Data Imbalance in Surveillance of Nosocomial Infections

Gilles Cohen[1], Mélanie Hilario[2], Hugo Sax[3], and Stéphane Hugonnet[3]

[1] Medical Informatics Division, University Hospital of Geneva,
1211 Geneva, Switzerland
Gilles.Cohen@dim.hcuge.ch

[2] Artificial Intelligence Laboratory, University of Geneva,
1211 Geneva, Switzerland
Melanie.Hilario@cui.unige.ch

[3] Department of Internal Medicine, University Hospital of Geneva,
1211 Geneva, Switzerland
{Hugo.Sax,Stephane.Hugonnet@hcuge.ch}

Abstract. An important problem that arises in hospitals is the monitoring and detection of nosocomial or hospital acquired infections (NIs). This paper describes a retrospective analysis of a prevalence survey of NIs done in the Geneva University Hospital. Our goal is to identify patients with one or more NIs on the basis of clinical and other data collected during the survey. In this classification task, the main difficulty resides in the significant imbalance between positive or infected (11%) and negative (89%) cases. To remedy class imbalance, we propose a novel approach in which both oversampling of rare positives and undersampling of the non infected majority rely on synthetic cases generated via class-specific subclustering. Experiments have shown this approach to be remarkably more effective than classical random resampling methods.

1 Introduction

Surveillance is the cornerstone activity of infection control, whether nosocomial[4] or otherwise. It provides data to assess the magnitude of the problem, detect outbreaks, identify risk factors for infection, target control measures on high-risk patients or wards, or evaluate prevention programs. Ultimately, the goal of surveillance is to decrease infection risk and consequently improve patients' safety. There are several ways to perform surveillance, each method having its advantages and drawbacks. The gold standard is hospital-wide prospective surveillance, which consists in reviewing on a daily basis all available information on all hospitalized patients in order to detect all nosocomial infections. This method is labor-intensive, infeasible at a hospital level, and currently recommended only

[4] A nosocomial infection is a disease that develops after a patient's admission to the hospital and is the consequence of treatment—not necessarily surgical—or work by the hospital staff. Usually, a disease is considered a nosocomial infection if it develops 72 hours after admission.

P. Perner et al. (Eds.): ISMDA 2003, LNCS 2868, pp. 109–117, 2003.

for high-risk, i.e., critically ill patients. As an alternative and more realistic approach, prevalence surveys are being recognized as a valid surveillance strategy and are becoming increasingly performed. Their major limitations are their retrospective nature, the dependency on readily available data, a prevalence bias, the inability to detect outbreak (depending on the frequency the surveys are performed), and the limited capacity to identify risk factors. However, they provide sufficiently good data to measure the magnitude of the problem, evaluate a prevention program, and help allocate resources. They give a snapshot of clinically active NIs during a given index day and provide information about the frequency and characterisitics of these infections. The efficacy of infection control policies can be easily measured by repeated prevalence surveys [4].

2 Data Collection and Preparation

The University Hospital of Geneva (HUG) has been performing yearly prevalence studies since 1994 [6]. The methodology of prevalence surveys is as follows. The investigators visit all wards of the HUG over a period of approximately three weeks. All patients hospitalized for 48 hours or more at the time of the study are included. Medical records, kardex, X-ray and microbiology reports are reviewed, and additional information eventually obtained by interviews with nurses or physicians in charge of the patient. All nosocomial infections are recorded according to modified Centres for Disease Control (CDC) criteria. Only infections still active at any point during the six days preceding the visit are included. Collected variables include demographic characteristics, admission date, admission diagnosis, comorbidities, McCabe score, type of admission, provenance, hospitalization ward, functional status, previous surgery, previous intensive care unit (ICU) stay, exposure to antibiotics, antacid and immunosuppressive drugs and invasive devices, laboratory values, temperature, date and site of infection, fulfilled criteria for infection. All this information (except those related to infection) are collected for infected and non-infected patients.

