Data Balancing using SVM as Pre-Processor

M.A.H. Farquad*, G. Latha*, Balaji Naik* and R. Kiranmai*

*Department of Computer Science and Engineering, SR International Institute of Technology, Rampally, Keesara, R.R. Dist – 501 301, Andhra Pradesh, INDIA
Email: {farquadonline, latha.gadepaka, balajinaik07, kirranmaai} @gmail.com
* Corresponding Author: +91 40-3257 7638

Abstract— Computational intelligence techniques are proved to be outperforming compared to standard statistical techniques, specifically when dealing with large, unbalanced and high dimensional data. In this paper we present Support Vector Machine as a pre-processor and decision tree is evaluated as a classifier. The proposed approach modifies the training data by replacing the actual target values with the predictions of the trained SVM. This modified training data is then used to train decision tree and the classification is done. The dataset analysed during this research study is COIL dataset which is highly unbalanced with 94:6 class distribution ratio. Extensive experimentation is done including various sampling techniques also to balance the data. Based on sensitivity measure, it is implied that pre-processing using SVM would balance the data well without compromising on the accuracy achieved by the classifier used for classification.

Index Terms— Decision Tree, Support Vector Machine, Pre-processing, Unbalanced data, COIL Data.

I. INTRODUCTION

The class imbalance problem has been recognized in almost every real world applications such as telecommunications [1], detection of oil spoils in satellite radar images [2], learning word pronunciation [3], text classification [4], risk management [5], information retrieval [6], medical diagnosis [7], intrusion detection [8] and fraud detection [9]. It is an evolving topic of research in machine learning. It is witnessed from the literature that traditional machine leaning algorithms are biased towards learning about majority class, thus producing poor prediction accuracy over the minority class and produce suboptimal models [2; 10-11].

Many real world applications have either almost all the records describing one class or very few records describing the other class (usually the more important class). Industrialists, banks and researchers have been involved in attempts to develop an algorithm to balance the data. Methods to deal with imbalanced problems include, resizing training set, adjusting misclassification costs and recognition based learning. Resizing training set is a simple strategy that includes, oversampling minority class samples [12] and downsizing majority class samples [13]. Cost sensitive classifiers [14] have been developed to handle the problem with different misclassification error costs, but may also be used for unbalanced dataset.

In the earliest stages of this research, undersampling using condensed nearest neighbour (CNN) [15], Edited Nearest Neighbor (ENN) [16], Selective undersampling using Tomak-Links concept [13], ENN with
Neighbourhood cleaning rule [17] are proposed. Further, Chawla et al., proposed SMOTE (Synthetic Minority Oversampling Technique) [18], where synthetic (artificial) samples are generated rather than oversampling by replacement.

A. Random Under-sampling

Undersampling is a technique in which some of the samples belonging to the majority class are removed randomly and combined with the minority class samples. For example, 25% undersampling means that the majority class is reduced by 25% of its original size, in other words, 25% of the available majority class instances are removed randomly from training data. 50% undersampling means that the majority class is reduced to 50% of its original size.

B. Random Over-sampling

Oversampling is a technique in which the samples belonging to the minority class are replicated a few times and combined with the majority class samples. For example, 100% oversampling means that the minority class instances are replicated once, in other words, minority class instances are doubled, and 200% oversampling means that the minority class is replicated twice.

C. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is an approach in which the minority class is oversampled by creating synthetic (or artificial) samples, rather than by oversampling with replacement of the existing samples. The minority class is oversampled by taking out each sample and introducing synthetic samples along the line segments that join any or all of the $k$ minority class nearest neighbors. SMOTE is used to widen the data region that corresponds to minority samples. This approach effectively forces the decision region of the minority class to become more general [18].

Machine learning algorithm i.e. Decision Tree is not an exception when it comes to being biased towards better learning and prediction of majority class when compared to learning and prediction about minority class. Further, DT is widely accepted and simplest algorithm for classification purposes. Hence, we select decision tree for calculating prediction accuracy for the problem at hand. Support Vector Machine is known and famous for providing global optimal solution for the classification problems. In this paper we present a data balancing procedure using SVM as pre-processor. The proposed approach obtains the predictions for training records using learnt SVM, the modified training data is then fed to train DT and the prediction accuracy is calculated. To evaluate the efficiency of the proposed data balancing approach various standard sampling techniques such as 25% under-sampling, 50% under-sampling, 100% over-sampling, 200% over-sampling and SMOTE are employed for balancing the data. It is observed that the proposed approach not only balances the data but also improves the prediction efficiency of decision tree.

