Artificial Intelligence in Medicine (2006) **37**, 7—18

ELSEVIER

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Learning from imbalanced data in surveillance of nosocomial infection

Gilles Cohen [a,*], Mélanie Hilario [b], Hugo Sax [c], Stéphane Hugonnet [c], Antoine Geissbuhler [a]

[a] *Medical Informatics Service, University Hospital of Geneva, Geneva, Switzerland*
[b] *Artificial Intelligence Laboratory, University of Geneva, Geneva, Switzerland*
[c] *Department of Internal Medicine, University Hospital of Geneva, Geneva, Switzerland*

Received 27 July 2004; received in revised form 8 March 2005; accepted 10 March 2005

KEYWORDS
Nosocomial infection;
Machine learning;
Support vector machines;
Data imbalance

Summary

Objective: 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.
Methods and material: Standard surveillance strategies are time-consuming and cannot be applied hospital-wide; alternative methods are required. In NI detection viewed as a classification task, the main difficulty resides in the significant imbalance between positive or infected (11%) and negative (89%) cases. To remedy class imbalance, we explore two distinct avenues: (1) a new resampling approach in which both oversampling of rare positives and undersampling of the noninfected majority rely on synthetic cases (prototypes) generated via class-specific subclustering, and (2) a support vector algorithm in which asymmetrical margins are tuned to improve recognition of rare positive cases.
Results and conclusion: Experiments have shown both approaches to be effective for the NI detection problem. Our novel resampling strategies perform remarkably better than classical random resampling. However, they are outperformed by asymmetrical soft margin support vector machines which attained a sensitivity rate of 92%, significantly better than the highest sensitivity (87%) obtained via prototype-based resampling.

* Corresponding author. Tel.: +41 22 372 7550; fax: +41 22 320 2927.
E-mail address: Gilles.Cohen@sim.hcuge.ch (G. Cohen).

0933-3657/$ — see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.artmed.2005.03.002

1. Introduction

Surveillance is the cornerstone activity of infection control, whether nosocomial[1] 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 with 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 (NIs). This method is labor-intensive, infeasible at a hospital level, and currently recommended only 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 survey frequency), 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 characteristics of these infections. The efficacy of infection control policies can be easily measured by repeated prevalence surveys [1]. However, whatever the strategy used, surveillance of NI is resource and labor-consuming, as it requires to assemble a wide range of data gathered from multiple sources. This calls for the development of alternative methods that would ultimately allow to constantly monitor infection risk across the hospital, and at a lower cost.

2. Background and motivation

The actual detection of NIs largely rely on manual methods. Infection control practitioners report infection rates using standard method (i.e. guidelines) elaborated by the Centers for Disease Control (CDC) [2]. Several teams have developed tools to assist physicians in detecting NIs, using computerized approaches. These tools typically work by searching clinical databases of microbiology and other data and producing a report that infection control physicians can then use to assess whether or not NI is present. Information technology is increasingly applied to surveillance of nosocomial infections in order to facilitate data collection at the bedside [3], to enhance data quality, or to render surveillance automatic for cost reasons [4]. So far, data mining techniques have only been rarely applied to surveillance [5—9]. None of these applications, however, can directly be compared to our approach to survey all endemic nosocomial infections using a wide range of patient data. Former studies using data mining were limited to outbreak detection, or merely based on microbiology results. Our long-term goal is to identify patients with a high risk of acquiring any kind of nosocomial infection in order to cut down on work load for manual review of patient records. This would, once more, render hospital-wide surveillance possible.

3. Data collection and preparation

The University Hospital of Geneva (HUG) has been performing yearly prevalence studies since 1994 [10]. These surveys are undertaken every year at the same period and last approximately 3 weeks. All patients hospitalized at time of the survey for more than 48 h are assessed for the presence of an active nosocomial infection. Data are extracted from medical records, kardex, X-ray and microbiology reports, and interviews with nurses and physicians in charge of the patient, if necessary. All nosocomial infections active during the 6 days preceding the day of survey are recorded and identified according to modified CDC criteria [2]. Collected variables include administrative information, demographic characteristics, admission diagnosis, comorbidities and severity of illness scores, type of admission, exposure to various risk factors for infection (surgery, intensive care unit stay, invasive devices, antibiotics, antacids, immunosuppressive treatments), clinical and paraclinical information, and data related to infection, when present.

