Local analgesia adverse effects prediction using multi-label classification

Guangzhi Qu a,*, Hui Wu a, Craig T. Hartrick b, Jianwei Niu c

a Computer Science and Engineering Department, Oakland University, Rochester, MI 48309, USA
b Anesthesiology Research, School of Medicine, Oakland University Rochester, MI 48309, USA
c School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Abstract

It is desirable to determine minimal effective initial local anesthetic bolus required to provide satisfactory analgesia following surgery. A way to predict potential adverse effects based on the type of anesthetic and initial bolus amount administered would be a significant contribution to personalized medicine. In this work, we propose new methods for multi-label classification to predict adverse effects in order to help doctors make appropriate treatment decisions. In this endeavor, the Pair-Dependency Multi-Label Bayesian Classifier (PDMLBC) and Complete-Dependency Multi-Label Bayesian Classifier (CDMLBC) models are proposed as classifiers that take into account the impact of features on the dependency between labels. We evaluated the proposed models on 36 patients who had recently received arthroscopic shoulder surgery. The experimental results show that the CDMLBC model outperforms other existing methods in multi-label classification.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Arthroscopic shoulder surgery is known to be a particularly painful stimulus that historically has often required postoperative hospitalization for intravenous opioid analgesia. Improvements in postoperative analgesia have permitted these procedures to be performed in the ambulatory setting [29]. Consequently, continuous interscalene infusion of ropivacaine using a disposable, programmable pump has become standard care following shoulder surgery.

Typically, relatively large boluses of local anesthetic (20–40 ml) are used to initiate the brachial plexus block and a continuous infusion of local anesthetic (4–8 ml/h) is then added to maintain the analgesia. This approach, while clearly effective from an analgesic perspective, is associated with a number of adverse effects. The close proximity of the phrenic nerve, recurrent laryngeal nerve, sympathetic chain, and other portions of the brachial plexus serving the distal extremity predispose patients to transient unwanted diaphragmatic paresis, dysphonia, dysphagia, Horner’s syndrome (miosis, ptosis, enophthalmos, conjunctival injection, nasal congestion), and hand numbness/weakness. While these annoying effects are usually tolerated, they occasionally result in hospitalization for symptom control. Patients with pre-existing pulmonary conditions may not tolerate the sympathetomy or the diaphragmatic paralysis, both of which may occur in more than 80% of subjects [34]. Respiratory compromise or poor pain control are the most common reasons for unscheduled hospitalization. Although reducing the initial local anesthetic bolus from 40 ml to 20 ml in one study still resulted in a 100% incidence of diaphragmatic paralysis [25], decreasing the initial bolus further may result in reduced spread to adjacent neural structures and presumably fewer adverse effects [22]. The reduced mass of local anesthetic also has the added safety benefit of reducing the potential for local anesthetic toxicity.

To properly decide on the initial bolus of anesthetic, doctors must consider many factors, including the patient’s demographic information, health history, pain location and intensity, etc. A way to predict potential adverse effects based on the type of anesthetic and initial bolus amount would be a significant contribution to personalized medication. Currently, some statistical methods are employed in studying the correlation between a patient’s pain symptoms, treatment (initial bolus amount) and specific adverse effects. When choosing a treatment plan, a doctor considers the effects the treatment may have on the patient based on the doctor’s knowledge and past experience. Though this strategy is widely used, it has many disadvantages. First, the treatment is mostly determined by the doctor’s empirical experience. If the doctor is not familiar with some kinds of effects, these effects will not be considered. If an overlooked effect is seriously adverse, there can be very dangerous consequences. Second, many adverse effects are correlated with each other. If the number of adverse effects is small, doctors can handle them easily, but if there are many, it is difficult or even impossible for a human being to consider all factors comprehensively. These problems are obstacles to the development of personalized medication.

Keywords:
Multi-label classification
Adverse effects prediction
Personalized medication
Anesthesiology
Pain medicine

* Corresponding author.
E-mail addresses: gqu@oakland.edu (G. Qu), hwu@oakland.edu (H. Wu), chartrick@beaumont.edu (C.T. Hartrick), niujianwei@buaa.edu.cn (J. Niu).
Intuitively, we can consider each adverse effect as a label to describe one aspect of response by a patient during the treatment. For each patient, the type and number of adverse effects vary based on the patient’s demographic information, historical assessment, and clinical lab results. In other words, the problem of predicting adverse effects can be converted into a typical multi-label classification problem. In this work, we propose new methods of multi-label classification to predict adverse effects in order to help doctors make appropriate treatment decisions. Multi-label classification methods have been widely used in many applications [31] such as text mining [16,7,12,15,26,33], bioinformatics [6,17,18], and scene classification [2]. One conventional approach of multi-label classification is binary relevance (BR) learning [3]. The basic idea of BR is to transform a multi-label classification problem into multiple single-label classification problems. During the training period, BR forms a training data set for each label. Data instances having the specified label are put in the positive group while the rest are put in the negative group. Based on the specified training data set, a distinct binary classifier is learned for each label. During testing, each binary classifier will yield a prediction confidence value, which will be fused into a combination model to make the final decision of whether the label can be assigned to the testing instance or not [1,11].

