Classification with Imbalance: A Similarity–based Method for Predicting Respiratory Failure

Harsh Shrivastava, Vijay Huddar, Sakyajit Bhattacharya, Vaibhav Rajan
Xerox Research Centre India, Bangalore, India
Email: {harsh.shrivastava, vijay.huddar, sakyajit.bhattacharya, vaibhav.rajan}@xerox.com

***Abstract*—Binary classification based methods are commonly used for designing predictive models in healthcare. A common problem in many healthcare datasets is that of imbalance, where there are far more observations in one class than the other during training. In such conditions, most classifiers do not have good predictive accuracy with respect to the under–represented class. We design a new similarity–based classifier to learn from imbalanced datasets, wherein input features are transformed using similarity with respect to a chosen subset of training points. We empirically demonstrate the superiority of our algorithm over state–of–the–art methods for imbalanced data classification in real and synthetic datasets. We also illustrate the application of our classifier in predicting Acute Respiratory Failure (ARF), a critical complication in Intensive Care Units (ICU), using semi–structured text contained in nursing notes recorded during a patient's ICU stay. Our experiments, on more than 800 patient records show that using our new classifier to learn from text–based features can effectively be used to predict ARF and, potentially, other complications in ICUs.**

I. Introduction

Supervised binary classification based methods are commonly used in clinical prediction models, such as in disease diagnostics [25], or in predicting risk of complications [5]. Most datasets in healthcare are imbalanced – containing significantly more samples from one class (the *majority* class) than the other (the *minority* class, usually the group of interest, e.g. patients at risk of a disease). Standard classifiers do not work well with imbalanced datasets, mainly because they attempt to reduce the overall misclassification error which biases the classifier towards the majority class in imbalanced datasets. As a result the ability of the classifier to identify test samples from the minority class (sensitivity) is poor. Classification on imbalanced datasets is recognized to be an important problem in machine learning. In this paper, we present a new similarity–based classification algorithm that is designed to learn from imbalanced datasets.

Similarity–based learning [7] uses the following two–stage approach for training a classifier: (1) choose a similarity measure and a subset of the training data (called *landmark points*) and compute the similarity between all training datapoints with the landmark points; (2) use any classifier to train on these features. When applied, as is, our experiments show that this technique does not work well on imbalanced datasets and results in the well–known problem of predicting all test samples into the majority class.

Our new algorithm, called **Q classifier** also uses similarities to landmark points as features. However, unlike previous similarity–based algorithms, the parameters of our similarity function are not fixed in advance but are learnt in a manner that accounts for imbalance in the data. We make several design choices – strategies for choosing landmark points and for parameter initialization – and analyze their effects on our algorithm's performance for imbalanced data. We evaluate the performance of our algorithm on a variety of synthetic and real datasets demonstrating improvement over state–of–the–art methods for handling imbalance.

Finally, we illustrate the use of our algorithm in a classification–based model for predicting postoperative Acute Respiratory Failure (ARF) in Intensive Care Units (ICU). ARF is a serious postoperative complication that occurs when the respiratory system fails in oxygenation and/or elimination of carbon dioxide. Like other postoperative complications it worsens patient outcome and mortality and often prolongs hospital stays leading to increased costs. Although it is not very common – it occurs postoperatively in about 3% of all surgical cases – death within 30 days occurs in nearly 26% of the cases [14]. Due to the rarity of the complication, datasets used for learning predictive models for ARF are usually imbalanced. Previous studies [16] have shown that patient information contained in various text sources that are regularly recorded in ICU (such as nursing notes, reports from radiology, biochemistry and other investigations) can provide valuable discriminatory signal for predictive models. The text–based classification model in [16] shows promising results but suffers from the problem of imbalance, leading to low sensitivity in identifying patients at risk for ARF. We show significant improvement in classification performance using Q classifier, outperforming previous methods [16] as well as many other methods for handling imbalance.

II. Related Work

Many techniques have been designed to improve classification for imbalanced datasets. They mainly fall under two broad categories: (1) preprocessing methods that attempt to balance the training data in various ways before using a standard classifier for training and (2) algorithmic techniques that modify the training phase of the classifier taking the imbalance into account. He and Garcia [15] and Sun et al [28] provide excellent reviews. Other studies on class imbalance and its effects on classification can be found in [22], [24].

In algorithmic techniques, such as cost–sensitive learning, costs of misclassification associated with minority and majority classes are adjusted in the training phase of the classification algorithm. For an overview, see [21]. Cost–sensitive versions of many standard classifiers exist, e.g. SVM [29]. The cost–sensitive method closest to ours is **Fast Boxes** [10], where relative costs are implicitly determined during

training. Data is assumed to lie in a grid. The minority class data is clustered and a minimal enclosing box on the grid is determined around each cluster. The feature space is partitioned based on the boxes and finally the boundaries are expanded using an exponential loss function. Goh and Rudin [10] also present another algorithm, Exact Boxes, but since it is not scalable, they recommend the use of Fast Boxes which is empirically found to be nearly as accurate as Exact Boxes.

In preprocessing techniques, the training data is re-sampled, in various ways, to minimize the class imbalance before training the classifier. These include SMOTE [6], where the minority class is inflated by adding synthetic samples that are similar to the data in the feature space, and under–sampling (over-sampling) the majority (minority) class to reduce the imbalance during training. See [1] for a review. These are independent of algorithmic techniques and can be applied in combination with them. For example, Guo and Viktor [13] combine boosting and SMOTE, Kang and Cho [18] combine an ensemble of SVMs with undersampling.

A comprehensive overview of similarity based classification methods can be found in [7]. The theory of similarity based learning is introduced by Balcan et al [4]. Over the years, similarity based classifiers have been explored in various settings and applications; e.g. in nearest neighbour and discriminant based learning [7], in efficient margin based methods [12] and in classifying sequence data [20]. Recent extensions to the theory and the most competitive algorithms are by Kar and Jain [19]. They use a diversity–based heuristic to select pairs of landmark points, one from the majority class and the other from the minority class. Similarities, based on the Gaussian kernel, of each training point are calculated with respect to the pair and the difference between the similarities are taken as features after a ramp–function based transformation. Finally SVM is used as a classifier on these features.

III. Our Classification Algorithm

Consider input $X_{n\times p}$ where n is the number of datapoints, each of dimension p. The j^{th} dimension of the i^{th} observation is denoted by x_{ij} and we skip the second subscript while referring to p–dimensional vectors. Vectors and matrices are shown in bold font. A general similarity–based classifier [7] has the following three steps:

1) Select l landmark points (from the training data).
2) For a choice of similarity function, f, find the similarity of each data point x_i to all the landmark points.
3) Using similarities as features, train a chosen classifier.