Although less time-consuming than prospective surveillance, a prevalence survey nevertheless requires considerable resources, i.e., approximately 800 hours for data collection and 100 hours for entering data in a electronic data base. Due to this important effort, we can afford to perform such studies only once a year. What is particularly time-consuming is the careful examination of all available information for all patients, in order to detect those who might be infected. The aim of this pilot study is to apply data mining techniques to data collected in the 2002 prevalence study in order to detect vulnerability to nosocomial infections on the basis of the factors described above.

The dataset consisted of 688 patient records and 83 variables. With the help of hospital experts on nosocomial infections, we filtered out spurious records as well as irrelevant and redundant variables, reducing the data to 683 cases and 49 variables. In addition, several variables had missing values, due mainly to erroneous or missing measurements. We replaced these missing values with the class-conditional mean for continuous variables and the class-conditional mode for nominal ones.

3 The Class Imbalance Problem

The major difficulty inherent in the data (as in many medical diagnostic applications) is the highly skewed class distribution. Out of 683 patients, only 75 (11% of the total) were infected and 608 were not. The class imbalance problem is particularly crucial in applications where the goal is to maximize recognition of the minority class[5] The issue of class imbalance has been actively investigated and remains largely open, but for lack of space we present the major trends very briefly. The interested reader can refer to [7] for a more comprehensive state of the art.

One solution to class imbalance is oversampling the majority class. Typically, cases from the minority class are replicated until the desired class proportions are attained. Recently, Chawla et al. [1] replaced straightforward case cloning by generating synthetic minority class cases from real ones, using a technique based on nearest neighbors. The opposite approach consists in undersampling, i.e., subsampling the majority class until its size matches that of the minority class. Although subsampling is often be done randomly, more guided strategies have been proposed; for instance, Kubat et al. [9] eliminate redundant, noisy and borderline cases to downsize the majority class. A third alternative, known as recognition-based learning, consists in simply ignoring one of the two classes and learning from a single class; one-class SVMs [10] illustrate this approach A fourth class of methods involves adjusting misclassification costs: failure to recognize a positive case (false negative) is penalized more than erroneously classifying a negative case as positive (false positive) [2]. Contrary to sampling approaches, cost-based approaches to imbalance involve modifying the learning algorithm's objective function. However, there are other ways of biasing the inductive process to boost sensitivity (i.e., capacity to recognize positives). Joshi et al. [8] decompose set-covering rule induction into a two-stage process: the first phase maximizes recall of the positive class, while the second phase refines results of the first phase in order to improve precision.

In this paper we propose an approach in which **both** oversampling and undersampling (and their combination) are performed using synthetic cases generated in the form of cluster prototypes. The first variant of this approach is K-means based undersampling of the majority class. This strategy appears unnecessary and even counterintuitive at first sight; one could indeed understandably question the need to generate artificial examples to represent an already over-represented class. The rationale is that since the artificial examples are built as centroids of subclusters of the majority class, they thus distill the essential discriminating properties of that class. For a given cardinality, one could therefore legitimately expect a set of these prototypes to be more informative than a set of real cases. To shrink the majority class, we ran K-means clustering on the training instances of this class with $K = N_{min}$, the size of the minority class. These N_{min} prototypes were then used as sole representatives of the minority

[5] For convenience we identify positive cases with the minority and negative cases the majority class.

class so that training was performed on equally distributed classes. The second variant involves oversampling the minority class using agglomerative hierarchical clustering (AHC). Partitional clustering methods like K-means are less adequate for this task due to the small number of clusters (and therefore of prototypes) that can be created. The number of clusters K should be considerably less than N_{min}; with K=N_{min} each cluster will have a single member which will naturally be its centroid. This is inacceptable since the idea is precisely to synthesize examples that are different from the existing cases (otherwise we revert to standard case duplication). Given this limit on K, the number of synthetic cases generated will be insufficient to attain inter-class equilibrium. Hierarchical clustering does not share this limitation, since the number of (eventually nested) clusters can be augmented at will by increasing the number of levels and varying the inter-cluster distance metrics used. We therefore turned to AHC using single- and complete-linkage in succession to vary the clusters produced. Clusters were gathered from all levels of the resulting dendograms. Their centroids were computed and concatenated with the original positive cases, thus upsizing the positive class to match the negative class. Finally, the third variant is the combination of AHC-based oversampling and K-means based undersampling. Experiments conducted to assess these variants are described in Section 4 and results are discussed in Section 5.

