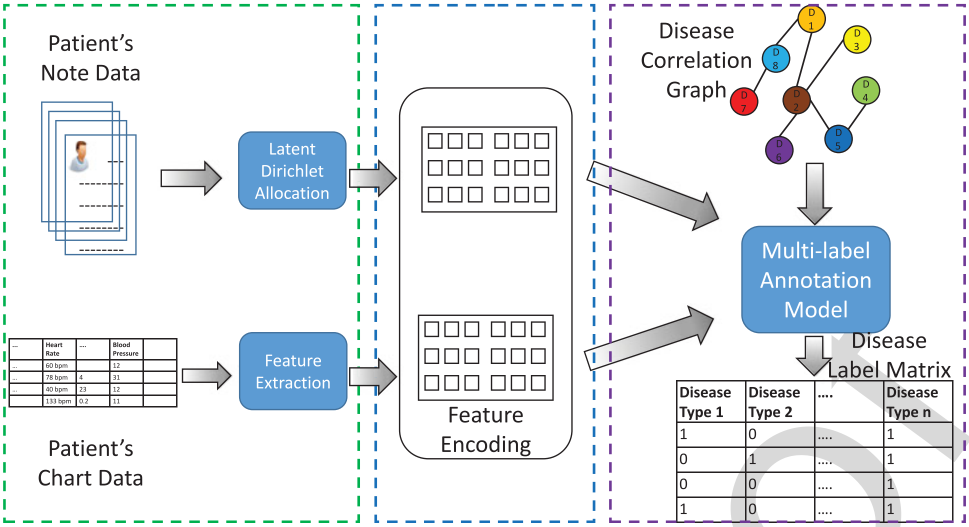
1 Introduction

Modern medical information systems, such as the Philips' CareVue system, records all patient data and stores them in relational databases for data management and related research activities. Clinicians and physicians often want to retrieve similar medical archives for a patient in ICU, with the aim of making better decisions. The simplest way is to input a group of disease codes that are diagnosed from the patient, into a system that can provide similar cases according to the codes. The most well-known and widely used disease code system is the International Statistical Classification of Diseases and Related Health Problems (commonly abbreviated as ICD) proposed and periodically revised by the World Health Organization (WHO). The latest version is ICD-10, which is applied with local clinical modifications in most of regions, e.g., ICD-10-AM for Australia. The goal of ICD is to provide a unique hierarchical classification system that is designed to map health conditions to different categories. In the United States, the ninth version of the International Classification of Disease (ICD9) has been pervasively applied in various areas where disease classification is required. For example, each patient in ICU will be associated with a list of ICD9 codes in the medical records purposes such as disease tracking, pathology, or medical record data management. By investigating the returned historical data, caregivers are expected to offer better treatments to the patient. Thus, complete and accurate disease labeling is very important.

The assignment of ICD codes to patients in ICU is traditionally done by caregivers in a hospital (e.g., physicians, nurses, and radiologists. This assignment may occur during or after admission to ICU. In the former case, ICD codes are separately labeled by multiple caregivers throughout a patient's stay in ICU as a result of different work shiftse duration of a patient's stay is usually much longer than the employment time shift of the medical staff in a hospital thus, different caregivers are prone to make judgments according to the latest conditions. It is more desirable to assign a disease label to the patient by taking the entire patient record into account. when assignment is conducted after admission to ICU, the ICD codes are allocated by a professional who examines and reviews all the records of a patient. However, it is still impossible for an expert to remember the correlations of diseases when labeling a list of disease codes, which sometimes leads to missing code or inaccurate code categorization. In fact, some diseases are highly correlated. Correlations between diseases can improve the multi-label classification results. For instance, *Hypertensive disease* (ICD9 401-405) correlates highly with *Other forms of heart disease* (ICD9 420-429) and *Other metabolic and immunity disorders* (ICD9 270-279). When considering the occurrence of the latter two disease labels in relation to the patient's condition, the possibility that a positive decision will be made will be much increased if *Hypertensive disease* is found in the

- S. Wang is with the School of Information and Communication Technology, Griffith University, Southport, QLD. E-mail: sen.wang@uq.edu.au.
- X. Chang and G. Long are with the Centre for Quantum Computation and Intelligent Systems (QCIS), University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: x.chang@uq.edu.au, guodong.long@uts.edu.au.
- X. Li is with the School of Information Technology and Electrical Engineering, The University of Queensland, St. Lucia, QLD. E-mail: huang@itee.uq.edu.au.
- L. Yao is with the School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia. E-mail: lina.yao@unsw.edu.au.
- Q.Z. Sheng is with the School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia. E-mail: michael.sheng@adelaide.edu.au.

Manuscript received 28 Aug. 2015; revised 12 July 2016; accepted 21 Aug. 2016. Date of publication 0 . 0000; date of current version 0 . 0000.
Recommended for acceptance by A. Banerjee.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2016.2605687



...	Heart Rate	....	Blood Pressure	
...	60 bpm		12	
...	78 bpm	4	31	
...	40 bpm	23	12	
	133 bpm	0.2	11	

Disease Type 1	Disease Type 2	....	Disease Type n
1	0	....	1
0	1	....	1
0	0	....	1
1	0	....	1

Fig. 1. Workflow demonstration of the proposed framework. The green box on the left contains the data pre-processing, Latent Dirichlet Allocation (LDA) topic modeling, and feature extractions. The blue central box mainly encodes the features using a Bag-of-Words model on both extracted chart and note features. The purple box on the right shows the main contribution of this work. A multi-label classification algorithm is proposed to assign patients' disease codes by correctly incorporating a structural graph that reflects disease correlations into the sparsity-based framework.

patient's record. Therefore, it is desirable to produce a system that can overcome the problems mentioned above.

The focus in this work is to assign disease labels to patients' medical records. Rather than predicting the mortality risk of an ICU patient, as in some previous works [1], [2], our work can be regarded as a multi-label prediction problem. In other words, mortality risk prediction is a binary classification problem in which the label indicates the probability of survival. Class labels in a multi-label problem, on the other hand, are not exclusive, which means the patient, according to the medical records, is labeled as belonging to multiple disease classes. The multi-label classification problem has always been an open but challenging problem in the machine learning and data mining communities. Some researchers [3], [4], [5], [6] extract features from patients and use supervised learning models to recognize disease labels without any consideration of disease correlations. In our model, we pay great attention to both the medical chart and note data of patients. Medical chart data is also termed structured data because their structure is normally fixed. In the ICU, some well-known health condition measurement scores (i.e., SAPS II) are manually determined by staff in the ICU, according to the patient's health condition. In contrast, medical chart data are raw recordings extracted from the monitoring devices attached to a patient. The chart data therefore reflect the physiological conditions of a patient at a lower level. Note data has no structure because it is derived from textual information. Therefore, it is commonly termed *free-text note data*. The advantages of these types of data are that they are descriptive and informative since they are summarized or determined by professionals. However, medical note data are very difficult to handle by most of the existing machine learning algorithms because none of the structures in the notes can be directly recognized as patterns. Medical notes are quite noisy, and their quality is often corrupted by misspellings or abbreviations. In addition, the contents of medical notes are not always consistent with the metrics. For example, different caregivers take notes in different metrics when recording a parameter. Some prefer to use English units while others use the American system (e.g., patient's temperature in Celsius versus Fahrenheit). Thus, compared to structured data, it is difficult to extract accurate and consistent features from notes. It is consequently difficult for medical notes to be utilized by machine learning algorithms.

To address the aforementioned problems, we propose a framework that will assign disease labels automatically while simultaneously considering correlations between diseases. We first extract medical data from two different views, structured and unstructured. Structured data can describe patients' raw health conditions from medical devices at a lower level, while unstructured data consist of more semantic information at a higher level which has proven to be helpful for characterizing features of patients for some prediction tasks [1]. We use a BoW model to convert features of different lengths into a unique representation for each patient. In this way, similarity comparison can be conducted by supervised learning algorithms. To step further, we propose an algorithm to classify disease labels with the help of the underlying correlations between diseases. Our work incorporates a graph structure which is derived from huge numbers of medical records to improve multi-label prediction results. The demonstration of the proposed framework is shown in Fig. 1. The main contributions of this work can be summarized as follows:

- We extract raw features from patients' chart data to characterize their conditions at a low level. A latent variable model, i.e., LDA, is used in this work to extract topic distributions in medical notes as descriptive features. BoW is proposed to encode both chart and note data for unique representation.
- We propose an algorithm to assign disease codes with joint consideration of disease correlations. This is achieved by incorporating a graph structure that reflects the correlations between diseases into a sparsity-based objective function. We propose the use of $\ell_{2,1}$-norm to exploit the correlations. Due to the

convexity of the objective function, the global optima are guaranteed.

- Extensive experiments have been conducted on a real-world ICU patient database. A large number of patient records are applied on this database in the evaluation. The experimental reports have shown that our proposed method is more effective for performing multi-label classification than the compared approaches. Effectiveness and efficiency evaluations have also been conducted.

The rest of this paper is organized as follows: Related work will be reviewed in Section 2. We will elaborate our method in detail in Section 3, followed by evaluation reports in Section 4. We conclude the paper in Section 5.