The key novelty of our approach, called **Q classifier**, is in the use of a parameterized similarity function, which makes it differ from the general approach above in both steps 2 and 3. The classifier (in step 3) has to be adapted to learn these parameters of the similarity function. We describe these details in the following using a Gaussian kernel as the similarity function and logistic regression as the classifier. Note that our general idea can be applied to other similarity functions and classifiers, *mutatis mutandis*. In addition, we also use new strategies for landmark point selection and parameter initialization, which are empirically demonstrated to be beneficial for imbalanced datasets in section IV. Algorithm 1 shows the entire method and below we describe each step.

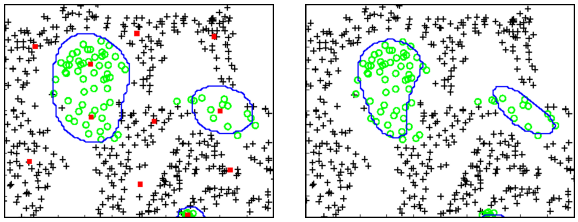

Fig. 1. Classification boundary obtained by Q (left) and SVM (right); minority class points (green circles), majority class points (black crosses) and landmark points (red squares).

Landmark Point Selection. We experiment with several strategies for landmark point selection and find that separately clustering the majority and minority class datapoints and using the cluster centres as landmark points gives us the best results. We use K–Means for clustering, for which the number of landmark points for majority ($K = l_{maj}$) and minority ($K = l_{min}$) classes are provided as input. The total number of landmark points $l = l_{min} + l_{maj}$ is chosen from the range $[p, 5p]$ using cross–validation on the training data, where p is the dimensionality of the dataset, and the ratio $l_{min} : l_{maj}$ is maintained equal to the *imbalance ratio* – the ratio of the total number of datapoints in the majority class to that in the minority class in the training data.

Parameterized Similarity Function. Instead of using a *fixed* similarity function, we find that learning the parameters of the similarity function during training improves classification performance for imbalanced datasets. We choose the Gaussian kernel as our similarity function between training points and landmark points. The similarity function between the i^{th} p-dimensional data point $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{ip}]$ and u^{th} landmark point $L_u = [L_{u1}, L_{u2}, \cdots, L_{up}]$ is given by

$$f_{iu} = e^{-\frac{1}{2}(\mathbf{x}_i - L_u)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}_i - L_u)} \tag{1}$$

where, $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1p}\sigma_1\sigma_p \\ \rho_{21}\sigma_2\sigma_1 & \sigma_2^2 & \cdots & \rho_{2p}\sigma_2\sigma_p \\ \vdots & \vdots & \cdots & \vdots \\ \rho_{p1}\sigma_p\sigma_1 & \rho_{p2}\sigma_p\sigma_2 & \cdots & \sigma_p^2 \end{bmatrix}$ is the corresponding deviance matrix, σ_j is the standard deviation along the j^{th} dimension when the deviance is measured from the landmark points, ρ_{qr} is the correlation coefficient between the q^{th} and the r^{th} feature where the deviance is measured from the landmark point.

The intuition behind this is illustrated in figure 1 (left: Q classifier). We also plot the boundary obtained by SVM (right). Classification boundaries (in blue) are determined by the similarity of the datapoints to the nearest landmark points (in red). By adaptively varying the parameters of the similarity functions (the width of the distance kernel) around each landmark point, our classifier captures the variability in the data. The similarity function in the region of the minority class learns the characteristics of the minority class samples.

Note that the application of the similarity function on the data (through equation 1) can be viewed as a transformation of the original dataset $X_{n\times p}$ into a transformed dataset $F_{n\times l}$ where each dimension now represents similarity with respect to a landmark point. We denote each point after this transformation as f_{iu}, where $i = 1, \ldots, n$ and $u = 1, \ldots, l$.

Classification Model. Since the parameters of the Gaussian kernel are not fixed, we cannot use an off–the–shelf classifier (as in step 3). We adapt logistic regression and use a gradient descent based approach that simultaneously learns the similarity function parameters as well as the regression parameters. Let h denote the logistic function: $h\left(\mathbf{f}_{\mathbf{\Sigma}}, \boldsymbol{\vartheta}\right) = \text{sigmoid}(\theta_0 + \theta_1 \cdot \mathbf{f}_1 + \cdots + \theta_l \cdot \mathbf{f}_l)$, where $\boldsymbol{\vartheta} = [\theta_0, \theta_1, ..., \theta_l]$, θ_0 is the bias term and θ_u is the corresponding regression coefficient for the u^{th} similarity value $\mathbf{f}_u$ and $\mathbf{f}_{\mathbf{\Sigma}} = [\mathbf{f}_1, \ldots, \mathbf{f}_l]$, parameterized by $\mathbf{\Sigma}$.

Parameter Estimation. To estimate the parameters $\mathbf{\Sigma}$ and $\boldsymbol{\vartheta}$ we minimize the following cost function (the total misclassification error): $J(\boldsymbol{\vartheta}, \mathbf{\Sigma}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log\left(h\left(\mathbf{f}_{\mathbf{\Sigma}}, \boldsymbol{\vartheta}\right)\right) + (1-y_i)\log\left(1 - h\left(\mathbf{f}_{\mathbf{\Sigma}}, \boldsymbol{\vartheta}\right)\right)\right]$ where y_i is the class label corresponding to the i-th observation $\mathbf{x}_i$ ($i = 1, 2, ...n$) taking values 0 or 1. Optimization is done using gradient descent. Learning $l+1$ logistic regression coefficients ($\boldsymbol{\vartheta}$), and p^2 terms in $\mathbf{\Sigma}$ is computationally expensive and so we estimate the correlations ρ only once in the beginning and iteratively estimate σ. Thus only $l+p+1$ parameters need to be learnt during gradient descent. This approach of initializing $\mathbf{\Sigma}$ (based on the correlations of input datapoints to the landmark points) during gradient descent is found to be better than random initialization as shown in our experiments. Update rules for the gradient descent are shown in Algorithm 1.

Prediction. After the logistic regression curve is fitted on the training data, a threshold t ($0 < t < 1$) has to be determined to classify test samples. For a test datapoint x_i, if $h\left(\mathbf{f}_{\mathbf{\Sigma}}, \boldsymbol{\vartheta}\right) > t$, we predict class 1, else we predict class 0. The threshold value t can be varied to obtain the complete Receiver Operating Characteristic (ROC) of the method.

Computational Complexity. The time complexity of training is $\mathcal{O}(k(n_t l p^2 + p^3))$ (where k is the number of iterations, n_t is the number of training observations, each of dimension p and l is the number of landmark points) which is dominated by the computation of $\frac{\partial J^{(k)}}{\partial \sigma_j}$ (for each p). The time–complexity of testing a dataset with n observations is $\mathcal{O}(nlp^2)$.

