6-29-2016

Oversampling Methods for Imbalanced Dataset Classification and their Application to Gynecological Disorder Diagnosis

Iman Nekooeimehr
University of South Florida, nekooeimehr@mail.usf.edu

Oversampling Methods for Imbalanced Dataset Classification and their Application to Gynecological Disorder Diagnosis

by

Iman Nekooeimehr

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Industrial and Management Systems Engineering
College of Engineering
University of South Florida

Major Professor: Susana Lai-Yuen, Ph.D.
Ali Yalcin, Ph.D.
Bo Zeng, Ph.D.
Mingyang Li, Ph.D.
Alfredo Weitzenfeld, Ph.D.
Stuart Hart, M.D.

Date of Approval:
June 3, 2016

Keywords: Binary Classification, Ordinal Regression, Pelvic Organ Prolapse, Object Tracking, Trajectory Analysis

Dedication

Dedicated to my beloved mother, father and brother.

Acknowledgments

I would like to deeply appreciate my advisor, Dr. Susana Lai-Yuen, for her academic advice and inspirations during my Ph.D. She is an excellent researcher, inspiring advisor and a perfect mentor every graduate student would like to work with.

I would like to also thank my committee members Dr. Ali Yalcin, Dr. Bo Zeng, Dr. Mingyang Li, Dr. Alfredo Weitzenfeld, and Dr. Stuart Hart for their valuable advice in my research. Finally, I would like to show my sincere appreciation to Dr. Paul Bao for his kind support during my Ph.D. May his soul rest in peace.

Table of Contents

List of Tables

List of Figures

Abstract

In many applications, the dataset for classification may be highly imbalanced where most of the instances in the training set may belong to some of the classes (majority classes), while only a few instances are from the other classes (minority classes). Conventional classifiers will strongly favor the majority class and ignore the minority instances. The imbalance problem can occur in both binary data classification and also in ordinal regression. Ordinal regression is a supervised approach for learning the ordinal relationship between classes. Extensive research has been performed for addressing imbalanced datasets for binary classification; however, current methods do not address within-class imbalance and between-class imbalance at the same time. Similarly, there has been very little research work on addressing imbalanced datasets for ordinal regression. Although current standard oversampling methods can be used to improve the dataset class distribution, they do not consider the ordinal relationship between the classes.

The class imbalance problem is a big challenge in classification problems. Most of the clinical datasets are highly imbalanced, which can weaken the performance of classifiers significantly. In this research, the imbalanced dataset classification problem is also examined in the context of a clinical application, particularly pelvic organ prolapse diagnosis. Pelvic organ prolapse (POP) is a major health problem that affects between 30-50% of women in the U.S. Although clinical examination is currently used to diagnose POP, there is still little evidence on specific risk factors that are directly related to particular types of POP and their severity or stages (Stage 0-IV). Data from dynamic MRI related to the movement of pelvic organs has the potential

to improve POP prediction but it is currently analyzed manually limiting its exploration and use to small datasets. Moreover, POP is a disorder with multiple stages that are ordinal and whose distribution is highly imbalanced.

The main goal of this research is two-fold. The first goal is to design new oversampling methods for imbalanced datasets for both binary classification and ordinal regression. The second goal is to automatically track, segment, and classify the trajectory of multiple organs on dynamic MRI to quantitatively describe pelvic organ movement. The extracted image-based data along with the designed oversampling methods will be used to improve the diagnosis of POP. The proposed research consists of three major objectives: 1) to design a new oversampling technique for binary imbalanced dataset classification; 2) to design a novel oversampling technique for ordinal regression with imbalanced datasets; and 3) to design a two-stage method to automatically track and segment multiple pelvic organs on dynamic MRI for improving the prediction of multi-stage POP with imbalanced datasets.

The proposed research aims to provide robust oversampling techniques and image processing models that can (1) effectively handle highly imbalanced datasets for both binary classification and ordinal regression, and (2) automatically track and segment multiple deformable structures for feature extraction from low contrast and nonhomogeneous images and classify them using the resulted trajectories. This research will set the foundation towards a computer-aided decision support system that can automatically extract and analyze image and clinical data to improve the prediction of disorders where the dataset is highly imbalanced through personalized and evidence-based assessment.

