Multi-Class Protein Fold Classification Using a New Ensemble Machine Learning Approach

Aik Choon Tan[1] actan@brc.dcs.gla.ac.uk
David Gilbert[1] drg@brc.dcs.gla.ac.uk
Yves Deville[2] yde@info.ucl.ac.be

[1] Bioinformatics Research Centre, Department of Computing Science, University of Glasgow, 17 Lilybank Gardens, Glasgow, G12 8QQ, Scotland, United Kingdom
[2] Department of Computing Science and Engineering, Université catholique de Louvain Place Sainte Barbe 2, B-1348 Louvain-la-Neuve, Belgium

Abstract

Protein structure classification represents an important process in understanding the associations between sequence and structure as well as possible functional and evolutionary relationships. Recent structural genomics initiatives and other high-throughput experiments have populated the biological databases at a rapid pace. The amount of structural data has made traditional methods such as manual inspection of the protein structure become impossible. Machine learning has been widely applied to bioinformatics and has gained a lot of success in this research area. This work proposes a novel ensemble machine learning method that improves the coverage of the classifiers under the multi-class imbalanced sample sets by integrating knowledge induced from different base classifiers, and we illustrate this idea in classifying multi-class SCOP protein fold data. We have compared our approach with PART and show that our method improves the sensitivity of the classifier in protein fold classification. Furthermore, we have extended this method to learning over multiple data types, preserving the independence of their corresponding data sources, and show that our new approach performs at least as well as the traditional technique over a single joined data source. These experimental results are encouraging, and can be applied to other bioinformatics problems similarly characterised by multi-class imbalanced data sets held in multiple data sources.

Keywords: ensemble machine learning, multi-class protein fold classification, imbalanced data, learning from multiple data types

1 Introduction

Classification and prediction of biological entities has long been a central research theme in bioinformatics. In recent years, increasing quantities of high-throughput biological data have become available that can be used to understand the relationship between the protein sequence, structure, and function. These data have been distributed and maintained in different databases by different research groups. The problem is that each of these database resources contains a different subset of a set of specific biological knowledge that can only answer questions (queries) within its own domain but not questions that span domain boundaries [26]. The current release of the Protein Data Bank (PDB, Aug 2003) contains more than 22,000 proteins with experimentally determined 3D structures. The number of these protein structures is increasing rapidly as a result of several international structural genomics initiatives, but there is still a huge gap when compared to the over 1 million (PIR, July 2003) known protein sequences. Elucidating the similarities (or differences) between protein structures is very important in understanding the relationship between protein sequence, structure and function, and for the analysis of possible evolutionary relationships.

Experimental protein structure determination is expensive and time consuming, and therefore sophisticated computational methods have been developed and applied to detect, search for and compare

remote protein homology (at the sequence level) in the hope that knowledge can be transferred to the new unknown protein (e.g. inference about function). Most computational tools developed in protein fold prediction are primarily based on sequence similarity searches. If a new protein sequence (with an unknown structure) has high sequence similarity with a protein (with a known structure), then the new protein may share a similar fold with this structure. These approaches have improved the prediction accuracy that is capable of detecting proteins that have high sequence similarity [14] but have failed to perform well with low sequence similarities for closely related structures. Machine learning is one such computational approach that has been widely used in the development of automatic protein structure classification and prediction ([1, 24] and references from therin).

One of the aims of structural genomics is to enhance the understanding of the relationship between an amino acid sequence and its corresponding protein fold. Symbolic machine learning has been widely applied to protein fold recognition especially in deriving rules to assist human experts in understanding "folding rules" [6, 17, 25, 29]. Although statistical learning methods (e.g. neural networks, support vector machines) consistently exhibit better performance than symbolic machine learning techniques (e.g. decision trees, rule-based systems), the resulting complex models are very hard to interpret and therefore do not easily lead to new "insights" into this problem. One of the advantages of using symbolic machine learning approaches for this purpose is the generation human understandable classifiers (rules) from some biological background knowledge that can explain the relationship between sequence and structure.

The SCOP database [20] is a manually derived comprehensive hierarchical classification of known protein structures, organised according to their evolutionary and structural relationships. The database is divided into four hierarchical levels: Class, Fold, Superfamily and Family. For SCOP 1.61 (Nov 2002), the 44327 protein domains were classified into 701 folds, resulting in an average of 64 domains per fold. The number of domains per fold varies in SCOP, where some of the folds are highly populated (e.g. TIM barrels) while some of the folds contain only a few examples (e.g. the HSP40/DnaJ peptide-binding fold only contains one protein). Thus, when performing learning over the SCOP folds, the common one-versus-others approach (two-class problem) results in learning with an imbalanced data set. This imbalanced proportion of examples in each fold contributes to the poor performance of classical machine learning techniques (e.g. decision trees). Existing machine learning approaches tend to produce a strong discriminatory classifier (high accuracy) with very low sensitivity (also called completeness) when learning on these types of problems.