IV. Experiments: Algorithm Performance

Our experiments, in this section, are designed to answer the following questions:

- A What are the effects of various design choices in the algorithm on its performance?
- B How does the performance of our algorithm vary with increasing sample size and data dimensionality?
- C How does the performance of our algorithm compare with that of other state–of–the–art classification algorithms designed for imbalance, on synthetic and real datasets?

Performance Metric: We use the Area under the Convex Hull of ROC Curve (AUH) [26] as our performance metric. AUH is found to be a robust metric for comparing classifiers, is recommended for imbalanced classification problems [26] and used in recent related work (e.g. [10]). For each classifier a parameter can be controlled that changes the relative importance of the positive and negative classes during training (thus achieving a trade-off between sensitivity and specificity). The ROC curve plots True Positive Rate (sensitivity or recall) and False Positive Rate (1 - specificity) for different values of

Algorithm 1 Q Classifier

TRAINING:

Input: Dataset $X_{n\times p}$, number of landmark points l, termination criterion ϵ.

Notation: Indices $j, q, r = 1, ..., p$ over dimensions; $i = 1, \ldots, n$ over observations; $u = 1, \ldots l$ over landmark points and superscript (k) denotes iteration k.

Initialize: Choose l landmark points $(L_1, L_2, ..., L_l)$ using K–Means on minority and majority class datapoints separately. Set:

$$\mu_j = 1/l \sum_{u=1}^{l} L_{uj}, \quad {\sigma_j^{(0)}}^2 = 1/n \sum_{i=1}^{n} \left(x_{ij} - \mu_j\right)^2$$

$$\rho_{qr} = 1/n \sum_{i=1}^{n} \left(x_{iq} - \mu_q\right)\left(x_{ir} - \mu_r\right) / \sigma_q^{(0)} \sigma_r^{(0)}$$

$$\mathbf{\Sigma}^{(0)} = \begin{bmatrix} {\sigma_1^{(0)}}^2 & \rho_{12}\sigma_1^{(0)}\sigma_2^{(0)} & \cdots & \rho_{1p}\sigma_1^{(0)}\sigma_p^{(0)} \\ \rho_{21}\sigma_2^{(0)}\sigma_1^{(0)} & {\sigma_2^{(0)}}^2 & \cdots & \rho_{2p}\sigma_2^{(0)}\sigma_p^{(0)} \\ \vdots & \vdots & \cdots & \vdots \\ \rho_{p1}\sigma_p^{(0)}\sigma_1^{(0)} & \rho_{p2}\sigma_p^{(0)}\sigma_2^{(0)} & \cdots & {\sigma_p^{(0)}}^2 \end{bmatrix}_{p\times p}$$

$$\boldsymbol{\vartheta}^{(0)} = [\theta_0^{(0)}, \theta_1^{(0)}, \ldots, \theta_l^{(0)}] = \left[\frac{1}{l+1}, \ldots, \frac{1}{l+1}\right]_{1\times(l+1)}$$

$$f_{iu}^{(0)} = e^{-\frac{1}{2}\left(\mathbf{x}_i - L_u\right)^\top {\mathbf{\Sigma}^{(0)}}^{-1} \left(\mathbf{x}_i - L_u\right)}$$

repeat

$$\{\text{for the } (k+1)^{st} \text{ iteration}\}$$

$$\theta_u^{(k+1)} = \theta_u^{(k)} - \alpha_{\boldsymbol{\vartheta}} \times \partial J^{(k)} / \partial \theta_u$$
$$\sigma_j^{(k+1)} = \sigma_j^{(k)} - \alpha_\sigma \times \partial J^{(k)} / \partial \sigma_j, \quad \alpha_{\boldsymbol{\vartheta}}, \alpha_\sigma : \text{learning rates}$$

$$\frac{\partial J^{(k)}}{\partial \theta_u} = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i - h\left(\mathbf{f}_i^{(k)}, \boldsymbol{\vartheta}^{(k)}\right)\right) f_{iu}^{(k)}$$

$$\frac{\partial J^{(k)}}{\partial \sigma_j} = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - h\left(\mathbf{f}_i^{(k)}, \boldsymbol{\vartheta}^{(k)}\right)\right)\left[\theta_1^{(k)} f_{i1}^{(k)}, \cdots, \theta_l^{(k)} f_{il}^{(k)}\right] \Delta_{ij}$$

$$\text{where } \Delta_{ij} = \begin{bmatrix} \left(\mathbf{x}_i - L_1\right)^\top \cdot D_j^{(k)} \cdot \left(\mathbf{x}_i - L_1\right) \\ \vdots \\ \left(\mathbf{x}_i - L_l\right)^\top \cdot D_j^{(k)} \cdot \left(\mathbf{x}_i - L_l\right) \end{bmatrix}_{l\times 1}$$

$$D_j^{(k)} = -{\Sigma^{(k)}}^{-1} \frac{\partial \Sigma^{(k)}}{\partial \sigma_j} {\Sigma^{(k)}}^{-1}$$

until $J(\boldsymbol{\vartheta}^{(k)}, \mathbf{\Sigma}^{(k)}) < \epsilon$, where J is the cost function:

$$J(\boldsymbol{\vartheta}^{(k)}, \mathbf{\Sigma}^{(k)}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log\left(h\left(\mathbf{f}_i^{(k)}, \boldsymbol{\vartheta}^{(k)}\right)\right) + (1-y_i)\log\left(1 - h\left(\mathbf{f}_i^{(k)}, \boldsymbol{\vartheta}^{(k)}\right)\right)\right]$$

$$\text{where } h\left(\mathbf{f}_i^{(k)}, \boldsymbol{\vartheta}^{(k)}\right) = \text{sigmoid}(\theta_0^{(k)} + \theta_1^{(k)} f_{i1}^{(k)} + \cdots + \theta_l^{(k)} f_{il}^{(k)})$$
$$\text{and } f_{iu}^{(k)} = e^{-\frac{1}{2}\left(\mathbf{x}_i - L_u\right)^\top {\mathbf{\Sigma}^{(k)}}^{-1} \left(\mathbf{x}_i - L_u\right)}$$

Output: Estimates of similarity function parameters $\mathbf{\Sigma}$ and logistic regression parameters $\boldsymbol{\vartheta}$.

PREDICTION: Use $\mathbf{f}_{\mathbf{\Sigma}}$ to compute features and classify using logistic regression function $h\left(\mathbf{f}_{\mathbf{\Sigma}}, \boldsymbol{\vartheta}\right)$ with chosen threshold t.

this parameter in its range. The AUH is then computed from convex hull of the the points on the ROC curve. A higher AUH indicates better overall performance and the highest value is 1, using normalized units.