2 Related Work

2.1 Medical Feature Encoding

Most of the existing research works aim to mine interesting patterns from medical records that are most frequently stored in text and images. Due to the huge success of the Bag-of-Words model in Natural Language Processing (NLP) and computer vision, BoW and its variants have been pervasively utilized to encode features in medical applications to accomplish various tasks such as classification and retrieval. In [7], a method is proposed to convert the entire clinical text data into UMLS codes using NLP techniques. The experiments show that the encoding method is comparable to or better than human experts. Ruch et al. [8] evaluate the effects of corrupted medical records, i.e., misspelled words and abbreviations, on an information retrieval system that uses a classical BoW encoding method. To classify physiological data with different lengths, modified multivariate BoW models are used to encode patterns in [9]. In addition, the 1-Nearest Neighbour (1NN) classifier predicts acute hypotensive episodes. Recently, Wang et al. [10] propose a Nonnegative Matrix Factorization based framework to discover temporal patterns over large amounts of medical text data. Similar to the BoW representation, each patient in that work is represented by a fixed-length vector encoding the temporal patterns. The evaluation is conducted on a real-world dataset that consists of 21K diabetes patients. Types of diabetes diagnosis coded by ICD9 are treated as ground-truth.

2.2 Multi-Label Learning in Medical Applications

Multi-label classification has been well studied recent years [11], [12], [13], [14], [15], [16], [17], [18], [19] in the machine learning and data mining communities. Due to the omnipresence of multi-label prediction tasks in the medical domain, multi-label classification has attracted more and more research attention to this domain in the past few years. Perotte et al. [20] propose to use a hierarchy-based Support Vector Machines (SVM) model on MIMIC II dataset to conduct automated diagnosis code classification. Zufferey et al. [21] compare different multi-label classification algorithms for chronic disease classification and point out the hierarchy-based SVM model has achieved superior performance than other methods when accuracy is important. In [22], Ferrao et al. use Natural Language Processing to deal with structured electronic health record, and apply Support Vector Machines to separately learn each disease code for each patient. Pakhomov et al. [23] propose an automated coding system for diagnosis coding assignment powered by example-based rules and naive Bayes classifier. Lita et al. [4] assign diagnostic codes to patients using a Gaussian process-based method. Even though the proposed method is conducted over a large-scale medical database of 96,557 patients, the method does not consider the underlying relationships between diseases. Many theoretical studies on multi-label classification have already pointed out that effectively exploiting correlations between labels can benefit the multi-label classification performance. In light of this, Kong et al. [24] apply heterogeneous information networks on a bioinformatic dataset to for two different multi-label classification tasks (i.e., gene-disease association prediction and drug-target binding prediction) by exploiting correlations between different types of entities.

Prior-based knowledge incorporation by a regularization term is an effective way to exploit correlations between classes. In a scenario of medical code classification, Yan et al. [25] introduce a multi-label large margin classifier that automatically uncovers the inter-code structural information. Prior knowledge on disease relationships is also incorporated into their framework. In the reported results, underlying disease relationships are discovered and are beneficial to the multi-label classification results. All the evaluations are conducted over a quite small and clean dataset that consists of only 978 samples of patient visits. This approach is feasible for small dataset but is questionable in a real-world dataset. The most recent research on computational phenotyping in [26] tackles a small multi-label classification problem on a real-world ICU dataset by applying two novel modifications to a standard DCNN. Che et al. investigate two types of prior-based regularization methods. In the first method, they use the hierarchical structure of ICD9 code classification at two levels, and embed the hierarchical structure in an adjacency graph into the framework; The second method is to utilize the prior information extracted from labels of training data. Che et al. explore the label co-occurrence information with a co-occurrence matrix, and embed the matrix into their deep neural network to improve the prediction performance. Similar to the prior-based regularization methods, we also embed an affinity graph derived from data labels in the framework to exploit correlations between disease codes. However, we do not directly apply the label correlation matrix, also called label co-occurrence matrix in [26], to improve the performance of multi-label classification. Instead, we further learn and utilize the structural information among classes by a sparsity-based model, which has been largely ignored by most of the existing works on diagnosis code assignment. As pointed out in [27], sparsity-based regularizers such as ℓ_1-norm and combination of ℓ_1-norm and ℓ_2-norm have virtues on structure exploitation, which can extract useful information from high-dimensional data. Moreover, many existing works [28], [29], [30], [31], [32] beyond medical domain have shown sparsity-based $\ell_{2,1}$-norm on regularization plays an important role when exploiting correlated structures in different applications. To this end, we model the correlations between diseases using the affinity graph, and incorporate the topological constraints of the graph using a novel graph structured sparsity-based model, which can capture the hidden class structures in the graph.

TABLE 1
Summarization of MIMIC II Database

	Size	Total	#per patient	Dim.
Charts	17 Gb	196,156,501	8390.29	500
Notes	618 Mb	599,128	25.63	500
Training Data	11,689	23,379	N.A.	500
Testing Data	11,790			500

3 Methods

In this section, we will first introduce the details of the database and data pre-processing methods used in this paper. Feature extractions from both chart and note data will be elaborated, followed by a description of the encoding method that is investigated in this paper. An algorithm that is able to incorporate correlations between diseases is subsequently proposed to solve the aforementioned problems.

3.1 Database and Data Pre-Processing

Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC II) [33] is a real-world medical database that is publicly available. Thanks to the efforts of academia, industry and clinical medicine, the database has successfully collected 32,535 ICU patients over seven years (from 2001 to 2007) at Boston's Beth Israel Deaconess Medical Center. To the best of our knowledge, MIMIC II is the largest ICU published database with comprehensive types of patient data in the world. Before releasing the database to the public, data scientists completely removed all protected health information (PHI) to protect the privacy of patients. A variety of data sources have been recorded in this database: 1) patient data recorded from bedside monitors, e.g., waveforms and trends; 2) data from clinical information systems; 3) data from hospital electronic archives; 4) mortality information. In this paper, we have used two parts of the database: chart event data and medical note data. Since chart data comes from device recordings made by caregivers, it reflects the health conditions of patients at a low level, whereas medical note data comes from medical doctors, registered nurses, and other professionals, and contains high-level semantic information summarized by experts. [1] has proven that extracting features based on topic modeling from note data is able to predict the mortality risk of patients.

Because only adult patient data are considered in this work, patients younger than 18 are excluded in the first step. We need both charts and notes as the raw data of a patient, so all those patients whose chart and note data are either empty and nearly empty or corrupted for unknown reasons, are ruled out. Patients without ICD9 records are also removed since their ground-truth information is uncertain. After patient filtering in three rounds, we obtain 23,379 adult patients out of 32,535. To train and test our algorithm, we randomly split the dataset into two parts, training data and testing data. Table 1 shows the data specifications. Note that the numbers in the fourth column (#per patient) are based on the total number of patients $(11,689 + 11,690 = 23,379)$. For example, the number of charts per patient is calculated by $196,156,501/23,379 \approx 8390.29$.

The ICD9 codes for each patient are stored in a list in the patient's medical record. We utilize ICD9 codes as ground-truth to train and test our models in experiments. According to its hierarchical structure, there are 19 categories at the upper level for the most general classification and 129 categories at the lower level for more specific classifications. Fig. 2 represents the hierarchical structure of ICD9, of which we use two levels, i.e., *high level* and *low level*. For example, all codes ranging from 460 to 519 are classified as *diseases of the respiratory system*, which is a general class label. There are six subclasses in this general class group: *acute respiratory infections* (460-466), *other diseases of the upper respiratory tract* (470-478), *pneumonia and influenza* (480-488), *chronic obstructive pulmonary disease and allied conditions* (490-496), *pneumoconioses and other lung diseases due to external agents* (500-508), and *other diseases of respiratory system* (510-519). Since they are hierarchically organized in two levels, we use them as label information in different two schemes. We name two different classification schemes c_0 and c_1. c_0 is for the general groups of disease while c_1 is the specific version. We exclude one class and its corresponding subclasses that are designed for neonates (*certain conditions originating in the perinatal period* (760-779)) in c_0 and c_1. We use the top δ-classes that can be observed in the medical records of ICU patients. We set $\delta = \{5, 10, 15\}$ for c_0 and $\delta = \{5, 10, 15, 20, 25\}$ in c_1. The reason for not including all classes is that the majority rarely occur in the ICU database.

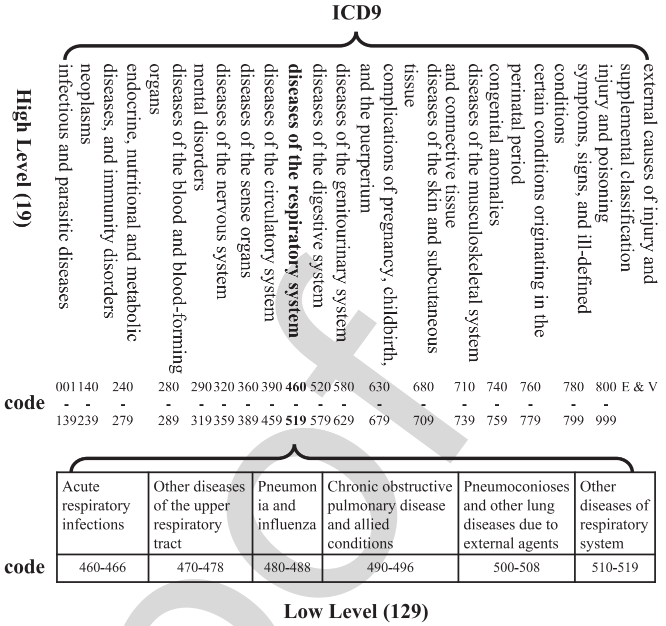

ICD9
High Level (19)
infectious and parasitic diseases
neoplasms
endocrine, nutritional and metabolic diseases, and immunity disorders
diseases of the blood and blood-forming organs
mental disorders
diseases of the nervous system
diseases of the sense organs
diseases of the circulatory system
diseases of the respiratory system
diseases of the digestive system
diseases of the genitourinary system
complications of pregnancy, childbirth, and the puerperium
diseases of the skin and subcutaneous tissue
diseases of the musculoskeletal system and connective tissue
congenital anomalies
certain conditions originating in the perinatal period
symptoms, signs, and ill-defined conditions
injury and poisoning
external causes of injury and supplemental classification
code
001-139
140-239
240-279
280-289
290-319
320-359
360-389
390-459
460-519
520-579
580-629
630-679
680-709
710-739
740-759
760-779
780-799
800-999
E & V
Acute respiratory infections
Other diseases of the upper respiratory tract
Pneumonia and influenza
Chronic obstructive pulmonary disease and allied conditions
Pneumoconioses and other lung diseases due to external agents
Other diseases of respiratory system
code
460-466
470-478
480-488
490-496
500-508
510-519
Low Level (129)

Fig. 2. The hierarchical structure of ICD9. There are two levels, *high level* and *low level*, to describe the disease codes. High level includes more generic disease classification groups (19 groups) while low level codes are more specific (129 groups). Note that we do not use the group of *certain conditions originating in the perinatal period* (760-779) and its related sub-codes because only adult patients are considered.

