

# Managing Imbalanced Data Sets in Multi-label Problems: A Case Study with the SMOTE Algorithm

Andrés Felipe Giraldo-Forero<sup>1</sup>, Jorge Alberto Jaramillo-Garzón<sup>1,2</sup>,  
José Francisco Ruiz-Muñoz<sup>1</sup>, and César Germán Castellanos-Domínguez<sup>1</sup>

<sup>1</sup> Signal Processing and Recognition Group, Universidad Nacional de Colombia,  
Campus la Nubia, Km 7 vía al Magdalena, Manizales, Colombia

<sup>2</sup> Grupo de Máquinas Inteligentes y Reconocimiento de Patrones - MIRP,  
Instituto Tecnológico Metropolitano, Cll 54A No 30-01, Medellín, Colombia  
{[afgiraldofo](mailto:afgiraldofo), [jaramilloj](mailto:jaramilloj), [jfruizmu](mailto:jfruizmu), [cgcastellanosd](mailto:cgcastellanosd)}@unal.edu.co

**Abstract.** Multi-label learning has been becoming an increasingly active area into the machine learning community since a wide variety of real world problems are naturally multi-labeled. However, it is not uncommon to find disparities among the number of samples of each class, which constitutes an additional challenge for the learning algorithm. SMOTE is an oversampling technique that has been successfully applied for balancing single-labeled data sets, but has not been used in multi-label frameworks so far. In this work, several strategies are proposed and compared in order to generate synthetic samples for balancing data sets in the training of multi-label algorithms. Results show that a correct selection of seed samples for oversampling improves the classification performance of multi-label algorithms. The uniform generation oversampling, provides an efficient methodology for a wide scope of real world problems.

## 1 Introduction

Multi-label learning refers to classification problems where each sample can be associated to more than one class at the same time. A high number of real world applications such as image classification [1] or protein sequence annotation [2] are multi-labeled. Methods for classifying multi-label data can be grouped into two categories: transformation of the problem and adaptation of the algorithm [3]. The former kind of methods searches to transform the multi-label problem into a single-label one. For that purpose, they employ intuitive strategies such as considering each different set of labels in the multi-label data set as a single label [1] or learning one binary classifier for each different label (one-against-all) [4]. Although those strategies have reached acceptable performances and they are very commonly used nowadays, none of them considers existing correlations among classes, thus discarding potentially useful information that could help to properly solve the problem.

On the other hand, methods based on adaptation of the algorithm are intended to modify existing algorithms in order to explode those correlations among classes.

Some of the most recent and successful general purpose algorithms in this category are ML-KNN [5], ML-LOC [6], and TRAM [7]. However, it is common to find that the number of elements belonging to each class significantly differ from each other. This “class imbalance” is a well known problem for most standard algorithms that assume balanced class distributions and thus, when presented with complex imbalanced data sets, fail to properly represent the distributive characteristics of the data and provide unfavorable accuracies [8]. Common strategies to manage imbalanced data comprise sub-sampling and over-sampling methods. Among them, SMOTE ( Synthetic Minority Over-sampling Technique) [9] is an over-sampling tool that has been successfully applied to several real world problems, but has not been applied so far to multi-label algorithms. Moreover, standard SMOTE does not consider correlations among classes and therefore it only could generate synthetic samples belonging to single classes. Although some studies have tackled the balance of classes in multi-label problems [10,11], these follow a scheme one vs all to adapt the balancing technique, losing information about the correlations that exist between the classes. In [12] the imbalance is treated from the classification point of view, causing lack of flexibility of the method when this is intended to be adapted to other classifiers.

This work presents a series of strategies for applying class balance tools in conjunction with multi-label algorithms. The strategies are tested over several real-world problems, proving to be a valuable tool to improve classification performance in multi-label problems. The rest of the paper is organized as follows: section II presents the basis of SMOTE; section III explains the proposed strategies to apply SMOTE in conjunction with multi-label algorithms. Experimental framework and results are shown in section IV and discussion and conclusions are presented in sections V and VI, respectively.

