Balancing Training Data for Automated Annotation of Keywords: a Case Study

Gustavo E. A. P. A. Batista[1], Ana L. C. Bazzan[2], and Maria Carolina Monard[1]

[1] Instituto de Ciências Matemáticas e de Computação, USP
Caixa Postal 668, 13560–970, São Carlos, Brazil
{gbatista,mcmonard}@icmc.usp.br
[2] Instituto de Informática, UFRGS
Caixa Postal 15064, 91501–970, Porto Alegre, Brazil,
bazzan@inf.ufrgs.br

Abstract. There has been an increasing interest in tools for automating the annotation of databases. Machine learning techniques are promising candidates to help curators to, at least, guide the process of annotation which is mostly done manually. Following previous works on automated annotation using symbolic machine learning techniques, the present work deals with a common problem in machine learning: that classes usually have skewed class prior probabilities, *i.e.*, there is a large number of examples of one class compared with just few examples of the other class. This happens due to the fact that a large number of proteins is not annotated for every feature. Thus, we analyze and employ some techniques aiming at balancing the training data. Our experiments show that the classifiers induced from balanced data sampled with our method are more accurate than those induced from the original data.

1 Introduction

Automatic annotation in genomics and proteomics is raising increasing interest among researchers and database curators. Each day the volume of data which has to be analyzed (mostly manually) increases to unmanageable levels. Thus, there is a clear need for automated tools to generate or at least support such an annotation. Following previous works on automated annotation using symbolic machine learning techniques, the present work deals with a common problem in machine learning: that classes frequently have skewed class distributions. This is especially the case in bioinformatics in general, and in automated annotation in particular. This happens due to the fact that a large number of proteins is not annotated for every feature. In this work we analyze some techniques to balance the data before applying standard machine learning techniques. The aim of this procedure is twofold: to improve the annotation regarding keywords (which is far from adequate), and to test those techniques for dealing with skewed class distributions in domains which deal with huge amounts of data and attributes. Our proposal is illustrated on databases related to proteins and families of proteins and concern the *Arabidopsis thaliana*, a model organism for plants.

This work is organized as follows. The next section describes related works concerning automated annotation using machine learning techniques. Section 3 explains the data collection procedure used and Section 4 discuses the problem of learning with imbalanced data sets. Methods commonly used to treat that problem as well as our approach are detailed in Section 5. Experiments and the results achieved so far are presented in Section 6. Finally, Section 7 concludes and outlines future research directions.

2 Related Work

There has been an explosion of data, information and computational tools coming out from the diverse genome projects. In some databases, this implies that an increasing amount of data must be manually analyzed before they are made available to the community. Although several sources of data are used, our concern is with the data on proteins and families of proteins which can be found in the SWISS–PROT database. Data on proteins are important to people working on bionformatics because one of the research goals is to understand how proteins interact in order to produce drugs, for instance. Moreover, SWISS–PROT is a *curated* database. The current release of SWISS–PROT has information on around 120 thousand entries (proteins). Thus, it is a challenge for any machine learning approach aiming for automated annotation or other goal.

Automatic annotation and machine learning are combined in [4] where the authors describe a machine learning approach to generate rules based on already annotated keywords of the SWISS–PROT database. Such rules can then be applied to unannotated protein sequences. Since this work has actually motivated ours, we provide here a brief introduction to it. Details can be found in [4].

In short, the authors have developed a method to automate the process of annotation regarding keywords in SWISS–PROT, based on the supervised learning algorithm C4.5 [8]. This algorithm works on training data, in this case, previously annotated keywords regarding proteins. Such data comprise mainly taxonomy entries, INTERPRO classification, and PFAM and PROSITE patterns. Given these data in the attribute-value format, C4.5 derives a classification for a target class, in this case, a given keyword.

Since dealing with the whole data in SWISS–PROT at once would be prohibitive due to its size, it was divided into protein groups according to the INTERPRO classification. Afterwards, each group was submitted to an implementation of the learning algorithm C4.5 contained in the software package Weka[1]. Rules were generated and a confidence factor for each rule was calculated. Confidence factors were calculated based on the number of false and true positives, by performing a cross–validation, and by testing the error rate in predicting keyword annotation over the TrEMBL database. The resulting framework (called Spearmint) can be accessed at http://www.ebi.ac.uk/spearmint.