This type of hospital-wide prevalence survey has been favored over prospective surveillance, as it is less time-consuming. However, it still requires considerable resources, as about 800 h are needed for data collection only. Consequently, we cannot afford to perform this surveillance more than once a year. The aim of this pilot study is to apply data

[1] 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 48 h after admission.

mining techniques to data collected in the 2002 prevalence study in order to detect nosocomially infected patients 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 [11], due to mainly erroneous or missing measurements. These values were assumed to be missing at random, as domain experts did not detect any clear correlation between the fact that they were missing and the data (whether values of the incomplete variables themselves or of others). We replaced these missing values with the class-conditional mean for continuous variables and the class-conditional mode for nominal ones. These preprocessing operations are often necessary in such retrospective analyses where data collection has not been engineered specifically for data mining purposes.

4. The imbalanced data 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 problem of imbalanced datasets is particularly crucial in applications where the goal is to maximize recognition of the minority class.[2] The issue of class imbalance has been actively investigated and remains largely open; it is handled in a number of ways [12] such as resampling (either by upsizing the minority class [13] or downsizing the majority class) [14], building cost-sensitive classifiers [15] that assign a higher cost to misclassification of minority class members, and rule-based methods that attempt to learn high confidence rules for the minority class [16].

All solutions to data imbalance that have been proposed to date can be roughly split into two main categories: the first consists in pre-processing the data to reestablish class balance whereas the second involves modifying the learning algorithm itself to cope with imbalanced data. In this paper we investigate solutions under each of these two categories. Section 5 discusses these two approaches; experiments conducted to assess them are described in Section 6 and results are discussed in Section 7.

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

5. Strategies for handling imbalanced data

In this section we present two distinct approaches to the imbalanced data problem. The first is aimed at eliminating or at least attenuating class imbalance *before* the learning process whereas the second adjusts the learning algorithm's bias to allow it to learn *despite* the handicap of imbalanced data. In the first approach, we decompose each class into fine-grained clusters and generate artificial cases in the form of cluster prototypes; these synthetic cases are used to drive the preliminary resampling process. In the second approach, we modify the inductive process itself in order to favor the positive minority and thus boost sensitivity [11]; this is done through the use of asymmetrical soft margins in support vector machines.

5.1. Prototype-based resampling

Resampling approaches appeared as the earliest and remain the most popular methods for coping with imbalanced data. Class rebalancing can be performed in one of two ways. In the undersampling approach, one eliminates instances to downsize the majority class; cases to retain or eliminate are usually drawn at random, but more informed strategies for subsampling have been proposed (see for instance [14]). In the oversampling approach, the minority class is upsized, typically by duplicating randomly selected class members. More recently, attempts have been made to augment the minority class by generating synthetic instances. For instance, Chawla et al. [13] generate synthetic cases from real ones using a technique based on nearest neighbors.

We pursue the resampling strategy by exploring a new way of generating synthetic examples. A selected class is subclustered and the resulting prototypes are reintroduced as synthetic cases. In one variant of the proposed strategy, we use the artificially created examples to downsize the majority class. This approach is novel and may appear 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 key difference is that in the downsizing approach, the synthetic cases are used to *replace* all the original majority class members. 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 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 in acceptable 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.

5.2. Asymmetrical margin support vector machines

In this section, we show how a learning algorithm like support vector machines can be adapted to learn in the presence of imbalanced datasets. To make this paper self-contained, we start with a brief overview of support vector classification before describing how asymmetrical margins provide a solution to the problem of skewed class distributions. The reader familiar with statistical learning theory can go directly to Section 5.2.2.

5.2.1. Overview of support vector classification

Support vector machines [17,18] (SVM) are learning machines based on the *Structural Risk Minimization* (SRM) principle from statistical learning theory. The SRM principle seeks to minimize an upper bound of the generalization error rather than minimizing the training error (Empirical Risk Minimization (ERM)). This approach results in better generalization than conventional techniques generally based on the ERM principle.

Consider a labeled training set $\{x_i, y_i\}$, $i = 1, \ldots, n$, $y_i \in \{-1, +1\}$, $x_i \in \mathcal{R}^d$. For a separable classification task, there exists a separating hyperplane, defined by $(w \cdot x) + b$, with w the weight vector, *b* the bias and where $(\cdot)$ denotes the inner product, which maximizes the *margin* or distance between the hyperplane and the closest data points belonging to the different classes. This optimum separating hyperplane is given by the solution to the problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|w\|^2 \\ &\text{subject to} && y_i(w \cdot x_i + b) \geq 1, \quad i = 1, \ldots, n \end{aligned} \tag{1}$$

where $b/\|w\|$ is the distance between origin and hyperplane. This is a quadratic programming problem (QP), solved by Karush–Kuhn–Tucker theorem. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ be the *n* nonnegative Lagrange multipliers associated with the constraints, the solution to the problem is equivalent to determining the solution of the *Wolfe dual*[19] problem:

