Zhang *et al. Health Information Science and Systems* 2013, **1**:15
http://www.hissjournal.com/content/1/1/15

HEALTH INFORMATION SCIENCE AND SYSTEMS

RESEARCH Open Access

On the application of multi-class classification in physical therapy recommendation

Jing Zhang[1], Peng Cao[1], Douglas P Gross[2] and Osmar R Zaiane[1*]

Abstract

Recommending optimal rehabilitation intervention for injured workers that would lead to successful return-to-work (RTW) is a challenge for clinicians. Currently, the clinicians are unable to identify with complete confidence which intervention is best for a patient and the referral is often made in trial and error fashion. Only 58% recommendations are successful in our dataset. We aim to develop an interpretable decision support system using machine learning to assist the clinicians. We proposed an alternate ripper (ARIPPER) combined with a hybrid re-sampling technique, and a balanced weighted random forests (BWRF) ensemble method respectively, in order to tackle the multi-class imbalance, class overlap and noise problem in real world application data. The final models have shown promising potential in classification compared to human baseline and has been integrated into a web-based decision-support tool that requires additional validation in a clinical sample.

Introduction

Decision support systems (DSS) in clinical prognosis have received increased attention from researchers. In this paper, we develop a system to help clinicians categorize injured workers and recommend appropriate rehabilitation programs based on the unique characteristics of individual worker.

Our system is a web application consisting of a user interface and a knowledge base. Unlike many DSS using knowledge bases developed manually by domain experts, we use rule-based machine learning algorithms to learn a set of rules from data. The rules can be further modified and tuned by the experts. By doing so, the experts can inject their own knowledge into the discovered rule set. The major challenge of generating the knowledge base is the presence of imbalance class distribution and multi-class classification in our real clinical dataset. Recently, the class imbalance learning has been recognized as a crucial problem in machine learning and data mining [1-3]. The issue occurs when the training data is not evenly distributed among classes. Imbalanced data learning is growing in importance and has been identified as one of the 10 main challenges of Data Mining [4]. This problem is also especially critical in many real applications, such as fraud detection and medical diagnoses. In these cases, standard classifiers generally perform poorly, rule-based classifiers are particularly sensitive to class imbalance. Classifiers usually tend to be overwhelmed by the majority class and ignore the minority class examples. Most classifiers assume an even distribution of examples among classes, and are designed to maximize accuracy, which is not a good metric to evaluate effectiveness in the case of imbalanced training data. Therefore, we need to improve traditional algorithms so as to handle imbalanced data.

Most existing imbalance data learning techniques so far are still limited to the binary class imbalance problem. There are fewer solutions for multi-class imbalance problems, which exist in real-world applications. They have been shown to be less effective or even to cause a negative effect in dealing with multi-class tasks [5]. The experiments in [6] imply that the performance decreases as the number of imbalanced classes increases.

Moreover, many studies have shown that for certain imbalanced data sets, the degree of imbalance between class prior distribution is not the only factor influencing performance of classification of imbalanced data. Data set complexity including overlapping, lack of representative data as well as presence of noisy instances [2]. The issues

*Correspondence: zaiane@cs.ualberta.ca
[1]Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada
Full list of author information is available at the end of the article

of overlapping and noisy exists in our datasets, therefore it is not realistic to simply balancing the dataset with complex structure of data.

In this paper we consider a series of rule-based classifiers combined with sampling techniques from imbalanced data. We proposed two different methods, alternate ripper combined with hybrid re-sampling techniques and balanced weighted random forest (BWRF) in the context of learning from imbalanced, overlapping as well as noisy multi-classes data, and empirically investigate and compare various common classifiers and the state of the art methods for imbalanced data learning. We find that both of them can achieve a better result and ARIPPER can produce meaningful recommendation rules as evaluated by our domain expert. Moreover, the combination of class decomposition and data processing method can help the classification on the minority class examples.

Background

Work-related musculoskeletal conditions are some of the most burdensome health conditions in terms of personal, societal and economic costs [7-9]. Low back pain is a leading cause of work disability and was recently identified as the sixth most disabling health condition worldwide in terms of overall disease burden [10].

In general, each injured worker receives a return-to-work (RTW) assessment and a following rehabilitation treatment. This is a classification process which involves assigning patients to appropriate rehabilitation programs that lead to successful return-to-work (RTW) based on their clinical and work-related characteristics (obtained from the assessment). There are five types of rehabilitation programs in total labeled as prog0, prog3, prog4, prog5 and prog6. The feature of each program is listed as follows:

- "Other" intervention (prog0): No rehabilitation or a single service provider. Basically it means a patient's health condition does not require any treatment or no treatments could help the patient return to work.
- Complex Service (prog3): Comprehensive pain management program for patients with chronic pain and multiple complex barriers to RTW.
- Provider Site Based Service (prog4): Interdisciplinary rehabilitation at a designated rehabilitation facility
- Work Site Based Service (prog5): The intervention takes place at a worksite instead of at a rehabilitation facility.
- Hybrid (prog6): A hybrid program of prog4 and prog5.

Each rehabilitation program has two possible outcomes:

- The program leads to successful return-to-work at a pre-determined time.
- An unsuccessful result at that pre-determined time followed by subsequent rehabilitation programs.

Although it is possible that multiple rehabilitation programs can lead to return-to-work for a patient, we cannot determine them since we cannot possibly let a patient go through multiple programs at once to observe the outcomes.

Therefore, an important assumption is that for each patient there exists only one appropriate program. If patients are correctly categorized into the true appropriate program, they return to work. Otherwise, there will be no successful RTW. Under the assumption above, we could determine a patient's return-to-work status in advance based on the classification result. The main idea here is to build a classification model that categorizes an injured worker into the appropriate rehabilitation program leading to RTW.

Preliminaries

Multi-class classification

Multi-class pattern recognition is a problem of building a system that accurately maps an input feature space to an output space of more than two pattern classes. Multi-class pattern recognition has a wide range of applications including handwritten digit recognition, categorization, bioinformatics and speech tagging and recognition. The task of multi-class classification is to assign a class k from a set of possible classes K to vectors (data samples) $x = (a_1, a_2, ..., a_m)$ consisting of n attribute values $a_i \in A_i$. The data mining algorithm is supposed to learn a relation (called classifier) $C : A_1 \times A_2 \times ... \times A_m \rightarrow K$ which can map an unseen instances to a corresponding class $k \in K$.

Imbalanced data learning

A common problem faced in data mining is dealing with class imbalance. A dataset is said to be imbalanced if one class (called the majority, or negative class) vastly outnumbers the other (called the minority, or positive class). The class imbalance problem is only said to exist when the positive class is the class of interest. This is due to the fact that if the positive, minority, class is not of interest, then it can be safely ignored. In most practical applications, such as medical diagnosis, spam filter, intrusion detection, etc. The minority class is the class of interest, and therefore the class imbalance problem must be addressed.

These imbalanced data learning methods can be grouped into two categories: the data perspective [11-15] and the algorithm perspective [5,6,16-18]. The methods with the data perspective try to balance out the class distribution by re-sampling the data space, either over-sampling instances of the minority class or under-sampling

instances of the majority class. The most common method with the algorithm perspective is cost-sensitive learning, which tries to learn more characteristics of samples with the minority class by setting a high cost to the misclassification of a minority class sample. Many empirical results have shown that the exact theoretical connection between re-sampling methods and cost-sensitive learning [18], and re-sampling methods can be competitive to cost-sensitive learning [19]. In addition, the re-sampling is not dependent on the classifier and simple to implement. Hence, the re-sampling is most straightforward and effective method for dealing with imbalance. In this paper, we focus our attention on the problem of re-sampling technique for imbalanced data classification.