Simulated Data: We use simulated datasets with two classes $\mathcal{C}_1$ and $\mathcal{C}_2$, the clusters chosen such that there is dependency within each cluster. Each data point, of dimension $p = 5$, is a product of a sample from a Multivariate Normal (MVN) distribution and another distribution as outlined in table I. Data in cluster $\mathcal{C}_1$ ($\mathcal{C}_2$) is sampled from distribution f_1 (f_2) (I_5 denotes the 5×5 Identity matrix, Δ is a 5×5 matrix with (i, j)-th element $0.9^{|i-j|}$. Unif: Uniform$(0, 1)$ distribution. t(df $= 9$): t-distribution with 9 degrees of freedom). $\mathcal{C}_1$ has 1000 observations and $\mathcal{C}_2$ has 100 observations. Sampling from these, we generate 75 datasets with imbalance ratios 100:1, 50:1, 10:1 (25 datasets each). In each dataset we randomly choose 80% of the data for training and remaining 20% for testing. Results show averages and standard deviations of AUH over 25 simulations.

$$f_1 = \text{MVN}(0.5, I_5/2) \times \text{Unif}(0, 1)$$
$$f_2 = \text{MVN}(2, \Delta) \times \text{t(df} = 9)$$

TABLE I. PARAMETER SETTINGS IN EACH CLUSTER; f_i: DISTRIBUTION IN CLUSTER $\mathcal{C}_i, i = 1, 2$. SEE TEXT FOR DETAILS.

A. Design choices in the algorithm

We study the effects of the following three design choices: (1) strategy for initializing Σ, (2) strategy for selecting landmark points, and (3) effect of using a parameterized similarity function (as opposed to a fixed similarity function). We also study the effect of imbalance in each of the above three cases. In each set of experiments the other choices are retained as described in algorithm 1 and section III.

1) Parameter Initialization: A common approach for initializing Σ during gradient descent is to initialize the matrix randomly. Instead we initialize the matrix in the following manner that captures the correlations of the datapoints with respect to the landmark points.

$$\mu_j = 1/l \sum_{u=1}^{l} L_{uj}, \quad {\sigma_j^{(0)}}^2 = 1/n \sum_{i=1}^{n} (x_{ij} - \mu_j)^2$$
$$\rho_{qr} = 1/n \sum_{i=1}^{n} (x_{iq} - \mu_q)(x_{ir} - \mu_r) / \sigma_q^{(0)} \sigma_r^{(0)}$$

In figure 2, for all three imbalance ratios, we see the improvement this initialization results in, compared to random initialization. For the latter, we use 10 random starts and average the results. The AUH achieved with our initialization is 0.9. Random initialisation, on the other hand, is unstable. Sometimes we see that cost function increases and the algorithm diverges, resulting in decrease in performance. Overall, both its sensitivity and specificity are lesser by 10% yielding an AUH of 0.7. The improvement due to our initialization strategy remains the same at all three imbalance ratios.

2) Choice of landmark points: We study the performance of our algorithm with the following four strategies of choosing landmark points. (1) Randomly choose points in the feature space. (2) Randomly choose some of the input data points as landmark points. (3) Cluster the input datapoint and select the cluster centres as landmark points. (4) Train an SVM on the training data and use the support vectors as landmark points.

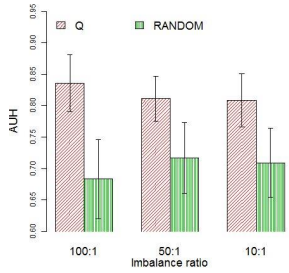

Fig. 2. AUH with our initialization (Q), and random initialisation (RANDOM) at three different imbalance ratios.

Figure 3 shows the performance of the different strategies. We observe that at all three imbalance settings, the performance of our strategy remains superior to the rest. Previous studies on similarity–based learning [19] recommend the "diversification" of landmark points. Clustering seems to achieve the diversification needed in our algorithm and at the same time improves the classification further, by selecting landmark points useful to both classes. Using K–Means clustering gives the best results: the average AUH is more than 0.8 at all settings. Other clustering algorithms can also be used; empirically we did not find improvements in classification performance on using DBSCAN or other variants of K–Means.

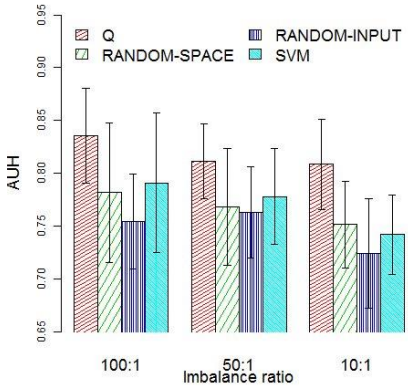

Fig. 3. AUH achieved with different landmark points selection strategies, at three different imbalance ratios.

3) Effect of a Parameterized Similarity Function: In a general similarity–based classifier, similarity function values with respect to landmark points are computed (just once) and used as features with any off–the–shelf classifier. We compare our algorithm, where the similarity function parameters are not fixed but learnt during training, with this standard approach. For all the classifiers we use the same landmark points (selected using K–Means) and the same similarity function (Gaussian kernel with Σ initialized as described in our algorithm 1). Note that Σ changes during training in our algorithm whereas it is computed just once to obtain similarity–based features in other algorithms. We use five different classifiers in the similarity–based framework: SVM-RBF, SVM-LINEAR, SVM-POLY, SVM-SIGMOID (denoting SVMs with four different kernels) and Logisitic Regression (LR). These classifiers are applied after obtaining similarity–based features using the same Gaussian kernel.

Figure 4 shows the performance of all the classifiers, av-

eraged over all simulations at three different imbalance ratios. We observe that all the SVM–based methods are significantly less accurate. LR is closest to our algorithm wherein all steps are identical, except that of training the similarity function parameters. We note that its performance is comparable to our algorithm – the initialization and landmark point selection playing a role in obtaining similarity based features – but the average AUH remains lower than our method at all settings. This suggests that modifying the similarity function parameters in the manner described in our algorithm has a significant effect on the classifier's ability to address imbalance.

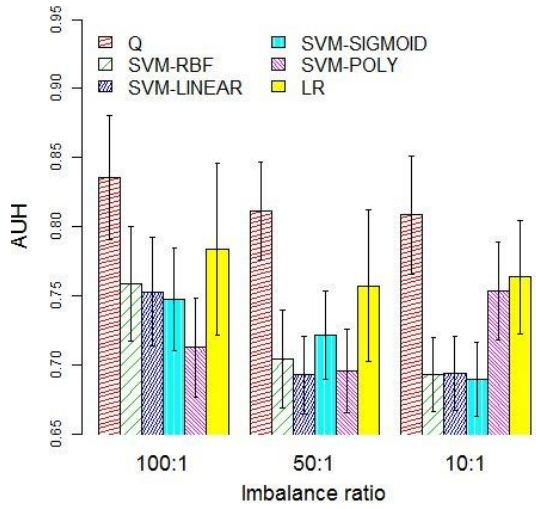

Fig. 4. AUH achieved by Q, which learns similarity function parameters during training and other classifiers where similarity function is computed only once before training.