Our work is motivated by Ding and Dubchak's [9] analysis where they applied support vector machines and neural networks to construct one-versus-others and all-versus-all methods for classifying multi-class SCOP fold from sequence data. As observed in their paper, the classical learning methods perform badly due to the imbalanced proportion of the data or the so-called well-known "False Positives" problem [9]. Furthermore, the protein sequence data types may be distributed in different data sources. This is a common characteristic of Bioinformatics where it is often necessary to use data from a variety of independently curated and maintained databases. Applying learning techniques to infer over the multiple data sources remains one of the research challenges in both the machine learning [21] and bioinformatics communities [26].

We investigate the following problems in the context of classifying multi-class SCOP folds:
1. Can we improve the coverage of classifiers when learning from imbalanced data sets, where the protein examples from one class heavily outnumber those from the other classes (e.g. the negative examples are over 95%)?
2. Can we maintain the independence of different protein sequence data sources, but at the same time exploit the information induced from these data types by combining at the *pattern* level?

We have devised eKISS (**e**nsemble **K**nowledge for **I**mbalance **S**ample **S**ets), an ensemble learning method to solve these types of problems. The objective of eKISS is to generate one-verus-others classifiers which are capable of learning over multi-class examples under the skewed normal distribution of the training examples, as well as providing explanation to the user. In addition, we have extended our

method to learn over multiple data types, whilst preserving the independence of their corresponding data sources, and we show that our new approach performs at least as well as the traditional technique over a single joined data source.

The organisation of this paper is as follows: section 2 provides the machine learning background and related work for our approach; section 3 describes the eKISS framework; section 4 presents the training and test sets used in this study; section 5 describes the experimental designs of this work; section 6 discusses the results and the last section concludes this paper.

2 Machine Learning Background

2.1 Problem Formulation and Notations

For a multi-class supervised classification problem, a set of training data (positive and negative examples) in the form of {x, y | x ∈features, y ∈ classes} is provided to the learner. The learner's task is to induce a set of rules that can discriminate positive examples (E+) from negative ones (E−), and thus propose a classification for new instances. The common approach of treating multi-class learning is to transform the K classes into a set of two-class problems, which is also known as one-versus-others (OvsO) method. This approach faces one serious problem when learning over multi class problems: when we transform the K classes into K two-class problems, the positive examples of a class C_1 will be under-represented compared to the large number of negative examples for class C_2,...,C_K. The presence of a large amount of negative examples in the training data poses several pitfalls for classical machine learning systems.

The major problem of applying discriminative classical machine learning techniques in this situation is that they either generate a trivial rejector classifier, which classifies everything as a negative class (due to the negative examples being the majority class), or they overfit the positive examples (minority class) by generating large decision trees or highly complex neural networks. Most discriminative learning approaches apply recursive partitioning of the instance space into regions labelled with the majority class in that region. Furthermore, the heuristic of stopping and pruning for the partitioning procedure is constructed to avoid 'overfitting' the training examples and is solely based on the overall accuracy or the overall error rate of the classifier, representing a weak measurement under the imbalanced data. This heuristic, known as Occam's razor in the machine learning literature, suggests that a learning algorithm should prefer "simpler" to more "complex" classifiers in order to avoid overfitting the training examples. Wolpert's "No-free-lunch" (NFL) theorems point out that all such heuristics fail as often as they succeed in supervised learning problems [33]. Hence, most classical machine learning methods suffer the above drawbacks and perform poorly under the two-class imbalanced data situation. This scenario is described as the "curse of imbalanced data" in machine learning terminology [18].

2.2 Related Work

Multi-class learning. Another approach for handling multi-class problems is to generate all possible pair-wise two-class classifiers between K classes from the training examples. This approach is known as the all-versus-all (AvsA) method in which, given K classes of training examples, the machine learning methods will generate two-class classifiers for all the $K(K-1)/2$ classifiers. The unseen proteins are classified by these classifiers; every classifier provides a vote for the class label, and the majority voted class will be the predicted class for the new proteins. In the ideal case, the correct class will get the maximum votes for all the class-paired classifiers. In our case, we observed that this approach does not perform well due to the votes of the correct class being randomly distributed among other classes. Most classifiers will be a trivial rejector which votes for a negative class. This problem is also observed by [9] where they described the votes for the most popular voted class decreasing gradually from maximum to minimum and simply returning the class with the highest vote. The other disadvantage of this approach is the large number of classifiers, which is very difficult to analyse for the

purpose of providing insights into the protein sequence-structure relationships.

Sampling methods for imbalanced data sets. Recently, some attempts have been proposed in the machine learning community to overcome the two-class imbalanced data set problem, which primarily focus on sampling over the training examples. This is due to the analysis work by Weiss and Provost [31] where they concluded that the natural class distribution is often not the best distribution for learning a classifier. These sampling methods involve either (i) under-sampling - reducing the negative class by randomly removing the negative examples from training set, or (ii) over-sampling - increasing the positive class by replicating the positive examples. Several studies [4, 10] observed that over-sampling with replication does not always improve the minority (positive) class prediction. This is due to the classifier becoming very specific in the minority class decision region and leading to overfitting the examples [4]. Drummond and Holte [10] have shown that under-sampling approach performs better than the over-sampling method. The under-sampling approach forces the learning algorithm to focus on different degrees of the class distribution, at the same time increasing the presence of the minority class in the training examples, which can generate a more robust classifier. Although these sampling approaches appear to be appealing for solving imbalanced data problems, at the moment most of these techniques are mainly experimented in two-class problems [4, 10] as well as on artificial/synthetic data [15]. Removing or increasing the training examples is not suitable in this research domain due to the multi-class nature of the training examples as well as the limited amount of real protein data. Furthermore, in the protein fold classification problem, we would like to learn sequence-fold relationships from the sequence features by using non-redundant protein examples with low sequence similarities. Hence, we would like to preserve all the original training examples and propose a method that is capable of performing learning over these multi-class imbalanced data.