## 2 Synthetic Minority Oversampling Technique - Smote

Basic strategies to manage class imbalance are: random sub-sampling and random over-sampling. However, these techniques provide disappointing results in several cases because the former one causes lose of potentially useful information, while the latter induces overfitting due to the exact replication of samples. As an alternative to improve these limitations, SMOTE [9] is an over-sampling strategy that avoids the overfitting because synthetic samples are not exact copies of the original ones. Instead, synthetic samples are interpolated along the line segments connecting seed samples, forcing the decision region of the minority class to become more general. Algorithm 1 describes this procedure.

## 3 Proposed Framework for Balancing Multi-label Data

Generating synthetic data from multi-labeled instances requires a careful selection of inputs for the SMOTE algorithm. For explanatory purposes, consider a set of training samples associated to a set of three possible labels  $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$ ,  $i = 1, 2, \dots, m$ , with  $\mathbf{x}_i \in \mathbb{R}^d$  and  $\mathbf{y}_i \subseteq \mathcal{Y}$ , being  $\mathcal{Y} = \{a, b, c\}$ . Let  $\mathcal{T}_\gamma \subset \mathcal{T}$ ,

---

**Algorithm 1.** SMOTE ( $\mathcal{S}, r, k$ )

---

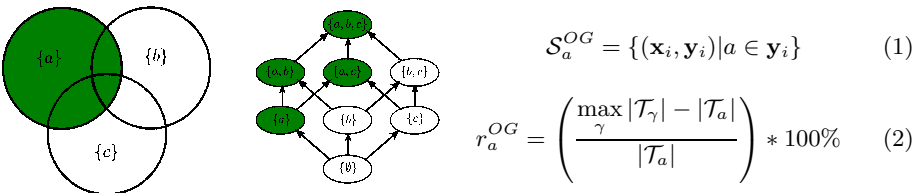
**Input:**  $\mathcal{S}$ : Seed samples, samples of the minority class  $\mathbf{x}_i \in \mathbb{R}^d, i = 1, 2, \dots, m$   
**Input:**  $r$ : Imbalance percentage  
**Input:**  $k$ : Number of nearest neighbors  
1: **for**  $i = 1, 2, \dots, m$  **do**  
2:   Compute distances  $\|\mathbf{x}_i - \mathbf{x}_j\|_2, \forall i \neq j$   
3:   Find the  $k$  nearest neighbors associated to the  $k$  minimum distances  
4:   Compute the number of synthetic samples to be generated from  $\mathbf{x}_i$ ,  
    $n = \text{round}(r/100)$   
5:   **for**  $z = 1, 2, \dots, n$  **do**  
6:     Select a random integer  $\varepsilon$  between 1 and  $k$   
7:     Draw a random vector from a uniform multivariate distribution  $\boldsymbol{\lambda} \sim \mathcal{U}_d(0, 1)$   
8:     Compute the synthetic sample  $\mathbf{s}_i^z = \boldsymbol{\lambda} \circ (\mathbf{x}_i - \mathbf{x}_\varepsilon) + \mathbf{x}_i$  where  $\circ$  is the  
   Hadamard product between vectors  
9:   **end for**  
10: **end for**  
11: **return** The set of  $n \times m$  synthetic samples  $\{\mathbf{s}_i^z\}, i = 1, 2, \dots, m, z = 1, 2, \dots, n$

---

$\gamma \in \mathcal{Y}$ , be the set of samples associated to a given class,  $\mathcal{T}_\gamma = \{(\mathbf{x}_i, \mathbf{y}_i) | \gamma \in \mathbf{y}_i\}$ . Balancing the classes means generating a number of synthetic samples of the three classes such that  $|\mathcal{T}_a| = |\mathcal{T}_b| = |\mathcal{T}_c|$ , where  $|\cdot|$  denotes the number of elements of the set.

Three strategies are proposed in order to apply SMOTE to imbalanced multi-label data. In the first place, it is important to properly define the set of seed samples for each class,  $\mathcal{S}_\gamma \subseteq \mathcal{T}_\gamma, \gamma \in \mathcal{Y}$ . Strategies are depicted with graphs in figures 1 2 3, highlighting in each case the nodes of the graph corresponding to seed samples of the class  $a$ . After defining the set of seed samples, the second input parameter of the SMOTE algorithm is the imbalance percentage  $r$ . The imbalance percentage  $r_a$  drives the decision of how many synthetic samples from class  $a$  must be generated (known as over-sampling). Equations (2), (4) and (6) show the calculus for strategies OG, PG and UG, respectively.