Though many classification methods have been proposed for multi-label processing, the correlations among labels, and especially the impact the features have on these correlations, are rarely considered. In this endeavor, we propose Bayesian based multi-label classification models that take into account the dependency between labels. Specifically, the Pair-Dependency Multi-Label Bayesian Classifier (PDMBBC) and the Complete-Dependency Multi-Label Bayesian Classifier (CDMLBC) are designed to provide accurate multi-label classification.

The rest of the paper is organized as follows. Section 2 introduces the related works in multi-label classification and the motivation behind the proposed new models. In Sections 3 and 4, we present the methodology and propose the models, i.e., PDMBBC and CDMLBC. Section 5 describes the evaluation metrics, evaluation procedure, experimental results, and conclusions.

2. Related work

Recently, multi-label classification problems have drawn a lot of attention from both academic and industrial sectors due to the existence of multiple labels on data records. Consequently, many traditional single-label classification techniques have been extended to attempt classification of multiple labels. Typical approaches include probabilistic models, logistic regression, K-nearest neighbors (KNN) and decision trees.

Among approaches that use probabilistic models, research focuses on how to represent correlations between labels. For example, Zhang et al. assume independence between labels and the desired labels are chosen according to their confidence levels [37]. Other approaches assume each label is subject to a specific distribution [36] such as conditional random field (CRF) [13], Diriclet distribution [19], multinomial distribution [33], or Gaussian distribution [28]. In McCallum’s approach [16], every document is modeled by a two-layer generative process. The first layer creates a label distribution for each document, and the second layer combines the label distribution and each document’s word distribution to generate words for this document. The model aims at determining the document’s original label distribution:

\[
\hat{\mathbf{z}} = \arg \max_{\mathbf{z}} P(\mathbf{z} | d)
\]

where \(d\) denotes a testing document, and \(\hat{\mathbf{z}}\) is a binary bit-vector. If one label is involved in document \(d\), its corresponding value in \(\hat{\mathbf{z}}\) is equal to 1; otherwise, its value is 0. By applying Bayes’ rule, Eq. (1) is transformed to

\[
P(\mathbf{z} | d) = P(E_{I_l} | \mathbf{z}) \prod_{c \in C} \alpha_c \cdot P(W_c | \mathbf{z})
\]

where \(C\) is the set of all labels and \(\alpha_c\) is \(c\)’s weight value in the label distribution \(\mathbf{z}\). Expectation–Maximization (EM) was used to estimate these parameters in their approach.

Logistic regression was also introduced as a means of solving the multi-label classification problem. Cheng and Hüllermeier combined KNN and logistic regression for multi-label classification [5], improving KNN quality by considering vote similarities. In the traditional KNN method, all votes are weighted equally. In their improved approach, a kernel function is formed to evaluate the weight of each vote. Higher similarity among neighbors helps boost the weights of the votes. Another improvement was achieved by combining the vote weights of all neighbors forming the intercept, which is a constant in logistic regression. By this method, a binary classifier was built for each label. They finally proposed a multi-label classifier in which the correlations among labels are organized by the linear equation of logistic regression. If label \(l_i\) has a positive impact on label \(l_j\), the coefficient of \(l_j\) in \(l_i\)’s logistic regression is positive; otherwise, it is negative. Fujino and Isozaki [10] studied the multi-label issue in patent mining task. They built a multi-label classifier similarly by using logistic regression as a binary classifier.

KNN has been extended for multi-label classification by many researchers as well. Spyromitros et al. made two extensions to the traditional BR + KNN multi-label classification: BRK-KNN-a and BRK-KNN-b [27]. BRK-KNN-a resolves the disadvantage of empty output in BR + KNN, and BRK-KNN-b utilizes the average size of \(k\) nearest neighbors’ multi-labels to determine the size of output. Brinker and Hüllermeier also used KNN-based binary relevance learning for multi-label ranking [4]. ML-KNN, proposed by Zhang and Zhou in [38], combines KNN and Bayesian rule in the following way:

\[
\mathbf{y}^* = \arg \max_{\mathbf{y} \in \{0,1\}^{|Y|}} P(H_t^{b} | E_{I_{l_i}}^c | C_t) \cdot l \in Y
\]

where \(C_t\) denotes the number of \(t\)’s neighbors which can be classified as the class labeled \(l\), and \(Y\) is the entire label set. \(H_t\) is the event that test instance \(t\) has label \(l\). If \(b\) is equal to 1, it means that \(t\) has label \(l\); otherwise, \(H_t^b\) indicates \(t\) does not has label \(l\). Since \(b\) has only two values, then if \(P(H_t^c | E_{I_{l_i}}^c ) \geq P(H_t^b | E_{I_{l_i}}^c )\), label \(l\) can be assigned to the testing instance \(t\). According to their experimental results, ML-KNN works better than many ensemble based multi-label classification approaches including Boostexter [26], multi-label decision tree ADTBoost. MH [7] and the multi-label kernel method Rank-SVM [8]. Hence, in our experimental evaluation, we compare our approach mainly with the ML-KNN method to demonstrate the effectiveness of the new approach.