Learning from multiple data sources. One of the primary goals in bioinformatics is to design tools or systems that integrate multiple data sources, to perform learning and reasoning over these data, and to support inference and annotation mechanisms of new sequences. It is easy to imagine a single biological database containing all the information collected from diverse data sources, but the implementation of such unified database is non-trivial in practice. The problem is that these database resources contain different subsets of biological knowledge, and are maintained and upgraded regularly by different research groups [26]. There are various ways in which bioinformatics groups have tried to integrate biological databases but they generally fall into one of these categories: (i) link integration; (ii) view integration and (iii) data warehousing [26]. Recent reviews on some technologies for integrating biological data can be found in [26, 34]. A common and direct approach to performing machine learning over multiple data sources is to combine all the data into a joint table and then to apply learning techniques on this joint table to discover meaningful and/or discriminative patterns. This approach suffers from two major problems: ignoring or destroying data variation [16, 28], and increasing the learning time and memory size. As suggested by Stein [26], a better solution (unfortunately a non-trivial one) is to maintain the scientific and political independence of these databases, as well as allow the information that they contain to be easily integrated to enable cross-database queries. Maintaining the independence of multiple data sources whilst performing integration at the pattern level has motivated us to undertake this study. We would like to investigate the possibility of performing symbolic machine learning over multiple data types and then combining the decision rules in some way using our proposed ensemble learning method to classify protein folds.

3 Methods

In this paper, we have devised eKISS, an ensemble machine learning approach, which integrates the classifiers generated from the both one-versus-other and all-versus-all approaches to improve the coverage of the positive protein examples under the multi-class imbalanced data. Ensemble machine learning can be loosely defined as a set of classifiers whose individual decisions are combined in some

way to classify new examples [8]. Several empirical studies have shown that the performance of ensemble machine learning approaches is better than that of single methods due to the drawbacks discussed in the background section as well as the existing NFL theorems in the individual learning algorithms [2, 7, 23, 27]. The ensemble approach that we have applied in eKISS differs from the classical ones; is that instead of combining decisions from different base classifiers, we combine the rules of the base classifiers to generate new classifiers for final decision making. Furthermore, eKISS preserves the original distribution of the training data, making the resulting classifiers sensitive to the imbalanced situation.

3.1 The eKISS Method

The eKISS approach consists of combining rules of base classifiers to generate new classifiers. In this study, we have applied the PART rule-based machine learning technique to generate the base classifiers for our ensemble learning system. PART [12] is a rule-induction algorithm that avoids global optimisation, and generates accurate and compact rule sets by combining the paradigms of "divide-and-conquer" (C4.5, [22]) and "separate-and-conquer" (RIPPER, [5]). PART adopts a separate-and-conquer strategy in that it builds a rule, removes the covered instances, and continues constructing rules recursively by generating a partial decision tree from the remaining instances. The rule generated by PART is fewer in number and more compact compared to RIPPER and C4.5. We have performed a one-against-others procedure to generate K two-class classifiers (K is the number of classes) and also an all-against-all approach to produce $K(K-1)/2$ classifiers. We then combined these $K+K(K-1)/2$ base classifiers to generate one new classifier per class, called the ensemble classifiers. For this protein fold classification problem, the ensemble could combine $25 + (25 \times 24)/2 = 325$ base classifiers. Since PART is a rule-based learning system, each PART classifier contains a set of decision rules. To simplify the presentation, we assume that each base PART classifier contains k positive decision rules, denoted R_{i1}, R_{i2}, ..., R_{ik} for the base classifier number i.

Classical machine learning methods generate a classifier by performing a heuristic search through the possible classification rules (true hypotheses) of the given instance space, trying to find rules that can "best" approximate the true classification of the instance space. Since the heuristics employed so far are not suitable for multi-class imbalanced data sets, the classical machine methods suffer the "curse of learning in imbalanced data" [18] and most of the time return a near optimal trivial rejector classifier.

The basic idea of eKISS is to consider any rule R_{ij} as a potential candidate rule for each of the new ensemble classifiers. The main assumption made in eKISS is that all the rules generated by the PART learning algorithm represent possible classification rules, hence enlarging the search space. The eKISS search strategy is to find all the rules that correctly classify the examples in the positive class, hence improving the coverage of the positive examples under the multi-class imbalanced data situation. We also believe these positive rules are useful for providing insights to the human expert in understanding the relationships between protein structure and sequence information compared to a trivial rejector classifier. Technically, a rule will be included in the new ensemble classifier of a given class if it correctly classifies the positive examples of that class. As a decision measure, we use the normalised confidence measurement, $\mathrm{cf\_norm} = (\mathrm{TP} - 0.5)/(\mathrm{TP} + \mathrm{FP}(|\mathrm{E}+|/|\mathrm{E}-|))$ as the cut-off point for rule selection. The rules of the new classifier for class C_i are all the rules that satisfy the cut-off point. The normalised confidence measurement derived by [22] has been applied in evaluating the goodness of the decision rules derived from the decision trees. This measurement takes into account the ratio of the positive and negative examples and thus produces a much more sensitive measurement for computing the accuracy of the rules in an imbalanced data situation. Obviously, some (but not all) of the rules of the base classifier of a class will be in the new classifier of the same class, as well as rules from other base classifiers. In our system, cf_norm represents the tuning parameter for trade-off between the coverage of the positive examples (TP-rate) and the precision (positive predicted value). For each class, eKISS allows the user to select the classifier that best suits his/her classification purpose by