The previous three experiments show that each of our design choices – parameter initialization, landmark points selection strategy and a parameterized similarity function – in the algorithm plays an important role in the performance of our algorithm.

B. Varying sample size and dimension

To study the effect of sample size n and dimensionality p, we simulate datasets with five different choices of p: $(2, 5, 10, 20, 50)$ and five different choices of n: (440,1100, 2200,3300,4400). For each choice of n, there are two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ as defined in table I where $\mathcal{C}_1$ is the majority class and $\mathcal{C}_2$ is the minority class, the imbalance ratio fixed at 10:1.

Figure 5 shows the performance of our algorithm at different choices of n and p. We observe that the AUH remains close to 0.8 at all settings except at $p = 50$. This is due to "the curse of dimensionality" – at higher dimensions the algorithm requires much larger number of observations to learn well.

C. Performance on Benchmark Datasets

Datasets: We analyze the performance of our algorithm and other baselines on benchmark datasets from various public repositories. Datasets corner, castle3D, corner3D, diamond3D, flooded3D are simulated data that are publicly available at [9]. Datasets Skin Segmentation, Bank Note, User Knowledge,

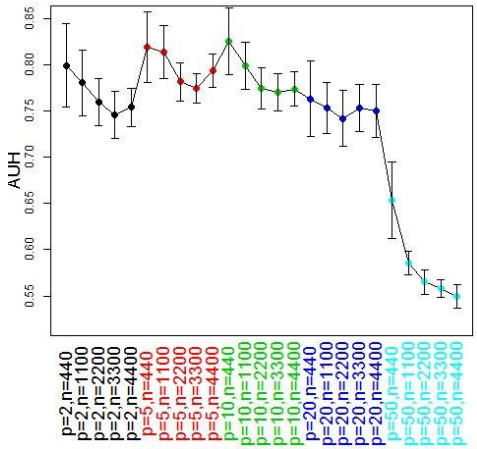

Fig. 5. AUH achieved by Q at different choices of sample size (n) and data dimensionality (p), averaged over 25 simulations.

Glass, Seeds, Leaf and Ionosphere are from the UCI repository [3]. The remaining datasets are from the KEEL repository [2]. Total number of observations, dimensions and imbalance ratio for each dataset are shown in table II.

Baseline Methods: Among methods designed for imbalance, we consider **Fast Boxes** [10] and **cost–sensitive** versions of classifiers: Logistic Regression (**LR**), Decision Trees (**CART**), Random Forest (**RF**), **SVM** (with RBF kernel), **Adaboost** (with decision trees). The imbalance weighting parameter for each of these algorithms are set to values $[0.1, 0.2, \ldots, 1]$ and the best performer is chosen. These results are reproduced from [10]. Since our method is in the family of similarity–based methods, we also compare our method with state–of–the–art similarity–based methods (although these are not designed to address imbalance explicitly). We use the method of Kar and Jain [19], denoted by **PJ**, as the main baseline. In addition, we use two other similarity–based methods. Using landmark points selected by K–Means (as in our method), we use Manhattan kernel and Sigmoid kernel to compute similarities (used as features) and then use SVM to train the classifiers. We denote these methods by **SIMm** and **SIMs** respectively.

Results: Classification results for our algorithm and baselines, averaged over 10–fold cross–validation (with the standard deviations), are presented in Table II. We observe that our algorithm outperforms all the baselines in 18 out of 25 datasets and is close to the best performing baseline in 5 of the remaining datasets.

V. Respiratory Failure Prediction

Postoperative Acute Respiratory Failure (ARF) is commonly defined as the inability to be extubated 48 hours after surgery [8]. Predictive models for postoperative ARF have been designed based on preoperative clinical and demographic factors by several authors [14], [17], [23]. Here we follow the approach in [16] that uses text data to generate the features for classification. Using labelled historical data from two classes A (those who develop ARF during their ICU stay) and B (those who do not), a classifier is trained that can distinguish between the two classes. The classifier can be used to predict the class