Re-sampling methods only manipulate the original training datasets; therefore it provides a convenient and effective way to deal with imbalanced learning problems using standard classifiers by balancing the instances of the classes. The purpose of the re-sampling methods is generated a new and more balanced dataset D^* based on the original dataset D, on which the classifier has a better performance. We aim to find the optimal re-sampling procedure S: $D \rightarrow D^*$. The re-sampling techniques include the under-sampling and over-sampling.

Too small number of examples in the minority class in comparison to the number of examples in the majority classes (expressed by an imbalance ratio) is not the only problem while creating classifiers from imbalanced data [20]. Other data-related factors, which make the learning task even more difficult, include overlapping of the minority and majority classes, the presence of noisy or rare examples.

In many domain, there is often occurs that some samples from different classes have very similar characteristics in the feature space. The problem is recognized as the so-called class overlapping problem.

Data are said to be noisy if they contain erroneous data values. These erroneous values can occur in the independent (attribute noise) or dependent (class noise) variables in a data set [21]. While real-world data often contain both types of noise the latter one is generally more detrimental to classification performance [21,22]. Noisy data can confuse a learning algorithm, blurring the decision boundaries that separate the classes or causing models to overfit to accommodate incorrect data points.

Class imbalance, overlapping and noise are well-established data characteristics encountered in a wide range of data mining and machine learning, and they often occur in some real application at the same time. Therefore, when we design rule-based learning model to deal with imbalanced data, we need to consider these factors which have adverse effects on the recognition of the minority classes.

We present some common re-sampling algorithms:

SMOTE

A popular and effective over-sampling method is the synthetic minority over-sampling technique (SMOTE) [11]. The SMOTE algorithm creates artificial data based on the feature space similarities between existing minority instances. Specifically, for minority class dataset C_{min}, consider the K nearest neighbors for each instance $x_i \in C_{min}$. To create a synthetic sample, randomly select one of the K nearest neighbors, then multiply the corresponding feature vector difference with a random number between [0,1], and finally, add this vector to the x_i:

$$x_{new} = x_i + (x_i^k - x_i) \times \delta \quad (1)$$

where x_i^k is one of the K-nearest neighbors for x_i, and $\delta \in [0, 1]$ is a random number. Therefore, the resulting synthetic instance according to (1) is a point along the line segment joining x_i under consideration and the randomly selected K nearest neighbor x_i^k.

Figure 1 shows an example of the SMOTE procedure.

There exist many methods based on the SMOTE for generating more appropriate instances. For instance, borderline-SMOTE [23] selects minority examples which are considered to be on the border of the minority decision region in the feature-space and only performs the SMOTE technique to oversample those instances, since borderline instances are more easily misclassified than others.

Tomek links

If two data examples from different classes are the 1 nearest neighbors to each other, they form a Tomek Link [14]. Given an instance pair:(x_i, x_j) where $x_i \in S_{min}, x_j \in S_{max}$ and $d(x_i, x_j)$ is the distance between x_i and x_j, then the (x_i, x_j) pair is called a Tomek link if there is no instance x_k,such that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$.

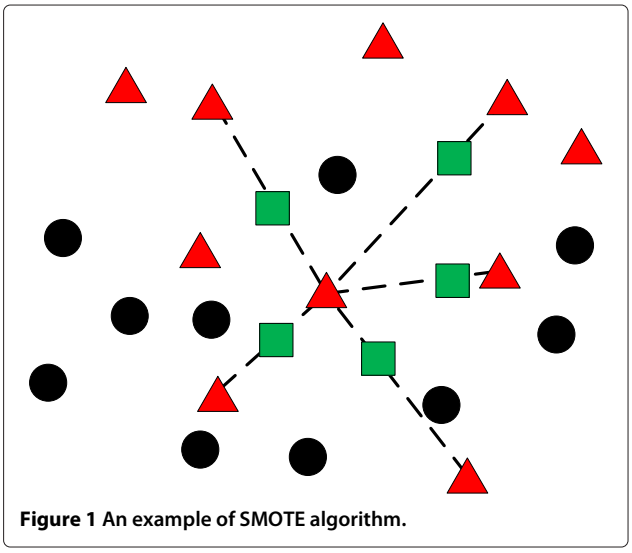

Figure 1 An example of SMOTE algorithm.

Figure 2 shows a typical procedure of using Tomek links to clean the overlapping data points.

This data cleaning technique has been effectively applied to remove the overlapping that is introduced from sampling methods. Either both of them are borderline points, or one of them is noise invading the data space of the other class.

Edited Nearest Neighbor, ENN

Edited Nearest Neighbor (ENN) [12] Rule is an undersampling method that removes data examples whose class label differs from that of at least two of its three nearest neighbors.

Neighborhood Cleaning Rule, NCR

Unlike ENN, NCR [15] finds each data example whose class label differs from the class of at least two of its three nearest neighbors. If this example belongs to the majority class, remove it. Otherwise, remove its nearest neighbors which belong to the majority class.

Rule-based classifier

Learning classification rules from examples is one of the most popular tasks in machine learning and data mining. Each classification rule has a conjunction of attribute values as antecedent and a class as consequent. Generally speaking, such rules are represented as symbolic expressions of the following form:

$$r : (\textit{Rule conditions}) \rightarrow \textit{targetclass} \quad (2)$$

where conditions are formed as a conjunction of elementary tests on values of attributes describing learning examples, and the rule consequence indicates the assignment of an example satisfying the condition part of the rule to a given class. A rule r covers an instance x if $x's$ attribute values satisfy the rule antecedent.

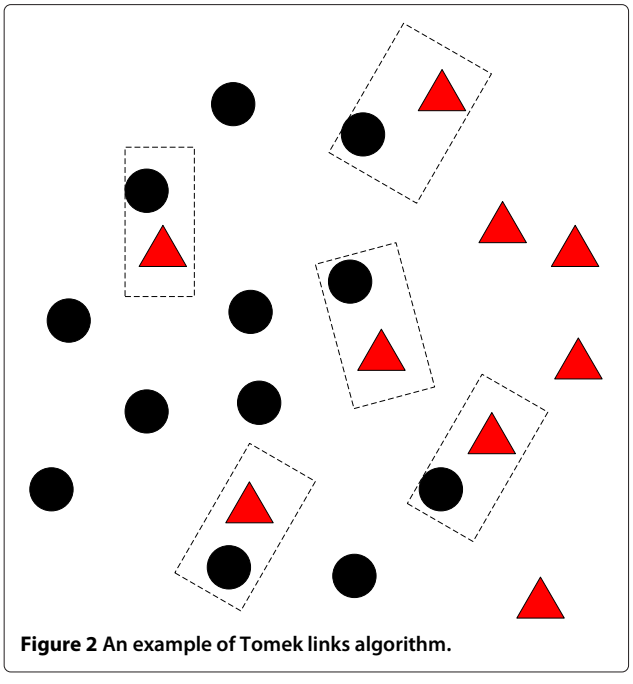

Figure 2 An example of Tomek links algorithm.

Rules are one of the most popular symbolic representations of knowledge discovered from data. They are more comprehensible and can be easily analysed by human experts, in particular "black boxes" models like neural networks or SVM. However, most rule-based classifiers are biased towards the majority classes and they have difficulties with correct recognition of the minority class. Such comprehensibility and explicability of the rule representation is highly appreciated when constructing intelligent systems, where these features often result in increased willingness of decision makers to accept provided suggestions and solutions.