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j (x_i . x_j) y_i y_j \\ &\text{subject to} && \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \end{aligned} \tag{2}$$

The solution for *w* is

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3}$$

There is a Lagrange multiplier α_i for each training point and only those training examples that lie close to the decision boundary have nonzero α_i. These vectors are called the *support vectors*. The classifier decision function $f(x)$ is:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i (x \cdot x_i) + b \right) \tag{4}$$

5.2.1.1. Soft margin hyperplanes. While the above method is fine for separable data points, very often noisy data or sampling problems will lead to no linear separation in the feature space. Very often, the data points will be almost linearly separable in the sense that only a few of the members of the data points cause it to be nonlinearly separable. Such data points can be accommodated into the theory with the introduction of slack variables that allow particular vectors to be misclassified. The hyperplane margin is then relaxed by penalizing the training points misclassified by the system. Formally

the optimal hyperplane is defined to be the hyperplane which maximizes the margin and minimizes some functional $\theta(\xi) = \sum_{i=1}^{n} \xi_i^{\sigma}$, where σ is some small positive constant. Usually the values $\sigma = 1$ and 2 are used since it is a QP and for $\sigma = 1$ the corresponding dual does not involve ξ and therefore offers a simple optimization problem. The constraint in (1) now assumes the form

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall_i \xi_i \geq 0 \tag{5}$$

The optimization problem becomes

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i^{\sigma} \\ &\text{subject to} && y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & && \forall_i \xi_i \geq 0. \end{aligned} \tag{6}$$

where ξ_i is a positive slack variable that measures the degree of violation of the constraint. The penalty C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the training error. This is called the soft margin approach. Again, instead of solving directly optimization problem (6) we consider the corresponding dual problem. We will consider the soft margin case for $\sigma = 1$ and 2.

1-Norm soft margin. If we select $\sigma = 1$ the dual problem becomes

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j (x_i \cdot x_j) y_i y_j \\ &\text{subject to} && 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n \\ & && \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{7}$$

The only difference with respect to (2) is that the Lagrange multipliers are upper bounded by C. The KKT conditions imply that nonzero slack variables can only occur for $\alpha_i = C$. For the corresponding points the distance from the hyperplane is less than $1/\|w\|$ as can be seen from the first constraint in (6). For α_i between 0 and C the corresponding points lie on one of the two margin hyperplanes as depicted in Fig. 1 (A).

2-Norm soft margin. If we select $\sigma = 2$the dual problem becomes

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j ((x_i.x_j) + \frac{1}{C}\delta_{i,j}) y_i y_j \\ &\text{subject to} && \alpha_i \geq 0, \quad i = 1, \ldots, n \\ & && \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{8}$$