3.2 Feature Extractions and Encodings

Different parameters will be recorded by medical staff at different time points for each patient, as mentioned above. There are 4,832 parameters in total that can be recorded in the chart, including textual and numerous properties. Only a small set of parameters will be simultaneously recorded at a certain time. 2,158 textual parameters are excluded since it is difficult for most of them to reflect the health conditions of patients even if they are digitalized. The remaining 2,674 numerous parameters are extracted as attributes of the structured data for patients. They can be viewed as the low-level descriptors of the health conditions. Similar to the ICD9 code, only a small number of these parameters are frequently recorded by caregivers in ICU. Thus, we rank the frequencies of parameter occurrences and select the top 500 most often recorded parameters to form the structured data for patients. In this way, chart feature extraction of the ith patient will produce a feature matrix $\boldsymbol{C}_i$. Each row, $\boldsymbol{c}_i$ is a 500-dimensional vector, $i = 1, \ldots, n_i$. n_i is the number of unique time points throughout the entire ICU stay of the ith patient. c_{pq} stores the qth parameter at the pth time point. Note that $\boldsymbol{C}_i$ is often sparse.

Besides the low-level numerous parameters, there are huge volumes of clinical notes for ICU patients in the MIMIC II database. Generally, they are of four types: *radiology reports*, *nursing/other notes*, *medical doctor notes*, and *discharge summary reports*. We use a similar pipeline in [1] to construct note features by using Latent Dirichlet Allocation (LDA) [34]. However, discharge notes are not excluded from our work, which is different to [1]. The reason for this is that explicit mortality outcome does not exert much influence on the ICD9 code classification. According to the pipeline settings, stop words are removed at the beginning of note data pre-processing, followed by a TF-IDF learning that picks out the 500 most informative words from the notes of each patient. The overall dictionary is built upon the amalgamation of the informative words of all patients. The number of topics is set as 50, resulting in a 50-dimensional vector for each patient for each note. Given a note feature matrix $\boldsymbol{N}_i$ for the ith patient, its entry n_{pq} is the proportion of topic q in the pth note. Another difference from [1] is that we do not use weights for each topic because the mortality information is not taken into account in our scenario.

Once feature extractions have been done, two feature matrices for the ith patient are obtained, $\boldsymbol{C}_i$ and $\boldsymbol{N}_i$ representing chart and note features respectively, since two arbitrary patients have different numbers of chart records and medical notes. To make a similarity comparison between two patients, e.g., $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$, a unique representation is achieved by encoding the feature matrices into two vectors of the same length. For simplicity and good performance, the BoW model and its variants, e.g., spatial-temporal pyramid BoW, are pervasively applied to represent text, image and video data in the tasks of retrieval or classification. BoW is a histogram-based statistical method that first requires a dictionary to be created using a clustering algorithm, often KMeans Clustering. The number of centers, also known as the size of the dictionary, are usually set by experiment. The BoW model will first compute a number of distance pairs between each feature and each center. Each feature will be assigned the label of the nearest center. The occurrences of centers will then be counted to form a vector as a unique representation. The size of the vector is the size of the dictionary. A descriptive representation is required to encode the numerous features in MIMIC II. In light of this, we apply BoW as a representation model to encode the features in this work. We have tested different sizes of dictionary, including 50, 100, 200, 300, 500, 1,000, 2,000, and 5,000. We find 500 is a trade-off between effectiveness and efficiency for both chart and note features and fix the dimensions of both chart and note data representations at 500 (shown in Table 1).

3.3 Proposed Algorithm

The notations used in this paper are first summarized to give a better understanding of the proposed algorithm. Matrices and vectors are written as boldface uppercase letters and boldface lowercase letters, respectively. We use the notational convention that defines each data as $d+1$ dimensional, i.e., the intercept term $x_0 = 1$. Therefore, a training dataset is denoted as $\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{(d+1)\times n}$, where n is the number of training samples. Correspondingly the class indicator matrix is represented as $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^T \in \mathbb{R}^{n\times c}$. c is the number of classes. $\boldsymbol{y}_i \in \{0,1\}^c$ is a c dimensional vector. If $\boldsymbol{x}_i$ belongs to the jth class, y_{ij} is 1, otherwise $y_{ij} = 0$, $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, c\}$. A structural incorporating framework can be represented as

$$\min_{\boldsymbol{W}} L(\boldsymbol{X}^T\boldsymbol{W}, \boldsymbol{Y}) + \gamma\Omega(\boldsymbol{W}), \tag{1}$$

where $L(\cdot)$ is a loss function. $\Omega(\cdot)$ is a regularization term while $\gamma \geq 0$ is the regularization parameter. $\boldsymbol{W} \in \mathbb{R}^{(d+1)\times c}$ is a coefficient matrix and its ith row and jth column are denoted as $\boldsymbol{w}^i$ and $\boldsymbol{w}_j$, respectively. To capture intrinsic relationships between features and labels, a sparsity-based norm is usually applied to the regularization term, $\Omega(\boldsymbol{W})$. Thus, if we can properly incorporate a graph structure that reflects the correlations between diseases, the multi-label classification performance can be improved. With this motivation, we need our objective function to have two properties: First, the loss function $L(\cdot)$ should be suitable for multi-label learning and be easy to implement in a large-scale scenario; Second, the sparsity-based norm on $\Omega(\boldsymbol{W})$ should be convex because of the computational issues and global optima. To satisfy these requirements, we design our objective function as follow:

$$\begin{aligned}\min_{\boldsymbol{w}_i} \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} \log\left(1 + \exp(-y_{ij}\boldsymbol{w}_i^T\boldsymbol{x}_j)\right) \\ +\gamma\sum_{i=1}^{c}\sum_{j=1}^{c} a_{ij}\|[\boldsymbol{w}_i, \boldsymbol{w}_j]\|_{2,1},\end{aligned} \tag{2}$$

The $\ell_{2,1}$-norm of the matrix $\boldsymbol{W}$ is defined as $\|\boldsymbol{W}\|_{2,1} = \sum_{i=1}^{d}\|\boldsymbol{w}^i\|_2$. In Eq. (2), we use logistic loss because of its simplicity and suitability for binary classification. Various loss functions have been applied to multi-label learning problems in other works, e.g., least squared loss; however, discussion on the choice of loss function is beyond the scope of this paper. a_{ij} is the entry of an affinity matrix $\boldsymbol{A} \in \mathbb{R}^{c\times c}$ which reflects the relationships between two arbitrary classes (diseases). In the label space, we use cosine similarity to represent the relationships between two arbitrary classes. Recall that the class indicator matrix is defined as $\boldsymbol{Y} \in \mathbb{R}^{n\times c}$. To define the cosine similarity between two classes, we denote $z_i \in \mathbb{R}^n$ as the ith column of $\boldsymbol{Y}$. $\boldsymbol{Y} = [z_1, \ldots, z_c]$. Note that z_i indicates the distribution of the ith class over the training data. Thus, the entry of the affine matrix is defined as follows:

$$a_{ij} = \cos(z_i, z_j) = \frac{< z_i, z_j >}{|z_i| \cdot |z_j|}, \tag{3}$$

where $i, j \in \{1, \ldots, c\}$. In the regularization term, a_{ij} can be regarded as a weight. According to Eq. (3), the more correlated the ith and the jth diseases are, the higher the value of a_{ij} will be. In Eq. (2), a higher a_{ij} will lead to more punishment to $[\boldsymbol{w}_i, \boldsymbol{w}_j]$ with the $\ell_{2,1}$ norm. Optimization will make $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ become more similar in columns and sparse in rows. To fully employ this constraint, the second term in Eq. (2) goes over the entire affinity matrix of the disease correlation. In this way, disease correlation is incorporated into the framework to improve the multi-label classification. Similar ideas have been explored in [35], [36]. Ma et al. characterize different degree of relevance between concepts and events by minimizing $\|[\boldsymbol{w}_i, \boldsymbol{w}_j]\|_{2,p}$. They did not consider utilizing relational graph to improve subsequent performance.