Chapter 1

Introduction

In this chapter, the motivation and research objectives are presented. The intellectual merit, and broader impact are then presented followed by the dissertation outline.

1.1. Motivation

Many datasets in various applications are imbalanced where some classes contain many more instances than others. Some examples where imbalanced datasets need to be classified include detection of fraudulent bank account transactions or telephone calls [1, 2], biomedical diagnosis [3, 4], text classification [5], information retrieval and filtering [6] and college student retention [7]. In two-class problems, the class that contains many instances is the *majority* class whereas the class that contains fewer instances is the *minority* class. When the dataset is imbalanced, conventional classifiers typically favor the majority class thus failing to classify the minority observations correctly and resulting in performance loss [8]. When the training data is highly imbalanced, the minority class may not even be detected. This kind of imbalance that exists between two different classes is called *between-class* imbalance. Another kind of imbalance that results in performance loss is *within-class* imbalance, which happens when the minority or majority instances have more than one concept (sub-cluster of data) and some of these concepts have less number of instances than others. In addition, the presence of high overlapping among the concepts is another factor that leads to classifiers' performance loss on minority instances [9]. However, current methods developed for imbalanced problems do not address both within-class imbalance and between-class imbalance at the same time. Most of them also worsen the overlapping among the concepts after trying to issue the imbalance problem.

Traditionally, the objective of supervised learning is to optimize the accuracy for the whole dataset, which may cause the classifier to ignore the performance on each individual class. In particular, in an imbalanced dataset, if a random classifier predicts all instances as the majority class, a very high accuracy can be achieved despite incorrectly classifying all minority instances. Therefore, it is strongly suggested to use measurements that are suitable for imbalanced dataset classification.

Ordinal regression is a supervised approach for learning ordering or ranking patterns, and has the properties of both multi-class classification and metric regression. It has properties of multi-class classification because the outcome is a finite set but it considers the ordinal relationship between classes. Ordinal regression also has properties of metric regression as it assumes the outcome variable is a latent continuous variable where the number of ranks is finite and the difference between ranks is not defined. Ordinal regression has applications in many areas such as information retrieval [10], credit rating [11], medical risk assessment [12], and preference learning [13] because very often, people represent their preferences via ranks and ordered classes. As an example, consider a clinical diagnosis where patients can be categorized to stages ranging from 0 to IV. Higher stages indicate higher severity of the condition so the misclassification error between different stages should be penalized differently. For instance, the misclassification error between stages 0 and IV should be much higher than the error between stages 0 and I. On the other hand, the stages are not continuous and the difference between adjacent stages is not equal making this problem different from regular regression.

Highly imbalanced datasets can be found in many applications. In this research, we focus on the problem of imbalanced datasets in clinical applications. In particular, we address the prediction of pelvic organ prolapse stages, which is a gynecological condition with multiple stages of severity and highly imbalanced datasets. Pelvic Organ Prolapse (POP) is a major health problem that affects up to 30-50% of women [14] with direct costs of about $1 billion per year [15]. POP is a herniation of the female pelvic floor organs (bladder, uterus, small bowel and rectum) into the vagina. This condition can cause significant problems including a bothersome vaginal bulge, and incomplete bowel and bladder emptying. POP is normally diagnosed through clinical examination since there are few associated symptoms. However, very little is known about the risk factors of POP even though it is one of the most common reasons for gynecological surgery according to the National Center for Health Statistics [16]. This makes POP a common but poorly understood condition.

In an effort to better understand the risk factors of POP and improve its diagnosis, data obtained through dynamic magnetic resonance imaging (MRI) of the pelvic floor has been studied as it has the potential to offer information not evident on clinical examination [17-20]. However, data from MRI is currently extracted manually resulting in a time consuming and reader subjective process. This has limited the amount, type, and usefulness of MRI data analyzed in population-based studies of POP.