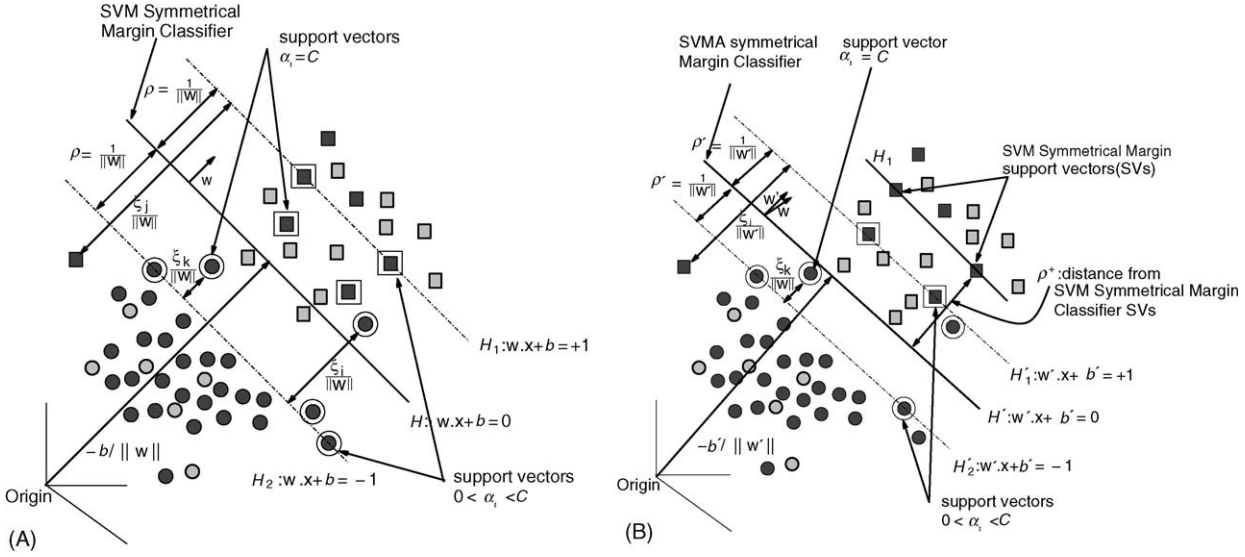

Figure 1 Schematic 2D overview of the asymmetrical soft margin principle on a toy classification problem. Squares represent positive and circles negative examples; dark symbols stand for training and grey for test examples. The left graph (A) shows the decision boundary induced by a symmetrical margin SVM, i.e., with a single penalty factor C for both positive and negative examples. The right graph (B) shows the new boundary obtained by introducing two penalty factors C^+ and C^- respectively for positive and negative examples. Observe that in (B) the decision boundary from (A) has been pushed away from the positive training examples (i.e. H_1 in (A)). This can be viewed as increasing the positive margin while reducing the negative margin of (A). This margin adjustment results in improved generalization performance as shown by the two positive examples (grey squares) which are misclassified in (A) and correctly classified in (B). This approach must not be confused with a naive postprocessing method which consists in moving the decision boundary by adjusting b: $f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i (x \cdot x_i) + b + \Delta b\right)$. In fact we can see that the direction of the decision boundary in (B) is different from the one in (A) ($w \neq w'$, as shown in (B) by the nonnull angle between these vectors).

where $\delta_{i,j}$ is the Kronecker symbol defined to be 1 if $i = j$ and 0 otherwise. The only difference w.r.t. the 1-norm is the addition of $1/C$ to the diagonal of the Gram matrix[3] $G = (x_i \cdot x_j)$.

5.2.1.2. Nonlinear support vector classification. The entire construction can be extended rather naturally to include nonlinear decision boundaries. Given a mapping $\phi : \mathcal{X} \to \mathcal{H}$ each data point x in input space is mapped onto a vector $z = \phi(x)$ in a higher dimensional feature space $\mathcal{H}$. We can then substitute the dot product $(\phi(x) \cdot \phi(x_i))_{\mathcal{H}}$ in feature space with a nonlinear function $K(x, x_i)$, also called a *kernel*. Conditions for a function to be a kernel are expressed in a theorem by Mercer [20,21]. The kernel function behaves like an inner product in $\mathcal{H}$, but can be calculated as a function in $\mathbb{R}^d$. Thus choosing a kernel will implicitly define a mapping ϕ. Most common kernels are polynomial $K(x, z) = ((x \cdot z) + 1)^p$and RBF Gaussian $K(x, z) = \exp(-\|x - z\|/2\sigma^2)$. The final classifier $f(x)$ is then expressed in term of $K(x, x_i)$

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b\right). \tag{9}$$

5.2.2. Asymmetrical margin support vector classification

The above formulation of the SVM is inappropriate in two common situations: in the case of unbalanced distributions, or whenever misclassifications must be penalized more heavily for one class than for the other. Generally in these cases the training set is unrepresentative of the whole dataset and the resulting classifier learned may have poor generalization performance. To illustrate the problem, Fig. 1 shows a toy binary classification problem where the training set is imbalanced. In this case the decision boundary induced by the maximal-margin SVM classifier predicts poorly since it misclassifies some positive (unseen) examples.

In order to adapt the SVM algorithm to these cases [22–24] the basic idea is to introduce different error weights C^+ and C^- for the positive and the negative class respectively, which results in a bias for larger multipliers α_i of the critical class. This induces a decision boundary which is more distant from the smaller class than from the other, resulting in better generalization performance. Let $i_+ = \{i | y_i = -1\}$ and $i_- = \{i | y_i = +1\}$. This transforms (6) into the following optimization problem:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C^- \sum_{i \in i_-} \xi_i^- + C^+ \sum_{i \in i_+} \xi_i^+ \tag{10}$$

$$\begin{aligned} \text{subject to} \quad & (w \cdot x_i + b) \geq 1 - \xi_i^+, \quad i = 1, \ldots, n \\ & (w \cdot x_i + b) \leq -1 + \xi_i^-, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{aligned} \tag{11}$$