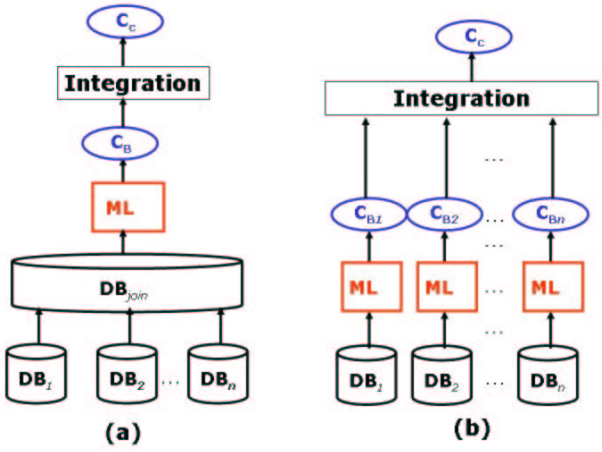


Figure 1: Schematic designs for the eKISS systems. (a) eKISS-ALL: the sequence data types are joined in a single table and using eKISS learning strategy over it. C_B represents the $K + K(K-1)/2$ base classifiers generated in this experiment, where K represents the 25 SCOP fold classes. C_c represents the final classifiers generated by eKISS. (b) eKISS-Multiple: the data types are learned independently and their corresponding $n \times (K + K(K-1)/2)$ base classifiers are integrated to generate the final classifiers C_c.

modifying the cf_norm value. Furthermore, in order to assist the user in selecting his/her choice of classifiers, the system can automatically generate ROC (Receiver Operating Characteristic) curves for each class, thus providing visualisation tool to facilitate the selection process.

3.2 Extending eKISS for Learning from Multiple Data Sources

As discussed in previous sections, the common approach to perform learning over multiple data sources is to join these data sources in a single joined table. The eKISS method can then be applied on the joined table, as illustrated in Figure 1(a). We refer to this approach as eKISS-ALL in the remainder of this paper. However, it is possible to use eKISS for the basis of a new ensemble knowledge approach in learning over multiple data sources as motivated by [26]. This approach, called eKISS-Multiple, is illustrated in Figure 1(b). Here, we maintained each data source as an independent entity instead of integrating them at the first instance. For each data source, one-versus-others and all-versus-all base classifiers are generated independently. The rules of all these base classifiers are then considered as potential candidates for the generation of the ensemble classifiers. Hence, if we have n different data sources, the potential candidate rules will be found in $n \times (K + K(K-1)/2)$ base classifiers, thus expanding the total search space for the eKISS system. We show later that by enlarging the search space in this way, more specific rules can be obtained for certain classes, hence improving the coverage of the minority classes.

Both the eKISS approaches have been designed to increase the sensitivity (positive coverage) of the classifiers. One would then expect the methods to have a reduced specificity (also called soundness). As will be shown in the results, this approach is useful when the ratio E+/E− is very low, and also when the initial classifiers yield little sensitivity. In that case, the loss of specificity is small compared to the increase of sensitivity, yielding more useful classifiers. Obviously, for some classes the base classifiers may be preferred to the new one.

4 Data Set

We have applied our method to the protein data set studied by Ding and Dubchak [9], which can be obtained from `http://www.nersc.gov/~cding/protein/`. The original data [9] contains a training

Table 1: Summary of the physico-chemical amino acid sequence data types used in this study.

Data types
Amino acid composition
Hydrophobicity
Polarity
Polarizability
Predicted Secondary Structure
Normalised van der Waals volume

set (N_{train}) and a test set (N_{test}). This training set was extracted from the PDB_select sets [13] and comprises 313 proteins from 27 most populated SCOP folds (more than seven examples for each fold) with all of the pair-wise sequence identities being less than 35%. The test set was extracted from PDB_40D [19] which contains 386 representatives of the same 27 SCOP folds with sequence similarity less than 35% (excluding the proteins in N_{train}). The features used in the learning system are extracted from protein sequences according to the method described in Dubchak *et al.* [11], where a protein sequence is represented by a set of parameter vectors on various physico-chemical and structural properties of amino acids along the sequence. These properties are hydrophobicity, polarity, polarizability, predicted secondary structure, normalised van der Waals volume and the amino acid composition of the protein sequence. Each sequence properties contained 21 continuous features. Since these properties are extracted individually from the corresponding protein sequences, one may treat these features as different data types stored in corresponding individual data sources [9, 11]. Table 1 summarises the protein sequence properties used in this study.