Vens et al. studied decision trees for hierarchical multi-label classification [35]. They classified related methods into three categories. One is the single-label classification (SC) approach, where there is a binary classifier corresponding to each label. The second category is Hierarchical Single-label Classifier (HSC). The difference between SC and HSC is that for a given class \(c\), the training set of SC is formed by labeling the instances in \(c\) with 1 and all the others 0, while HSC labels its instances and their descendant labels’ instances with 1 and the others 0. Basically,
HSC utilizes the Hierarchical Category to label instances. The third category, called HMC, relabels every example only once and uses the Hierarchical Category to determine all labels for an instance. Clare and King changed the feature selection criterion of C4.5 to deal with multi-label classification [6].

Besides the above techniques, support vector machine [14,9,8,24] and ensemble methods [26,7,32,21,20] based approaches have also been proposed to solve multi-label classification problem.

3. Problem statement

Let \( F = \{ f_1, f_2, \ldots, f_J \} \) be a feature set, \( C = \{ c_1, c_2, \ldots, c_m \} \) (\( m \in \mathbb{R} \)) be a label set, \( C^* \) be the power set of \( C \) except \( 0 \), and \( W = (w_1, w_2, \ldots, w_n) \in F^n \) (\( n \in \mathbb{R} \)) be an instance. We define two tasks of multi-label classification.

Task 1 is to choose a proper set of labels for an instance. For example, doctors need to know the possible adverse effects of a treatment on a patient. In this scenario, the selected adverse effects should reflect the patient's response to the local anesthetic injection. We noticed that the number of adverse effects could be different for each individual patient. We formulate the objective function of Task 1 as follows:

\[
\arg \max_{C^* \in C^*} P(C^* | W)
\]  

(4)

Task 2 is to select a predefined number of labels for an instance. For example, doctors may need to examine the top \( k \) (\( k = 3 \)) adverse effects for a patient. In this scenario, the objective function is defined as

\[
\arg \max_{C^* \subset C, |C^*| = k} P(C^* | W)
\]  

(5)

where \( k \) is the demanded number of labels.

In the following sections, we will extend the conventional Naive Bayesian classifier to handle these two tasks separately.

4. Multi-label Bayesian classifier

We claim that there exists dependency among the labels and these correlations are affected by the feature values of the data instance. For example, different BMI (body mass index) values may quantitatively impact the correlation between adverse effects Hypophonia and Tinnitus. In this work, we propose two models named the Pair-Dependency Multi-Label Bayesian Classifier (PDMMLBC) and the Complete-Dependency Multi-Label Bayesian Classifier (CDMLBC) to accomplish Task 1 and Task 2, respectively. In the PDMMLBC model, the dependency is considered between any two labels, which is similar as the idea proposed by Ghamrawi and McCallum [13]. In the CDMLBC model, the dependency between more than two labels is considered. Sections 4.1 and 4.2 will describe these two approaches in detail.

Firstly, we make the assumption that all features of an instance are independent, which is a basic assumption for Naive Bayesian classification.

4.1. Pair-Dependency Multi-Label Bayesian Classifier (PDMMLBC)

One simple solution to Task 1 is to check each element in label power set \( C^* \), then choose the label set with the highest probability. However, this method is prohibitively costly in terms of computational overhead as the size of the label set increases. Instead of finding the exact solution, we propose the PDMMLBC model and employ a greedy search algorithm to compute the approximate solution. Mathematically, we can describe the iterative process as

\[
C^+ \leftarrow C^+ \cup \left\{ c \mid \arg \max_{c' \in C^+} \sum_{i=1}^{n} P(c | l_i, W) \right\}
\]  

(6)

where \( C^+ \) is the working set of predicted labels for testing instance \( W \) at each iteration. \( C^+ \) is initialized by

\[
C^+ = \arg \max_{c \in C} P(c | W)
\]  

(7)

Using Bayes’ Rule, we can do the following transformation on \( P(c | W) \):

\[
P(c | W) = \frac{P(c, W)}{P(W)} = \frac{P(c) \cdot P(W | c)}{P(W)}
\]

\[
= \frac{P(c) \cdot \prod_{i=1}^{n} P(w_i | c)}{P(W)} \cdot \prod_{i=1}^{n} P(w_i | c) / P(c)
\]

\[
= \frac{P(c) \cdot \prod_{i=1}^{n} \sum_{y \in C} P(w_i | c) \cdot P(y | c, W)}{P(W)}
\]

\[
= \frac{P(c) \cdot \prod_{i=1}^{n} \sum_{y \in C} P(w_i | c) \cdot P(y | c, W)}{P(W)}
\]

\[
= \frac{P(c) \cdot \prod_{i=1}^{n} \sum_{y \in C} P(w_i | c) \cdot P(y | c, W)}{P(W)}
\]

(8)

To simplify computation, we take logarithms on both sides of Eq. (8), giving us

\[
\log(P(c | W)) = \log(P(c)) + \sum_{i=1}^{n} \log \left( \sum_{y \in C} P(w_i | c) \cdot P(y | c, c) \right) / \log(P(W))
\]

(9)

According to Eq. (9), the first label for instance \( W \) can be determined easily. After acquiring the first label, the other labels will be chosen in an iterative process according to Eq. (6).