System design and implementation

System requirements

The system we are developing has the following requirements:

- The classification model should be interpretable. The users should be able to see the evidence supporting the recommendations made by the system. Rule-based algorithms are more desirable.
- The system should provide multiple predictions with support evidence (e.g., supporting rules or guidelines) so the users can choose the most appropriate one under different considerations.
- The system should include a limited number of variables.

Data analysis

The dataset is from an outcome evaluation database managed by the Workers' Compensation Board Alberta. This includes data on workers in the province of Alberta who filed compensation claims for musculoskeletal injuries and who were referred to rehabilitation facilities for Return-to-Work assessment. WCB-Alberta's administrative database was augmented by clinical data from rehabilitation providers who are contracted to file reports at time of claimants' admission and discharge from rehabilitation programs.

The data in our research are extracted from a Canadian provincial compensation database at the Workers Compensation Board (WCB) Alberta. All samples in our dataset are labeled by the assessing clinician. The 'label' is the appropriate rehabilitation program for each claimant and the predictive features are the various measures available on each claimant at time of assessment. The dataset of mainly the year of 2010 contains 14484 cases of injured workers, of which 8611 were unique cases and included in further analysis. Not all records have a successful RTW

result. If an intervention was unsuccessful, the claimant would have to go through subsequent interventions. In order to train a classification model that predicts successful interventions, the machine learning algorithms need to learn from only the successful records in the data instead of all of them. In other words, the algorithms should mine the relationship between the patients' characteristics and their rehabilitation program in records with positive outcome. Therefore, we extracted 4876 successful records for the algorithms to learn a positive classification model. The successful outcome is when the injured worker receives no compensation at 30 days after the assessment admission. The new dataset is highly-skewed as shown in Table 1. There are three minority classes (prog3, prog5 and prog6) and two majority classes (prog0 and prog4). Moreover, the class prog3 and prog5 has rare instances, which contain only 84 and 96 samples respectively. Furthermore, the data is obtained from different hospital and clinical, the criterion of evaluation and diagnosis are not the same. The nature of the rehabilitation program is also somewhat responsible for the misclassification between the majority classes. For example, for one patient without severe symptom, one hospital may predict it as prog0, another hospital may regard it as prog4. It results in the overlapping between classes. As we are informed by the experts, prog0 and prog4 are similar to each other. A large portion of people receiving prog4 actually does not need prog4. But in order to make sure that people do return to work, they are assigned with prog4 in the end. Additionally, prog6 is a hybrid program of prog4 and prog5, which makes it even more complicated in classification. The data visualization in later section also confirms this issue as we can see the overlap between these classes.

Additionally, it can't avoid the occurrence of noisy data. The rest of the data consisting of unsuccessful cases is used to train a negative model. The dataset includes 200 features. We consulted the experts from the Department of Physical Therapy to check each variable and eliminate those that are absolutely irrelevant from the perspective of clinical practice. 59 features are selected for further investigation.

Methods

RIPPER

RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [24] is an inductive rule-based learner that generates Disjunctive Normal Form (DNF) rules to identify classes while minimising the error defined by the number of misclassified training example by the rule.

Table 1 Class distribution of the final dataset

Class	prog0	prog3	prog4	prog5	prog6
num of records	1828	84	2286	96	582

RIPPER forms rules through a process of repeated growing and pruning, and the representation of the rules generated can be more powerful because it is not constrained by the arborescent structure of the tree. During the growing phase, the rules are made more restrictive in order to fit the training data as closely as possible. During the pruning phase, the rules are made less restrictive in order to avoid overfitting, which can cause poor classification performance.

Concisely, the algorithm proceeds iteratively starting with an empty rule set, and in each iteration, the training data is split into a growing set and a pruning set, then a rule is grown from the growing set and immediately pruned or simplified based on the pruning set. If the error rate of the new rule on the pruning set does not exceed some threshold, the rule is added to the rule set representing the learned model and all examples in the training data covered by this rule are removed before being split again for the next repetition. Otherwise, the iteration is stopped and the rule set is returned. The RIPPER algorithm builds a single rule in the following steps:

1. Split currently uncovered examples into a growing and pruning set.
2. On the growing set, it starts with an empty rule (an rule with no antecedent).
3. Add a new condition into the rule antecedent as long as this addition maximize FOIL's information gain criterion
4. Repeat Step 3 until no negative examples from the growing set are covered by this rule.
5. Prune this immediately on the pruning set.

In a multi-class situation, the rules generated from the RIPPER algorithm are ranked in ascending order based on the number of examples in the class. An unknown instance is tested against the rules in that order. The first rule that covers the test instance "fires" and the testing phase ends. The RIPPER algorithm for multi-class classification is described in the following steps:

1. RIPPER sort the classes in ascending order based on the class size.
2. It chooses the smallest class as the positive class and the rest is considered as the negative class.
3. A rule set for the positive class is learned.
4. Repeat step 2 and 3 for the next smallest class.

Alternate RIPPER, ARIPPER

Our CDSS need to provide multiple recommendations for clinicians to choose, the default RIPPER algorithm makes only one prediction. To make multiple predictions, we make the following modifications and refer to the modified algorithm as ARIPPER (Alternate RIPPER) in the rest of the paper: for each test instance, we gather all the rules

covering it and group the rules together that predict the same program and rank these predictions based on their quality. Such quality can be computed by measuring the quality of the underlying supporting rules. We consider four types of measurements:

- **Highest Average Rule Confidence (HAvgRCF)**: calculate the average rule confidence of all rules supporting each recommendation. The one with the highest average rule confidence is the final prediction.
- **Single Rule with Highest Confidence (SRHCF)**: the rule with the highest confidence makes the final prediction.
- **Highest Average Weighted Chi-Square (HAvgCS)**: HAvgCS is a measurement adopted from CMAR, i.e., Classification based on Multiple Association Rules [25]. It calculates the weighted rule Chi-Square value of all rules supporting each recommendation. The weighted Chi-Square measure is defined in CMAR [25]:

$$weighted\chi^2 = \sum_{k=1}^{n} \frac{\chi^2\chi^2}{max\chi^2} \tag{3}$$

where n is the number of rules in a group. χ^2 stands for the Chi Square value of a single rule. $max\chi^2$ represents the upper bound of χ^2 and is defined in CMAR [25]:

$$max\chi^2 = (minsup(p), sup(c) - \frac{sup(p)sup(c)}{|T|})^2|T|e \tag{4}$$

where $|T|$ stands for the total number of data instance in the training data. For each rule $R : p \rightarrow c$, sup(p) and sup(c) stand for the support of the rule body p and the support of the class label c respectively. Additionally,

$$e = \frac{1}{sup(p)sup(c)} + \frac{1}{sup(p)(|T| - sup(c))} + \frac{1}{(|T| - sup(p)sup(c))} + \frac{1}{(|T| - sup(p))(|T| - sup(c))} \tag{5}$$

- **Single Rule with Highest Weighted Chi-Square (SRHCS)**: SRHCS looks at the Chi-Square of each single rule and uses the one with the highest Chi-Square value as the quality of a recommendation.

Random forests

As far as predictive performance is concerned, in recent years, a number of works have reported that ensembles of base learners exhibit substantial performance improvement over single base learners. Ensemble systems have drawn more and more attention because of their flexible characteristics. Not only multiple classifiers could have better answer than a single one, but also the ensemble framework provides diversity for avoiding the overfitting of some algorithms.