For $\sigma = 1$ with the same computations as above we obtain the same formulation as in (7) except that

$$0 \leq \alpha_i \leq C^+ \text{ for } y_i = +1,$$

$$0 \leq \alpha_i \leq C^- \text{ for } y_i = -1$$

For $\sigma = 2$ we obtain the following dual formulation

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \times \left((x_i \cdot x_j) + \mathbb{I}_{i \in i_+} \frac{1}{C^+} \delta_{ij} + \mathbb{I}_{i \in i_-} \frac{1}{C^-} \delta_{ij}\right) \tag{12}$$

where $\mathbb{I}$ is the indicator function. This can be interpreted as a change in the Gram matrix G. Then the balance between sensitivity and specificity can be controlled by adding $1/C^+$ to the elements of the diagonal of G which correspond to examples of the positive class and $1/C^-$ to those corresponding to examples of the negative class. Writing the 2-norm soft margin in a kernel-based version, we get:

$$K(x, z) = \begin{cases} K(x_i, z) + \dfrac{1}{C^-} \delta_{x_i z} & \text{for } y_i = -1 \\ K(x_i, z) + \dfrac{1}{C^+} \delta_{x_i z} & \text{for } y_i = +1 \end{cases} \tag{13}$$

Veropoulos et al. [23] have shown that regularization of kernel matrix diagonal elements produces similar results for both 1-norm and 2-norm. The experiments reported in the next section have thus been restricted to the 1-norm case.

6. Experimental setup

6.1. Learning algorithms

For the preprocessing strategy 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 [25]. Naive Bayes computes the posterior probability of each class given a new case, then assigns the case to the most probable

[3] Given a set of instances $X = \{x_i, y_i\}_{i=1}^n$, the Gram matrix is the matrix of all possible inner-products of pairs from X, $G = (g_{ij}) = (x_i . x_j)$.

class. IB1 is basically a *K*-nearest-neighbors [26] classification algorithm, while Adaboost builds a single-node decision tree iteratively, focusing at each step on previously misclassified cases [27]. Support vector machines have been described more extensively in Section 5.2.1 because of the central role they play in our experiments. Aside from taking part in the study of the impact of preliminary resampling on five learning algorithms, they are used to illustrate how the inductive bias of learning algorithms can be modified to ensure satisfactory performance despite data pathologies such as class imbalance.

6.2. Performance metrics

In classification tasks, the performance of a classifier is commonly quantified in terms of its predictive accuracy, i.e. the fraction of correctly classified test cases. However, highly skewed class distributions can make this metric close to meaningless. 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 [11], is defined as: sensitivity: $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, whereas the accuracy rate on the negative class, also known as specificity [11], is: specificity: $\mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$. Classification accuracy is simply: accuracy: $(\mathrm{TP}+\mathrm{TN})/N$, where $N=\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FP}$ 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

$$gm=\sqrt{\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\times\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}}=\sqrt{\text{sensitivity}\times\text{specificity}} \tag{14}$$

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

$$F_\alpha=\frac{PR}{\alpha R+(1-\alpha)P} \tag{15}$$

where *R* (recall) is no other than sensitivity and *P* (precision) is defined as $P=\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, i.e., the proportion of true positives among all predicted positives. The α 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

$$\mathrm{cwa}=\frac{1}{\sum_{i=1}^{k}w_i}\sum_{i=1}^{k}w_i\,\mathrm{accu}_i \tag{16}$$

where $w_i\in\Re$ 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 $0\le w_i\le 1$ and $\sum w_i=1$, we get $\mathrm{cwa}=\sum_{i=1}^{k}w_i accu_i$ which simplifies to

$$\mathrm{cwa}=w_i\times\text{sensitivity}+(1-w_i)\times\text{specificity} \tag{17}$$

in binary classification.

6.3. ROC curves

In medical diagnosis [28], biometrics and recently machine learning [29], the usual way of assessing a classification method is the receiver operating characteristic (ROC) curve. A ROC curve plots sensitivity versus 1 − specificityfor different thresholds of the classifier output. Based on the ROC curve, one can decide how many false positives (respectively false negatives) one is willing to tolerate and tune the classifier threshold to best suit a certain application. A random assignment of classes to data would result in a ROC curve in the form of a diagonal line from (0, 0) to (1, 1).

6.4. Evaluation strategy

The experimental goal was to measure (1) the relative performance of different approaches to adjusting class distribution and (2) the performance of an SVM asymmetrical soft margin approach to cope

with uneven datasets. Given the limited amount of data, we adopted stratified five-fold cross-validation in all the experiments. To evaluate the resampling 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 4. 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.

To train our SVM classifiers we use a radial basis kernel of the form

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (18)$$

To obtain the optimal values for the hyperparameters σ, C, C^+ and C^- we experimented with different SVM classifiers using a range of values. The performance of the selected SVMs was quantified based on sensitivity, specificity and accuracy.

For our experiments we fixed C^- at 1, and to determine the best C^+ parameter we built several SVM classifiers using different values for C^+.