Dataset	n, p (imbalance)	LR	SVM	CART	Adaboost	RF	Fast Boxes	PJ	SIMm	SIMs	Q
corner	10000,2	0.9871	0.9948	0.9488	0.6984	0.6828	0.9891	0.733	0.8678	0.995	**0.9952**
	(99:1)	(0.0129)	(0.0005)	(0.2717)	(0.0449)	(0.0265)	(0.0001)	(0.2334)	(0.0900)	(0.0005)	**(0.0004)**
castle3D	545, 3	0.5449	1	0.9532	0.9272	0.9455	1	0.9956	0.9923	0.6412	**1**
	(7.2:1)	(0.0324)	(0)	(0.0347)	(0.0499)	(0.0563)	**(0)**	(0.0036)	(0.0073)	(0.0175)	**(0)**
corner3D	1000, 3	0.8448	0.9225	0.8481	0.6245	0.5657	**0.9736**	0.9238	0.7288	0.9499	0.9726
	(28.4:1)	(0.0316)	(0.0463)	(0.0504)	(0.03927)	(0.0309)	**(0.0091)**	(0.0216)	(0.1103)	(0.0069)	(0.0051)
diamond3D	1000, 3	0.5449	0.7962	0.7372	0.5492	0.5957	0.9516	0.9661	0.7095	0.7513	**0.9876**
	(33.5:1)	(0.0324)	(0.0917)	(0.0347)	(0.0499)	(0.0309)	(0.0119)	(0.0202)	(0.0337)	(0.0331)	**(0.0051)**
square3D	1000, 3	0.5	0.9626	0.9106	0.8703	0.879	0.9578	0.9611	0.579	0.71	**0.9749**
	(7:1)	(0)	(0.0156)	(0.0306)	(0.01451)	(0.0234)	(0.009)	(0.0164)	(0.0209)	(0.0238)	**(0.0096)**
flooded3D	1000, 3	0.5	0.7912	0.7724	0.5471	0.5489	0.9233	0.9258	0.6205	0.7133	**0.9703**
	(26.8:1)	(0)	(0.0781)	(0.0902)	(0.0329)	(0.044)	(0.0307)	(0.0297)	(0.0310)	(0.0240)	**(0.0076)**
abalone19	4174, 9	0.5188	0.5	0.5382	0.5	0.5	0.6882	0.7684	0.7728	0.8173	**0.8396**
	(129.4:1)	(0.0182)	(0)	(0.0261)	(0)	(0)	(0.0583)	(0.0634)	(0.0735)	(0.0754)	**(0.0113)**
yeast6	1484, 8	0.8503	0.8649	0.7995	0.7126	0.7277	0.8609	0.8433	0.8008	0.8172	**0.9322**
	(41.4:1)	(0.0341)	(0.0246)	(0.0624)	(0.0536)	(0.0581)	(0.0585)	(0.1110)	(0.06722)	(0.07541)	**(0.0668)**
yeast5	1484, 8	0.9499	0.9229	0.9197	0.8305	0.8061	0.9767	0.9611	0.7357	0.9573	**0.9771**
	(32.7:1)	(0.0479)	(0.0339)	(0.0575)	(0.0859)	(0.0616)	(0.0092)	(0.0242)	(0.0522)	(0.023)	**(0.0197)**
yeast4	1484, 8	0.8001	0.7836	0.7595	0.6131	0.5922	0.8794	0.9039	0.8166	0.8525	**0.9139**
	(28.1:1)	(0.0309)	(0.048)	(0.041)	(0.0326)	(0.0326)	(0.0274)	(0.0456)	(0.0808)	(0.0904)	**(0.0612)**
abalone918	731, 9	0.8849	0.678	0.7427	0.6117	0.558	0.7171	**0.9237**	0.8165	0.8277	0.893
	(16.4:1)	(0.027)	(0.0391)	(0.0517)	(0.0456)	(0.03213)	(0.0603)	**(0.0585)**	(0.08294)	(0.09783)	(0.0782)
ecoli4	336, 7	0.8926	0.9176	0.8809	0.7965	0.8494	0.9202	0.9714	0.9653	0.8976	**0.9847**
	(15.8:1)	(0.0615)	(0.0424)	(0.0593)	(0.0775)	(0.0775)	(0.0622)	(0.0346)	(0.0638)	(0.1005)	**(0.0261)**
yeast17	459, 7	0.7534	0.6905	0.7481	0.5382	0.5529	0.7033	0.8578	0.7746	0.8134	**0.8647**
	(14.3:1)	(0.0611)	(0.0386)	(0.0713)	(0.0225)	(0.0359)	(0.0547)	(0.0766)	(0.0661)	(0.0732)	**(0.0854)**
haberman	306, 3	0.6589	0.6898	0.6699	0.6004	0.613	0.529	**0.771**	0.6846	0.7088	0.7353
	(2.8:1)	(0.1713)	(0.0427)	(0.0276)	(0.0323)	(0.0318)	(0.0265)	**(0.0865)**	(0.0309)	(0.0492)	(0.0663)
yeast1	1484, 8	0.7836	**0.7991**	0.7641	0.6859	0.613	0.5903	0.7908	0.6502	0.6459	0.7952
	(2.5:1)	(0.0184)	**(0.015)**	(0.0133)	(0.0219)	(0.0318)	(0.0286)	(0.0512)	(0.0555)	(0.0583)	(0.0530)
iris0	150, 4	**1**	0.998	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	(2:1)	**(0)**	(0.0063)	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**
wisconsin	683, 9	0.9746	0.9735	0.9741	0.9611	0.9672	0.8054	0.9905	**0.9973**	0.9972	0.9837
	(1.9:1)	(0.0093)	(0.0073)	(0.0075)	(0.0124)	(0.0072)	(0.1393)	(0.0112)	**(0.0027)**	(0.0027)	(0.0164)
ecoli01	220, 7	0.9728	0.985	0.984	0.9828	0.9855	0.9433	0.9846	0.9979	0.9673	**0.9985**
	(1.8:1)	(0.014)	(0.0091)	(0.0105)	(0.0063)	(0.0097)	(0.03)	(0.0271)	(0.0020)	(0.0509)	**(0.0032)**
Skin Segmentation	245057, 3	0.9656	0.9932	0.9514	0.9628	0.9914	0	0.9669	0.9311	0.7867	**0.9992**
	(3.8:1)	(0.0243)	(0.0024)	(0.0803)	(0.0346)	(0.0166)	(0)	(0.0286)	(0.0597)	(0.0545)	**(0)**
Bank Note	862, 5	0.945	0.9997	0.9678	0.9992	0.9996	0.9423	0.9992	0.8617	0.9473	**1**
	(7.6:1)	(0.0140)	(0.0030)	(0.0136)	(0.0003)	(0.0005)	(0.0277)	(0.0014)	(0.0354)	(0.0138)	**(0)**
User Knowledge	403, 5	0.987395	0.9990	0.955882	0.984163	0.997253	0.9978	0.9980	0.877828	0.650291	**0.9998**
	(7.1:1)	0	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	**(0)**
Glass	214, 9	0.9817	0.9535	0.9064	0.9540	0.9789	0.9400	0.9378	0.9767	0.9546	**0.988649**
	(7.4:1)	(0.0362)	(0.0722)	(0.1076)	(0.0827)	(0.0394)	(0.0872)	(0.0580)	(0.0200)	(0.0604)	**(0.0139)**
Seeds	160, 7	0.9598	**0.9902**	0.7937	0.9803	0.9715	0.8687	0.9589	0.8848	0.9429	0.972321
	(7:1)	(0.0322)	**(0.0139)**	(0.1545)	(0.0242)	(0.0383)	(0.0407)	(0.0563)	(0.1223)	(0.0974)	(0.0339)
Leaf	340, 15	0.9492	0.9892	0.7184	0.9646	0.9523	0.7861	0.9100	0.9461	0.7885	**0.99231**
	(27.3:1)	(0.0521)	(0.0089)	(0.1587)	(0.0353)	(0.0572)	(0.0572)	(0.0748)	(0.0466)	(0.0583)	**(0.0153)**
Ionosphere	260, 16	0.5476	0.84	0.8863	0.9365	0.9577	**0.9806**	0.8470	0.8552	0.7352	0.9619
	(6.4:1)	(0.0656)	(0.0738)	(0.0576)	(0.0609)	(0.0574)	**(0.0289)**	(0.0369)	(0.1148)	(0.0567)	(0.0600)

TABLE II. CLASSIFIER PERFORMANCE: AVERAGE AUH (STANDARD DEVIATION BELOW) OVER 10–FOLD CROSS VALIDATION. THE SECOND COLUMN SHOWS NUMBER OF OBSERVATIONS (n), DIMENSIONS (p) AND IMBALANCE RATIO (BELOW IN PARENTHESES). BEST RESULTS IN BOLD.

label of a new patient based on his/her data. If the predicted label is A, the patient is considered to be at risk for ARF.