4 Experimental Setup

4.1 Learning Algorithms

We compared alternative solutions to the class imbalance problem using five learning algorithms with clearly distinct inductive biases. Decision trees such as those built by C4.5 are models in which each node is a test on an individual variable and a path from the root to a leaf is a conjunction of conditions required for a given classification [11]. Naive Bayes computes the posterior probability of each class given a new case, then assigns the case to the most probable class. IB1 is basically a K-nearest-neighbors [3] classification algorithm, while Adaboost builds a single-node decision tree iteratively, focusing at each step on previously misclassified cases [5]. Support vector machines (SVMs) [12] represent a powerful learning method based on the theory of Structural Risk Minimisation (SRM). SVMs learn a decision boundary between two classes by mapping the training data onto a higher dimensional space and then finding the maximal margin hyperplane within that space.

4.2 Performance Metrics

In classification tasks, the most commonly used performance metric by far is predictive accuracy. This metric is however close to meaningless in applications with significant class imbalance. To see this, consider a dataset consisting of 5% positive and 95% negatives. The simple rule of assigning a case to the majority class would result in an impressive 95% accuracy whereas the classifier would have failed to recognize a single positive case—an inacceptable situation in medical diagnosis. The reason for this is that the contribution of a class to the overall accuracy rate is a function of its cardinality, with the effect that rare positives have an almost insignificant impact on the performance measure.

To discuss alternative performance criteria we adopt the standard definitions used in binary classification. TP and TN stand for the number of true positives and true negatives respectively, i.e., positive/negative cases recognized as such by the classifier. FP and FN represent respectively the number of misclassified positive and negative cases. In two-class problems, the accuracy rate on the positives, called sensitivity, is defined as $TP/(TP+FN)$, whereas the accuracy rate on the negative class, also known as specificity, is TN/(TN+FP). Classification accuracy is simply $(TP+TN)/N$, where $N = TP+TN+FP+FP$ is the total number of cases.

To overcome the shortcomings of accuracy and put all classes on an equal footing, some have suggested the use of the geometric mean of class accuracies, defined as $gm = \sqrt{\frac{TP}{TP+FN} * \frac{TN}{TN+FP}} = \sqrt{sensitivity * specificity}$. The drawback of the geometric mean is that there is no way of giving higher priority to the rare positive class. In information retrieval, a metric that allows for this is the F-measure $F_\beta = \frac{PR}{\beta P + (1-\beta)R}$, where R (recall) is no other than sensitivity and P (precision) is defined as $P = TP/(TP+FP)$, i.e., the proportion of true positives among all predicted positives. The β parameter, $0 < \beta < 1$, allows the user to assign relative weights to precision and recall, with 0.5 giving them equal importance. However, the F-measure takes no account of performance on the negative class, due to the near impossibility of identifying negatives in information retrieval. In medical diagnosis tasks, however, what is needed is a relative weighting of recall and specificity. To combine the advantages and overcome the drawbacks of the geometric mean accuracy and the F-measure, we propose the mean class-weighted accuracy (CWA), defined formally for the K-class setting as $cwa = \frac{1}{\sum_{i=1}^{k} w_i} \sum_{i=1}^{k} w_i accu_i$, where $w_i \in \aleph$ is the weight assigned to class i and $accu_i$ is the accuracy rate computed over class i. If we normalize the weights such that $0 \leq w_i \leq 1$ and $\sum w_i = 1$, we get $cwa = \sum_{i=1}^{k} w_i accu_i$ which simplifies to $cwa = w_i * sensitivity + (1 - w_i) * specificity$ in binary classification.