The rest of the paper is organized as follows. Section 2 briefs about Decision Tree and Support Vector Machine. Section 3 explains the architecture of proposed pre-processing procedure. Section 4 presents the experimental setup which includes dataset description followed by experimental methodology. Empirical analysis is then presented in Section 5. Finally Section 6 concludes the paper.

II. INTELLIGENT ALGORITHMS: A BRIEF

A. Decision Tree

Decision trees form an integral part of ‘machine learning’ an important sub-discipline of artificial intelligence [19]. C4.5 uses a divide-and-conquer approach to growing decision trees that was pioneered by Hunt and his co-workers [20]. Only a brief description of the method is given here; see Quinlan [19] for a more complete treatment.

The following algorithm generates a decision tree from a set $D$ of cases:

- If $D$ satisfies a stopping criterion, the tree for $D$ is a leaf associated with the most frequent class in $D$.
  
  One reason for stopping is that $D$ contains only cases of this class, but other criteria can also be formulated.

- Some test $T$ with mutually exclusive outcomes $T_1$, $T_2$, ..., $T_k$ is used to partition $D$ into subsets $D_1$, $D_2$, ..., $D_k$, where $D_i$ contains those cases that have outcome $T_i$. The tree for $D$ has test $T$ as its root with one sub tree for each outcome $T_i$ that is constructed by applying the same procedure recursively to the cases in $D_i$. 

717
Provided that there are no cases with identical attribute values that belong to different classes, any test $T$ that produces a non-trivial partition of $D$ will eventually lead to single-class subsets as above. However, in the expectation that smaller trees are preferable (being easier to understand and often more accurate predictors), a family of possible tests is examined and one of them chosen to maximize the value of some splitting criterion. The default tests considered by C4.5 are:

- $A=?$ for a discrete attribute $A$, with one outcome for each value of $A$.
- $A\leq t$ for a continuous attribute $A$, with two outcomes, true and false. To find the threshold $t$ that maximizes the splitting criterion, the cases in $D$ are sorted on their values of attribute $A$ to give ordered distinct values $v_1, v_2, \ldots, v_N$. Every pair of adjacent values suggests a potential threshold $t = (v_i + v_{i+1})/2$ and a corresponding partition of $D$. The threshold that yields the best value of the splitting criterion is then selected.

In some situations, every possible test splits $D$ into subsets that have the same class distribution. All tests then have zero gain, and C4.5 uses this as an additional stopping criterion. The recursive partitioning strategy above results in trees that are consistent with the training data, if this is possible. In practical applications data are often noisy—attribute values are incorrectly recorded and cases are misclassified. Noise leads to overly complex trees that attempt to account for these anomalies. Most systems prune the initial tree, identifying sub trees that contribute little to predictive accuracy and replacing each by a leaf. Figure 1 displays an example decision tree to PLAY or DON’T PLAY tennis.

![Example Decision Tree for PLAY](image)

**Figure 1: Example Decision Tree for PLAY**

### B. Support Vector Machine (SVM)

The support vector machine (SVM) is a universal constructive learning procedure based on the statistical learning theory [21-22]. SVMs are an inductive machine learning techniques based on the structural risk minimization principal that aims at minimizing the true error and performs classification by constructing an $N$-dimensional hyper plane that optimally separates the data into two categories. The main objective of SVM is to find an optimal separating hyperplane that correctly classifies data points as much as possible and separates the points of two classes as far as possible, by minimizing the risk of misclassifying the training samples and unseen test samples. Figure 2 shows the hyperplane boundary learnt by SVM and the Support Vectors at the decision boundary.

**Properties of Support Vector Machines:**

- SVM provides the flexibility in choosing a similarity (distance) function.
- Solution can be achieved by using sparse training data i.e. a few samples are usually considered “important” by the algorithm. It is then crucial for SVM to keep number of support vectors as small as possible without compromising the accuracy of the classification.
- SVMs have the ability to handle large feature spaces.
- No probability density estimation is done.
- Perhaps the biggest limitation of the support vectors approach is choice of the kernel, speed and size.
III. PROPOSED PRE-PROCESSING USING SVM

Based on the inferences it is observed that standard machine learning algorithms are biased towards better learning about majority class and decision tree is not an exception. In this paper we present a data balancing procedure using SVM as a pre-processor which in-turn improves the performance of decision tree. SVM is one of the most effective classification techniques proposed in literature. The proposed approach obtains the SVM model using available training data. Later, corresponding actual target values of the training data are replaced by the predictions obtained by trained SVM model. Further, this modified dataset is then used to train DT and prediction accuracy is calculated. The proposed approach is presented in Figure 3. It is observed that use of SVM prediction prior to train DT helps DT to learn better about the customers.