The features used to train the classifier are derived from nursing notes recorded for each patient. These notes describing the patient's condition are periodically (approximately once in 3–4 hours) recorded by attending clinical staff in ICUs. For ARF patients we consider only the notes until the diagnosis of ARF is mentioned, for both training and prediction. Note that all discharge summaries are excluded from our analysis since they are written at the end of the patient's stay and cannot be used in a real-time prediction system within the ICU. This also makes the problem harder since discharge summaries contain comprehensive information of patients' past and current medical history which nursing notes lack. Discharge summaries are well–formed documents and their grammatical structure could be used for feature generation. In comparison nursing notes (see figure 6 for an example) are informally written and contain non–standard and inconsistently used abbreviations.

A. Data

The source of our data is MIMIC II [27], a publicly available database, part of Physionet [11], containing clinical data of more than 2300 patients in Critical Care. We restrict our study to surgery patients in the database for whom clinical notes are present. From this set, we identify patients with postoperative respiratory failure through ICD9 code 518.5. Our dataset contains 806 patients' data out of which 122 patients were diagnosed with ARF and 684 patients were not diagnosed with ARF.

B. Text Preprocessing and Feature Extraction

We follow the steps of preprocessing and feature generation from such nursing notes that were found to be effective in [16]. The key idea lies in the observation that the data is not completely unstructured but is structured into various headings such as "NEURO", "PULM" etc. See figure 6 for an example. Since these headings are not consistent (for example, "CARDIO" is also written as "CV" and "CARD" in some

```
7a-7p
CV:  Afib, occassional PVCs.  Pt has brief periods (3-6 seconds) that HR drops to
paced beats.  Afebrile.  Weaning levo as BP toleates.  Milrinone decreased, SVO2 (
milrinone now and recheck SVO2 and CI later.  Only aline is femoral.

PULM:  4L/NC, good sats.  Coughs, raises thick tan or clear sputum multiple times
airleak.

NEURO:  Alert, oriented.  Denies pain.  Wife in to visit.  Turned side to side in

GU:  Foley, one time small lasix dose given.  Did not start on scheduled lasix be

GI:  Active bowel sounds, eating 100% of meals. No BM.

ENDO:  Treating BG with RISS.

PLAN:  Recheck SVO2 and CI in a few hours, wean levo to off, pulmonary toilet.
```

Fig. 6. Sample de–identified nursing notes from an ICU

notes), a synonyms dictionary is created and synonymous words are replaced by consistent headings. The significance of the same word differs when it is under different headings and so we extract features for each heading separately and assign an importance value to each word based on its frequency of occurrence in the training data as described below.

We denote by *text observation* all the text data for a single patient (until the diagnosis of ARF, for ARF patients) concatenated together. Words within the same heading are processed together for each text observation. Stemming, stop word removal and punctuation removal are performed to obtain a list of stem words under each heading (for each text observation). Let $n_w(C, H)$ be the number of text observations from class C wherein the word w occurs under heading H. The importance of a word is computed as $I_w(H) = n_w(A, H) - n_w(B, H)$ for classes A and B. Thus words that are more frequent in class A are positive and those for class B are negative and the importance value is an approximate measure of the word's discriminatory power. For each heading H, we sort the words with respect to their importance values $I_w(H)$, select the top and bottom 5% (thus selecting from both the most negative and most positive values), and discard the rest. Within each heading, each of these words forms a feature and the number of occurrences of the word within a text observation is the feature value. A patient's data consists of a feature vector containing all the feature values (for all the headings). With these preprocessing steps we obtain 6417 features, from which we retain the ten most discriminatory principal components.

C. Experimental Results

Figure 7 (left) shows the AUH achieved by our Q classifier and the baseline algorithms, averages and standard deviation shown over 5–fold cross validation. Q achieves the highest average AUH of 0.81. Previous work on ARF prediction [16] had used classifiers LR, RF and SVM, all of which are outperformed by Q classifier. Similarity–based approaches PJ, SIMm and SIMs also do not perform well: Q achieves higher AUH. We also compare Q classifier with methods designed to address the class imbalance of 6:1 in the dataset. Q outperforms Fast Boxes and other cost–sensitive methods (results of other classifiers with lesser AUH than Fast Boxes are not shown here due to space constraint; CART and Adaboost shown in figure 8). All these baselines are used in the same manner as described in section IV-C. In 4 out of the 5 folds, the AUH achieved by Q is higher than all other classifiers; only in the fifth fold, Fast Boxes achieves marginally better AUH (0.77 compared to Q's 0.76) and the performance of all other classifiers is much lower.

We also use a sampling–based preprocessing technique, SMOTE, that is commonly used (before training a classifier) for imbalanced datasets in combination with the classifiers. The performance of all the classifiers, with and without SMOTE, remains comparable and Q + SMOTE achieves higher average AUH than all the other classifiers as seen in figure 7 (right).

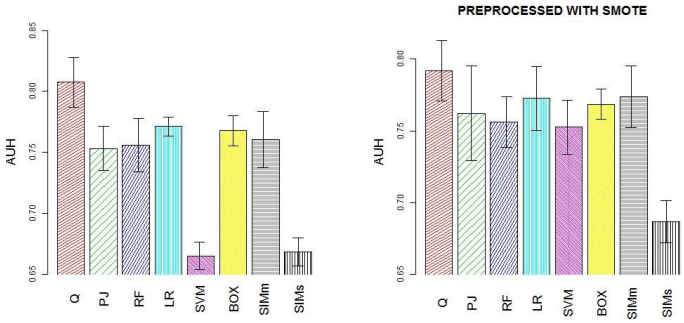

Fig. 7. Classification performance (AUH) on ARF dataset averaged over 5–fold cross validation. Left: No preprocessing with SMOTE, Right: Training data preprocessed with SMOTE.

Varying Imbalance Ratio

We then artificially introduce further imbalance in the dataset by removing samples from the minority class and study the performance of all the classifiers, both with and without preprocessing using SMOTE. Figure 8 shows the AUH of all the classifiers as the imbalance increases from 6:1 to 30:1. The performance of Q classifier remains between 0.8 and 0.85 in all the settings whereas for all other methods, we see a decline in performance.

VI. Conclusions

We present a new similarity–based learning algorithm, called Q, for imbalanced data classification. In our experiments we analyze various design choices in the algorithm and select strategies for landmark point selection and initialization that are effective for imbalanced data. Empirical results show that our algorithm outperforms other similarity–based methods as well as state–of–the–art cost–sensitive methods like Fast Boxes designed for imbalanced datasets on a wide variety of benchmark datasets. To our knowledge this is the first algorithm in the similarity based learning framework that is designed for class imbalance. Two areas where our algorithm can be improved are in handling very high–dimensional and categorical data. Theoretical aspects such as generalization bounds in settings where the parameters are learnt during training would also be interesting to study.

Extensive applications of such a classifier can be found in clinical datasets where imbalance is common – more often than not, subjects with the disease being studied form the minority class and data of healthy subjects are available in plenty (the majority class). We show an application in predicting postoperative Acute Respiratory Failure in ICU patients where a classification–based approach is used with features generated from nursing notes that are periodically recorded in ICUs. The predictive performance of Q classifier surpasses that of many other competitive methods designed for imbalance as well as other similarity–based classifiers.