**One-against-All Generation** The first strategy, depicted in figure 1, is defined as a “one-against-all generation” (OG). In this case, all the samples belonging to the minority class are considered as seed samples, that is, seed samples of class  $a$  are defined on equation (1)



**Fig. 1.** one-against-all generation

This strategy is inspired from transformation method called binary relevance, used in [4]. This approach is widely employed, however, it is criticized for not taking into account the correlation between classes.

**Pure Samples Generation** The second strategy, termed “pure samples generation” (PG) is depicted in figure 2. It takes as seed samples only those samples associated to a single label. Thus, the seed samples for class  $a$  are selected according to equation 3.

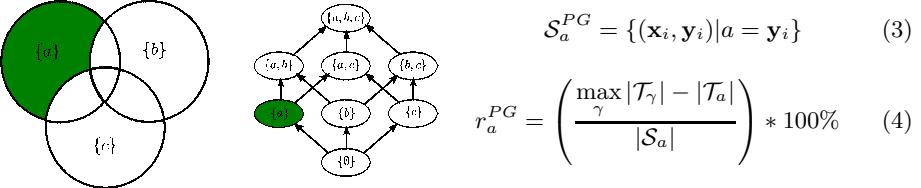


Fig. 2. pure samples generation

This strategy is inspired from transformation method called PT2 described in [3], those were used successfully in problems with low cardinality levels.

**Uniform Generation.** The third strategy, depicted in figure 3 divides the whole set  $\mathcal{T}_a$  into several subsets, and over-samples each set individually. The subsets are defined on , by each different set of labels and are depicted in figure 3 with different intensities of green. This strategy is termed “uniform generation” (UG).

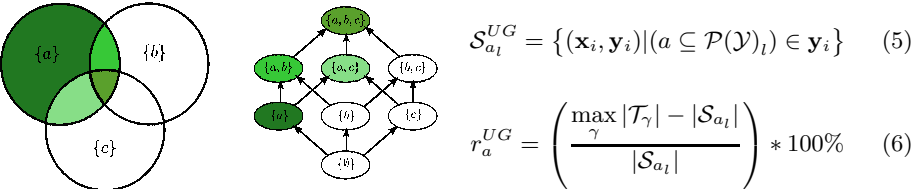


Fig. 3. uniform generation

## 4 Experimental Setup

Experiments were performed over four datasets from the MULAN [13] repository and a fifth dataset from [14]. The first dataset, *Emotion*, comprises 593 songs belonging to several music genres. The second dataset, *Scene*, is a natural scene classification problem [1]. It consists of 2407 natural scene images belonging to six different classes. The third dataset, the *Enron* database, is a subset of the Enron email corpus [15], including 1702 emails with 53 possible labels. Each message was labeled by two people, but no claims of consistency, comprehensiveness, nor generality are made about these labelings. Due to the large number of classes, only those that have more than 300 samples were selected. Also, the principal

component analysis (PCA) was used in order to decrease the number of features. The fourth dataset, *Image*, consists of 2000 natural scene images, where a set of labels is artificially assigned to each image. The number of images belonging to more than one class (e.g. sea+sunset) comprises over 22% of the dataset [14]. Finally, the *Yeast* database is designed to predict functional classes in the genome of yeast *Saccharomyces cerevisiae*. The whole data set has 2417 instances of genes and 14 possible class labels; only the classes 1, 3, 4, 5 and 6 are considered and the first 250 principal components were used as features. Table 1 summarizes the datasets. The cardinality of the database is an average of the number of labels associated to each sample, given an estimate of the difficulty of the multi-label problem.