7. Results

Table 1 summarizes performance results on the original skewed class distribution and illustrates clearly the inadequacy of the accuracy criterion for this task. For instance, AdaBoost exhibits the highest accuracy of 90% but actually performs more poorly than Nave Bayes in detecting positive cases of nosocomial infections. In fact, Nave 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.

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.

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 4, 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

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
Nave Bayes	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

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

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

Table 3 Oversampling and undersampling based on synthetic examples

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

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
Nave Bayes	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

the 0.19—0.57 interval on the original class distribution and especially from the 0.01 to 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.

We have explained (Section 4) 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 two out of the five 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 four out of five classifiers. For Nave Bayes, however, sensitivity reaches 0.87 and class-weighted accuracy 0.84, yielding the maximum performance level recorded over all our resampling preprocessing experiments.

Table 5 summarizes performance results for symmetrical and asymmetrical SVMs on the original skewed class distribution and illustrates clearly the inadequacy of the former for this task. These are the best results from a selection of configurations used for training the classifiers.

In the first experiment based on symmetrical margins, accuracy rates hover constantly around 90% whereas even the best sensitivity remains barely higher than 50% (see Fig. 2). This clearly illustrates the inadequacy of the symmetrical soft margin approach as well as the inappropriateness of accuracy as a performance criterion for the nosocomial application.

To explore the effect of asymmetrical soft margins, we trained SVMs with σ fixed at 0.1 and C^- fixed at 1 for a wide range of C^+ values. Fig. 3 illustrates the effect of upper bound C^+ on the α_i of the positive (i.e. infected) class. For example, as C^+

Table 5 Performance for different SVM configurations with RBF Gaussian kernel and width parameter $\sigma = 0.1$ (original class distribution: 0.11 pos, 0.89 neg)

SVM Classifier	Hyperparameters	Sensitivity	Specificity	CWA	Accuracy
Symmetrical margin	$C = 4$	0.026	1	0.27	0.893
Symmetrical margin	$C = 20$	0.44	0.964	0.58	0.906
Symmetrical margin	$C = 45$	0.506	0.944	0.62	0.896
Asymmetrical margin	$C^+ = 3$	0.586	0.912	0.67	0.876
Asymmetrical margin	$C^+ = 5$	0.76	0.837	0.78	0.828
Asymmetrical margin	$C^+ = 11$	0.88	0.809	0.86	0.816
Asymmetrical margin	$C^+ = 29$	0.92	0.722	0.87	0.744

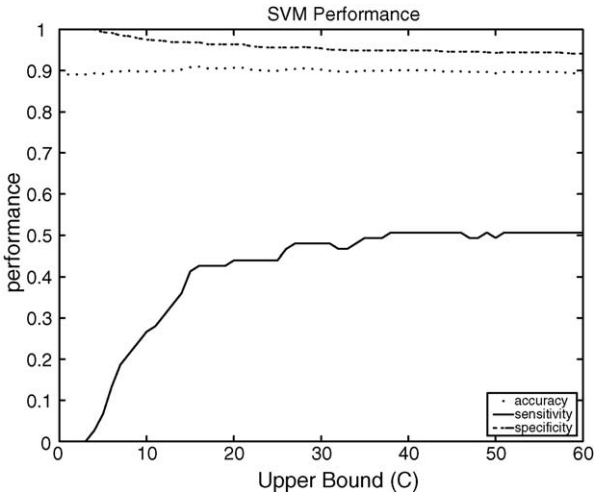

Figure 2 Generalization performance of the symmetrical margin SVM classifier against different C values.

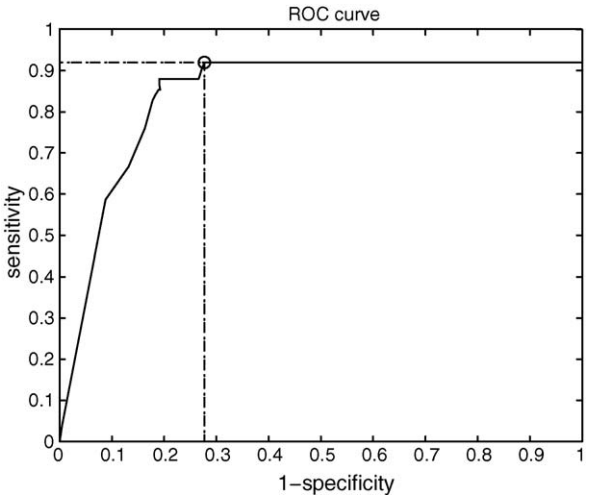

Figure 4 ROC curve for SVM classifiers varying error weight values for the positive class C^+.

increases, the number of false positives is increased but at the detriment of a decrease in the number of false negatives. Sensitivity increases while specificity decreases with increasing values of C^+ (at least up to 29), but as shown clearly in the figure, the gain in sensitivity far outdistances loss of specificity—a fact occluded by the concomitant decrease in accuracy.