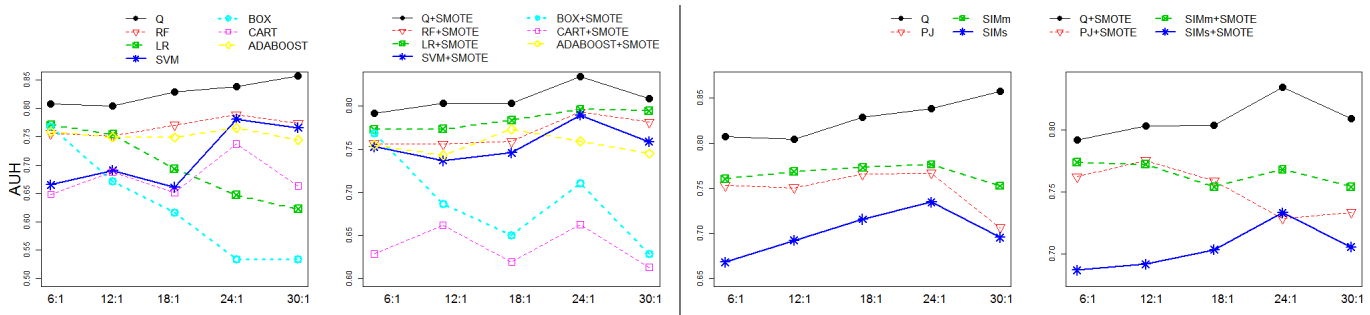

Fig. 8. Classifier Performance (AUH) on ARF dataset with varying imbalance ratios. Left: Comparison with cost–sensitive classifiers, without and with preprocessing using SMOTE; Right: Comparison with similarity–based classifiers, without and with preprocessing with SMOTE.

REFERENCES

[1] Naoki Abe. Sampling approaches to learning from imbalanced datasets: active learning, cost sensitive learning and beyond. In *Proc. of ICML Workshop: Learning from Imbalanced Data Sets*, volume 22, 2003.

[2] J Alcalá, A Fernández, J Luengo, J Derrac, S García, L Sánchez, and F Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2010.

[3] K. Bache and M. Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.

[4] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1-2):89–112, 2008.

[5] Sakyajit Bhattacharya, Vaibhav Rajan, and Vijay Huddar. A novel classification method for predicting acute hypotensive episodes in critical care. In *Proc. 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 43–52. ACM, 2014.

[6] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 853–867. Springer, 2005.

[7] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and Algorithms. *The Journal of Machine Learning Research*, 10:747–776, 2009.

[8] E Stanley Crawford, Lars G Svensson, Kenneth R Hess, Salwa S Shenaq, Joseph S Coselli, Hazim J Safi, Prita K Mohindra, and Victor Rivera. A prospective randomized study of cerebrospinal fluid drainage to prevent paraplegia after high-risk surgery on the thoracoabdominal aorta. *Journal of vascular surgery*, 13(1):36–46, 1991.

[9] S. T. Goh and C. Rudin. http://web.mit.edu/stgoh/www/imbalanceddatafolder/, 2014.

[10] Siong Thye Goh and Cynthia Rudin. Box drawings for learning with imbalanced data. In *Proc. 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342. ACM, 2014.

[11] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13).

[12] Thore Graepel, Ralf Herbrich, Bernhard Scholkopf, Alex Smola, Peter Bartlett, K-R Muller, Klaus Obermayer, and Robert Williamson. Classification on proximity data with LP-machines. In *Proc. International Conference on Artificial Neural Networks*, pages 304–309, 1999.

[13] Hongyu Guo and Herna L. Viktor. Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. *Bibliometrics*, 6(1):30–39, 2004.

[14] Himani Gupta, Prateek K Gupta, Xiang Fang, Weldon J Miller, Samuel Cemaj, R Armour Forse, and Lee E Morrow. Development and validation of a risk calculator predicting postoperative respiratory failure risk calculator predicting respiratory failure. *CHEST Journal*, 140(5):1207–1215, 2011.

[15] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284, 2009.

[16] Vijay Huddar, Vaibhav Rajan, Sakyajit Bhattacharya, and Shourya Roy. Predicting postoperative acute respiratory failure in critical care using nursing notes and physiological signals. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 2702–2705. IEEE, 2014.

[17] Robert Johnson, Ahsan Arozullah, Neumayer Leigh, William G. Henderson, Patrick Hosokawa, and Shukri F. Khuri. Multivariable predictors of postoperative respiratory failure after general and vascular surgery: Results from the patient safety in surgery study. *Journal of the American College of Surgeons*, 204(6):1188–1198, 2007.

[18] P. Kang and S. Cho. EUS SVMs: Ensemble of under–sampled SVMs for data imbalance problems. In *Neural Information Processing Systems (NIPS) , Lecture Notes in Computer Science*, volume 4232, pages 837–846. 2006.

[19] Purushottam Kar and Prateek Jain. Similarity-based learning via data driven embeddings. In *Advances in neural information processing systems*, pages 1998–2006, 2011.

[20] Li Liao and William Stafford Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of computational biology*, 10(6):857–868, 2003.

[21] C.X. Ling and V.S. Sheng. Cost–sensitive learning and the class imbalance problem. In *Sammut C (Ed) Encyclopedia of machine learning*. Springer, 2008.

[22] Xu-Ying Liu and Zhi-Hua Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Sixth IEEE International Conference on Data Mining (ICDM)*, pages 970–974, 2006.

[23] Kazuyo Nakahara, Kiyoshi Ohno, Junpei Hashimoto, shinichiro Miyoshi, Hajime Maeda, Akihide Matsumura, Takatoshi Mizuta, Akinori Akashi, katuhiro Nakagawa, and Yasunaru Kawashima. Prediction of postoperative respiratory failure in patients undergoing lung resection for lung cancer. *The Annals of Thoracic Surgery*, 46(5):549–552, 1988.

[24] Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. Class imbalances versus class overlapping: an analysis of a learning system behavior. In *MICAI 2004: Advances in Artificial Intelligence*, pages 312–321. Springer, 2004.

[25] Daryl Pregibon. Logistic regression diagnostics. *The Annals of Statistics*, pages 705–724, 1981.

[26] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine learning*, 42(3):203–231, 2001.

[27] Mohammed Saeed, C Lieu, G Raber, and RG Mark. MIMIC II: a massive temporal ICU patient database to support research in intelligent patient monitoring. In *Computers in Cardiology, 2002*, pages 641–644, 2002.

[28] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719, 2009.

[29] K. Veropoulos, C. campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60, 1999.