The approach by Kretschmann and colleague was the basis for an automated annotation tool dealing only with data on mycoplasmas [1], as a way to reduce

[1] http://www.cs.waikato.ac.nz/ ml/weka

the data set and also because annotating proteins related to mycoplasmas was the aim of that project. Since the interest was on the annotation of keywords for proteins related to the *Mycoplasmataceae* family, the generation of rules was based on a reduced set of proteins extracted from SWISS–PROT. In this way it was possible to consider all attributes at once, in a different way of that proposed by [4]. Moreover, a single rule for each keyword was generated, thus avoiding inconsistencies in the proposed annotation. The rules were evaluated using a set of proteins from the TrEMBL database. Results show that the quality of annotation was satisfactory: between 60% and 75% of the given keywords were correctly predicted.

The work in [1] left open some research directions: first, to improve the training data in order to obtain more accurate classifiers; second, to extend the method to include other machine learning techniques; third, to extend to other fields of the SWISS–PROT database.

The second direction was tackled in [9] by comparing symbolic methods to a neural network model. The first direction is the major motivation for the present paper: we want to test the hypothesis that a balanced input data could produce more accurate rules for automated annotation, as it will be detailed in the next sections.

3 Data Collection

In this section we briefly describe our approach to tackle the field "Keywords" in the SWISS–PROT database. The reader is directed to [1] for more details. While the focus of that paper was on annotation of keywords related to sequences regarding the family of *Mycoplasmataceae*, our current work focusses on the *Arabidopsis thaliana* due to the following: a high number of proteins in SWISS–PROT is classified as "hypothetical protein" (around 50% of them according to data obtained in February 2002). Besides, the proteins in TrEMBL, which are also potential targets for comparison, are poorly annotated regarding the Keywords field[2]. The problem of lack of classification is not so acute regarding the *Arabidopsis thaliana* since this is a model organism. It has a better level of annotation, as well as there are more cross-references among databses. This latter issue is very important to us since the cross-references build up the basis of attributes for the machine learning techniques. On the other hand, being well annotated means that there are many more attributes to consider, a major challenge for machine learning techniques.

The raw data was collected directly from the SWISS–PROT database making a query for Organism=*Arabidopsis thaliana* and Keyword=<desired keyword> (for instance DNA-binding).

This query also delivered:

1. for SWISS–PROT: AccNumber, keywords;

[2] In the data we collected previously [1], 378 out of 1894 had no keyword at all, while 896 had no attributes such as INTERPRO classification.

2. for INTERPRO: IPR AccNumber;
3. for PROSITE: PS AccNumber.

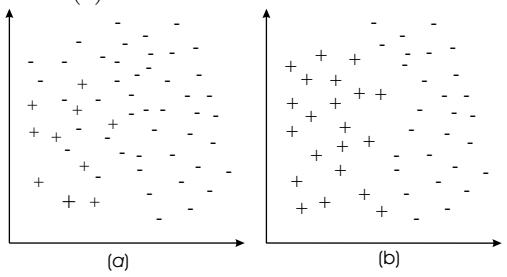
Since the *Arabidopsis thaliana* is well annotated regarding other attributes like PFAM and PRODOM classifications, we included these too. A typical input file describes the class (keyword); then a number of lines follow indicating how the attributes are mapped for all proteins in the training set.

4 Machine Learning and Imbalanced Data Sets

Learning from imbalanced data is a difficult task since most learning systems are not prepared to cope with a large difference between the number of cases belonging to each class. However, real world problems with these characteristics are common. Researchers have reported difficulties to learn from imbalanced data sets in several domains.