3.4 Optimization

In this section, we give an iterative approach to optimize the objective function. First, we write the objective function shown in Eq. (2) as follows:

$$\begin{aligned} \min_{\boldsymbol{w}_i} \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{n} \log\left(1 + \exp(-y_{ij} \boldsymbol{w}_i^T \boldsymbol{x}_j)\right) \\ + \gamma \sum_{i=1}^{c} \sum_{j=1}^{c} a_{ij} Tr([\boldsymbol{w}_i, \boldsymbol{w}_j]^T \boldsymbol{D}^{ij} [\boldsymbol{w}_i, \boldsymbol{w}_j]), \end{aligned} \tag{4}$$

where D^{ij} is a diagonal matrix with the dth diagonal element as $\frac{1}{2\|[\boldsymbol{w}_i, \boldsymbol{w}_j]^d\|_2}$. $Tr(\cdot)$ is the trace operation of a matrix. The second term in Eq. (4) can be simplified as follows:

$$\begin{aligned} & \sum_{i=1}^{c} \sum_{j=1}^{c} a_{ij} Tr([\boldsymbol{w}_i, \boldsymbol{w}_j]^T \boldsymbol{D}^{ij} [\boldsymbol{w}_i, \boldsymbol{w}_j]) \\ = & \sum_{i=1}^{c} \sum_{j=1}^{c} (a_{ij} \boldsymbol{w}_i^T \boldsymbol{D}^{ij} \boldsymbol{w}_i + a_{ij} \boldsymbol{w}_j^T \boldsymbol{D}^{ij} \boldsymbol{w}_j) \\ = & \sum_{i=1}^{c} \boldsymbol{w}_i^T \left(\sum_{j=1}^{c} a_{ij} \boldsymbol{D}^{ij} \right) \boldsymbol{w}_i + \sum_{j=1}^{c} \boldsymbol{w}_j^T \left(\sum_{i=1}^{c} a_{ij} \boldsymbol{D}^{ij} \right) \boldsymbol{w}_j \\ = & \sum_{i=1}^{c} \boldsymbol{w}_i^T \left(\sum_{j=1}^{c} a_{ij} \boldsymbol{D}^{ij} \right) \boldsymbol{w}_i + \sum_{j=1}^{c} \boldsymbol{w}_j^T \left(\sum_{i=1}^{c} a_{ij} \boldsymbol{D}^{ji} \right) \boldsymbol{w}_j \end{aligned}$$

Because of $a_{ij} = a_{ji}$ and $\boldsymbol{D}^{ij} = \boldsymbol{D}^{ji}$, we rewrite the above equation as:

$$\begin{aligned} & \sum_{i=1}^{c} \sum_{j=1}^{c} a_{ij} Tr([\boldsymbol{w}_i, \boldsymbol{w}_j]^T \boldsymbol{D}^{ij} [\boldsymbol{w}_i, \boldsymbol{w}_j]) \\ = & \sum_{i=1}^{c} \boldsymbol{w}_i^T \left(2 \sum_{j=1}^{c} a_{ij} \boldsymbol{D}^{ij} \right) \boldsymbol{w}_i \end{aligned} \tag{5}$$

By denoting $\boldsymbol{Q}^i = 2 \sum_{j=1}^{c} a_{ij} \boldsymbol{D}^{ij}$, the problem in Eq. (4) will arrive at:

$$\min_{\boldsymbol{w}_i} \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{n} \log\left(1 + \exp(-y_{ij} \boldsymbol{w}_i^T \boldsymbol{x}_j)\right) + \gamma \sum_{i=1}^{c} \boldsymbol{w}_i^T \boldsymbol{Q}^i \boldsymbol{w}_i \tag{6}$$

From the above equation, we observe that the problem in Eq. (6) is unrelated between different $\boldsymbol{w}_i$. Hence, we decouple it to solve the following problem for each $\boldsymbol{w}_i$:

$$\min_{\boldsymbol{w}_i} \frac{1}{n} \sum_{j=1}^{n} \log\left(1 + \exp(-y_{ij} \boldsymbol{w}_i^T \boldsymbol{x}_j)\right) + \gamma \boldsymbol{w}_i^T \boldsymbol{Q}^i \boldsymbol{w}_i \tag{7}$$

We denote $L(\boldsymbol{w}_i) = \frac{1}{n} \sum_{j=1}^{n} \log\left(1 + \exp(-y_{ij} \boldsymbol{w}_i^T \boldsymbol{x}_j)\right)$ and $\Omega(\boldsymbol{w}_i) = \boldsymbol{w}_i^T \boldsymbol{Q}^i \boldsymbol{w}_i$. By using gradient descent, we can update $\boldsymbol{w}_i$ as follows:

$$\boldsymbol{w}_i^{(t+1)} = \boldsymbol{w}_i^{(t)} + \eta \left\{ \nabla_{\boldsymbol{w}_i} L(\boldsymbol{w}_i) + \gamma \nabla_{\boldsymbol{w}_i} \Omega(\boldsymbol{w}_i) \right\} \tag{8}$$

$\eta > 0$ is the learning rate. t is the step index. Because both $L(\boldsymbol{w}_i)$ and $\Omega(\boldsymbol{w}_i)$ are differentiable with respect to $\boldsymbol{w}_i$, we summarize the detailed algorithm to optimize the proposed objective function in Algorithm 1.

Algorithm 1. Algorithm to Solve the Problem in Eq. (2)

```
Data: Data $X \in \mathbb{R}^{(d+1)\times n}$, Parameter $\gamma$, $k$, and label correlation
      matrix $A \in \mathbb{R}^{c\times c}$
Result: $W \in \mathbb{R}^{(d+1)\times c}$
Randomly initialize $W$;
repeat
    For each $i$ and $j$, calculate the diagonal matrix $D^{ij}$,
    where the $d$th diagonal element is$\frac{1}{2\|[w_i, w_j]^d\|_2}$;
    For each $i$, calculate the diagonal matrix $Q^i$ by
    $Q^i = 2\sum_j a_{ij} D^{ij}$;
    For each $i$, update $w_i$ in Eq. (8) using Gradient Descent;
until Convergence;
```

Because the logistic loss function and $\ell_{2,1}$-norm are all convex, the objective function in Eq. (2) converges to the global optima by Algorithm 1. The related proof can be found in Appendix, available in the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2016.2605687.

4 Experiments

In this section, descriptions of all the compared methods will first be given, followed by an introduction to the experiment settings. The experimental results will then be reported and analyzed.

4.1 Experiment Settings

In the experiments, we compare our proposed algorithm with the following approaches:

- *Binary Relevance SVM (BR-SVM)*: Binary Relevance (BR) is a transformation approach, which divides the multi-label classification problem into many binary classification problems. For the task of diagnosis code assignment, BR-SVM has achieved the best performance in terms of accuracy measured by Hamming loss in [21].
- *Hierarchy-based SVM (H-SVM)*: The hierarchy-based SVM considers the class hierarchical structures in learning processes and achieves comparable performance in terms of Hamming loss in [20], [21]. The

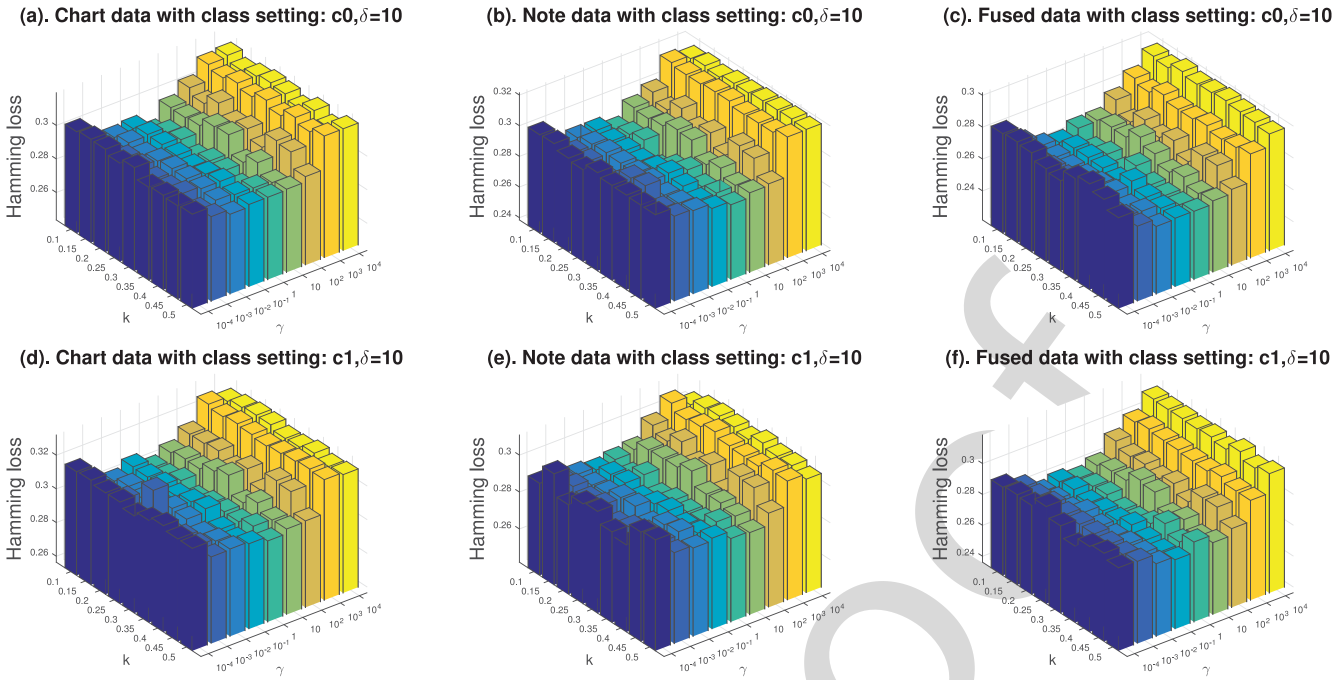


Fig. 3. Performance variations with the different combinations of γs and ks. Top 10 classes ($\delta = 10$) are used in different class settings, c_0 and c_1.

hierarchy of ICD9 codes is available from the NCBO BioPortal [37].