Random forests [26] are an ensemble of tree-type classifiers for classification or regression. It can handle large amount of training data efficiently and that are inherently suited for multi-class problems. They derive their strength from two aspects: using random subsamples of the training data (as in bagging) and randomizing the algorithm for learning base-level classifiers (decision trees). The base-level algorithm randomly selects a subset of the features at each step of tree construction and chooses the best among these.

Random Forests grows a number of such classification trees. Each tree is grown as follows:

1. A tree of maximal depth is grown on a bootstrap sample of size m of the training set. No pruning is performed.
2. A number m which is much smaller than the total number of variables p (typically $m = \sqrt{p}$)) is specified, such that at each node, m variables are sampled at random out of the p. The best split on these variables is used to split the node into two subnodes.

To classify a test instance, the Random Forests classifies the instance by simply combining all results from each of the trees in the forest. The final classification is given by majority voting of the ensemble of trees in the forest. In contrast to bagging, an additional layer of randomness is included in step 2 of the algorithm above. Instead of just constructing trees of different bootstrap samples, step 2 changes the way the individual trees are constructed, namely at each node the splitting variable is not chosen among all variables but the best possible split among a random subset of variables is performed.

Random forest generally exhibits a substantial performance improvement over the single tree classifier such as CART and C4.5. It yields generalization error rate that compares favorably to Adaboost, yet is more robust to noise. In addition, the Random Forests are computationally much less intensive than Adaboost.

Balanced and weighted random forests, BWRF

Ensemble systems is incorporated with re-sampling technique or weighting strategy to acquire better classification performance and generalization capability.

We present a method that uses an ensemble of random forest integrated with a re-balancing technique that

combines both over-sampling and under-sampling: balanced and weighted random forests (BWRF).

For each instance from minority class, the minority class (prog3, prog5 and prog6) is over-sampled with the SMOTE method to smooth the decision boundary. After that, we under-sample the majority class (prog0 and prog4) instances N times to generate N bootstrap samples so that each bootstrap sample has the same or similar size with the over-sampled positive instances. Then, each bootstrap sample (of the majority class) is combined with the over-sampled positive instances to form a training set to train an unprunned tree classifier. Finally, the N tree classifier are combined to make a prediction on a test example by casting a weighted vote from the ensemble of tree classifiers.

We utilized the out-of-bag (OOB) samples in determining different classifier's voting power, and then each base classifier is weighted when combined to create the final decision function. The goal is to assign weights that reflect the relative contribution of each classifier in improving the overall performance of the ensemble. It is known that the use of overall accuracy is not an appropriate evaluation measure for imbalanced data. Kubat et al. [27] suggested the G-mean defined as the geometric mean of accuracies measured separately on each class. G-mean measures the balanced performance of a learning algorithm between these two classes, and is commonly utilized when performance of both classes is concerned and expected to be high simultaneously. Therefore G-mean is chosen to be the metric for representing the performance of each classifier. G-mean is defined as follows:

$$G\text{-}mean = \sqrt{Sensitivity \times Specificity} \tag{6}$$

where

$$Sensitivity = \frac{TP}{TP + FN} \tag{7}$$

$$Specificity = \frac{TN}{TN + FP} \tag{8}$$

G-mean is typically defined for binary classes but can be expanded to the scenario of multiple classes as the geometric mean of recall values of each class in [28]:

$$G\text{-}mean = \sqrt[1/K]{\prod_{k}^{K} R_k} \tag{9}$$

where M is the number of classes, R_k is the recall for each class. As each accuracy value (R_k) representing the classification performance of a specific class is equally accounted, G-mean is capable to measure the balanced performance among classes of a classification output.

A pseudo code of BWRF construction is shown in **Algorithm 1**. In addition, Figure 3 illustrates the algorithm.

Algorithm 1 BWRF

Require:
Training Dataset, D_{train}
Ensemble size, N
Over-sampling ratio, R_{os}
Ensure:
Ensemble classifier, *Ensemble*

```
minority class subset prog3, prog5, prog6, minSubset
majority class subset prog0, prog4, majSubset
over-sampled minority class subset, osMinSubset
osMinSubset = {}
Ensemble = {}
for each class ∈ minSubset do
    Over-sampling the class using SMOTE with R_os
    Add the new over-sampled dataset into osMinSubset
end for
for i = 1; i < N; i++ do
    under-sampled majority class subset with same amount of instances to osMinSubset, usMinSubset
    usMinSubset = {}
    for each class ∈ majSubset do
        Randomly under-sampling with bootstrap
        Add the new under-sampled dataset into usMinSubset
    end for
    Construct balanced training subset BD_i with osMinSubset and usMinSubset
    Build a tree classifier C_i without pruning on the BD_k
    Calculate the weights of each classifier in Ensemble with OOB instances
    Ensemble = Ensemble ∪ C_i
end for
```

Since data often exhibits characteristics at a local rather than global level, the framework of random forests can find more valuable local data properties so as to improve the quality of sampling. Moreover, the different imbalanced data distribution in each random subset makes the ensemble classifier robust to the evolving testing distribution. Furthermore, random forests ensemble can alleviate the effect of class overlapping on the imbalanced data distribution, since the two classes may be separable in some reduced feature subspace. In random forests, construct a single unprunned tree using the strategy of random selection of features and bootstrap sampling are more robust w.r.t. attribute noise and class label noise individually.

A method for extracting rules from a decision tree is quite simple. A rule can be extracted from a path linking from the root to a leaf node. All nodes in the path

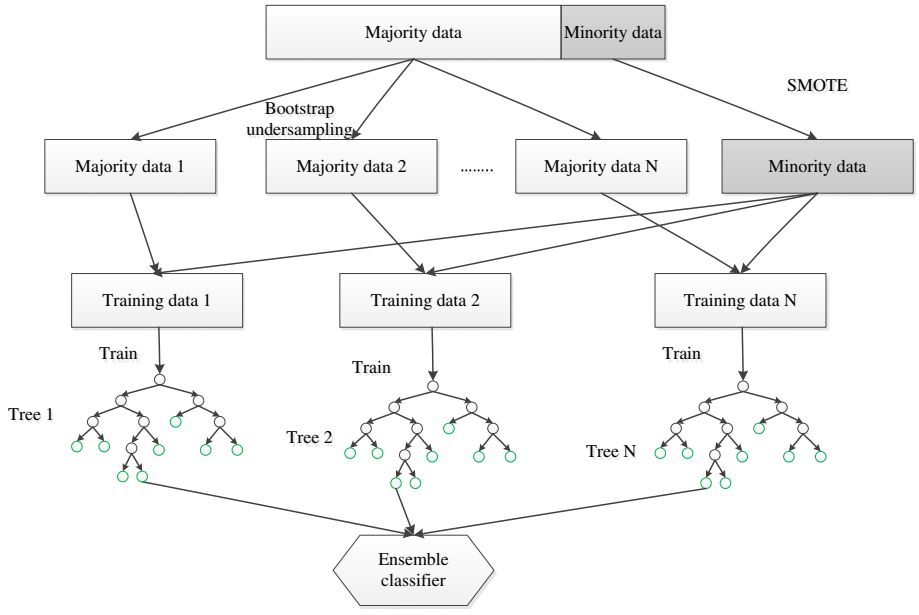

Figure 3 The procedure of BWRF.

are gathered and connected to each other using conjunctive operations. As for random forests, we need to extract and integrate the rules from random forests, remove these inconsistent conditions and output a new rule set which can be better applied to classify unseen data.

Ethical approval

The Health Research Ethics Board of the University of Alberta approved this research project.