In order to get a high quality set of labels, the following constraint is added to Eq. (6):

\[
\sum_{i=1}^{n} x(c, l_i, W) \geq \pi * |C^+|
\]  

(10)

where \( |C^+| \) represents the size of working set \( C^+ \) and \( x(\cdot) \) is a binary function of the instance \( W \), predicted label \( l \) and the new label \( c \) under consideration; \( x(\cdot) \) is defined as follows:

\[
x(c, l_i, W) = \begin{cases} 
1 & \text{if } P(c | l_i, W) > P(P(c) | l_i, W) \\
0 & \text{otherwise}
\end{cases}
\]

(11)

In Eq. (11), \( \pi \) denotes the event that the label \( c \) does not belong to the instance \( W \). Basically, Eqs. (10) and (11) define the logic in recruiting new labels. That is, a new label \( c \) can only be selected if at least \( \pi \) percentage of the already chosen labels agree to accept \( x(c, l_i, W) = 1 \) the new label.

Using similar derivation steps as in Eqs. (8)–(9), we can do the following transformation on \( P(c | l_i, W) \):

\[
P(c | l_i, W) = \frac{P(c, l_i, W)}{P(l_i, W)} = \frac{P(c, l_i, W) \cdot P(w_i | c, l_i)}{P(l_i, W)}
\]

\[
= \frac{P(c, l_i, W) \cdot \prod_{i=1}^{n} P(w_i | c, l_i)}{P(l_i, W)}
\]

(12)
We can take logarithms on both sides of Eq. (12) to get
\[
\log(P(c|l,W)) = \log(P(l)) + \log(P(c|l)) + \sum_{i=1}^{n} \log(P(w_{i}|c,l)) - \log(P(l,W))
\] (13)

Given both \(l\) and \(W\), \(\log(P(l))\) and \(\log(P(l,W))\) are constants in Eq. (13), so we can ignore them in the practical computation.

In the following, we will show that the inequality statement in Eq. (14) is true.

\[
\log(P(c|l)) + \sum_{i=1}^{n} \log(P(w_{i}|c,l)) > \log(1 - P(c|l)) + \sum_{i=1}^{n} \log \left( \frac{P(w_{i}|l) - P(w_{i}|c,l) \# P(c|l)}{1 - P(c|l)} \right)
\] (14)

**Proof.** We first take logarithms on both sides of Eq. (10) then we have

\[
\log(P(c|l,W)) > \log(P(c|l))
\]

By utilizing Eqs. (13), (15) can be transformed into

\[
\begin{align*}
\log(P(l)) + & \sum_{i=1}^{n} \log(P(w_{i}|c,l)) - \log(P(l,W)) \\
> & \log(P(l)) + \log(P(c|l)) \\
+ & \sum_{i=1}^{n} \log(P(w_{i}|l)) - \log(P(l,W)) \\
\downarrow \\
\log(P(c|l)) + & \sum_{i=1}^{n} \log(P(w_{i}|c,l)) \\
> & \log(P(c|l)) = \sum_{i=1}^{n} \log(P(w_{i}|c,l)) \\
\end{align*}
\]

Since the following conditions in Eq. (17) stand:

\[
P(c|l) + P(c|l) = 1 \\
P(w_{i}|c,l) \# P(c|l) + P(w_{i}|l) = P(w_{i}|l) \\
P(c,l) + P(\tau,l) = P(l)
\]

we get

\[
\begin{align*}
\log(P(c|l)) + & \sum_{i=1}^{n} \log(P(w_{i}|c,l)) \\
= & \log(1 - P(c|l)) + \sum_{i=1}^{n} \log \left( \frac{P(w_{i}|l) - P(w_{i}|c,l) \# P(c|l)}{P(c,l)} \right) \\
= & \log(1 - P(c|l)) + \sum_{i=1}^{n} \log \left( \frac{P(w_{i}|l) \# P(c|l)}{P(l) - P(c|l)} \right) \\
= & \log(1 - P(c|l)) + \sum_{i=1}^{n} \log \left( \frac{P(w_{i}|l) - P(w_{i}|c,l) \# P(c|l)}{P(l) - P(c|l)} \right) \\
= & \log(1 - P(c|l)) + \sum_{i=1}^{n} \log \left( \frac{P(w_{i}|l) - P(w_{i}|c,l) \# P(c|l)}{1 - P(c|l)} \right)
\end{align*}
\] (18)

Combine Eqs. (16) and (18), we can prove that the inequality in Eq. (14) holds. □

Using the derivation results from Eqs. (10) to (14), the PDMMLC model can be implemented with Algorithm 1. The main procedure of the PDMMLC model consists of two steps. In the first step, we find the most probable label for the testing instance, which is given by Line 1 in Algorithm 1. The second step is to search for other related labels, which is formulated in Line 3. The most significant benefit of PDMMLC is that it needs merely user-defined parameters, which makes it very easy to deploy in practical data mining tasks.