Before exploiting these data, we have analysed both the training and test sets and found some interesting errors in both data sets, especially in the training set (N_{train}). The first error is the inconsistency of the data sets. Ding and Dubchak [9] used the protein data from PDB_selects as the training set, at a time when the SCOP classification did not exist. Although [9] reclassified the training set according to the early SCOP database, we believe that the domain definitions in SCOP were still not well defined. Their test set was extracted from the more recent SCOP database (SCOP 1.48, Dec 1999) for which the domain definitions are well defined and which clearly contains major changes compared to the early SCOP version used to assign the training set. We found some protein examples in the training set which had not been assigned into domains at that time (due to the earlier SCOP domain definitions) but were present in the test set as different chopped domains. Probably this "dirty" data may have contributed to some poor performance of analysis [9]. At the same time, this also shows that the domain definition has evolved in the SCOP database over time by careful manual assignment; an automatic and intelligent system may facilitate this protein fold classification process.

Therefore, we cleaned the data set by removing the errors from both training and testing examples. We applied the protein fold classification according to SCOP 1.61 (Nov 2002, [20]) and Astral 1.61 [3] with sequence identity less than 40% (Nov 2002), removing those fold classes with less than 8 examples. After performing this cleaning stage, our protein fold data contained 582 examples distributed in 25 fold SCOP classes. We then randomly divided the data into a training set (408 protein examples) and a test set (174 protein examples).

5 Experimental Settings

We performed two different experiments with eKISS so as to answer the objectives of this study. In the first experiment, we joined all the sequence features into a single data source, and performed classification on this data. This straightforward combination approach provided 125 physico-chemical

Table 2: Average performance evaluation of eKISS-ALL (cf_norm = 0.69) and OvsO-PART.

SCOP Class	Method	Training Set (408 protein examples)				Test Set (172 protein examples)			
		Sn/TPR	FPR	PPV	F_1	Sn/TPR	FPR	PPV	F_1
All α	eKISS	0.537	0.258	0.059	0.106	0.764	0.342	0.060	0.111
(6 folds)	PART	0.068	0.025	0.043	0.053	0.045	0.035	0.046	0.045
All β	eKISS	0.491	0.270	0.067	0.118	0.865	0.435	0.073	0.135
(8 folds)	PART	0.025	0.034	0.016	0.019	0.008	0.042	0.009	0.009
α/β	eKISS	0.471	0.262	0.063	0.111	0.531	0.321	0.044	0.081
(8 folds)	PART	0.041	0.048	0.041	0.041	0.056	0.043	0.092	0.069
$\alpha + \beta$	eKISS	0.157	0.084	0.034	0.056	0.550	0.297	0.072	0.127
(2 folds)	PART	0	0.063	0	undef	0	0.033	0	undef
Small proteins	eKISS	1.000	0.496	0.166	0.285	1.000	0.640	0.112	0.202
(1 fold)	PART	0.100	0.073	0.100	0.100	0	0.053	0	undef

and structural properties of amino acids as our learning attributes. The goal of this experiment was two-fold. First, to evaluate eKISS in learning over multi-class imbalanced SCOP folds and to compare it with OvsO-PART- a classical rule-based system; second, to evaluate eKISS-ALL as an approach to perform learning over multiple data types using a single joined data source in classifying multi-class SCOP folds

The second experiment evaluated eKISS-Multiple, where we preserved the individual data sources, and performed base classification on each data source and then combined the rules using our approach. This has resulted in 6 different sequence data sources each containing 21 physico-chemical and structural properties of amino acids as our learning features (Table 1). Using both experiments, we were able to compare the performance of eKISS-Multiple with eKISS-ALL.

Standard measurements have been applied to evaluate the goodness of our classifiers compared to OvsO-PART: true positive rate (positive coverage or sensitivity, TPR = TP/(TP+FN)), false positive rate (FPR = FP/(FP+TN) or (1 − specificity)), positive predicted value (PPV = TP/(TP+FP)) and F_1-measure ((2TPR × PPV)/(TPR + PPV)) [30] which evaluates the trade off between sensitivity and positive predicted value.

We applied the WEKA machine learning package [32] to generate the base classifiers for eKISS. The eKISS ensemble system is written in Perl and the ROC curves are generated using gnuplot. We have compared eKISS with decision trees (J48), support vector machines, and neural networks from this package.

6 Results and Discussion

Comparison of eKISS-ALL and PART. We performed ten-fold cross-validation on the training data and tested on the test set by comparing the performance of eKISS-ALL and OvsO-PART. Table 2 summarises the performance of eKISS-ALL and PART on the training and test sets. From the results, eKISS-ALL outperforms PART on 20 out of 25 classes based on the F_1-measure. The results show that eKISS increases the sensitivity and also the positive predictive accuracy compared to PART. According to our experiments, eKISS-ALL performs better than OvsO-decision trees (20 classes), OvsO-SVM (24 classes), and artificial neural networks (21 classes). The goodness of eKISS classifiers can be improved by tuning the cf_norm values for each SCOP fold, which could lead to the generation of better classifiers. Although our method increases the true positive rate (TPR), as a trade-off it also increases the false positive rate (FPR). Since the objective of this study is to improve the rule coverage when classifying protein folds, we permit the rule-set to cover some false positives as a consequence of improving the positive coverage of classical machine learning. However, the results

show that the increase of TP-rate is higher than the corresponding increase of the FP-rate.