Experiments

Evaluation measurements

To evaluate the performance of the physicians (human baseline), we use the successful rate as our measurement. It is the only measurement we can use for the human baseline. The successful rate of the physicians is defined as the number of successful recommendations (patient returns to work by receiving this recommendation) made by a physician over the number of all cases in the dataset. This is similar to the overall classification accuracy measurement. With class imbalance, this is not a good measurement. However, since we do not know the true class label of unsuccessful cases, it is neither possible to obtain the confusion matrix to use other measurements like Precision and F-measure, nor to know the measurements of each class. Therefore, we can only use overall classification accuracy in comparison with the human baseline, however, we do include sensitivity, specificity and G-mean for completeness.

All the classifiers are trained and validated in 10-fold cross validations. In 10-fold cross validation, the dataset is broken into 10 disjoint sets such that each set has (roughly) the same distribution. The classifier is learned 10 times such that in each iteration a different set is withheld from the training phase, and used instead to test the classifier. Although the performance of these algorithms is very promising in the 10-fold cross validation, independent test evaluations are still required to examine their true classification ability. However, we were unable to get more minority class data for our research at that time. Independents test datasets will be provided in the future for additional evaluations.

Experiment design

We conducted a variety of experiments and only those giving meaningful results are presented here. All the methods are on the Weka platform.

SMOTE + Tomek Link + ARIPPER

To mitigate the class imbalance, we use a progressive sampling approach to change the class distribution. The main reason for this combination is that the synthetic data from a minority class might invade the majority class too deeply and with the cleaning of Tomek Links, we could avoid potential overfitting.

1. Choose one minority class and fix the rest.
2. Increase the size of the selected class by a certain percentage P.
3. Train an ARIPPER classifier on the sampled dataset. If the true positive rate of the selected class increases significantly, undo the sampling and repeat step 2

with a larger percentage P' and step 3. However, if the size of the sampled class is greater than that of the largest class in the dataset or the increase is less than 2%, stop the sampling process.

4. Choose P as the final sampling percentage.

The final sampling percentage obtained for each minority class is 900%, 900% and 300% respectively. Figure 4 shows the class distribution before and after the sampling. We can visualize the sampled dataset using Principal Component Analysis (PCA) with the first two components as shown in Figure 5-a. We can see that on the left side class 3 has a minor overlap with class 6 while class 5 has invaded class 6. On the right side, class 0 and 4 are mixed together.

Tomek Link method to overcome class overlaps after sampling in order to avoid minority data generated from SMOTE from invading the majority class too deeply and causing classification difficulties

To make it easier for any algorithm to build a good classification model, a data cleaning stage is desirable to clean up the borders between each class. We apply the Tomek Link Cleaning method to weed out noise. Data points from different classes that form a Tomek Link are considered as borderline or noisy points, and generally can be removed. The details of the cleaning process are stated as follows:

1. Extract each pair of classes.
2. Identify the Tomek links between these two classes.
3. Remove noise or borderline points. If such cleaning improves the overall performance of the model, merge the cleaned up classes back to the whole dataset. Otherwise, undo the cleaning.
4. Repeat step 1 to 3 until all possible pairs of classes is processed.

Figure 5-b visualizes the dataset after Tomek Link cleaning. We can see that data points from Class 0 are completely mixed with Class 4. It is possible that the current selected features cannot separate these two classes effectively. Since feature selection is data dependent, we further sampled on prog0 as a possible solution for the class overlap. It is possible that we can select new and effective features to separate Class 0 and 4. Sampling on Class 0 may cause further overlapping between Class 0 and 4. But those points will be removed later as noise while the useful examples will be reinforced. We choose to sample 60% on class 0 and apply the same procedures above. 19 features are selected and the visualization using PCA is shown in Figure 5-c.

Clearly, we can see that part of Class 0 is now separable from Class 4. We then apply the Tomek Link cleaning on the new dataset. Figure 5-d visualizes the dataset after the cleaning. Those points from class 0 mixing with class 4 are removed while those separable remain in the data space. We then build a model using both ARIPPER and associative classification learner on this dataset. The evaluation is detailed in the next section.

Class decomposition + SMOTE + NCR (OVA)+ ARIPPER

In this approach we first decompose the dataset into 5 binary datasets. Each binary dataset contains the data from one positive class and all other classes are considered as one negative class. We use SMOTE to sample on the minority class in each binary dataset. The size of the minority class should be close to but smaller than that of the majority class. Then we use Neighborhood Cleaning Rule (NCR) as a data cleaning method to clean the data space. After the cleaning, five binary classifiers are created using different learning algorithms. To make a prediction for an unknown instance, each classifier generates a probability of that instance belonging to the positive class. We

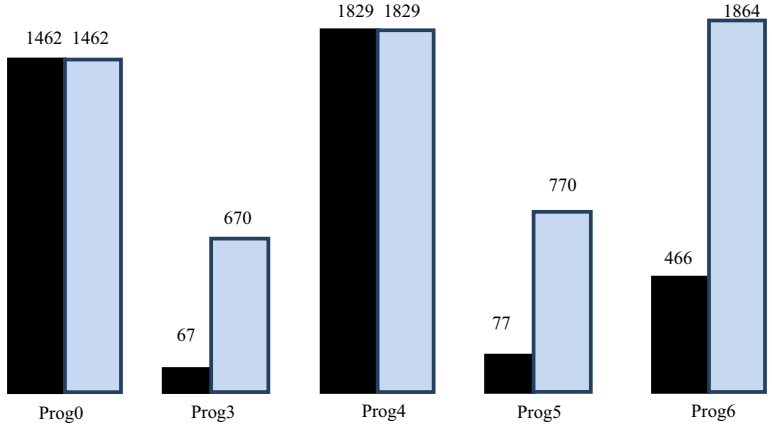

Figure 4 Class distribution before and after sampling-final dataset.