**Algorithm 1.** Pair-Dependency Multi-Label Bayesian Classifier (PDMMLC).

1. \(C^{+} \leftarrow \{c | \arg \max_{c \in C} \log(P(c)) + \sum_{i=1}^{n} \log(\sum_{y \in \mathcal{Y}} P(w_{i}|c,y) \# P(y|c))\}\)
2. repeat
3. \(c = \arg \max_{c \in C} \left( \sum_{i=1}^{n} \log \left( \frac{P(c|l) - P(c)}{1 - P(c|l)} \right) \right) + \sum_{i=1}^{n} \log \left( \frac{P(w_{i}|c,l) \# P(c|l)}{P(w_{i}|l) - P(w_{i}|c,l) \# P(c|l)} \right) \)
4. \(C^{+} \leftarrow C^{+} + \{c\}\)
5. until \(|c| = 0\)

4.2. Complete-Dependency Multi-Label Bayesian Classifier (CDMMLC)

As shown in the previous section, PDMMLC utilizes the dependency between any two labels in multi-label classification. The CDMMLC model, on the other hand, will further investigate the impact of the dependency among more \(\geq 3\) labels. Of the tasks defined in Section 3, the CDMMLC model will help resolve Task 2 directly. Using similar derivation steps to those described in Section 4.1, we can transform \(P(C^{+}|W)\) into

\[
\log(P(C^{+}|W)) = \log(P(C^{+})) + \sum_{i=1}^{n} \log \left( \sum_{y \in \mathcal{Y}} (P(s_{i}|C^{+},y) \# P(y|C^{+})) \right)
\]

(19)

For a specific data instance, \(\log(P(W))\) is a constant. Combining Eqs. (5) and (19), we can refine the objective function of Task 2 as

\[
\arg \max_{c \in C} \sum_{i=1}^{n} \log \left( \sum_{y \in \mathcal{Y}} (P(w_{i}|C^{+},y) \# P(y|C^{+})) \right)
\]

(20)

To get an optimal solution of Eq. (20), CDMMLC will check \(\binom{n}{k}\) sets of different multi-labels. The computation complexity depends on the specific application.

4.3. Analysis

In this section, we will further analyze the properties of the proposed models and compare them with existing approaches. Since we extend the classical Naive Bayesian classifier in our approach, the proposed models can handle both single-label classification and multi-label classification problems.

**Theorem 1.** If \(k = 1\), CDMMLC is a Naive Bayes classifier.

**Proof.** If \(k = 1\), then the objective function of CDMMLC is

\[
\arg \max_{c \in C} P(c|W)
\]
With Bayes’ Rule and the assumption that all features are independent, we get

\[
P(c|W) = \frac{P(c,W)}{P(W)} = \frac{P(c) \cdot P(W|c)}{P(W)} = \frac{P(c) \cdot \prod_{i=1}^{n} P(W_i|c)}{\prod_{i=1}^{n} P(W_i)}
\]

Obviously, this function is the formulation of the Naive Bayes classifier. So when \(k=1\), CDMLBC boils down to a Naive Bayes classifier.

**Theorem 1** shows that the Naive Bayes classifier is a special case of the CDMLBC model \((k=1)\). In other words, when considering the multi-label classification problem, the worst performance of the proposed CDMLBC model will be equal to that of the traditional Naive Bayes approach.

For the PDMLBC model, the first step degenerates into a Naive Bayes problem. Due to the low performance of the Naive Bayes classifier in single-label classification, the performance of the PDMLBC model will also be affected. Fortunately, the two steps in PDMLBC are separated, so we can employ other, better single-label classifiers to replace the Naive Bayes classifier in the first step, and other related labels can still be chosen through the iteration process according to Eq. (6).

The CDMLBC model considers a set of labels as a whole, so we have the following theorem.

**Theorem 2.** If labels \(c_i\) and \(c_j\) \((c_i \neq c_j, c_i,c_j \in C)\) do not appear together in any training data, then \(c_i,c_j\) cannot be predicted jointly by CDMLBC for any testing instance.

**Proof.** As labels \(c_i\) and \(c_j\) do not appear together in any training sample,

\[
P(C^+ = 0)
\]

where \(c_i,c_j \in C^+\).

Consider the objective function of CDMLBC,

\[
P(C^+ | W) = \frac{P(C^+)}{P(W)} \prod_{i=1}^{n} \left( \sum_{y \in C^+} P(W_i | C^+, y) \cdot P(y | C^+) \right) = 0
\]

Since \(P(C^+ | W) = 0\), then \(C^+\) cannot be predicted. Consequently, \(c_i,c_j\) cannot appear together in a multi-label set predicted by CDMLBC.

Though this property limits the variety of the multi-labels predicted by CDMLBC, it guarantees that all results are not arbitrarily generated. Furthermore, compared to the Label Power-set (LP) method, the CDMLBC model is more flexible. Let us use an example to illustrate this idea.

As shown in Table 1, there are three sample data instances and each of them has a multi-label with three labels. The candidate multi-label sets of LP and CDMLBC are shown in Table 2, each of the LP’s candidate multi-labels must appear in the training data set exactly. On the other hand, the CDMLBC model only requires the candidate multi-label to appear together either as a whole or as a subset of one of the multi-label sets.