Why learning under such conditions is so difficult? Imagine the situation illustrated in Figure 1, where there is a large imbalance between the majority class (-) and the minority class (+). It also shows that there are some cases belonging to the majority class incorrectly labelled (noise). Spare cases from the minority class may confuse a classifier like *k-Nearest Neighbor (k-NN)*. For instance, 1-NN may incorrectly classify many cases from the minority class (+) because the nearest neighbor of these cases are noisy cases belonging to the majority class. In a situation where the imbalance is very high, the probability of the nearest neighbor of a minority class case (+) be a case of the majority class (-) is near 1, and the minority class error rate will tend to 100%, which is unacceptable for many applications.

Fig. 1. Many negative cases against some spare positive cases (a) balanced data set with well-defined clusters (b).



Decision trees also suffer from a similar problem. In the presence of noise, decision trees may become too specialized (overfitting), i.e., the decision tree inducer may need to create many tests to distinguish the minority class cases (+) from noisy majority class cases. Pruning the decision tree does not necessarily alleviate the problem. This is due to the fact that pruning remove some branches

considered too specialized, labelling new leaf nodes with the dominant class on that node. Thus, there is a high probability that the majority class will also be the dominant class of those leaf nodes.

5 Treating Imbalanced Data Sets

One of the most direct ways for dealing with class imbalance is to alter the class distributions toward a more balanced distribution. There are two basic methods for balancing class distributions:

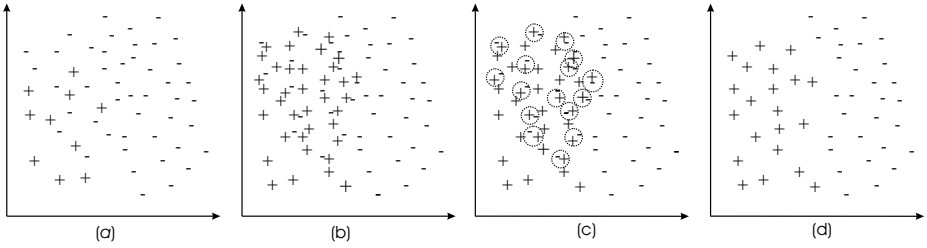
1. **Under-sampling**: these methods aim to balance the data set by eliminating examples of the majority class, and;
2. **Over-sampling**: these methods replicate examples of the minority class in order to achieve a more balanced distribution.

Both, under-sampling and over-sampling, have known drawbacks. Under-sampling can throw away potentially useful data, and over-sampling can increase the likelihood of occurring overfitting, since most of over-sampling methods make exact copies of the minority class examples. In this way, a symbolic classifier, for instance, might construct rules that are apparently accurate, but actually, cover one replicated example.

In this work we use a new method for balancing a data set that combines over and under-sampling techniques employed in the literature. The heuristics used by this method aim to overcome the drawbacks previously described.

The over-sampling technique used in this work, called Smote, was proposed in [2]. Its main idea is to form new minority class examples by interpolating between several minority class examples that lie together. Thus, the overfitting problem is avoided and causes the decision boundaries for the minority class to spread further into the majority class space.

Fig. 2. Balancing a data set: original data set (a); over-sampled data set (b); Tomek links identification (c); and borderline and noise examples removed (d).



The process of creating new minority class examples is illustrated in Figure 2. In (a) is shown the original data set, and in (b) is presented the same

data set with new minority class examples created artificially. The current implementation creates as many minority class examples as needed to balance the class distributions. This decision is motivated by the results presented in [11], in which is shown that allocating 50% of the training examples to the minority class, while it does not always yield optimal results, it generally leads to results which are no worse than, and often superior to, those which use the natural class distributions.

Although over-sampling minority class examples can balance class distributions, some other problems usually present in data sets with skewed class distributions are not solved. Frequently, class clusters are not well-defined since some majority class examples might be invading the minority class space. The opposite can also be true, since interpolating minority class examples can expand the minority class cluster, introducing artificial minority class examples too deep in the majority class space. Inducing a classifier under such situation can lead to overfitting. For instance, a decision tree classifier may have to create several branches in order to distinguish among the examples that lie in the wrong side of the decision border.