Given the plethora of potential risk factors for POP, it is very likely that this condition is caused by a combination of risk factors that are patient-specific. Unfortunately, there is currently very few data to predict the risk of development of this disorder and the variables associated with its development remain poorly understood [19]. DMRI has been used to analyze the displacement of the pelvic organs to complement clinical examination. Some studies have indicated some association between the movement of pelvic organs and POP [21, 22]. However, studies have been confined to small datasets limiting conclusive evidence. There is currently no automated or quantitative approach to measure multiple pelvic organ movement and their correlation with the severity of prolapse. The ability to predict prolapse would be extremely important to improve the understanding of POP and to potentially develop adequate preventive strategies. Major challenges in the prediction of POP are that current MRI data is extracted manually and is insufficient, and the distribution of stages is highly imbalanced preventing the development of robust prediction models.

In addition, some of the challenges of automating the analysis of multiple organ movements on DMRI are as follows: (1) many of the frames from the DMRI sequence do not provide additional information as the movement of pelvic organs is captured in only a few frames; (2) within the few frames that capture organ movement, organs sometimes move significantly between consecutive frames so their boundaries do not overlap across the frames; and (3) the trajectories of pelvic organs need to be modeled together to capture the interactions among the organs. Therefore, it is necessary to identify those frames that capture organ movement to reduce computation time, then correctly track organs whose boundaries do not overlap across consecutive frames, and finally, perform trajectory classification of multiple organs to quantitatively describe their movement and determine their potential association with POP.

1.2. Research Objectives

The main goal of this research is two-fold. The first goal is to design new oversampling methods for imbalanced datasets for both binary classification and ordinal regression. The second goal is to automatically track, segment, and analyze the trajectory of multiple organs on dynamic MRI to quantitatively describe pelvic organ movement. Clinical datasets are commonly highly imbalanced, where the majority of the data represents one or few classes. The class imbalance problem is a big challenge in classification problems and can significantly weaken the performance of classifiers. Moreover, in many applications, the classes are ordinal so in contrast with multi-class classification, the ordinal relationship between the classes needs to be considered. In this research, novel oversampling methods for ordinal regression are proposed. Ordinal regression is a supervised approach for learning ordering patterns, and has the properties of both multi-class classification and metric regression. It considers the ordinal relationship between classes and assumes the outcome variable as a latent continuous variable with finite number of ranks. Finally, the designed oversampling methods will be examined in the prediction of two-stage and multi-stage POP while automatically extracting data from pelvic organ movement from dynamic MRI.

The proposed research consists of three main research objectives:

1) To design a new oversampling technique for binary classification. A new method called Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) is proposed to more effectively balance the dataset for two-class classification problems. The proposed method clusters the minority instances using a semi-unsupervised hierarchical clustering approach and adaptively determines the size to oversample each sub-cluster using their classification complexity and cross validation. Then, the minority instances are oversampled depending on their Euclidean distance to the majority class. A-SUWO aims to identify hard-to-learn instances by considering minority instances from each sub-cluster that are closer to the borderline. It also avoids generating synthetic minority instances that overlap with the majority class by considering the majority class in the clustering and oversampling stages. Results demonstrate that the proposed method achieves significantly better results in most datasets compared with other sampling methods.

2) To design a novel oversampling technique for ordinal regression. A new oversampling method called Cluster-based Weighted Oversampling for Ordinal Regression (CWOS-Ord) is proposed for addressing imbalanced datasets in ordinal regression. The proposed CWOS-Ord method aims to address this problem by first clustering minority classes by considering the instances of other classes and oversampling them based on their distances and ordering relationship to other classes' instances. The final size to oversample the clusters depends on their complexity and their initial size so that more synthetic instances are generated for more complex and smaller clusters while fewer instances are generated for more complex and larger clusters. As a secondary contribution, existing oversampling methods for two-class classification have been extended for ordinal regression. Results demonstrate that the proposed method provides significantly better results compared to other methods based on the performance measures, particularly when used on datasets with higher imbalance ratio.