- *Label specIfic FeaTures (LIFT)* [38]: In the multi-label learning framework, LIFT will perform clustering on features with respect to each class, after which training and testing will be conducted by querying the clustering results. Using this method, label-specific features belonging to a certain class will be exploited.
- *Multi-Label kNN (MLkNN)* [13]: ML-kNN is used to learn multi-label k-nearest neighbor classifiers. We tune values of k in the range of $\{8, 9, 10, 11, 12\}$ according to [13] and report the best result in the experiment.
- *RankSVM* [39]: This algorithm is designed to handle multi-label classification problems by using a large margin ranking system. This system has a number of common features with traditional SVMs.
- *SubFeature Uncovering with Sparsity (SFUS)* [40]: This method considers both selecting the most distinctive features in the original feature space and exploiting shared structural information in a subspace. It has been applied in a multi-label learning application that automatically annotates multi-labels to web images.

Since noise may exist in disease correlations, we set a filter parameter k that controls the sparsity of the affinity matrix A. If $a_{ij} < k$, $a_{ij} = 0$. All medical data are randomly and evenly split into two parts for training and testing procedures. In the training phase, 5-fold cross validation and grid search scheme are applied to select the best parameters on training data. In our proposed algorithm, there are two parameters, k and γ. k is a filter parameter that controls the sparsity of the affinity matrix A, while γ is the regularization parameter. In the experiment, k is tuned in {0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5}, while γ is tuned in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$. The learning rate, η, in Eq. (8) is set at 0.001 in all experiments. After the parameter selection, we fit the model with the best parameters on the testing dataset, and report the corresponding results. The parameters of the compared methods are tuned in the same range of $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$, e.g., regularization parameter for the SVM-based methods. Because of the hierarchical structures of ICD9 codes mentioned before, we name c_0 as the most general classification label and c_1 as the more specific label. For c_0, there are 19 disease categories while there are 129 categorizations for c_1. Since only adult records are considered, we exclude the disease group that is designed for neonates at all levels, i.e., *certain conditions originating in the perinatal period* (769-779). As a result, the full class setting at c_0 level includes 18 classes. For c_1 level, we only consider top δ disease codes because some diseases are rarely diagnosed in ICU. We set $\delta = \{5, 10, 15, 18\}$ in c_0 and $\delta = \{5, 10, 15, 20, 25\}$ in c_1. The evaluations are thus conducted in different label settings; for example, the label setting $c_0\delta_5$ means the c_0 with $\delta = 5$ is in use.

Since there are two types of features that are extracted from chart and note data respectively, we concatenate the chart and note features to form the third fused features. All the algorithms are evaluated using the three type of features, i.e., chart features, note features, and their concatenated features. Note that there have so far been many feature fusion strategies, including: *early* fusion, *late* fusion and *multi-stage* fusion. In this paper, we only consider early fusion, in which two types of features, chart and note features, are concatenated. It is worth considering the underlying correlations between two features since the high-level note data are summarized and inferred from low-level chart data. However, this is not the focus of this paper and can be considered for future work.

To evaluate the performance, we have adopted two criteria that are widely used in multi-label learning: *Hamming loss* and *Ranking loss*. The former criterion is an example-based metric that evaluates the errors from either the predictions of

TABLE 2
Performance Comparison Between Our Algorithm and All Compared Methods Using Medical Chart Data Under Different Label Settings

Criteria	Settings		BR-SVM	H-SVM	LIFT	MLkNN	RankSVM	SFUS	Ours
Hamming loss ↓	c_0	δ_5	.369 ± .002	.330 ± .003	.336 ± .002	.339 ± .003	.345 ± .002	.329 ± .002	**.311 ± .001**
		δ_{10}	.333 ± .002	.311 ± .003	.317 ± .002	.317 ± .002	.328 ± .003	.302 ± .002	**.289 ± .011**
		δ_{15}	.275 ± .002	.262 ± .002	.269 ± .002	.269 ± .002	.284 ± .001	.260 ± .002	**.249 ± .001**
		δ_{18}	.237 ± .003	.229 ± .003	.232 ± .001	.224 ± .001	.246 ± .004	.229 ± .004	**.218 ± .001**
	c_1	δ_5	.392 ± .003	.376 ± .002	.375 ± .005	.378 ± .002	.434 ± .008	.405 ± .003	**.367 ± .002**
		δ_{10}	.318 ± .001	.312 ± .002	.313 ± .003	.313 ± .001	.332 ± .004	.337 ± .003	**.302 ± .001**
		δ_{15}	.276 ± .002	.269 ± .002	.271 ± .001	.271 ± .001	.288 ± .001	.302 ± .003	**.264 ± .001**
		δ_{20}	.241 ± .001	.238 ± .002	.237 ± .002	.237 ± .001	.252 ± .001	.266 ± .002	**.222 ± .001**
		δ_{25}	.209 ± .003	.202 ± .001	.210 ± .002	.209 ± .001	.221 ± .001	.239 ± .001	**.195 ± .002**
Ranking loss ↓	c_0	δ_5	.266 ± .001	.244 ± .002	.245 ± .001	.257 ± .001	.282 ± .002	.301 ± .013	**.236 ± .001**
		δ_{10}	.252 ± .001	.232 ± .001	.236 ± .001	.240 ± .001	.264 ± .004	.273 ± .002	**.221 ± .001**
		δ_{15}	.230 ± .002	.218 ± .002	.227 ± .001	.229 ± .001	.253 ± .002	.243 ± .001	**.204 ± .002**
		δ_{18}	.214 ± .001	.202 ± .003	.210 ± .001	.214 ± .003	.237 ± .003	.225 ± .001	**.184 ± .001**
	c_1	δ_5	.343 ± .001	.314 ± .005	.315 ± .003	.320 ± .002	.349 ± .001	.400 ± .002	**.310 ± .003**
		δ_{10}	.303 ± .002	.277 ± .001	.269 ± .002	.272 ± .001	.330 ± .003	.325 ± .006	**.269 ± .001**
		δ_{15}	.287 ± .001	**.256 ± .001**	.262 ± .002	.265 ± .002	.309 ± .004	.293 ± .001	.265 ± .003
		δ_{20}	.275 ± .003	.248 ± .001	**.246 ± .001**	.249 ± .001	.294 ± .002	.279 ± .002	.249 ± .003
		δ_{25}	.244 ± .001	.233 ± .002	.230 ± .001	.232 ± .001	.272 ± .002	.262 ± .001	**.224 ± .003**

Hamming loss and ranking loss are used as metric. The parameters k and γ are fixed at 0.25 and 1, respectively.

wrong labels or from missing predictions. From the definition, we can see that error-free performance will have zero *Hamming loss*, which means there is no difference between the predicted labels and the ground truth. In other words, the smaller the value of *the Hamming loss* is, the better the performance will be. *Ranking loss* takes into account the average fraction of label pairs that are mis-ordered for the object. Similarly, a smaller *Ranking loss* indicates a better performance result. More details of these two criteria can be found in [41]. We repeat the experiments five times and report the average results with standard deviations under eight different label settings for each hierarchy.

4.2 Evaluation Results

Since there are two parameters, i.e., k and γ, in our framework, we conduct an experiment to investigate performance variations with respect to different parameter combinations. Performance variations with different combinations of ks and γs are drawn in Fig. 3. Due to page limitations, we only select the top 10 classes ($\delta = 10$) in each ICD9 hierarchy (c_0 or c_1) for two features and their fusion version. We only consider *Hamming loss* as the metric in this experiment. k varies in a range of $\{0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.5\}$ while $\gamma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4\}$. Note that the smaller the *Hamming loss* value (the shorter bar in

TABLE 3
Performance Comparison Between Our Algorithm and All Compared Methods Using Medical Note Data Under Different Label Settings