**Table 1.** Databases summary

Database	Classes	Samples	Features	Cardinality	Database	Classes	Samples	Features	Cardinality
Emotion	6	593	72	1.869	Image	5	2000	135	1.24
Scene	6	2407	294	1.074	Yeast	14	2417	103	4.237
Enron	52	1702	1001	3.378					

The strategies are used in conjunction with three state-of-the-art multi-label methods: TRAM, ML-KNN and ML-LOC. Parameters are fixed for all the datasets as: 10 nearest neighbors for the first two methods (values suggested in [5, 7]), and  $\lambda_1 = 1$ ,  $\lambda_2 = 100$ ,  $m = 15$  for ML-LOC (suggested by [6]). A lineal kernel is used for evading extra tuning parameters. The number of nearest neighbors for SMOTE is set to 5, by literature recommendations [9].

All the reported results are obtained with a 10-fold cross-validation scheme. It is important to point out that, since the Yeast database has the highest cardinality and some of its classes have no pure samples (samples belonging exclusively to that class), the “pure samples regeneration” strategy (PG) could not be applied for such database.

**Table 2.** Results for the Emotion dataset

Classifier	Measure	WO	OG	PG	UG
ML-kNN	$H_{loss} \downarrow$	$0.197 \pm 0.020$	$0.255 \pm 0.021$	$0.239 \pm 0.019$	$0.207 \pm 0.019$
	$F_{measure} \uparrow$	$0.621 \pm 0.037$	$0.459 \pm 0.051$	$0.536 \pm 0.041$	$0.646 \pm 0.030$
	$G_{mean} \uparrow$	$0.700 \pm 0.031$	$0.567 \pm 0.046$	$0.635 \pm 0.035$	$0.734 \pm 0.029$
	$Recall \uparrow$	$0.596 \pm 0.050$	$0.401 \pm 0.053$	$0.480 \pm 0.047$	$0.646 \pm 0.051$
TRAM	$H_{loss} \downarrow$	$0.218 \pm 0.022$	$0.247 \pm 0.027$	$0.240 \pm 0.026$	$0.218 \pm 0.026$
	$F_{measure} \uparrow$	$0.640 \pm 0.032$	$0.540 \pm 0.045$	$0.579 \pm 0.044$	$0.652 \pm 0.040$
	$G_{mean} \uparrow$	$0.729 \pm 0.028$	$0.649 \pm 0.035$	$0.678 \pm 0.035$	$0.743 \pm 0.035$
	$Recall \uparrow$	$0.656 \pm 0.037$	$0.496 \pm 0.042$	$0.546 \pm 0.047$	$0.683 \pm 0.050$
MLLOC	$H_{loss} \downarrow$	$0.248 \pm 0.024$	$0.285 \pm 0.023$	$0.255 \pm 0.022$	$0.253 \pm 0.016$
	$F_{measure} \uparrow$	$0.450 \pm 0.076$	$0.231 \pm 0.074$	$0.424 \pm 0.070$	$0.471 \pm 0.045$
	$G_{mean} \uparrow$	$0.545 \pm 0.074$	$0.313 \pm 0.089$	$0.520 \pm 0.067$	$0.574 \pm 0.046$
	$Recall \uparrow$	$0.389 \pm 0.074$	$0.166 \pm 0.062$	$0.345 \pm 0.070$	$0.413 \pm 0.051$

**Table 3.** Results for the Scene dataset

Classifier	Measure	WO	OG	PG	UG
ML-kNN	$H_{loss} \downarrow$	0.086 ± 0.009	0.089 ± 0.008	0.087 ± 0.008	0.087 ± 0.008
	$F_{measure} \uparrow$	0.742 ± 0.028	0.731 ± 0.025	0.738 ± 0.026	0.741 ± 0.026
	$G_{mean} \uparrow$	0.812 ± 0.020	0.807 ± 0.018	0.815 ± 0.019	0.817 ± 0.019
	$Recall \uparrow$	0.691 ± 0.031	0.688 ± 0.026	0.701 ± 0.029	0.705 ± 0.030
TRAM	$H_{loss} \downarrow$	0.090 ± 0.010	0.093 ± 0.009	0.092 ± 0.010	0.092 ± 0.010
	$F_{measure} \uparrow$	0.746 ± 0.027	0.736 ± 0.027	0.740 ± 0.029	0.740 ± 0.028
	$G_{mean} \uparrow$	0.831 ± 0.019	0.825 ± 0.018	0.828 ± 0.018	0.829 ± 0.019
	$Recall \uparrow$	0.730 ± 0.028	0.725 ± 0.027	0.729 ± 0.026	0.731 ± 0.028
MLLOC	$H_{loss} \downarrow$	0.155 ± 0.009	0.153 ± 0.008	0.152 ± 0.008	0.151 ± 0.006
	$F_{measure} \uparrow$	0.355 ± 0.047	0.339 ± 0.051	0.356 ± 0.045	0.350 ± 0.043
	$G_{mean} \uparrow$	0.463 ± 0.049	0.441 ± 0.053	0.456 ± 0.048	0.453 ± 0.046
	$Recall \uparrow$	0.270 ± 0.045	0.256 ± 0.048	0.274 ± 0.043	0.265 ± 0.040