**Table 1**

<table>
<thead>
<tr>
<th>Example data instance</th>
<th>Label set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{c_1,c_2,c_3}</td>
</tr>
<tr>
<td>2</td>
<td>{c_1,c_2,c_3}</td>
</tr>
<tr>
<td>3</td>
<td>{c_1,c_3,c_4}</td>
</tr>
</tbody>
</table>

**Table 2**

<table>
<thead>
<tr>
<th>Label powerset</th>
<th>CDMLBC model</th>
</tr>
</thead>
<tbody>
<tr>
<td>{c_1, }</td>
<td>{c_1, }</td>
</tr>
<tr>
<td>{c_1, }</td>
<td>{c_1, }</td>
</tr>
<tr>
<td>{c_1, }</td>
<td>{c_1, }</td>
</tr>
<tr>
<td>{c_1, }</td>
<td>{c_1, }</td>
</tr>
<tr>
<td>{c_1, }</td>
<td>{c_1, }</td>
</tr>
</tbody>
</table>

**Table 3**

<table>
<thead>
<tr>
<th>Classification model</th>
<th>Label dependency</th>
<th>Label limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Label-feature</td>
<td>NO</td>
</tr>
<tr>
<td>BR</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ML-KNN</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>PDMLBC</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CDMLBC</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

As we know, in designing multi-label classifiers, the most important thing is to make full use of the correlation/dependency among labels. In single-label classification, labels are exclusive, and the goal is to learn a model that relates features to labels. In multi-label classification, however, more than one label can coexist in one instance. This coexistence is also related to some specific features. In conclusion, Table 3 summarizes the different aspects that existing models take into account when dealing with multi-label classification problems.

In Table 3, **Label dependency** indicates whether an algorithm in the row considers the dependency among labels. The difference between the columns **Label-label** and **Label-feature** is whether the feature is taken into account while considering the dependency among labels. The column **Label limitation** indicates the degree of consistency between the predicted label set and the existing label set in the training data set. **NO** means there is no requirement; **SOME** means there is some kind of requirement, the details of which are dependent on the specific algorithms; and **RESTRICTED** means the predicted multi-label must appear in the training multi-labels.

In the binary relevance (BR) method, a multi-label classification is transformed into multiple, independent single-label classifications. With this strategy, there is an individual binary classifier for each label during the training phase. During testing, each binary classifier gives its own confidence value on the prediction, and the final decision of the labels depends on these individual confidence values. The advantage of this method is its simplicity, and state-of-the-art single-label classifiers can be used directly. However, this method ignores the essential property of multi-label classification that labels are correlated with each other.

In ML-KNN, the authors focus on the dependency among labels without considering the impact from features. Compared with the classical KNN method, the ML-KNN approach uses Bayes Rule on label frequency to make the judgment on label assignment rather than counting the number of neighbor labels as in the KNN method. In ML-KNN, if labels \(c_i\) and \(c_j\) are predicted together (they may not have to appear together in the training data set), label \(c_i\) must appear in \(c_j\)’s neighbor labels, and vice versa. We need also to point out the instability in dependency among labels given by ML-KNN since a difference in neighbor relationship affiliation will change the dependency. Let us use an example to illustrate this. As shown in Fig. 1, there are three points \(A(1,2), B(2,2)\) and \(C(5,5), \ldots\)
Suppose we are looking for the neighbor of $B$ in terms of closest distance. If the distance is defined as Euclidean distance, then we have $d_{eu}(A,B)=1$ and $d_{eu}(C,B)=4.2426$. Therefore, $B$'s closest neighbor is $A$. On the other hand, if the distance is calculated by Cosine similarity, then the distances change to $d_{cos}(A,B)=0.0513$ and $d_{cos}(C,B)=0$. In this case, $B$'s closest neighbor is $C$.

5. Experiments and conclusions

In this section, we will first describe the metrics used in evaluating the newly proposed PDMLBC and CDMLBC models in multi-label classification, followed by the procedure used to evaluate these models.

5.1. Evaluation metrics

For evaluating the performance of multi-label classifiers, many criteria have been proposed [26,14]. As there is no ranking score of each label in multi-labels predicted by CDMLBC, then we focus on example-based metrics [30], including Hamming loss, Precision, Recall, F1 and Accuracy. To describe these metrics clearly, we will first establish some necessary notation. Let $T=(x_1,Y_1), (x_2,Y_2), \ldots, (x_n,Y_n)$ ($q \in R$) be a test data set, where $x_i$ ($1 \leq i \leq q$) is a test instance and $Y_i \in \mathcal{C}$ is $x_i$’s multi-label. In addition, let $h(x_i)$ be the predicted multi-label for $x_i$ by one multi-label classifier.

- **Hamming loss** calculates the percentage of mis-predicted labels:
  \[
  HL(T) = \frac{1}{q} \sum_{i=1}^{q} \frac{|h(x_i) \Delta Y_i|}{|Y_i|} 
  \]  
  (21)
  where $\Delta$ is the notation for differentiating two sets. As shown in Eq. (21), the mis-predicted labels for $x_i$ include the labels which appear in $Y_i$ but not in $h(x_i)$ and the labels which are in $h(x_i)$ but not in $Y_i$.