Criteria	Settings		BR-SVM	H-SVM	LIFT	MLkNN	RankSVM	SFUS	Ours
Hamming losss ↓	c_0	δ_5	.339 ± .003	.312 ± .001	.305 ± .004	.315 ± .002	.317 ± .007	.303 ± .002	**.295 ± .001**
		δ_{10}	.308 ± .001	.289 ± .002	.293 ± .001	.294 ± .002	.305 ± .002	**.280 ± .002**	.281 ± .002
		δ_{15}	.267 ± .001	.250 ± .001	.251 ± .003	.253 ± .002	.264 ± .002	**.243 ± .002**	.247 ± .001
		δ_{18}	.229 ± .002	.207 ± .002	.214 ± .002	.208 ± .003	.226 ± .003	.212 ± .001	**.195 ± .002**
	c_1	δ_5	.371 ± .002	.343 ± .002	.343 ± .001	.363 ± .002	.408 ± .004	.403 ± .001	**.325 ± .001**
		δ_{10}	.304 ± .003	.290 ± .002	.288 ± .001	.297 ± .001	.314 ± .001	.316 ± .002	**.285 ± .001**
		δ_{15}	.264 ± .002	.256 ± .001	.254 ± .001	.260 ± .001	.277 ± .001	.284 ± .001	**.253 ± .001**
		δ_{20}	.236 ± .004	.234 ± .001	.224 ± .001	.228 ± .001	.245 ± .002	.249 ± .001	**.223 ± .001**
		δ_{25}	.210 ± .007	.190 ± .002	.197 ± .001	.201 ± .001	.222 ± .001	.220 ± .001	**.197 ± .001**
Ranking loss ↓	c_0	δ_5	.227 ± .001	.221 ± .002	**.205 ± .003**	.205 ± .002	.219 ± .001	.256 ± .003	.219 ± .002
		δ_{10}	.238 ± .001	**.212 ± .003**	.218 ± .002	.216 ± .001	.237 ± .002	.220 ± .003	.221 ± .002
		δ_{15}	.219 ± .002	.206 ± .003	.198 ± .001	.197 ± .001	.221 ± .002	.212 ± .001	**.189 ± .001**
		δ_{18}	.197 ± .002	.190 ± .001	.181 ± .003	.182 ± .001	.205 ± .001	.194 ± .002	**.169 ± .002**
	c_1	δ_5	.324 ± .002	.309 ± .002	.302 ± .003	.302 ± .002	.313 ± .002	.406 ± .007	**.291 ± .004**
		δ_{10}	.276 ± .003	**.249 ± .003**	.251 ± .002	.250 ± .001	.278 ± .004	.260 ± .001	.267 ± .002
		δ_{15}	.265 ± .001	.240 ± .002	.244 ± .001	.243 ± .001	.284 ± .006	.248 ± .002	**.231 ± .004**
		δ_{20}	.246 ± .001	.225 ± .003	.227 ± .001	.227 ± .001	.265 ± .002	.250 ± .001	**.216 ± .003**
		δ_{25}	.223 ± .003	.203 ± .002	.207 ± .001	.206 ± .001	.239 ± .002	.233 ± .001	**.200 ± .004**

Hamming loss and ranking loss are used as metric. The parameters k and γ are fixed at 0.25 and 1, respectively.

TABLE 4
Performance Comparison Between Our Algorithm and All Compared Methods Using Fused Medical Data (Chart and Note) Under Different Label Settings

Criteria	Settings		BR-SVM	H-SVM	LIFT	MLkNN	RankSVM	SFUS	Ours
Hamming loss ↓	c_0	δ_5	.329 ± .002	.292 ± .001	.311 ± .004	.328 ± .002	.349 ± .004	.301 ± .002	**.282** ± **.001**
		δ_{10}	.303 ± .002	.270 ± .002	.298 ± .002	.306 ± .002	.295 ± .003	.275 ± .002	**.261** ± **.001**
		δ_{15}	.257 ± .002	.234 ± .001	.256 ± .002	.262 ± .002	.261 ± .001	.238 ± .002	**.231** ± **.001**
		δ_{18}	.213 ± .003	.203 ± .001	.219 ± .002	.217 ± .001	.223 ± .003	.207 ± .004	**.200** ± **.003**
	c_1	δ_5	.362 ± .003	.332 ± .001	.344 ± .003	.369 ± .004	.424 ± .007	.388 ± .002	**.318** ± **.001**
		δ_{10}	.300 ± .002	.297 ± .002	.290 ± .002	.305 ± .003	.337 ± .009	.307 ± .003	**.278** ± **.001**
		δ_{15}	.259 ± .001	.251 ± .003	.254 ± .001	.265 ± .002	.269 ± .002	.278 ± .002	**.247** ± **.002**
		δ_{20}	.235 ± .002	.230 ± .001	.225 ± .001	.232 ± .002	.239 ± .001	.244 ± .001	**.219** ± **.002**
		δ_{25}	.202 ± .002	.198 ± .002	.199 ± .002	.205 ± .002	.211 ± .001	.216 ± .001	**.194** ± **.001**
Ranking loss ↓	c_0	δ_5	.226 ± .002	.203 ± .001	**.199** ± **.002**	.200 ± .002	.226 ± .001	.265 ± .006	.214 ± .002
		δ_{10}	.238 ± .001	**.209** ± **.002**	.211 ± .001	.210 ± .001	.246 ± .003	.223 ± .005	.215 ± .001
		δ_{15}	.219 ± .001	**.182** ± **.002**	.194 ± .001	.193 ± .001	.230 ± .002	.217 ± .002	.188 ± .002
		δ_{18}	.202 ± .003	.171 ± .002	.177 ± .001	.178 ± .002	.214 ± .003	.199 ± .002	**.168** ± **.001**
	c_1	δ_5	.315 ± .002	.268 ± .002	.288 ± .002	.288 ± .002	.311 ± .002	.354 ± .006	**.274** ± **.002**
		δ_{10}	.273 ± .002	.242 ± .002	.240 ± .002	.238 ± .002	.285 ± .006	.328 ± .019	**.228** ± **.001**
		δ_{15}	.264 ± .003	.231 ± .001	.232 ± .002	.231 ± .002	.281 ± .005	.268 ± .005	**.223** ± **.003**
		δ_{20}	.248 ± .003	.217 ± .002	.218 ± .001	.217 ± .001	.263 ± .002	.258 ± .003	**.201** ± **.003**
		δ_{25}	.227 ± .002	.193 ± .001	.199 ± .001	.198 ± .001	.243 ± .002	.240 ± .004	**.189** ± **.003**

Hamming loss and ranking loss are used as metric. The parameters k and γ are fixed at 0.25 and 1, respectively.

Fig. 3), the better performance. From all the Figs. 3a, 3b, 3c, 3d, 3e, 3f, we can observe that highest and lowest values of γ (e.g., $10^{-4}, 10^{-3}, 10^3, 10^4$) are detrimental to the performance. Medium values of γ, such as $10^{-1}, 1, 10^1$ usually yield good performance results. On the contrary, there is not an obvious pattern for the filter parameter$k$. However, the best performance result (the shortest bar) is usually identified when $\gamma = 1$ and $k = 0.25$. We have observed similar trends and results for the other class settings. As a result, we fix $\gamma = 1$ and $k = 0.25$ as the best parameter combination in the rest of experiments. To consider the effectiveness of our algorithm, we compare all the algorithms detailed above and report the results of the different types of features in Tables 2, 3, and 4. Note that the parameters are fixed ($\gamma = 1$ and $k = 0.25$). Average results with standard deviations are represented in the tables. From the tables, we make the following observations: Irrespective of the type of features used, our proposed algorithm performs better than all others in terms of *Hamming loss* and *Ranking loss* in most of the different class settings. In each classification hierarchy (c_0 and c_1), it is interesting to find that both criteria mostly decrease for all algorithms with the increase of number of classes (e.g., δ varies from 5 to 18 in c_0). For example, the *Hamming loss* of the proposed method is 0.311 when $\delta = 5$ in