**Table 4.** Results for the Enron dataset

Classifier	Measure	WO	OG	PG	UG
ML-kNN	$H_{loss} \downarrow$	0.325 ± 0.021	0.393 ± 0.017	0.356 ± 0.019	0.336 ± 0.023
	$F_{measure} \uparrow$	0.458 ± 0.036	0.283 ± 0.033	0.385 ± 0.032	0.533 ± 0.041
	$G_{mean} \uparrow$	0.528 ± 0.033	0.391 ± 0.034	0.496 ± 0.026	0.616 ± 0.031
	$Recall \uparrow$	0.402 ± 0.044	0.255 ± 0.034	0.301 ± 0.027	0.509 ± 0.049
TRAM	$H_{loss} \downarrow$	0.268 ± 0.021	0.361 ± 0.020	0.321 ± 0.025	0.288 ± 0.016
	$F_{measure} \uparrow$	0.623 ± 0.026	0.491 ± 0.029	0.554 ± 0.029	0.633 ± 0.021
	$G_{mean} \uparrow$	0.664 ± 0.023	0.607 ± 0.023	0.646 ± 0.026	0.696 ± 0.018
	$Recall \uparrow$	0.618 ± 0.024	0.500 ± 0.024	0.550 ± 0.032	0.670 ± 0.025
MLLOC	$H_{loss} \downarrow$	0.299 ± 0.016	0.347 ± 0.025	0.317 ± 0.021	0.306 ± 0.020
	$F_{measure} \uparrow$	0.556 ± 0.042	0.425 ± 0.051	0.507 ± 0.039	0.572 ± 0.039
	$G_{mean} \uparrow$	0.601 ± 0.043	0.530 ± 0.046	0.588 ± 0.038	0.624 ± 0.039
	$Recall \uparrow$	0.540 ± 0.047	0.340 ± 0.045	0.446 ± 0.050	0.572 ± 0.055

### 5 Results and Discussion

For comparison purposes, all test are also performed without over-sampling (WO). Tables 2, 3, 4, 5 and 6 show the results for the *Emotions*, *Scene*, *Enron Image* and *Yeast* datasets, respectively. Tables show Hamming Loss ( $H_{loss}$ ), F-measure ( $F_{measure}$ ), Geometric mean ( $G_{mean}$ ) and Recall ( $Recall$ ) defined below.

$$F_{measure} = \frac{1}{Q} \sum_{i \in \mathcal{T}} 2 \frac{|h(x_j) \cap \mathbf{Y}_i|}{|\mathbf{Y}_i \setminus h(x_j)| + 2|h(x_j) \cap \mathbf{Y}_i| + |h(x_j) \setminus \mathbf{Y}_i|}$$

$$H_{loss} = \frac{1}{Q(n-m)} \sum_{i \in \mathcal{T}} |h(x_j) \Delta \mathbf{Y}_i| \quad Recall = \frac{1}{Q} \sum_{i \in \mathcal{T}} \left( \frac{|h(x_j) \cap \mathbf{Y}_i|}{|h(x_j) \cap \mathbf{Y}_i| + |\mathbf{Y}_i \setminus h(x_j)|} \right)$$

$$G_{mean} = \frac{1}{Q} \sum_{i \in \mathcal{T}} \sqrt{\frac{|h(x_j) \cap \mathbf{Y}_i| |h(x_j)^c \cap \mathbf{Y}_i^c|}{(|h(x_j) \cap \mathbf{Y}_i| + |\mathbf{Y}_i \setminus h(x_j)|) (|h(x_j) \setminus \mathbf{Y}_i| + |h(x_j)^c \cap \mathbf{Y}_i^c|)}}$$

where  $\setminus$ ,  $\Delta$ ,  $^c$ ,  $|\cdot|$ ,  $h(x_j)$ ,  $\mathbf{Y}_i$  stands for the difference and symmetric difference between two sets, complement and the cardinality of the set, the set of labels predicted, the set of real labels, respectively; and  $Q$  is the number of labels.