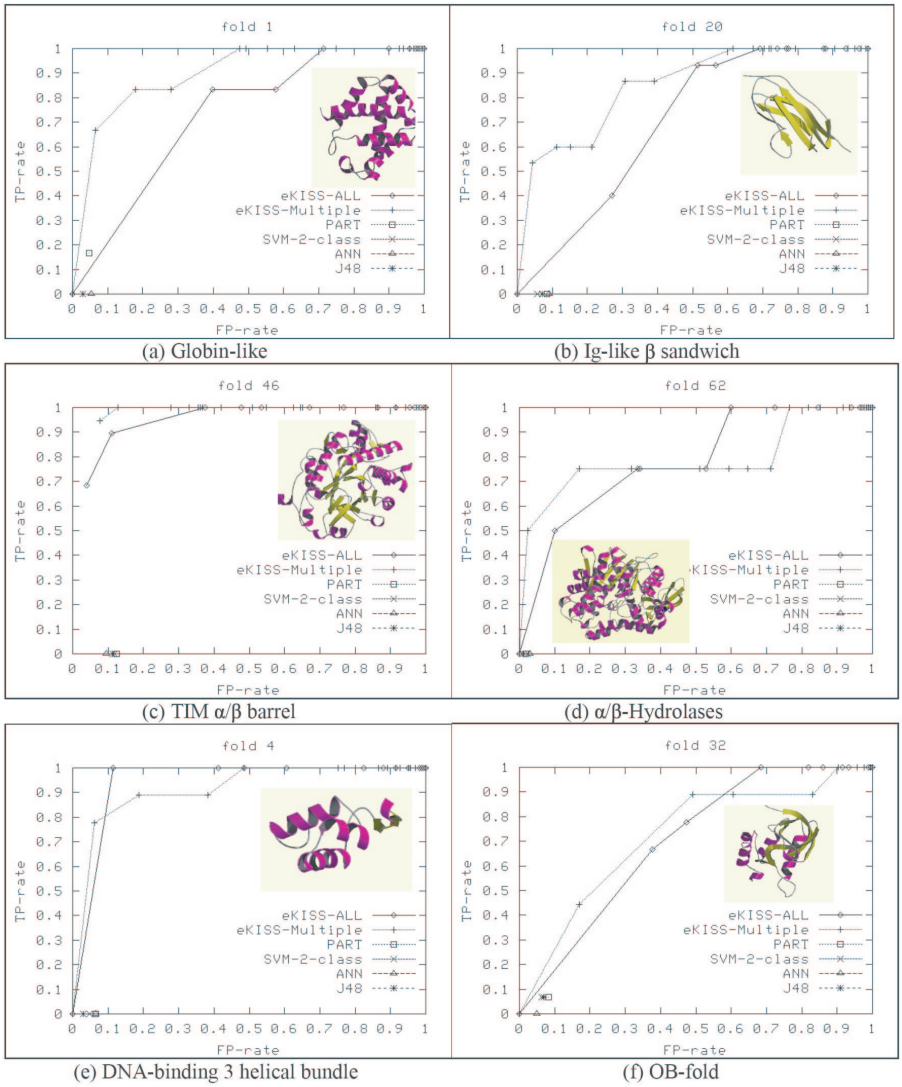
Comparison of eKISS-ALL and eKISS-Multiple. Figure 2 presents the ROC curves of selected classes generated from the test set by comparing eKISS-ALL (learning from single joined data source) and eKISS-Multiple (learning from multiple data sources). The ROC curves represent the trade-off between coverage (TPR on the y-axis) and the error rate (FPR on the x-axis) of a classifier. A good classifier will be located at the top left corner of the ROC graph, illustrating that this classifier has high coverage of true positives with few false positives. A trivial rejector will be at the bottom left corner of the ROC graph and a trivial acceptor will be at the top right corner of the graph. The two ROC curves in Figure 2 represent eKISS-Multiple and eKISS-ALL respectively for different cf_norm values. The higher the cf_norm in both curves is shifted further to the top left corner. These ROC curves clearly show that the eKISS-Multiple improved the classification performance of the joined data sources because the $\text{ROC}_{Multiple}$ curve is constantly higher than the ROC_{ALL} curve. These figures show that the classical machine learning tends to produce a trivial rejector under the imbalanced data set (left bottom corner), while both eKISS approaches greatly improved the classifier's sensitivity under different cf_norm values. These figures also show that the eKISS-Multiple perform better or at least as good as eKISS-ALL.

As expected, the four classical machine learning methods produce a trivial rejector located at the bottom left corner of the ROC curves (Figure 2). The same results were observed by [9] where they found that using multiple data types and applying majority voting on the results lead to better prediction accuracy. Our approach is different from [9] in two different ways: (i) we integrate all the (possible) decision rules from the base classifiers in some way to construct our new classifiers rather than the base classifiers' decisions; (ii) our final classifiers represent all the patterns (decision rules) learned from each individual data source compared to the classifiers formed by a majority voting system. From this observation, obviously eKISS-Multiple performed better than PART since it is better than eKISS-ALL which has been shown to be better than PART.