4.3 Evaluation Strategy

The experimental goal was to measure the relative performance of different approaches to adjusting class distribution. Given the limited amount of data, we adopted 5-fold stratified cross-validation in all the experiments. To evaluate our approach, we ran the five learning algorithms (1) on the original class distribution, then on training data balanced via (2) random subsampling,(3) random oversampling, and (4) different variants of our approach as described in Section 3. All learned models were validated on a test set with the original class distribution. In this way, it was ensured that the validation stage was not influenced by any bias introduced by the various class resampling strategies.

5 Results

Table 1 summarizes performance results on the original skewed class distribution and illustrates clearly the inadequacy of accuracy for this task. For instance, AdaBoost exhibits the highest accuracy of 90% but actually performs more poorly than Naive Bayes in detecting positive cases of nosocomial infections. In fact, Naive Bayes ranks last in terms of accuracy rate due to its poor performance on the majority class (specificity of 0.88, lower than all the others) but attains the highest sensitivity, 12% higher than that of AdaBoost. Accuracy clearly underestimates the merit of recognizing rare positives.

Table 1. Baseline performance (original class distribution: 0.11 pos, 0.89 neg)

Classifier	Sensitivity	Specificity	CWA	Accuracy
IB1	0.19	0.96	0.38	0.88
NaiveBayes	0.57	0.88	0.65	0.85
C4.5	0.28	0.95	0.45	0.88
AdaBoost	0.45	0.95	0.58	0.90
SVM	0.43	0.92	0.55	0.86

We then tested classical methods of random undersampling and oversampling. At each cross-validation cycle, the training set contained 60 positive cases and 486 negative cases. A random sample of 60 negative cases was drawn and used with the 60 available positive cases to train the classifiers. In a separate experiment, positive cases were randomly duplicated until the size of the minority class matched that of the majority class. Table 2 (a) and (b) show performance measures obtained on test data with the original class distribution by classifiers trained on the adjusted class distribution.

Table 2. Random subsampling and oversampling (0.5 pos, 0.5 neg)

(a) Random subsampling					(b) Random oversampling				
Classifier	Sens	Spec	CWA	Accu	Classifier	Sens	Spec	CWA	Accu
IB1	0.01	0.99	0.26	0.88	IB1	0.19	0.96	0.38	0.88
NaiveBayes	0.21	0.96	0.40	0.88	NaiveBayes	0.68	0.83	0.72	0.81
C4.5	0.00	1.00	0.25	0.89	C4.5	0.49	0.87	0.59	0.83
AdaBoost	0.04	1.00	0.28	0.89	AdaBoost	0.73	0.87	0.77	0.85
SVM	0.05	0.99	0.29	0.88	SVM	0.60	0.89	0.67	0.86

The results are contrasted: while random subsampling drastically degraded prediction of positives with respect to the original imbalanced data, random oversampling clearly improved the sensitivity and CWA of all the classifiers except (understandably) IB1. Note that contrary to CWA, accuracy misleadingly decreases with random oversampling.

As explained in Section 3, our approach differs from these random approaches in its principled generation of synthetic samples. In the first variant, we use K-means clustering to subsample the majority class. Results shown in Table 3 (a) support clearly the efficacy of K-means based subsampling. Sensitivity ranges from 0.56 for IB1 to 0.83 and 0.84 for SVM and Adaboost respectively—a visible leap from the 0.19-0.57 interval on the original class distribution and especially from the 0.01-0.21 range attained with random subsampling. More remarkably, specificity did not degrade considerably, so that CWA rates vary between 0.67 and 0.81, definitely better than all previous performance.

Table 3. Oversampling and undersampling based on synthetic examples

(a) K-means subsampling 0.5 pos 0.5 neg					(b) AHC oversampling 0.38 pos 0.62 neg				
Classifier	Sens	Spec	CWA	Accu	Classifier	Sens	Spec	CWA	Accu
IB1	0.56	0.88	0.64	0.84	IB1	0.33	0.91	0.48	0.85
NaiveBayes	0.75	0.78	0.76	0.78	NaiveBayes	0.64	0.85	0.69	0.82
C4.5	0.72	0.67	0.71	0.68	C4.5	0.45	0.87	0.56	0.83
AdaBoost	0.84	0.74	0.81	0.75	AdaBoost	0.65	0.89	0.71	0.86
SVM	0.83	0.74	0.81	0.75	SVM	0.53	0.88	0.62	0.84