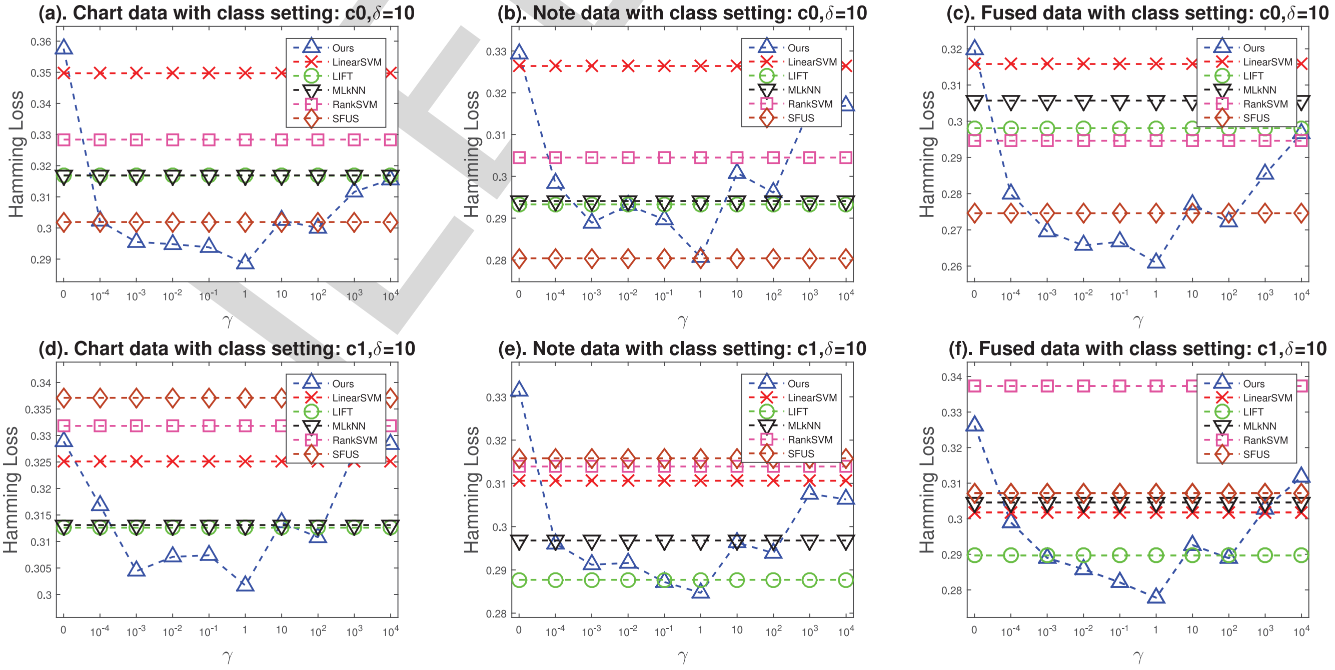
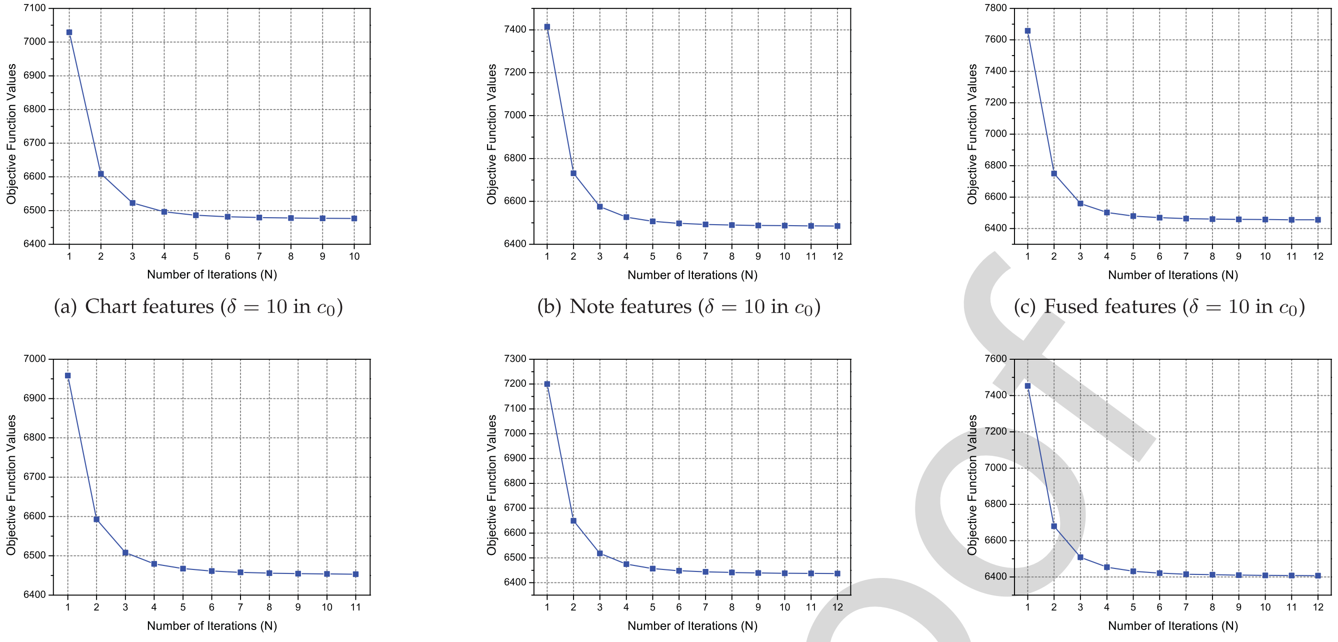


Fig. 4. Performance variations with respect to different γs. We test chart, note, and fused data under all class setting. Due to the page limit, we only show the results under class setting $c_0\delta_{10}$ and $c_1\delta_{10}$. k is fixed at 0.25. Performance results of all compared methods are also drawn in each figure.



(a) Chart features ($\delta = 10$ in c_0)

(b) Note features ($\delta = 10$ in c_0)

(c) Fused features ($\delta = 10$ in c_0)

(d) Chart features ($\delta = 10$ in c_1)

(e) Note features ($\delta = 10$ in c_1)

(f) Fused features ($\delta = 10$ in c_1)

Fig. 5. The convergence curves of the objective function values in (2) using algorithm 1 on MIMIC II. We test chart, note, and fused data under all class setting. Due to the page limit, we only show the results under class setting $c_0\delta_{10}$ and $c_1\delta_{10}$. k is fixed at 0.25.

c_0. However, the value decreases to 0.249 and 0.218 when $\delta = 15$ and $\delta = 18$, respectively. Exceptions can be observed for note and fused features ($\delta = 10$ in $c0$) measured by *Ranking loss*. In most cases, note features have better results than chart features. This may because the note data contain descriptive and predictive information from medical experts. On the other hand, our method achieves better performance by using fused features than by using each of them separately. For instance, the biggest margin is observed at *Ranking loss* when $\delta = 10$ in chart or note features have higher values (i.e., 0.2690 for the chart data and 0.2665 for the note data, respectively). However, the improvement achieved by feature fusion in all settings is sometimes limited, which is the result of the simple early fusion strategy (the concatenation of two features). From the results in Tables 2, 3, and 4, we can observe that our algorithm performs much better than BR-SVM, which does not consider the correlation between diseases. Compared to the other methods, which take correlation into account, it is worth noting that our proposed algorithm still yield better performance results in the most cases. To validate the effectiveness of the disease correlation embedding via a graph structure, we fix k as 0.25 and add $\gamma = 0$ as in the previously tested range. When $\gamma = 0$, there is no contribution from disease correlation mining in the objective function in Eq. (2). The entire work is then equivalent to a standard logistic regression model for a multi-label classification problem. In Fig. 4, the classification performance of all the methods is drawn in each sub-figure to give a better understanding of how and when our method achieves superior performance than its counterparts. Note that none of the other compared algorithms change their performance with the variation of γ. They are shown as horizontal dashed lines in the figures. In all the sub-figures, our proposed method has a higher *Hamming loss* when $\gamma = 0$. With the changes to γ, the value is minimized at a certain γ (usually $\gamma = 1$). However, dramatic increases are observed when much bigger γs are engaged. This experiment validates that a proper fraction of disease correlation embedding is indeed beneficial to multi-label learning. With this graph structure, our framework stands out against all other algorithms in most cases.

Lastly, we conduct empirical experiments to demonstrate the convergence of our proposed algorithm. We first test the number of iterations of our algorithm and report the results in Fig. 5. Due to page limitations, we only select the top 10 classes ($\delta = 10$) in each ICD9 hierarchy (c_0 or c_1) for two features and their fusion version. From the experiments, we see that the objective function value converges within a few steps (approximately 12 iterations in most cases). To test the efficiency of the proposed algorithm, we fix two parameters ($k = 0.25$ and $\gamma = 1$) under the full class setting ($c_0\delta_{18}$). We

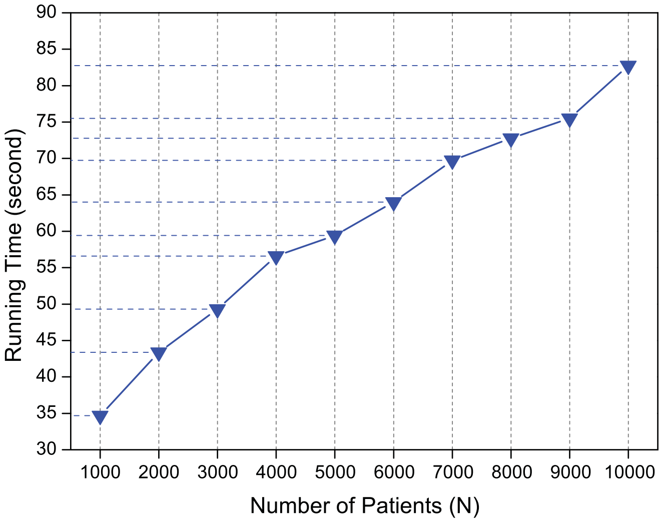


Fig. 6. Averaged runtime records with the increase in the number of patient data. *X*-axis is the number of data, while *Y*-axis denotes the corresponding runtime of our algorithm in second.

TABLE 5
Objective Function Value Variance w.r.t. Different Initializations of W Under All Class Settings: Setting Diagonal Elements of W to 0.5, 1, 2 (0 for Other Elements), and Setting All the Elements of W to 0.5, 1, 2, and Random Values

	$c_0\delta_5$	$c_0\delta_{10}$	$c_0\delta_{15}$	$c_0\delta_{18}$	$c_1\delta_5$	$c_1\delta_{10}$	$c_1\delta_{15}$	$c_1\delta_{20}$	$c_1\delta_{25}$
1st init.	3260.7	6455.7	8524.4	9078.7	3590.1	6407.6	8691.4	10580.3	11773.9
2nd init.	3260.7	6455.7	8524.4	9078.7	3590.1	6407.6	8691.4	10580.3	11773.9
3rd init.	3260.7	6455.7	8524.4	9078.7	3590.1	6407.6	8691.4	10580.3	11773.9
4th init.	3260.7	6455.7	8524.4	9078.7	3590.1	6407.6	8691.4	10580.3	11773.9
5th init.	3260.7	6455.7	8524.4	9078.7	3590.1	6407.6	8691.4	10580.3	11773.9
6th init.	3260.7	6455.7	8524.4	9078.7	3590.1	6407.6	8691.4	10580.3	11773.9
7th init.	3260.7	6455.7	8524.4	9078.7	3590.1	6407.6	8691.4	10580.3	11773.9

In this experiment, all the class settings are tested.

increase the number of patient data from 1,000 to 10,000 and record the corresponding running time of the algorithm. In each run, the max iteration number is set to 20. We repeat the test 10 times and report the averaged result in Fig. 6. To empirically demonstrate that our algorithm converges to a global optima, we design an experiment which tests different initializations of $\boldsymbol{W}$ in Algorithm 1. We initialize W in different seven ways: setting all the diagonal elements of W to 0.5, 1, 2 (0 for other elements), and setting all the elements of W to 0.5, 1, 2, and random values. All the class settings are tested. From Table 5, we can see the objective function values of different seven initialization ways are the same for each class setting. It can be seen that our algorithm always converges to the global optimum regardless of the different initializations.