We believe that these unique features of eKISS provide better rules for understanding the relationships between the physico-chemical protein sequence properties and their corresponding fold. In addition, the eKISS framework may be an alternative solution for database integration problems currently faced by the bioinformatics community. Instead of creating a unified data warehouse for storing data sources from different origins, this approach allows the data sources to be stored in different locations and perform learning each data sources individually. The integration part is performed at the pattern (decision rule) level which is separate from the joined data level. Thus, if a data source has been updated, the eKISS system only needs to update its patterns by performing learning over the updated data source. Furthermore, using this approach, it is easy to 'plug-in' new protein data sources that could enhance the classifier's performance. For the purposes of this study, we have assumed that data types are internally consistent; in reality this will need to be ensured by a combination of mechanised and hand consistency checking as achieved for example by the approach of [35].

In order to verify the hypothesis that the set of rules from all the base classifier forms a useful search space for the generation of the new classifiers, we used a set of random rules (obtained by applying PART on a randomly generated data set). The performances of the resulting random classifiers were clearly worse than the performance of eKISS. This observation suggested that the decision rules that were selected to construct the eKISS classifiers are relevant in discriminating between the classes.

In general, eKISS performs well in learning from a small set of positive examples compared to the negative examples because eKISS is capable of generating a softer boundary for the classifier. It thus avoids problems connected with the strong discriminative boundary generated by classical learning systems. One of the essential conditions for ensemble methods to perform better than any of its individual classifier members is the diversity of the base classifiers. The reason for this is if several base classifiers make the wrong prediction (no diversity), the ensemble of these classifiers will not improve the prediction. But if the base classifiers make different predictions (diversity), the ensemble



(a) Globin-like (b) Ig-like β sandwich

(c) TIM α/β barrel (d) α/β-Hydrolases

(e) DNA-binding 3 helical bundle (f) OB-fold

Figure 2: ROC curves of the eKISS system on learning over multiple sequence data types (eKISS-Multiple) and the joined data (eKISS-ALL) compared with OvsO-PART, OvsO-SVM, neural networks and OvsO-decision trees on the test data.

may predict correctly by considering the majority votes from the diverse base classifier's decision. We believe that the base classifiers of eKISS are made diverse by combining the one-against-others and the all-against-all PART classifiers. Re-selecting the appropriate rules from these base classifiers creates the diversity of the ensemble and hence improves the positive coverage of eKISS. Obviously, the base classifiers for eKISS-Multiple are more diverse than eKISS-ALL since they are generated from different data sources. We believe some of the decision rules derived from eKISS-Multiple are more discriminative due to the fact that they represent the specific rules induced from individual data types, thus contributing to the better performance when they are combined in an ensemble at the latter stage.

Another interesting finding from this experiment is that the rule sets generated from eKISS are much smaller than those of the original PART system. We would have expected eKISS rule-sets to contain more rules compared to PART due to "collecting" additional rules from other classifiers, but it turns out they were fewer rules. We believe that the rule-sets of eKISS are useful for classifying protein folds and thus can assist wet experimental biologists in understanding the co-relationships between amino acid physico-chemical properties and functions. Compared with the two statistical

learning methods that we investigated in this study, both resulting models are hard to interpret and the performance maybe improved by specific tuning on the available parameters (e.g. different types of kernels for SVM and the number of nodes in neural networks).

7 Conclusion and Future Work

We have described eKISS, an ensemble method specifically designed to increase the sensitivity (positive coverage) of classifiers without losing corresponding specificity when learning over multi-class imbalanced data sets where examples from one class heavily outnumber those from other classes. We have applied this approach to the classification of 25 SCOP protein folds and our results show that this approach is useful when the ratio of E+/E− is very low, and also when the initial classifiers yield little sensitivity. In both cases, the loss of specificity is small compared to the increase in sensitivity, thus yielding more useful classifiers. We have also shown that eKISS is capable of learning from multiple data types, attaining as least as good performance compared with combining all data types in one table. This approach is very useful when learning over data obtained from independently curated and maintained databases. Furthermore, an advantage of the rules generated by eKISS compared with those of PART is that they are shorter and provide hints for understanding the relationship between the amino acid physico-chemical properties of a sequence and its constituted fold. The encouraging results of this approach can be applied to a wide range of other bioinformatics problems, and we plan to evaluate eKISS on other data sets with similar characteristics. Another extension that we would like to explore is to create larger and diverse base classifiers by incorporating decision rules generated from different rule-based systems that employ different inductive biases compared to PART.

Acknowledgments

We would like to thank Gilleain Torrance and Ali Al-Shahib for their useful comments. AC Tan was funded by a University of Glasgow studentship.

References

[1] Baldi, P. and Brunak, S., *Bioinformatics: The Machine Learning Approach*, MIT Press, 2001.

[2] Bauer, E. and Kohavi, R., An empirical comparison of voting classification algorithms: bagging, boosting, and variants, *Machine Learning*, 36:105–142, 1999.

[3] Chandonia, J.-M., Walker, N.S., Lo Conte, L., Koehl, P., Levitt, M. and Brenner, S.E., ASTRAL compendium enhancements, *Nucleic Acids Res.*, 30:260–263, 2002.