A comparison of Table 5 and Tables 2—4 shows that the asymmetrical margin approach leads to better sensitivity than all our proposed resampling methods, provided that the appropriate hyperparameters are used. The best sensitivity rate in these previous experiments was 0.87, attained by Naive Bayes coupled with hybrid over/undersampling via prototype generation. SVMs using asymmetrical margins and a C^+ parameter of 29 perform remarkably better with a sensitivity rate of 0.92.

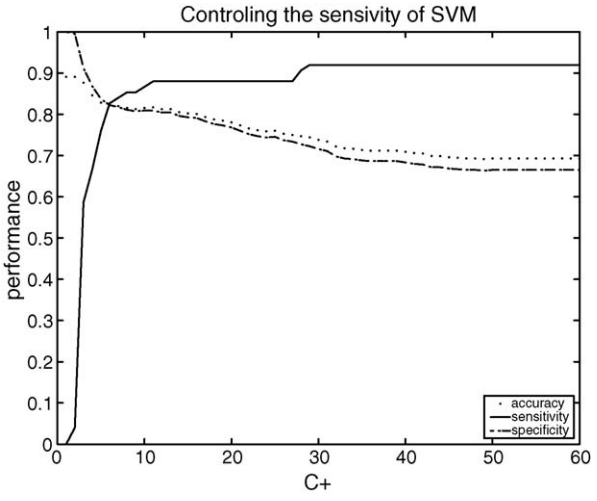

Figure 3 Generalization performance of the asymmetrical margin SVM classifier against different C^+ values.

In order to visualize and assess the behavior of the SVM classifiers throughout a whole range of the output threshold values, the ROC curve shown in Fig. 4 has been produced. This allows experts to easily choose the model best suited to their purpose. The model corresponding to the circled point on the ROC curve (Fig. 4) has been retained by Geneva Hospita experts for the NI classification task. It corresponds to the highest sensitivity 92% reached for a specificity of 72.2% which has been judged completely acceptable.

8. Conclusion

We analyzed 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 two different approaches. The first is 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 encouraging: whereas the sensitivity range of the five 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.

The second approach uses an algorithm proposed by [22,23] where class-dependent regularization parameters are introduced in such a way as to obtain a larger margin on the side of the smaller class (asymmetrical soft margin). The results obtained are indeed promising: whereas the sensitivity range of symmetrical soft margin SVMs was 2.6—50.6%, it increased to 58.6—92% with asymmetrical soft margin SVMs. The maximal sensitivity rate of 92% represents a significant improvement over the best sensitivity of 87% attained by our first novel approach using class balancing with synthetic examples.

These encouraging results make us believe that effective pre-processing as illustrated by our first approach can complement learning algorithm adjustments such as the use of asymmetrical soft margins for SVMs. Despite these results, we intend to prospectively validate the best classification models obtained by our two approaches by performing in parallel a standard prevalence survey and then to improve it in order to classify site-specific infections. We also plan to improve accuracy of SVMs by enhancing the resolution in the support region boundaries via conformal transformation [30,31]. Overall we feel that the combination of asymmetrical soft-margin SVMs with data preprocessing for class skew correction is a promising approach to nosocomial infection detection.

Acknowledgement

The authors are grateful for the dataset provided by the infection control team at the University of Geneva Hospitals.