**Table 5.** Results for the Image dataset

Classifier	Measure	WO	OG	PG	UG
ML-kNN	$H_{loss}$ ↓	0.197 ± 0.010	0.201 ± 0.010	○ 0.200 ± 0.011	○ 0.202 ± 0.011
	$F_{measure}$ ↑	0.470 ± 0.029	0.458 ± 0.033	0.471 ± 0.037	0.481 ± 0.040
	$G_{mean}$ ↑	0.580 ± 0.025	0.575 ± 0.029	0.586 ± 0.032	0.598 ± 0.035 ●
	$Recall$ ↑	0.374 ± 0.033	0.380 ± 0.033	0.394 ± 0.040	● 0.404 ± 0.047 ●
TRAM	$H_{loss}$ ↓	0.218 ± 0.015	0.220 ± 0.014	○ 0.218 ± 0.016	○ 0.222 ± 0.017
	$F_{measure}$ ↑	0.531 ± 0.031	0.517 ± 0.029	○ 0.524 ± 0.034	○ 0.526 ± 0.035
	$G_{mean}$ ↑	0.657 ± 0.025	0.649 ± 0.023	○ 0.655 ± 0.027	0.658 ± 0.027
	$Recall$ ↑	0.496 ± 0.032	0.488 ± 0.029	0.494 ± 0.035	0.502 ± 0.036
MLLOC	$H_{loss}$ ↓	0.231 ± 0.011	0.230 ± 0.010	0.230 ± 0.009	0.228 ± 0.011
	$F_{measure}$ ↑	0.161 ± 0.063	0.153 ± 0.058	0.150 ± 0.058	0.181 ± 0.058
	$G_{mean}$ ↑	0.249 ± 0.068	0.240 ± 0.064	0.239 ± 0.060	0.273 ± 0.065
	$Recall$ ↑	0.104 ± 0.047	0.099 ± 0.046	0.096 ± 0.046	0.119 ± 0.045

**Table 6.** Results for the Yeast dataset

Classifier	Measure	WO	OG	UG
ML-kNN	$H_{loss}$ ↓	0.240 ± 0.013	0.273 ± 0.015	○ 0.269 ± 0.016
	$F_{measure}$ ↑	0.575 ± 0.028	0.505 ± 0.033	○ 0.590 ± 0.029 ●
	$G_{mean}$ ↑	0.648 ± 0.023	0.598 ± 0.027	○ 0.674 ± 0.024 ●
	$Recall$ ↑	0.491 ± 0.031	0.400 ± 0.037	○ 0.554 ± 0.039 ●
TRAM	$H_{loss}$ ↓	0.272 ± 0.016	0.295 ± 0.017	○ 0.288 ± 0.017
	$F_{measure}$ ↑	0.598 ± 0.024	0.551 ± 0.028	○ 0.601 ± 0.023
	$G_{mean}$ ↑	0.671 ± 0.02	0.645 ± 0.023	○ 0.685 ± 0.020 ●
	$Recall$ ↑	0.609 ± 0.023	0.518 ± 0.027	○ 0.623 ± 0.025 ●
MLLOC	$H_{loss}$ ↓	0.319 ± 0.019	0.346 ± 0.017	○ 0.319 ± 0.018
	$F_{measure}$ ↑	0.298 ± 0.068	0.153 ± 0.075	○ 0.307 ± 0.057
	$G_{mean}$ ↑	0.386 ± 0.070	0.245 ± 0.087	○ 0.394 ± 0.062
	$Recall$ ↑	0.237 ± 0.058	0.101 ± 0.055	○ 0.246 ± 0.053

All tables depict (mean ± sd) for each metric. Additionally, ●(○) indicate which strategy is significantly higher (lower) than WO strategy, based on paired  $t$ -test at 95% significance level. ↑(↓) implies the larger (smaller), the better.