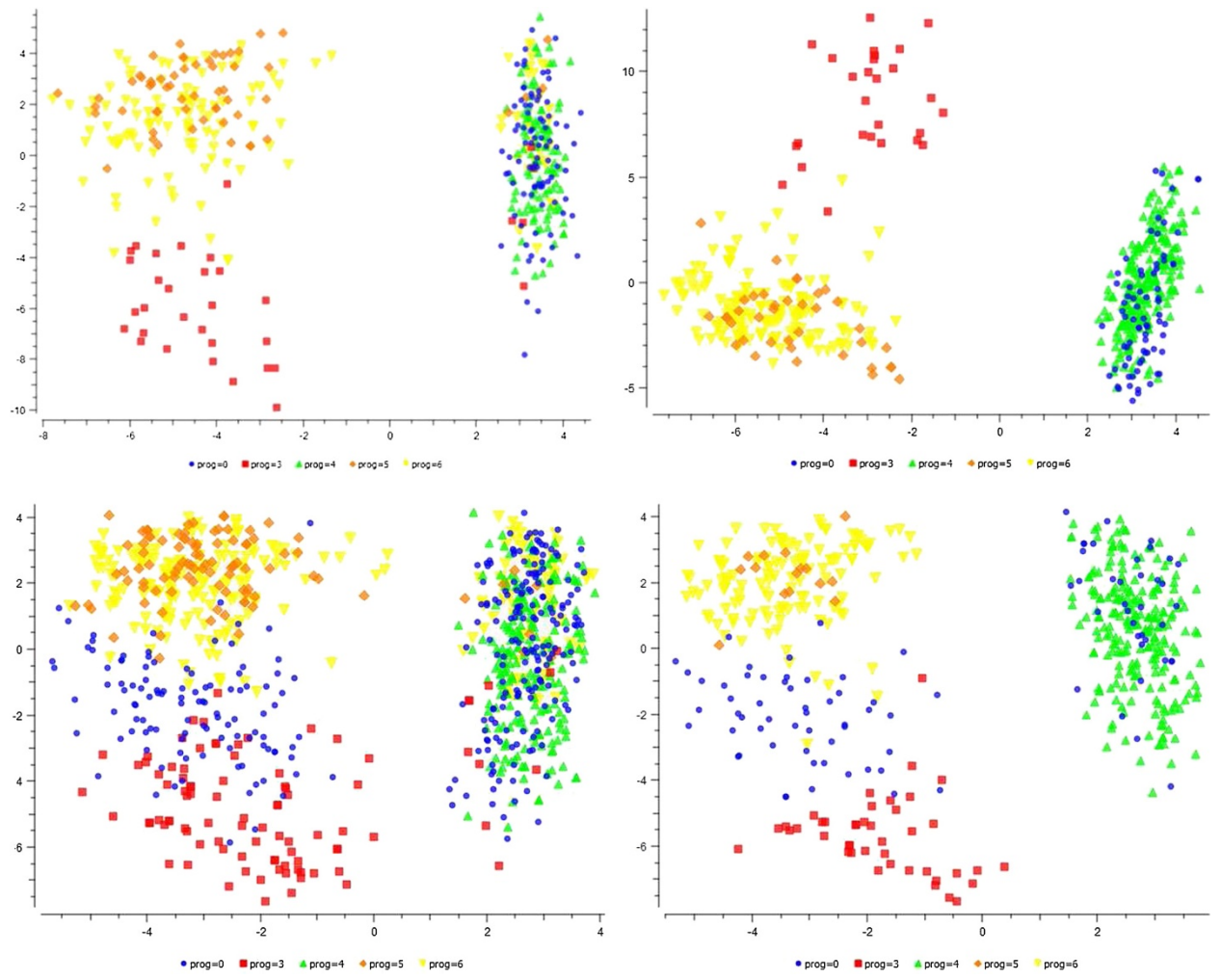

Figure 5 Dataset visualization a) left top: after SMOTE, b) right top: after Tomek Link cleaning, c) left bottom: after resampling, d) right bottom: after cleaning in second round.

use the imbalance rate to combine the probability prediction of all 5 classifiers: we take the product of prediction probability and the imbalance rate of its corresponding class as a final weight. The test instance belongs to the class with the highest weight.

BWRF

In our experiments reported below, we evaluate the effectiveness of our proposed BWRF algorithm show the different performance of BWRF with varying the ensemble size. Moreover, we test the different aggregation strategies: the weighted voting and the majority voting.

Experiment evaluation

As mentioned in the System requirements Section, our system should make multiple recommendations for the users to choose from. Since this is a Decision **Support** System, our goal is to help the physicians but not to replace them. The physicians can view the rules supporting the recommendations and make their own decisions from the recommendation pool.

However, from a computer science perspective, this is not sufficient. For a multi-class classification problem, the model has to finalize its prediction. Therefore, for each dataset obtained by using different data preprocessing strategies, we train a model from it using different algorithms and then evaluate their performance on the test set.

SMOTE + Tomek Link + ARIPPER (direct approach)

We first train a model using the ARIPPER algorithm. The rules obtained from this model were evaluated by experts from the Department of Physical Therapy and considered as meaningful rule sets. Our prototype system is implemented based on these rules. Table 2 shows the prediction evaluation on the test set using these four

Table 2 Evaluation on the test set (ARIPPER)

Criterion	HAvgRCF	SRHCF	HAvgCS	SRHCS	Potential
Accuracy	0.73	0.72	0.48	0.48	**0.78**

measurements: The "potential" means that if any of the predictions matches with the "true label", we count it as a correct prediction. Note that in rules generated from the RIPPER algorithm, there is a default rule with empty rule body and neither confidence nor Chi Square is applicable. So for each test instance, we assign to it both the selected prediction and the default prediction. If either of them matches with the true label, we count it as a correct prediction.

We then train three other classifiers using the Naive Bayes algorithm, C4.5 algorithm and ARC-BC [29] respectively. Table 3 shows the evaluation of each algorithm on the test set. The overall accuracy is 0.385, 0.478, and 0.470 respectively.

Class decomposition + SMOTE + NCR (OVA)

For the decomposition approach, we are using two base learners for each binary classifier Naive Bayes and RIPPER (original RIPPER). Table 4 show the confusion matrix of the evaluation on the test set using base learner Naive Bayes and RIPPER.

BWRF

we construct ensemble models with different ensemble size,and present the results on the test dataset. Table 5 and Figure 6 show the result of BWRF with weighted voting fusion, and Table 6 and Figure 7 show the result of BWRF with majority voting fusion. The over-sampling ratio in BWRF, R_{os} is empirically set to a moderate value, 200%, since that a under-sampling is performed for majority class subset when constructing each balanced training subset.

Furthermore, we also compared the common ensemble classifier, such as Adaboost, Bagging and RSM. The sizes of components are 100 in the all ensemble classifiers. Table 7 show the results.

Table 3 The results of common methods with 10 fold cross validation (the recall of each class, accuracy and G-mean)

Class	Naive Bayes	C4.5	ARC-BC
Prog0	0.071	0.05	0.132
Prog3	0.000	0.000	0.588
Prog4	0.969	0.980	0.856
Prog5	0.000	0.000	0.105
Prog6	0.000	0.000	0.069
Accuracy	0.482	0.478	0.471
G-mean	0.000	0.000	0.217

Table 4 The results of common methods with 10 fold cross validation (the recall of each class, accuracy and G-mean)

Class	Naive Bayes	ARC-BC
Prog0	0.080	0.846
Prog3	1.000	0.444
Prog4	0.284	0.314
Prog5	0.556	0.000
Prog6	0.060	0.000
Accuracy	0.201	0.473
G-mean	0.238	0.000

Comparison between common imbalanced data methods

We also empirically assessed BWRF against the state-of-the art methods for imbalanced data learning, such as Cost-Sensitive Classifier (CSC) [18], MetaCost [16], SMOTEBoost [13], Borderline-SMOTE (B-SMOTE) [23] and AdaBoost.NC combined with random over-sampling (ANCOS) [6]. These methods were considered because they are commonly used in research on class imbalance some from the algorithm perspective and some from the re-sampling perspective. We do not use the non-heuristic random re-sampling in our comparison since they are known to have drawbacks such as information loss or overfitting.

For the over-sampling methods including SMB and B-SMOTE, the amount of new data for each minority class (prog3, prog5 and prog6) is set to be 900%, 900% and 300% respectively. The sizes of components are 100 in the all ensemble classifiers. The penalty strength parameter in AdaBoost.NC is set 9. In the setting of CSC and MetaCost, the misclassification cost for majority classes is set to 1, and the one for each minority class (prog3, prog5 and prog6) is set to the size ratio between the largest class (prog0) and each class. The based classifier is chosen as RIPPER. Table 8 show the results.

Discussion

In this section, we discuss the evaluation in the former section. Note that the evaluation here has its own limitations as we mentioned in the earlier section.