In order to create more well-defined class clusters, *Tomek links* [10] were identified and removed from the data set. A Tomek link can be defined as follows: given two examples x and y belonging to different classes, and be $d(x, y)$ the distance between x and y. A (x, y) pair is called a Tomek link if there is not a case z, such that $d(x, z) < d(x, y)$ or $d(y, z) < d(y, x)$. If two examples form a Tomek link, then or one of these examples is noise or both examples are borderline.

Tomek links have already been successfully employed as an under-sampling technique [5]. In this work, only majority class examples that participate of a Tomek link were removed, since minority class examples were considered too rare to be discarded. In our work, as minority class examples were artificially created and the data sets are currently balanced, then both, majority and minority class examples that form a Tomek link, are removed.

In Figure 2 is also shown the identification of Tomek links (c), and the removal of those examples (d). The elimination of noise and borderline examples create well-defined majority and minority class clusters.

6 Results and Discussion

In the experiments, we have employed the C4.5 learning system [8]. We selected as target class the DNA-binding keyword and all attributes which appeared related to this keyword. In order to reduce the number of attributes, we performed the query so to get data only for this particular keyword. However this posed the bias of having many positive examples, *i.e.*, too many examples with the keyword DNA-binding annotated compared with just few examples that do not have this keyword annotated.

More precisely, this query returned 235 examples, 197 of them had the keyword DNA-binding annotated and 38 of them had not. An initial experiment

with C4.5 trained over the original data set showed a low *false-positive rate* (FP), where only 2.03% of the examples labelled with the DNA-binding keyword will be erroneously classified as non-DNA-binding. On the other hand, the *false negative rate* (FN) might be considered unacceptable since it is expected that 23.33% of the non-DNA-binding examples will be erroneously classified as DNA-binding.

The most widely used performance measure for learning systems is the overall error rate. However, overall error rate is particularly suspect as a performance measure when studying the effect of class distribution on learning since they are strongly biased to favor the majority class [6]. When classes are imbalanced, a more reliable performance measure is the area under the ROC curve (AUC). ROC[3] graphs [7] are widely used to analyze the relationship between FN and FP for a classifier, and they are consistent for a given problem even if the distribution of positive and negative examples is highly skewed. Due to lack of space, ROC graphs are not shown in this work, however, we use the area under the ROC curve (AUC) to represent the expected performance as a single scalar. The AUC has a known statistical meaning: it is equivalent to the Wilconxon test of ranks, and is equivalent to several other statistical measures for evaluating classification and ranking models [3]. Higher values of AUC indicates that a classifier will present a better average performance over all costs and class distributions.

Aiming to obtain a reference in which the proposed balancing methods could be compared to, we applied two non-heuristical balancing methods: the random over-sampling that randomly replicates the minority class examples until a balanced class distribution is reached; and the random under-sampling that removes majority class examples in order to obtain a more balanced distribution. In order to avoid learning with too few examples, majority class examples were removed until a ratio of two majority class examples for each minority class example was obtained. Table 6 presents the results obtained for the original data set, as well as for the data sets obtained after the application of the random and Smote balancing methods.

Table 1. FP rate, FN rate, Overall error rate and AUC after cross validation

Data set	FP	FN	Overall	AUC
Original	2.03%±0.83%	23.33%±4.61%	5.53%±1.10%	87.77%±0.55%
Random Under-sampling	2.03%±0.83%	23.33%±4.61%	5.53%±1.10%	87.77%±0.55%
Random Over-sampling	12.63%±2.53%	7.50%±5.34%	11.88%±1.61%	90.13%±0.40%
Smote Over-sampling	15.24%±2.02%	2.50%±2.50%	13.17%±1.46%	91.33%±0.09%
Smote Over-sampling + Tomek links	14.74%±2.22%	2.50%±2.50%	12.75%±1.66%	91.58%±0.12%

Random under-sampling did not provide any improvement over the original data set. Although this method provided the exact same performance results, the decision trees induced were slightly different. Random over-sampling was able

[3] ROC is an acronym for *Receiver Operating Characteristic*, a term used in signal detection to characterize the tradeoff between hit rate and false alarm rate over a noisy channel.

to reduce significantly the FN rate, although it also increased the FP rate. For most real world domains, there is a trade-off between FP and FN, and reducing both rates is unfeasible. Using the AUC as a reference metric that combines FP and FN, random over-sampling obtained a higher AUC.