Strategies OG and PG decrement the performance of classification, presumably due to the fact that this strategy is only capable of generating synthetic samples belonging to a single class instead of truly synthetic samples belonging to multiple classes. In addition PG presents a decrease respecting to the performance without over-sampling, possibly by the inability of this strategy to generate samples with multiple labels, fact that results in the lack of exploitation of the correlations between classes. Generation strategies based on OG, PG and UG showed similar behaviors when applied to datasets with low cardinality (low number of labels per sample), as it can be seen in Table 3, with variations of less 2% in the performance for three classifier. The experiments carried out in this paper show that the performance of ML-kNN can be significantly improved when a balance of classes is made, due to the fact that the prior probabilities on the Bayesian decision rule are computed from relative frequencies of samples and are thus the minority class loses prominence.

## 6 Conclusions

Three strategies for managing imbalanced data sets in multi-label problems are proposed. Experimental analyses on several real-world problems showed that “uniform generation” and classification by TRAM, achieved high performance on four of five tasks. Also, “uniform generation” is the most efficient strategy for multi-label datasets with high cardinality, while ‘pure samples generation” and “one against all” induce noise to the classification. It is important to note that the proposed strategies can also be implemented with other class-balance techniques based on classifier ensembles or undersampling. As future work, a broader study including several class-balance methods can be conducted. Also, further studies are needed for computing the optimal number of synthetic samples.

**Acknowledgements.** This work was partially funded by the Research office (DIMA) at the Universidad Nacional de Colombia at Manizales and (COLCIENCIAS) through grant No.111952128388.

## References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. *Pattern Recognition* 37(9), 1757–1771 (2004)
2. Elisseeff, A.: Kernel methods for multi-labelled classification and categorical regression problems. In: *Advances in Neural Information Processing* (2002)
3. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. *International Journal of Data Warehousing and Mining* 3(3), 1–13 (2007)
4. Jaramillo-Garzón, J.A., et al.: Predictability of protein subcellular locations by pattern recognition techniques. In: *EMBC-IEEE* (2010)
5. Zhang, M., Zhou, Z.: ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition* 40(7), 2038–2048 (2007)
6. Huang, S.J., Zhou, Z.H.: Multi-Label Learning by Exploiting Label Correlations Locally. In: *IAAA* (2012)
7. Kong, X., Ng, M., Zhou, Z.: Transductive Multi-Label Learning via Label Set Propagation. *IEEE Transactions on Knowledge and Data Engineering*, 1–14 (2011)
8. He, H., Garcia, E.: Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21(9), 1263–1284 (2009)
9. Chawla, N., Bowyer, K., Hall, L.: SMOTE: synthetic minority over-sampling technique. *Journal of Artificial* 16 (2002)
10. Tahir, M.A., Kittler, et al.: Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognition* (2012)
11. Dendamrongvit, S., Kubat, M.: Undersampling approach for imbalanced training sets and induction from multi-label text-categorization domains. In: Theeramunkong, T., Nattee, C., Adeodato, P.J.L., Chawla, N., Christen, P., Lenca, P., Poon, J., Williams, G. (eds.) *PAKDD Workshops 2009*. LNCS, vol. 5669, pp. 40–52. Springer, Heidelberg (2010)



12. Chen, K., Liang Lu, B.: Efficient classification of multilabel and imbalanced data using min-max modular classifiers. In: The International Joint Conference on Neural Networks (IJCNN 2006), pp. 1770–1775 (2006)
13. Tsoumakas, G., Vilcek, J., Spyromitros, E., Vlahavas, I.: Mulan: A java library for multi-label learning. *Journal of Machine Learning Research* 1, 1–48 (2010)
14. Zhou, Z.-H., Zhang, M.: Multi-instance multi-label learning with application to scene classification. In: *Advances in Neural Information Processing Systems* (2007)
15. Klimt, B., Yang, Y.: Introducing the Enron Corpus. *Machine Learning* (2004)