- **Precision** comes from the metrics for single-label classifiers in Information Retrieval (IR):
  \[
  P(T) = \frac{1}{q} \sum_{i=1}^{q} \frac{|Y_i \cap h(x_i)|}{|h(x_i)|} 
  \]  
  (22)

- **Recall** corresponds to recall in single-label metrics:
  \[
  R(T) = \frac{1}{q} \sum_{i=1}^{q} \frac{|Y_i \cap h(x_i)|}{|Y_i|} 
  \]  
  (23)

- **F1** combines precision and recall:
  \[
  F1(T) = \frac{\sum_{i=1}^{q} 2np(x_i) \cup h(x_i)}{\sum_{i=1}^{q} p(x_i) + \sum_{i=1}^{q} h(x_i)} 
  \]  
  (24)

- **Accuracy** is similar to accuracy in single-label metrics:
  \[
  A(T) = \frac{1}{q} \sum_{i=1}^{q} \frac{|Y_i \cap h(x_i)|}{|Y_i \cup h(x_i)|} 
  \]  
  (25)

Among these five metrics, only the Hamming loss score is inversely proportional to the classifier’s performance. The other four metrics return increasingly large scores as classifier performance increases.

5.2. Evaluation procedure

Subjects aged 18–60 years undergoing arthroscopic shoulder surgery will be randomized into three groups. Subjects and observers will be blinded as to group assignment. Unblinded investigators will administer the injection and establish the infusion in the preoperative holding area. In the operating room the quality of the block will be assessed by surgical manipulation. Sedation with intravenous propofol will be established. General anesthesia will be established if needed (based upon the response to manipulation and the extent of hypesthesia over the C5 and C6 vertebrae).
Fig. 2. Experimental results. (a) Hamming loss. (b) Precision. (c) Recall. (d) F1. (e) Accuracy.
dermatomes) according to standard practice with propofol for induction, a laryngeal mask airway (LMA) and sevoflurane for anesthetic maintenance. Supplemental postoperative analgesia will be provided as needed using intravenous fentanyl, intravenous ketorolac (30 mg, one time dose), and oral hydrocodone/acetaminophen as per the WBH acute pain algorithm (WBH-RO Form 6459). The Stryker Pain Pump II will be used for continuous infusion per WBH policy (WBH-RO: Anesthetic Infusion Device: Pain Pump; Policy 480, part II: pp. 2–7) with the following settings: Ropivacaine 0.2%; bolus: 3 ml; continuous infusion: 4 ml/h; lock-out: 20 min. A single dose of dexamethasone will be allowed for postoperative nausea and vomiting (PONV) or PONV prophylaxis.

5.2.2. Primary outcome measures

- **Pain measurements:** Categorical and Numeric Pain Rating Scales (NRS) will be used at baseline (pre-operatively), at discharge from PACU, and at 24 and 48 h. A change in NRS of 1.8 will be considered clinically significant.

- **Diaphragmatic excursion:** Diaphragmatic excursion at maximal effort for inspiration and exhalation will be assessed by a blinded ultrasonographer bilaterally, both preoperatively and prior to discharge from PACU, using a low frequency probe (4 Hz) posterolaterally at the midclavicular line. Because diaphragmatic excursion may decrease as a result of surgery, anesthesia, and opioid pain medications, the change in ipsilateral measurement will be normalized using a ratio to the contralateral excursion changes.

5.2.3. Secondary outcome measures

- **Dysphonia:** At discharge from PACU, 24 and 48 h the patient will be asked “Is your voice hoarse?” (yes/no).

- **Horner’s Syndrome:** At discharge from PACU, 24 and 48 h the patient will be asked “Do you have blurred vision or a droopy eyelid?” If either of these effects are noted the patient will be considered to have symptomatic Horner’s Syndrome and a positive finding (yes/no) will be recorded.

- **Hand weakness:** At discharge from PACU, 24 and 48 h grip strength will be assessed by asking the question, “Does your grip feel weak?” (yes/no).

- **Patient satisfaction:** A 24 and 48 h 5-point Likert categorical Helpfulness scale will be administered (Is your interscalene infusin: extremely harmful; harmful; neutral: not harmful, but not helpful; helpful; extremely helpful?).

- **Functional outcome:** At baseline and again at 12 weeks subjects will complete the Simple Shoulder Test (Table 4). This test is a series of 12 (yes/no) questions. This has been shown to be a valid, reliable and consistent for subjects up to and including 60 years of age when similar injuries (rotator cuff dysfunction) are assessed. The use of this assessment collected by telephone interview has been validated and is comparable to more complicated scales when converted to a 100-point scale according to the process described by Romeo et al. [23].

- **Local anesthetic toxicity:** At discharge from PACU, 24 and 48 h the presence of any one of the following will be considered a positive score: tinnitus, perioral numbness, feeling jittery (yes/no).