IV. EXPERIMENTAL SETUP

A. Dataset

The dataset analyzed in this paper is from COIL 2000 data mining competition. It contains insurance company’s customers data with the target to predict customers who buy caravan insurance and who do not. Per possible customer, 86 attributes are given: 43 socio-demographic variables which are derived via the customer's ZIP area code, and 43 variables about ownership of other insurance policies. This dataset contains 9822 records in total, out of which 5822 records are considered as training data and remaining 4000 records are considered to be test data. It is observed that only 6% of the customers buy the insurance policy which is the objective of the study, remaining data represents the customers who do not prefer to buy caravan insurance i.e. 94%, which makes the dataset unbalance.

B. Experimental Methodology

To evaluate the proposed data balancing procedure extensive experimentation is done. During the process we also employed standard sampling techniques to balance the data. The sampling techniques employed (only on
training data) are 100% over-sampling, 200% over-sampling, 25% under-sampling, 50% under-sampling and SMOTE. Table 1 presents the class distribution after applying standard sampling techniques. 6th column in Table 1 represents the distribution ratio of the majority and minority class after modifications using the proposed data balancing using SVM approach. It is observed that trained SVM modifies the data into balanced data.

<table>
<thead>
<tr>
<th>Data</th>
<th>Total</th>
<th>Good</th>
<th>Bad</th>
<th>Ratios</th>
<th>Ratios after applying Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>5822</td>
<td>5474</td>
<td>348</td>
<td>94 : 6</td>
<td>42 : 58</td>
</tr>
<tr>
<td>SMOTE</td>
<td>10948</td>
<td>5474</td>
<td>5474</td>
<td>50 : 50</td>
<td>35 : 65</td>
</tr>
<tr>
<td>25% Under-Sampling</td>
<td>4453</td>
<td>4105</td>
<td>348</td>
<td>92 : 8</td>
<td>41 : 59</td>
</tr>
<tr>
<td>50% Under-Sampling</td>
<td>3085</td>
<td>2737</td>
<td>348</td>
<td>89 : 11</td>
<td>45 : 55</td>
</tr>
<tr>
<td>100% Over-Sampling</td>
<td>6170</td>
<td>5474</td>
<td>696</td>
<td>89 : 11</td>
<td>48 : 52</td>
</tr>
<tr>
<td>200% Over-Sampling</td>
<td>6418</td>
<td>5474</td>
<td>944</td>
<td>85 : 15</td>
<td>40 : 60</td>
</tr>
</tbody>
</table>

| TEST DATA        | 4000  | 3762 | 238  | 94 : 6 |

V. Empirical Analysis

Based on the application selected in this paper, identifying the potential customers who can buy caravan insurance policy is the objective of the study. Hence, we place high emphasis on sensitivity alone which contributes towards the bottom-line of the insurance business. Consequently in this paper, sensitivity is accorded top priority ahead of specificity and accuracy. We used the SVM library viz., LibSVM [23] for SVM model building. RapidMiner4.5 community edition is used for employing DT (J48) [24]. The quantities employed to measure the quality of the classifiers are sensitivity, specificity and accuracy, which are defined as follows [25]:

Sensitivity is the measure of proportion of the true positives (customers buying caravan insurance policy in this study), which are correctly identified.

\[
\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}
\]

Specificity is the measure of proportion of the true negatives (customers who do not prefer buying caravan policy in this study), which are correctly identified.

\[
\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}
\]

Accuracy is the measure of proportion of true positives and true negatives, which are correctly identified.

\[
\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}
\]

Table 2 presents the results obtained using stand-alone SVM including standard sampling techniques and proposed approach as well. From the results obtained it implied that SVM perform equally well irrespective of the class distribution ratio of the data. It can be concluded that SVM is efficient enough to learn the best possible separating hyperplane in the available data. It is also observed that stand-alone SVM outperformed using 25% under-sampling data with 69.75% sensitivity whereas SMOTE data outperformed when overall performance is considered with 5699 AUC.