References

[1] French GG, Cheng AF, Wong SL, Donnan S. Repeated prevalence surveys for monitoring effectiveness of hospital infection control. Lancet 1983;2:1021—3.
[2] Garner JS, Jarvis WR, Emori TG, Horan TC, Huges JM. CDC definitions for nosocomial infections. Am J Infect Control 1988;16(3):128—40.
[3] Kelsey MC, Emmerson AM, Enstone JE. The second national prevalence survey of infection in hospitals: methodology. J Hosp Infect 1995;30(1):7—29.
[4] Trick WE, Zagorski BM, Tokars JI. Computer algorithms to detect bloodstream infections. Emerg Infect Dis 2004; 10(9):1612—20.
[5] Brossette SE, Sprague AP, Hardin JM, Waites KB, Jones WT, Moser SA. Association rules and data mining in hospital infection control and public health surveillance. J Am Med Inform Assoc 1998;5(4):373—91.
[6] Moser SA, Jones WT, Brossette SE. Application of data mining to intensive care unit microbiologic data. Emerg Infect Dis 1999;5(3):454—7.
[7] Brossette SE, Sprague AP, Jones WT, Moser SA. A data mining system for infection control surveillance. Meth Inf Med 2000;39(4):303—10.
[8] Ma L, Tsui FC, Hogan WR, Wagner MM, Ma H. A framework for infection control surveillance using association rules. In: AMIA Annual Symposium. Washington: AMIA; 2003. p. 410—4.
[9] Lamma E, Manservigi M, Mello P, Riguzzi F, Serra R, Storari S. A system for monitoring nosocomial infections. In: Brause Rüdigger Wum, Hanisch Ernstnt, editors. ISMDA, LNCS. Franckfurt, Germany: Springer; 2000. p. 282—92.
[10] Harbarth S, Ruef Ch, Francioli P, Widmer A, Pittet D. Swiss-Noso Network. Nosocomial infections in Swiss university hospitals: a multi-centre survey and review of the published experience. Schweiz Med Wochenschr 1999;129:1521—8.
[11] Perner P. Data mining on multimedia data. Berlin, Heidelberg: Springer Verlag, 2002.
[12] Japkowicz N. The class imbalance problem: a systematic study. Intell Data Anal J 2002;6(5):429—49.
[13] Chawla N, Bowyer K, Hall L, Kegelmeyer WP. SMOTE: Synthetic Minority Over-sampling TEchnique. J Artif Intell Res (JAIR) 2002;16:321—57.
[14] Kubat M, Matwin S. Addressing the curse of imbalanced data sets: one-sided sampling. In: Fisher Douglas Hul, editor. Proceedings of the 14th International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann; 1997. p. 179—86.
[15] Domingos P. Metacost: a general method for making classifiers cost-sensitive. In: Chaudhuri S, Madigan D, editors. Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: ACM Press; 1999. p. 155—64.
[16] Ali K, Manganaris S, Srikant R. Partial classification using association rules. In: Heckerman Davidvd, Mannila Heikkiik, Pregibon Darylrl, editors. Proceedings of the Third International Conference on Knowledge Discovery in Databases and Data Mining. Menlo Park, California: AAAI Press; 1997 . p. 115—8.
[17] Vapnik V. Statistical learning theory. New York: John Wiley & Sons, 1998.
[18] Cortes C, Vapnik V. Support vector networks. Mach Learn 1995;20(3):273—97.
[19] Fletcher R. Practical methods of optimization, 2nd ed, New York: John Wiley & Sons, 1987.
[20] Burges CJC. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 1998;2(2):121—67.
[21] Cristianini N, Taylor JS. An introduction to support vector machines. Cambridge, UK: Cambridge University Press, 2000
[22] Karakoulas G, Shawe-Taylor J. Optimizing classifiers for imbalanced training sets. In: Kearns Michael Jca, Solla Sara Ar , Cohn David Avd, editors. Advances in neural information processing systems (NIPS-99). Cambridge, MA: The MIT Press; 1999. p. 253—9.
[23] Veropoulos K, Cristianini N, Campbell C. Controlling the sensitivity of support vector machines. In: Thomas Dean, editor. Proceedings of the International Joint Conference on Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann; 1999. p. 55—60.
[24] Morik K, Brockhausen P, Joachims T. Combining statistical learning with a knowledge-based approach — a case study in intensive care monitoring. In: Bratko Ivana., Dzeroski Sasos., editors. Proceedings of the Sixteenth International Conference on Machine Learning (ICML99). San Francisco, CA, USA: Morgan Kaufmann; 1999. p. 268—77.
[25] Quinlan JR. C4.5: programs for machine learning. San Mateo, CA: Morgan Kaufmann, 1993.

[26] Duda R, Hart P, Stork D. Pattern classification. New York: John Wiley & Sons, 2000.
[27] Freund Y, Schapire RE. Experiments with a new boosting algorithm. In: Saitta Lorenzarn, editor. Proceedings of the 13th International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann; 1996. p. 148—56.
[28] Centor RM. Signal detectability: the use of ROC curves and their analyses.. Med Decis Making 1991;(11):102—6.
[29] Provost F, Fawcett T, Kohavi R. The case against accuracy estimation for comparing induction algorithms. In: Shavlik Juded., editor. Proceedings of the 15th International Conference on Machine Learning (ICML98). Madison, WI, USA: Morgan Kaufmann; 1998. p. 445—53.
[30] Amari S, Wu S. Improving support vector machine classifiers by modifying kernel functions. Neural Networks 1999; 12(6):783—9.
[31] Wu S, Amari S. Conformal transformation of kernel functions: a data-dependent way to improve support vector machine classifiers. Neural Process Lett 2002;15(1):59—67.