Finally, the proposed methods were applied to the data set. Firstly, Smote over-sampling was applied and later Tomek links were removed. The results obtained by both methods are shown in Table 6. Those methods obtained the lowest FN rate, 2.50%, but also the highest FP rate, 15.24% and 14.74%, respectively. Comparing with random over-sampling those methods present a 200% improvement in FN rate, with an increase of the FP rate in approximately 21% and 17%, respectively. Using the AUC as a reference metric, the data set over-sampled by the application of Smote over-sampling obtained the second highest AUC, only being surpassed by the proposed method that allies Smote over-sampling and Tomek links.

7 Conclusion and Future Work

This paper presents methods to deal with the problem of learning with skewed class distributions applied to automated annotation of keywords in the SWISS–PROT database. The use of similar methods was proposed in [4] and [1]. However, in neither work the problem of imbalance was tackled. Our data comes basically from databases of proteins and motifs, and are related to the organism *Arabidopsis thaliana*. We selected some target classes (the keywords) and all attributes which appeared in those databases for each class. In order to reduce the number of attributes, we performed the query so to get data only for a particular keyword. However this posed the bias of having too many positive examples; in reality, when performing a more general query (e.g. one that just queries the SWISS–PROT for a given organism), the opposite happens: one ends up with too many *negative* examples. This does not invalidate the methodology we follow here which showed good results; on the contrary: these methods can then be applied to more keywords at once just the same way we did here.

The obvious direction for this research is to apply the methods for balancing the data when one has several keywords, which brings two questions: the large number of negative examples, and the increase in the number of attributes, which might lead to the necessity of applying methods for attribute selection.

Other directions can be the use of other classifiers, the comparative study of the present results to those achieved when the data on all keywords is considered at once, and especially the question of missing data. This latter problem was reported for the *Mycoplasmataceae* training data and applies for some keywords related to the *Arabidopsis thaliana* as well.

Acknowledgements This research is partially supported by Brazilian Research Councils CAPES and FAPESP.

References

1. A. L. C. Bazzan, S. Ceroni, P. M. Engel, and L. F. Schroeder. Automatic Annotation of Keywords for Proteins Related to *Mycoplasmataceae* Using Machine Learning Techniques. *Bioinformatics*, 18(S2):S1–S9, 2002.
2. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Jounal of Artificial Intelligence Research*, 16:321–357, 2002.
3. D. J. Hand. *Construction and Assessment of Classification Rules*. John Wiley and Sons, 1997.
4. E. Kretschmann, W. Fleischmann, and R. Apweiler. Automatic Rule Generation for Protein Annotation with the C4.5 Data Mining Algorithm Applied on SWISS–PROT. *Bioinformatics*, 17:920–926, 2001.
5. M. Kubat and S. Matwin. Addressing the Course of Imbalanced Training Sets: One-Sided Selection. In *XIV International Conference in Machine Learning*, pages 179–186, San Francisco, CA, 1997. Morgan Kaufmann.
6. M. C. Monard and G. E. A. P. A. Batista. Learning with Skewed Class Distribution. In J. M. Abe and J. I. da Silva Filho, editors, *Advances in Logic, Artificial Intelligence and Robotics*, pages 173–180, São Paulo, SP, 2002. IOS Press.
7. F. J. Provost and T. Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Knowledge Discovery and Data Mining*, pages 43–48, 1997.
8. J. R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, CA, 1988.
9. L. F. Schroeder, A. L. C. Bazzan, J. Valiati, P. M. Engel, and S. Ceroni. A Comparison Between Symbolic and Non–Symbolic Machine Learning Techniques in Automated Annotation the "Keywords" Field of SWISS-PROT. In *I Workshop Brasileiro de Bioinformática*, pages 80–87, 2002.
10. I. Tomek. Two Modifications of CNN. *IEEE Transactions on Systems Man and Communications*, SMC-6:769–772, 1976.
11. G. M. Weiss and F. Provost. The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report ML-TR-44, Rutgers University, Department of Computer Science, 2001.