<table>
<thead>
<tr>
<th>Balancing Technique</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Unbalanced</td>
<td>63.45%</td>
<td>42.24%</td>
<td>43.5</td>
<td>5284.5</td>
</tr>
<tr>
<td>SMOTE</td>
<td>67.65%</td>
<td>46.33%</td>
<td>47.6%</td>
<td>5699</td>
</tr>
<tr>
<td>25% Under-Sampling</td>
<td>69.75%</td>
<td>40.59%</td>
<td>42.34%</td>
<td>5517</td>
</tr>
<tr>
<td>50% Under-Sampling</td>
<td>62.61%</td>
<td>44.34%</td>
<td>45.42%</td>
<td>5347.5</td>
</tr>
<tr>
<td>100% Over-Sampling</td>
<td>66.39%</td>
<td>49.04%</td>
<td>50.08%</td>
<td>5771.5</td>
</tr>
<tr>
<td>200% Over-Sampling</td>
<td>67.23%</td>
<td>40.32%</td>
<td>41.93%</td>
<td>5377.5</td>
</tr>
</tbody>
</table>

Results yielded using stand-alone decision tree are presented in Table 3. From the results obtained it is implied that decision tree is biased towards learning better about majority class and its performance degraded very badly for minority class (the objective is to predict minority instances correctly). It is observed that
decision tree using SMOTE data yielded best sensitivity of 53.78% in this category, because the number of instances provided to decision tree about majority class and minority class are equally balanced.

### Table III: Results Yielded Using Decision Tree

<table>
<thead>
<tr>
<th>Balancing Technique</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Unbalanced</td>
<td>6.72%</td>
<td>98.33%</td>
<td>92.8%</td>
<td>5252.5</td>
</tr>
<tr>
<td>SMOTE</td>
<td>53.78%</td>
<td>64.22%</td>
<td>63.6%</td>
<td>6391</td>
</tr>
<tr>
<td>25% Under-Sampling</td>
<td>0%</td>
<td>99.97%</td>
<td>94.02%</td>
<td>4998.5</td>
</tr>
<tr>
<td>50% Under-Sampling</td>
<td>0%</td>
<td>99.81%</td>
<td>93.88%</td>
<td>4990.5</td>
</tr>
<tr>
<td>100% Over-Sampling</td>
<td>7.98%</td>
<td>97.13%</td>
<td>93.12%</td>
<td>5255.5</td>
</tr>
<tr>
<td>200% Over-Sampling</td>
<td>8.4%</td>
<td>97.32%</td>
<td>92.02%</td>
<td>5286</td>
</tr>
</tbody>
</table>

Results obtained using the proposed data balancing approach is presented in Table 4. It is observed that the performance of decision tree is enhanced efficiently when the proposed modified data is used to train it. Even the sensitivity of the SMOTE data is improved from 53.85% to 68.49%. Similarly the results obtained for other datasets (after applying standard balancing techniques) also imply the improvement in the performance of decision tree whereas the efficiency of stand-alone SVM is not achieved by using proposed data balancing approach. When the comparison is done for stand-alone decision tree and the SVM-DT, it is observed that SVM-DT improves its performance irrespective of the balancing technique applied. It is also observed that trained SVM predicts most of the available training instances as minority class instance, resulting in more minority class instances for decision tree to learn from.

### Table IV: Results Yielded Using SVM-DT

<table>
<thead>
<tr>
<th>Balancing Technique</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Unbalanced</td>
<td>60.92</td>
<td>43.41</td>
<td>44.45</td>
<td>5216.5</td>
</tr>
<tr>
<td>SMOTE</td>
<td>68.49</td>
<td>43.65</td>
<td>45.12</td>
<td>5607</td>
</tr>
<tr>
<td>25% Under-Sampling</td>
<td>67.23</td>
<td>39.71</td>
<td>41.35</td>
<td>5347</td>
</tr>
<tr>
<td>50% Under-Sampling</td>
<td>55.88</td>
<td>47.77</td>
<td>48.25</td>
<td>5182.5</td>
</tr>
<tr>
<td>100% Over-Sampling</td>
<td>66.39</td>
<td>47.61</td>
<td>48.72</td>
<td>5700</td>
</tr>
<tr>
<td>200% Over-Sampling</td>
<td>62.61</td>
<td>39.47</td>
<td>40.85</td>
<td>5104</td>
</tr>
</tbody>
</table>

### VI. Conclusions

It is well known that standard machine learning algorithms are biased towards learning about majority class when dealing with unbalanced data and decision tree algorithm is not an exception. In this paper SVM is employed as pre-processor to balance the data and the prediction is done using decision tree. The proposed approach makes use of the training data and builds an SVM model and later the actual target values of training set are replaced by the predictions of this trained SVM. Finally this modified data is used to train decision tree. COIL data mining competition data is analyzed in the present study which is unbalanced with 94:6 class distribution ratio. Various standard data balancing techniques are also implemented for extensive research study. From the results obtained it is concluded that when the predictions of trained SVM are considered, decision tree performs much better than that of using actual training set (including only standard data balancing techniques). For future analysis feature selection, other machine learning classifiers and other unbalanced datasets can be considered for extensive study.

### References