5.2.4. Additional secondary outcomes

Additional secondary outcomes including unscheduled admission, total amount of local anesthetic used (read from pump), type and amount of supplemental analgesics, nausea/vomiting, requirement for general anesthesia (defined as the use of an LMA and/or inhalational anesthesia, i.e., sevoflurane), early termination (catheter malfunction/dislodgement prior to 48 h) will be recorded.

5.3. Experimental results

In evaluating our proposed models (i.e., PDMLBC and CDMLBC), analysis is performed based on the data gathered from the 36 patients randomized into these three bolus groups. The features we investigate included the patient’s demographics information (sex, nationality, BMI), patient medical history (ASA status, Categorical Scale, and NRS), pre-op functional outcome—Simple Shoulder Test (SST) (Q₁–Q₃ in Table 4), Dominant hand, Shoulder Evaluated, Anesthesia Type used, Categorical Scale and NRS right prior to Discharge from PACU, and Initial Bolus Amount. The adverse effects are the labels under study, which are Dysphonia(1), Horner’s Syndrome(2), Dyspnea(3), Tinnitus(4), Perioral numbness(5), Feeling jittery(6), PONV(7). For ease of explanation, numbers were assigned to each adverse effect (indicated in the parentheses).

We tested 9-fold testing with both PDMLBC and CDMLBC models and compared the results to the BR+NB method and the ML-KNN method. Fig. 2 shows the comparative results with the metrics Hamming loss, Precision, Recall, F1, and Accuracy. Overall, the CDMLBC model won in all five aspects. For labels 1 (Dysphonia), 3 (Dyspnea), and 5 (Perioral numbness), ML-KNN has

<table>
<thead>
<tr>
<th>Metric/method</th>
<th>PDMRCB</th>
<th>CDMLBC</th>
<th>NB+BR</th>
<th>ML-KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamming loss</td>
<td>0.0121 ± 0.00005</td>
<td>0.0112 ± 0.00003</td>
<td>0.0125 ± 0.00006</td>
<td>0.0125 ± 0.00000</td>
</tr>
<tr>
<td>Precision</td>
<td>0.9180 ± 0.0028</td>
<td>0.9246 ± 0.0020</td>
<td>0.9153 ± 0.0031</td>
<td>0.9206 ± 0.0017</td>
</tr>
<tr>
<td>Recall</td>
<td>0.9180 ± 0.0028</td>
<td>0.9246 ± 0.0020</td>
<td>0.9193 ± 0.0028</td>
<td>0.9206 ± 0.0017</td>
</tr>
<tr>
<td>F1</td>
<td>0.9180 ± 0.0028</td>
<td>0.9246 ± 0.0020</td>
<td>0.9167 ± 0.0029</td>
<td>0.9206 ± 0.0017</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.9180 ± 0.0028</td>
<td>0.9246 ± 0.0020</td>
<td>0.9153 ± 0.0031</td>
<td>0.9206 ± 0.0017</td>
</tr>
</tbody>
</table>
better performance in terms of Hamming loss. For labels 3 (Dyspnea) and 5 (Perioral numbness), ML-KNN has the best performance in Precision, Recall, F1, and Accuracy. However, it loses in the labels 2 (Horner’s Syndrome), 4 (Tinnitus), 6 (Feeling jittery), and 7 (PONV) in all aspects.

On the other hand, CDMMLBC also works better than ML-KNN according to Table 5. For Hamming loss, CDMMLBC can achieve a score of 0.0112, while the ML-KNN has a Hamming loss of 0.0125. For other metrics (Precision, Recall, F1, Accuracy), the CDMMLBC model has a uniform increase of about 0.4% from 0.9206 to 0.9246. It should be noticed that ML-KNN does have smaller variances. This fact reveals the disadvantage of probability-based models: these models closely rely on the scale of training samples.

References


Guangzhi Qu received his PhD in Computer Engineering from The University of Arizona in 2005. He is an assistant professor in Computer Science and Engineering Department at the Oakland University. His research interests include data mining, machine learning, complex networks, medical informatics, health care computing and discrete event simulation.

Hui Wu received his MS in Computer Science from Beijing University of Aeronautics and Astronautics. He is now a PhD student in CSE Department at Oakland University. His research interests include information retrieval, data mining and ontology learning.
Craig T. Hartrick MD, DABPM, FIPP, is a Professor of Biomedical Sciences and Anesthesiology at the Oakland University, William Beaumont School of Medicine (OUWBSM), and a Clinical Professor of Health Sciences at the Oakland University. Currently he serves as Editor-in-Chief of the Journal Pain Practice, Discipline Director for Pharmacology for OUWBSM, and Director of Anesthesiology Research at Beaumont Health System-Troy.

JianWei Niu received his Ph.D. degree in Computer Science from the Beijing University of Aeronautics and Astronautics (BUAA, now renamed as Beihang University) in 2002. He is a professor in School of Computer Science and Engineering, BUAA. He is an ACM/IEEE member and has published more than 80 referred papers and filed more than 20 patents in mobile and pervasive computing. He has been on various chairs and TPC members for many international conferences. He served as the Program Chair of IEEE SEC 2008. His current research interests include mobile and pervasive computing.