5 Conclusions

The aim of this paper has been to learn ICU patient diagnosis labels and automatically conduct annotation according to the patient data. We extracted medical chart and note data from a publicly available large-scale Intensive Care Unit database, i.e., MIMIC II. The Bag-of-words model was applied to encode both chart and note features. With the goal of achieving acceptable multi-label classification performance, we proposed an algorithm based on sparsity regularization to exploit and utilize disease correlations via a graph structure. The entire framework is convex and leads to a guaranteed global optima. Our algorithm improves multi-label classification performance by capturing the disease correlations. Extensive experiments demonstrate that the proposed method, with the help of successful disease correlation embedding, learns the diagnostic codes of patients more effectively than all other compared approaches.

Acknowledgments

This work was supported by the Australian Research Council Discovery Project under Grant No. DP 140100104. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Australian Research Council.

References

[1] M. Ghassemi, et al., "Unfolding physiological state: Mortality modelling in intensive care units," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Min.*, 2014, pp. 75–84.

[2] A. E. Johnson, A. A. Kramer, and G. D. Clifford, "A new severity of illness scale using a subset of acute physiology and chronic health evaluation data elements shows comparable predictive accuracy," *Critical Care Med.*, vol. 41, no. 7, pp. 1711–1718, 2013.

[3] C. K. Loo and M. Rao, "Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1589–1593, Nov. 2005.

[4] L. V. Lita, S. Yu, R. S. Niculescu, and J. Bi, "Large scale diagnostic code classification for medical patient records." in *Proc. Int. Joint Conf. Natural Lang. Process.*, 2008, pp. 877–882.

[5] O. Frunza, D. Inkpen, and T. Tran, "A machine learning approach for identifying disease-treatment relations in short texts," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 801–814, Jun. 2011.

[6] Y. Park and J. Ghosh, "Ensembles of α-trees for imbalanced classification problems," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 131–143, Jan. 2014.

[7] C. Friedman, L. Shagina, Y. Lussier, and G. Hripcsak, "Automated encoding of clinical documents based on natural language processing," *J. Amer. Med. Informat. Assoc.*, vol. 11, no. 5, pp. 392–402, 2004.

[8] P. Ruch, R. Baud, and A. Geissbhler, "Evaluating and reducing the effect of data corruption when applying bag of words approaches to medical records," *Int. J. Med. Informat.*, vol. 67, no. 13, pp. 75–83, 2002.

[9] P. Ordonez, T. Armstrong, T. Oates, and J. Fackler, "Using modified multivariate bag-of-words models to classify physiological data," in *Proc. IEEE Int. Conf. Data Min. Workshop*, Dec. 2011, pp. 534–539.

[10] F. Wang, N. Lee, J. Hu, J. Sun, and S. Ebadollahi, "Towards heterogeneous temporal clinical event pattern discovery: A convolutional approach," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Min.*, 2012, pp. 453–461.

[11] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[12] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, 2005, pp. 195–200.

[13] M.-L. Zhang and Z.-H. Zhou, "ML-kNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[14] M.-L. Zhang, "Ml-rbf: RBF neural networks for multi-label learning," *Neural Process. Lett.*, vol. 29, no. 2, pp. 61–74, 2009.

[15] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learning*, vol. 85, no. 3, pp. 333–359, 2011.

[16] G. Tsoumakas, I. Katakis, and L. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[17] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.

[18] X. Chang, H. Shen, S. Wang, J. Liu, and X. Li, "Semi-supervised feature analysis for multimedia annotation by mining label correlation," in *Proc. 18th Pacific-Asia Conf. Advances. Knowl. Discovery Data Min.*, 2014, pp. 74–85.

[19] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

[20] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and N. Elhadad, "Diagnosis code assignment: Models and evaluation metrics," *J. Amer. Med. Informat. Assoc.*, vol. 21, no. 2, pp. 231–237, 2014.

[21] D. Zufferey, T. Hofer, J. Hennebert, M. Schumacher, R. Ingold, and S. Bromuri, "Performance comparison of multi-label learning algorithms on clinical data for chronic diseases," *Comput. Biol. Med.*, vol. 65, pp. 34–43, 2015.

[22] J. C. Ferrao, F. Janela, M. D. Oliveira, and H. M. Martins, "Using structured EHR data and SVM to support ICD-9-CM coding," in *Proc. IEEE Int. Conf. Healthcare Informat.*, 2013, pp. 511–516.

[23] S. V. Pakhomov, J. D. Buntrock, and C. G. Chute, "Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques," *J. Amer. Med. Informat. Assoc.*, vol. 13, no. 5, pp. 516–525, 2006.

[24] X. Kong, B. Cao, and P. S. Yu, "Multi-label classification by mining label and instance correlations from heterogeneous information networks," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Min.*, 2013, pp. 614–622.

[25] Y. Yan, G. Fung, J. G. Dy, and R. Rosales, "Medical coding classification by leveraging inter-code relationships," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Min.*, 2010, pp. 193–202.

[26] Z. Che, D. Kale, W. Li, M. T. Bahadori, and Y. Liu, "Deep computational phenotyping," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2015, pp. 507–516.

[27] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Ann. Statist.*, pp. 3468–3497, 2009.

[28] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l2,1-norms minimization," in *Proc. Advances Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[29] S. Wang, Y. Yang, Z. Ma, X. Li, C. Pang, and A. G. Hauptmann, "Action recognition by exploring data distribution and feature correlation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1370–1377.

[30] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 723–742, Apr. 2012.

[31] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, and A. G. Hauptmann, "Multi-feature fusion via hierarchical regression for multimedia analysis," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 572–581, Apr. 2013.

[32] X. Chang, F. Nie, Y. Yang, and H. Huang, "A convex formulation for semi-supervised multi-label feature selection," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1171–1177.

[33] M. Saeed, et al., "Multiparameter intelligent monitoring in intensive care II (mimic-II): A public-access intensive care unit database," *Critical Care Med.*, vol. 39, no. 5, 2011, Art. no. 952.

[34] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learning Res.*, vol. 3, pp. 993–1022, 2003.

[35] Z. Ma, Y. Yang, N. Sebe, and A. G. Hauptmann, "Knowledge adaptation with partially shared features for event detection using few exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1789–1802, Sep. 2014.

[36] X. Cai, F. Nie, W. Cai, and H. Huang, "New graph structured sparsity model for multi-label image annotations," in *Proc. IEEE Int. Conf. Comput. Vis. Australia*, 2013, pp. 801–808.

[37] P. L. Whetzel, et al., "Bioportal: Enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications," *Nucleic Acids Res.*, vol. 39, no. suppl. 2, pp. W541–W545, 2011.

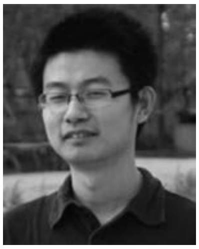
[38] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1609–1614.

[39] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. Advances Neural Inf. Process. Syst.*, 2001, pp. 681–687.

[40] Z. Ma, F. Nie, Y. Yang, J. R. Uijlings, and N. Sebe, "Web image annotation via subspace-sparsity collaborated feature selection," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1021–1030, Aug. 2012.

[41] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Tech. rep., Oregon State University, Corvallis, OR, 2010.

Sen Wang received the ME degree in computer science from Jilin University, China, and the PhD degree from The University of Queensland, in 2014. He is currently a lecturer in the School of Information and Communication Technology, Griffith University. His research interests include machine learning, data mining, biomedical application, and social media mining.



Xiaojun Chang is working toward the PhD degree in the Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Australia. He has been working as a visiting student in the Language Technologies Institute, Carnegie Mellon University, since March, 2014. His research interests include machine learning, data mining, and computer vision.

Xue Li received the MSc and PhD degrees from The University of Queensland (UQ) and the Queensland University of Technology, in 1990 and 1997 respectively. He is currently a full professor in the School of Information Technology and Electrical Engineering, UQ, Brisbane, Queensland, Australia. His major areas of research interests and expertise include data mining, multimedia data security, database systems, and intelligent web information systems. He is a member of the ACM, the IEEE, and the SIGKDD.

Guodong Long received the BS and MS degrees in computer science from the National University of Defence Technology, Changsha, China, in 2002 and 2008, respectively, and the PhD degree in information technology from the University of Technology, Sydney (UTS), in 2014. He is currently a research lecturer in the Research Centre for Quantum Computing and Intelligence Systems (QCIS), UTS. His research interests include data mining, machine learning, and database and cloud computing.

Lina Yao received the BE degree from Shandong University and the MSc and PhD degrees in computer science from the University of Adelaide. She is currently a lecturer in the School of Computer Science and Engineering, University of New South Wales. Her research interests include data mining, Internet of Things, ubiquitous and pervasive computing, and service computing.

Quan Z. Sheng received the PhD degree in computer science from the University of New South Wales, in 2006. He is a full professor in the School of Computer Science, University of Adelaide, and head of the Advanced Web Technologies Research Group. His research interests include big data analytics, distributed computing, Internet computing, and Web of Things. He received the Australian Research Council Future Fellowship in 2014, Chris Wallace Award for outstanding research contribution in 2012, and Microsoft Research Fellowship in 2003. He is the author of more than 210 publications.