We have explained (Section 3) why we chose agglomerative hierarchical clustering to create prototypical instances for oversampling. By combining multilevel clusterings based on single and complete linkage, we were able to compute a total of 234 synthetic instances of the minority class. Added to the 60 original training positives and 486 negatives, they produced a 0.38-0.62 class distribution for training. Results of this operation are shown in Table 3 (b). Here again, sensitivity rates improve significantly over the baseline for all classifiers. However, AHC oversampling improves sensitivity over random oversampling for only 2 out of the 5 classifiers. This can be explained by the fact that in random oversampling positives are as numerous as negatives while they remain outnumbered in 0.38-0.62 AHC distribution.

Finally, we investigated the impact of combining AHC based oversampling and K-means based subsampling. As seen in Table 4, sensitivity and class-weighted accuracy improve over simple AHC oversampling for all classifiers but degrade over K-means subsampling for 4 out of 5 classifiers. For Naive Bayes, however, sensitivity reaches 0.87 and class-weight accuracy 0.84, yielding the maximum performance level recorded over all our experiments.

Table 4. Combined AHC oversampling and K-Means subsampling (0.5 pos 0.5 neg)

Classifier	Sensitivity	Specificity	CWA	Accuracy
IB1	0.49	0.86	0.59	0.82
NaiveBayes	0.87	0.74	0.84	0.75
C4.5	0.68	0.79	0.71	0.78
AdaBoost	0.77	0.85	0.79	0.84
SVM	0.69	0.82	0.73	0.81

6 Conclusion

We analysed the results of a prevalence study of nosocomial infections in order to predict infection risk on the basis of patient records. The major hurdle, typical in medical diagnosis, is the problem of rare positives. We addressed this problem via a novel approach based on the generation of synthetic instances for both oversampling and undersampling. Generation of artificial cases must however meet a hard constraint: the synthetic cases generated must remain within the frontiers of a given class. This constraint is met by the use of prototypes of class subclusters. Results are indeed promising: whereas the sensitivity range of the 5 classifiers was [0.19-0.57] on the original class distribution, it increased to [0.49-0.87] after combined AHC based oversampling and K-means based subsampling. This suggests that both oversampling and undersampling become more effective when performed using synthetic samples instead of the true instances.

Acknowledgements

The authors thank Profs. A. Geissbuhler and D. Pittet (University of Geneva Hospitals) for the opportunity to pursue this study.

References

[1] N. Chawla, K. Bowyer, L. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. In *International Conference on Knowledge-Based Systems*, 2000.

[2] P. Domingos. A general method for making classifiers cost-sensitive. In *Proc. 5th International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[3] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley, 2000.

[4] G. G. French, A. F. Cheng, S. L. Wong, and S. Donnan. Repeated prevalence surveys for monitoring effectiveness of hospital infection control. *Lancet*, 2:1021–23, 1983.

[5] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proc. 13th International Conference on Machine Learning*, 1996.

[6] S. Harbarth, Ch. Ruef, P. Francioli, A. Widmer, D. Pittet, and Swiss-Noso Network. Nosocomial infections in swiss university hospitals: a multi-centre survey and review of the published experience. *Schweiz Med Wochenschr*, 129:1521–28, 1999.

[7] N. Japkowicz. The class imbalance problem: A systematic study. *Intelligent Data Analysis Journal*, 6(5), 2002.

[8] M. V. Joshi, R. C. Agarwal, and V. Kumar. Mining needles in a haystack: Classifying rare classes via two-phase rule induction. In *ACM-SIGMOD*, 2001.

[9] M. Kubat and S. Matwin. Addressing the curse of imbalanced data sets: One-sided sampling. In *Procsȯf the Fourteenth International Conference on Machine Learning*, pages 179–186, 1997.

[10] L. M. Manevitz and M. Youssef. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2, December 2001.

[11] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[12] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.