In this paper we propose two ways to construct classification model using machine learning that categorizes

Table 5 The results of BWRF with weighted voting when varying the ensemble size with 10 fold cross validation (accuracy and G-mean)

Ensemble size	10	30	50	100	150	200
Accuracy	0.497	0.553	0.581	0.577	**0.585**	0.569
G-mean	0.124	0.161	0.197	0.263	0.297	**0.324**

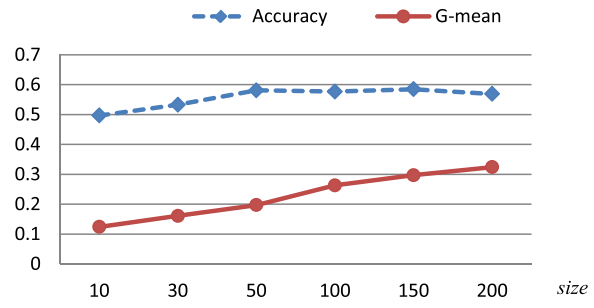

Figure 6 The result of BWRF with weighted majority voting.

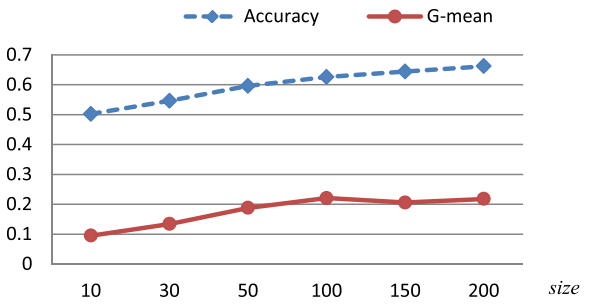

Figure 7 The result of BWRF with majority voting.

an injured worker into his own appropriate rehabilitation program, which is a real medical application problem with multiclass imbalanced data classification. From the comprehensive empirical evaluation, we show that both of our proposed methods encountered severe class imbalance and overlap in our data. Our proposed methods can be applied on many binary class or multiple class datasets, even if they are not imbalanced, such as loan recommendation, fraud prevention, spam detection, intrusion detection, climate data analysis, etc.

Moreover, the comparative results demonstrate that our methodologys have a better predictive ability than other single and ensemble classifiers in the context of imbalanced data with complex data characteristic. In order to learn from this complex dataset, both of the proposed methods are based on the re-sampling technique. The re-sampling technique is a popular method not only in dealing with class-imbalance, but also in filtering the noisy instances, so as to achieve a balanced and clean data distribution for use in the construction of rule-based classification models. From Table 2, we can find different quality measurements (e.g., HAvgRCF, SRHCF, HAvgCS and SRHCS) that can lead to different potential classification accuracy. We can see that by choosing the prediction with rule confidence measurement, the prediction accuracy reaches around 72%. Therefore, HAvgRCF is the best criterion for the ARIPPER algorithm on predicting unseen instance. From Table 3 and 4, it is apparent that the ARIPPER with hybrid progressive re-sampling outperforms the other common single classifier in terms of global accuracy. Unfortunately, since each instance has two predictions, we cannot analyze other measurement using the confusion matrix. The accuracy on test set using Naive Bayes, C4.5 and ARC-BC algorithms is lower than the human baseline.

However, ARC-BC does slightly better than the other two on predicting minority class examples. The standard classifier, such as C4.5, Naive Bayes, cannot solve the imbalanced data classification. Their biases are not suitable to the imbalanced data. For C4.5, for example, when building decision trees, the class label associated with a leaf is found by examining the training cases covered by the leaf and choosing the most frequent class. In the presence of the class imbalance problem, decision trees may be prone to ignore the minority class.

Since the pruning is based on the predicted error, there is a high probability that some branches that predict the small classes are removed and the new leaf node is labeled with a dominant class. For the class decomposition approach, the overall accuracy is also lower than the human baseline. That is because the decomposition strategy of the one class versus the other classes will worsen the imbalanced distribution even more for the small classes. However, one thing worth noticing is that by using Naive Bayes as the base learner, the model makes good predictions of the minority classes. But it is difficult to ensure good classification results on both the majority and minority classes at the same time. This is a common trade-off with the presence of class imbalance.

In addition to re-sampling methods, ensemble methods have also been used to improve performance on imbalanced datasets. They combine the power of multiple classifiers trained on similar datasets to provide accurate predictions for future instances. From Table 6, our proposed ensemble model is better than the other three

Table 6 The results of BWRF with majority voting when varying the ensemble size with 10 fold cross validation (accuracy and G-mean)

Ensemble size	10	30	50	100	150	200
Accuracy	0.502	0.546	0.596	0.626	0.644	**0.662**
G-mean	0.095	0.134	0.188	**0.221**	0.206	0.218

Table 7 The comparison between BWRF and other common ensemble classifiers with 10 fold cross validation (accuracy and G-mean)

	Adaboost	Bagging	RSM	BWRF
Accuracy	0.51	0.526	0.547	**0.577**
G-mean	0	0	0	**0.263**

Table 8 The results of the state-of-the-art methods for imbalanced data learning with 10 fold cross validation (accuracy and G-mean)

	SMOTEBoost	B-SMOTE	MetaCost	CSC	ANCOS
Accuracy	0.307	0.321	**0.412**	0.364	0.327
G-mean	0.104	0.084	0.133	0.156	**0.197**

ensemble models with the same amount of base classifier. The diversity is an important property for achieving a good performance from ensembles. The hybrid re-sampling strategy combined with random forest model not only injects more diversity into the ensemble via the learning algorithm, but also via the bias of the re-sampling algorithm.

From Table 7, we can find that our two methods perform very well when compared to the other four state-of-the-art methods for imbalanced data learning. SMOTEBoost and B-SMOTE only over-sample the minority class, hence they cannot solve it well if the noisy instances exist. The cost sensitive learning performs slightly worse than our methods. It may be because the ratio misclassification cost based on the size ratio between two classes is not appropriate, resulting in obtaining an unexpected performance. All the results in our experiments demonstrate the effects of using the hybrid sampling procedure combined with rule-based classifier to improve the performance of the common methods in terms of global accuracy and G-mean.

Furthermore, we focus on the rule representation. More precisely, we are interested in classification problems where discovered knowledge represents a function mapping objects (examples), described by a fixed set of attributes (features), to decision classes (concepts). The rule-based classifiers are capable of constructing a meaningful, editable and interpretable knowledge base (rules set) automatically without human inputs and have the potential of discovering something the human experts have overlooked.

The purpose of our project was to develop a classification algorithm and accompanying computer-based tool to help categorize individuals who were not working due to a wide variety of musculoskeletal disorders. Currently our prototype system implements the ARIPPER model trained from the first experiment since the rules are considered to be very meaningful from clinical perspective,so that they can make better decisions and possibly increase the efficiency of the decision-making process. This rule set shows a high "potential" on the test evaluation. As a decision support system, this should be sufficient since the clinician is the one who makes the final decision. To further evaluate the system, we need to do additional validations in real clinical settings. As for BWRF, we should integrate rules from multiple trees in a Random Forest which can help improve the comprehensiveness of the rules in the future.

Summary

In this work we build a decision support system with a knowledge base generated by machine learning algorithms. To tackle the multi-class imbalance and class overlap due to the nature of this clinical dataset, we apply several data re-sampling techniques to make it easier for the learning stage. Our results show that ARIPPER combined with hybrid re-sampling techniques and BWRF achieves a better performance, and ARIPPER combined with hybrid re-sampling techniques generates a meaningful rule-based model whose prediction ability is comparable to the clinicians. The Figure 8 shows a screen capture of providing multiple recommendations. Moreover, combining class decomposition with data re-sampling is a better way to effectively classify minority class examples than applying the data re-sampling directly.

Since our system provides human readable rules and presents these rules as evidence of any recommendation, a feedback loop is conceivable allowing an expert user to change these rules by directly injecting domain knowledge in the model initially automatically derived from the data.