[4] Chawla, N.V., C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure, *Workshop on Learning from Imbalanced Datasets II, ICML*, 2003.

[5] Cohen, W.W., Fast effective rule induction, *Proc. 12th ICML*, 115–123, 1995.

[6] Cootes, A.P., Muggleton, S.H., and Sternberg, M.J.E., The automatic discovery of structural principles describing protein fold space, *J. Mol. Biol.*, 330:839–850, 2003.

[7] Dietterich, T.G., An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, *Machine Learning*, 40:139–157, 2000.

[8] Dietterich, T.G., Ensemble methods in machine learning, *Proc.1st International Workshop on MCS*, LNCS 1857:1–15, 2000.

[9] Ding, C.H.Q. and Dubchak, I., Multi-class protein fold recognition using support vector machines and neural networks, *Bioinformatics*, 17:349–358, 2001.

[10] Drummond, C. and Holte, R.C., C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, *Workshop on Learning from Imbalanced Datasets II, ICML*, 2003.

[11] Dubchak, I., Muchnik, I., and Kim, S.-H., Protein folding class predictor for SCOP: approach based on global descriptors, *Proc. 5th ISMB*, 104–107, 1997.

[12] Frank, E. and Witten, I.H., Generating accurate rule sets without global optimisation, *Proc.15th ICML*, 144–151, Morgan Kaufmann, 1998.
[13] Hobohm, U. and Sander, C., Enlarged representative set of proteins, *Protein Sci.*, 3:522–524, 1994.
[14] Holm, L. and Sander, C., Protein folds and families: sequence and structure alignments, *Nucleic Acids Res.*, 27:244–247, 1999.
[15] Japkowicz, N., Class imbalances: are we focusing on the right issue?, *Workshop on Learning from Imbalanced Datasets II, ICML*, 2003.
[16] Kell, D.B. and King, R.D., On the optimisation of classes for the assignment of unidentified reading frames in functional genomics programmes: the need for machine learning, *Trends in Biotechnology*, 18:93–98, 2000.
[17] King, R.D., Clark, D.A., Shirazi, J., and Sternberg, M.J.E., On the use of machine learning to identify topological rules in the packing of β-strands, *Protein Engineering*, 7:1295–1303, 1994.
[18] Kubat, M. and Holte, R.C., and Matwin, S., Machine learning for the detection of oil spills in satellite radar images, *Machine Learning*, 30:195–215, 1998.
[19] Lo Conte, L., Ailey, B., Hubbard, T.J.P., Brenner, S.E., Murzin, A.G., and Chothia, C., SCOP: a structural classification of proteins database, *Nucleic Acids Res.*, 28:260–262, 2000.
[20] Lo Conte, L., Brenner, S.E., Hubbard, T.J.P., Chothia, C., and Murzin, A.G., SCOP database in 2002: refinements accommodate structural genomics, *Nucleic Acids Res.*, 30:264–267, 2002.
[21] Mitchell, T.M., Machine learning and data mining, *Communications of the ACM*, 42:31–36, 1999.
[22] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
[23] Quinlan, J.R., Bagging, boosting, and c4.5, *Proc. 13th National Conference on Artificial Intelligence*, 725–730, 1996.
[24] Rost, B., Neural networks predict protein structure: hype or hit?, In *Artificial Intelligence and Heuristic Methods in Bioinformatics*, Fransconi, P. and Shamir, R. (Eds.), IOS Press, 2003.
[25] Selbig, J. and Argos, P., Relationships between protein sequence and structure patterns based on residue contacts, *Proteins*, 31:172–185, 1998.
[26] Stein, L.D., Integrating biological databases, *Nature Reviews Genetics*, 4:337–345, 2003.
[27] Tan, A.C. and Gilbert, D., An empirical comparison of supervised machine learning techniques in bioinformatics, *Proc. 1st Asia Pacific Bioinformatics Conference*, 219–222, 2003.
[28] Ting, K.M. and Low, B.T., Model combination in the multiple-data-batches scenario, *Proc. 9th ECML*, 250–266, 1997.
[29] Turcotte, M., Muggleton, S.H., and Sterberg, M.J.E., Automated discovery of structural signatures of protein fold and function, *J. Mol. Biol.*, 306:591–605, 2001.
[30] Van Rijsbergen, C.J., *Information Retrieval*, Butterworths, 1979.
[31] Weiss, G.M. and Provost, F., Learning when training data are costly: the effect of class distribution on tree induction, *J. Artificial Intelligence Research*, 19:315–354, 2003.
[32] Witten, I.H. and Frank, E., *Data Mining: Practical machine learning tools and techniques with java implementations*, Morgan Kaufmann, 2000.
[33] Wolpert, D.H., The supervised learning no-free-lunch Theorems, *Proc. 6th Online World Conference on Soft Computing in Industrial Applications*, 2001.
[34] Wong, L., Technologies for integrating biological data, *Briefing in Bioinformatics*, 3:389–404, 2002.
[35] Yeh, I., Karp, P.D., Noy, N.F., and Altman, R.B., Knowledge acquisition, consistency checking and concurrency control for gene ontology (go), *Bioinformatics*, 19:241–248, 2003.