As for future study, we plan to find a solution to determine the right prediction between the default prediction

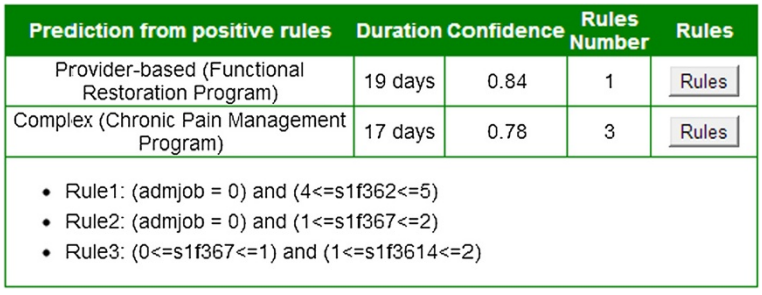

Figure 8 A screen capture of providing multiple recommendations.

and the other one as discussed in experiment 1. Building a binary classifier between these two predictions would be a good start. Another extension to our work is to integrate the negative model into the evaluation of the positive model such as canceling conflicting predictions under certain circumstances. To evaluate the system from a clinical perspective, additional validation in random clinical trials is required. In addition, we hope that we could get more data for the minority classes in later stage of this research.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
JZ: proposal and presentation of the method of ARIPPER, performance of the experiments related with ARIPPER, and implementation of the web application system. 50% of manuscript editing. PC: literature research, proposal and presentation of the random forests based ensemble method (BWRF), and performance of the experiments related with BWRF as well as the comparison with common ensemble classifiers and the state-of-the-art methods. 50% of manuscript editing. DG: Data acquisition and interpretation. ORZ: supervision and guidance all the model design as well as experiments procedure. All authors read and approved the final manuscript.

Acknowledgements
This research was funded by the Workers' Compensation Board of Alberta who also assisted with data acquisition.

Author details
[1]Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada. [2]Department of Physical Therapy, University of Alberta, Edmonton, Alberta, Canada.

Received: 28 June 2013 Accepted: 19 November 2013
Published: 4 December 2013

References

1. Chawla NV, Japkowicz N, Kolcz A: **Editorial: special issue on learning from imbalanced data sets.** *SIGKDD Explorations Spec Issue Learn Imbalanced Datasets* 2004, **6:**1–6.
2. He H, Garcia E: **Learning from imbalanced data.** *IEEE Trans Knowl Data Eng* 2009, **21**(9):1263–1284.
3. Kotsiantis S, Kanellopoulos D, Pintelas P: **Handling imbalanced datasets: a review.** *GESTS Int Trans Comput Sci Eng* 2006, **30:**25–36.
4. Yang Q, Wu X: **10 challenging problems in data mining research.** *Int J Inf Technol Decis Mak* 2006, **5**(4):597–604.
5. Zhou ZH, Liu XY: **Training cost-sensitive neural networks with methods addressing the class imbalance problem.** *IEEE Trans Knowl Data Eng* 2006, **18**(1):63–77.
6. Wang S, Yao X: **Multiclass imbalance problems: analysis and potential solutions.** *IEEE Trans Syst, Man, Cybernet, Part B* 2012, **42**(4):1119–1130.
7. Martin BI, Deyo RA, Mirza SK, Turner JA, Comstock BA, Hollingworth W, Sullivan SD: **Expenditures and health status among adults with back and neck problems.** *J Am Med Assoc* 2008, **299:**656–664.
8. Hadler NM: *Occupational musculoskeletal disorders*. Philadelphia, Pennsylvania, USA: Lippincott Williams & Wilkins, Wolters Kluwer; 2005.
9. Lane R, Desjardins S: *Canada. population and public health branch. Strategic policy directorate. Policy research division. Economic burden of illness in Canada Ottawa*. Ottawa, Canada: Health Canada; 2002.
10. Murray CJL, Vos T, Lozano R, Naghavi M, et al: **Disability-adjusted life years (DALYs) for 291 diseases and injuries in 21 regions, 1990–2010: a systematic analysis for the global burden of disease study 2010.** *The Lancet* 2013, **380:**2197–2223.
11. Chawla NV, Bowyer K, Hall L, Kegelmeyer W: **SMOTE: synthetic minority over-sampling technique.** *J Artif Intell Res* 2002, **16:**341–378.
12. Wilson DL: **Asymptotic properties of nearest neighbor rules using edited data.** *IEEE Trans Syst Man Cybernet* 1972, **3:**408–421.
13. Chawla N, Lazarevic A, Hall L, Bowyer K: **SMOTEBoost: Improving prediction of the minority class in Boosting.** In *Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)*. Heidelberg: Springer Berlin; 2003:107–119.
14. Tomek I: **Two modifications of CNN.** *IEEE Trans Syst, Man Cybernet* 1976, **6**(11):769–772.
15. Laurikkala J: **Improving identification of difficult small classes by balancing class distribution.** In *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*. London, UK: Springer-Verlag; 2001:63–66.
16. Domingos P: **Metacost: a general method for making classifiers cost-sensitive.** In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 99)*. New York, NY, USA: ACM; 1999:155–164.
17. Cao P, Zhao DZ, Zaïane O: **A optimized cost-sensitive SVM for imbalanced data learning.** In *Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining(PAKDD)*. Heidelberg: Springer Berlin; 2013:280–292.
18. Zadrozny B, Langford J, Abe N: **Cost-sensitive learning by cost-proportionate example weighting.** In *Proceedings of the 3rd IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society; 2003:435–442.
19. Weiss G, McCarthy K, Zabar B: **Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs.** In *Proceedings of international conference on data mining (ICDM 07)*. Washington, DC, USA: IEEE Computer Society; 2007:35–41.
20. Napierala K, Stefanowski J: **BRACID: a comprehensive approach to learning rules from imbalanced data.** *J Intell Inf Syst* 2012, **39**(2):335–373.
21. Zhu X, Wu X: **Class noise vs. attribute noise: a quantitative study.** *Artif Intell Rev* 2004, **22**(3):177–210.
22. Khoshgoftaar TM, Hulse JM, Napolitano A: **Comparing boosting and bagging techniques with noisy and imbalanced data.** *IEEE Trans Syst, Man Cybernet, Part A: Syst Hum* 2011, **41**(3):552–568.
23. Han H, Wang WY, Mao BH: **Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning.** In *Proceedings of the 2005 international conference on Advances in Intelligent Computing*. Heidelberg: Springer Berlin; 2005:878–887.
24. Cohen WW: **Fast effective rule induction.** In *Proceedings of the 12th International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann; 1995:115–123.
25. Li W, Han J, Pei J: **CMAR: Accurate and efficient classification based on multiple class-association rules.** In *Proceedings of the 1st IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society; 2001:369–376.
26. Leo B: **Random forests.** *Mach Learn* 2001, **45:**5–32.
27. Kubat M, Matwin S: **Addressing the curse of imbalanced training sets: one-sided selection.** In *Proceedings of the 4th International Conference on Machine Learning (ICML 97)*. San Francisco, CA: Morgan Kaufmann; 1997:179–186.
28. Sun Y, Kamel M, Wang Y: **Boosting for learning multiple classes with imbalanced class distribution.** In *Proceedings of the 6th IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society; 2006:592–602.
29. Zaïane O, Antonie ML: **Classifying text documents by associating terms with text categories.** In *Proceedings of the 13th Australasian database conference*. Darlinghurst, Australia: Australian Computer Society, Inc.; 2002:215–222.

doi:10.1186/2047-2501-1-15
Cite this article as: Zhang *et al.*: On the application of multi-class classification in physical therapy recommendation. *Health Information Science and Systems* 2013 **1**:15.