3) To design a new contour tracking method is presented to automatically track and segment pelvic organs on DMRI followed by a multiple-object trajectory classification method to improve the diagnosis of pelvic organ prolapse. A model is presented to automatically track, segment and analyze multiple pelvic organ movement in DMRI. The outcome of this model will provide quantitative data on organ movement to be used in improving the prediction of POP. A contour tracking method is proposed to automatically track and segment multiple pelvic organs from a sequence of DMRI images. The proposed method first tracks the pelvic organs over the frame sequence to generate initial adaptive curves for subsequent organ segmentation and to identify those frames that contain significant changes in organ movement. Given that segmentation is a computationally expensive process, reducing the

number of frames to be segmented to a representative set without losing information aims to reduce computation time. On the second stage, pelvic organs are segmented using the generated initial adaptive curves on the representative frames. Finally, a new Coupled Switched Hidden Markov Model (CSHMM) is proposed as a new dynamic Bayesian model to analyze multiple trajectories and their interactions. This model aims to analyze pelvic organ movement and define MRI-based features for prolapse diagnosis.

1.3. Intellectual Merit

The proposed research aims to address the current challenges regarding the imbalance dataset problem in binary data classification and ordinal regression through novel oversampling techniques. The proposed research also aims to provide new techniques for tracking, segmenting, and analyzing the movement of multiple deformable structures on images for automated feature extraction. For binary classification with imbalanced datasets, a new method called ASUWO is proposed that identifies hard-to-learn instances and avoids generating synthetic minority instances that overlap with the majority class. ASUWO is then extended to address the current challenges in ordinal regression with imbalance datasets. The new method called CWOS-Ord considers the ordering relationship to other classes' instances to oversample the clusters based on their complexity. Finally, a two-stage method is proposed to automatically track and segment multiple pelvic organs from dynamic MRI from a sequence of images. Then, trajectory classification is proposed to quantify organ movement to predict the risk of development of POP.

Clinically, the proposed research is expected to provide a quantitative model to better predict the risk of development of POP in women while increasing our understanding of the risk factors related to the different types and stages of POP. The proposed techniques will set the foundation towards a computer-aided decision support system that can automatically extract and analyze image and clinical data to improve the prediction of POP.

1.4. Broader Impact

The outcome of the proposed research will be two new oversampling methods for imbalanced datasets for binary classification and ordinal regression. The second outcome is a method to automatically track, segment, and classify the trajectory of multiple organs on dynamic MRI to quantitatively describe pelvic organ movement. There are a number of broader impacts that are expected as a result of this research. First, two generic methods are presented to overcome the problem of imbalance datasets for binary classification and ordinal regression. These two methods can be applied as the pre-processing stage in any imbalanced datasets with numerical features and binary or ordinal outcomes. Experiments on publicly available datasets with different imbalance ratios demonstrate the effectiveness of these two methods in addressing the imbalance problem and improving classification performance.

Another broader impact is the ability to automate the process of analyzing the movement of multiple deformable structures on images. The proposed method is expected to improve automatic tracking, segmentation, and trajectory analysis of multiple deformable structures from a sequence of images. The ability to automate the process of tracking, segmentation and classification of moving organs in Dynamic MRI is expected to improve the prediction of the stages of POP and increase our understanding of risk factors that are directly related to the development of a specific stage of POP. This aims to improve the prediction of POP in predisposed patients to possibly develop personalized preventive strategies and reduce healthcare costs. Although the proposed research focuses on the pelvic region, the proposed techniques are applicable to other problems where images have low contrast, high inhomogeneity, and noise.

Furthermore, it can be applied to other areas such as motion analysis, gesture recognition, and automation and robotics.

1.5. Outline

The remaining chapters of the dissertation are organized as follows: Chapter 2 discusses the state-of-the-art on imbalanced data classification, ordinal regression with imbalanced datasets, POP and current diagnosis, and deformable object tracking and trajectory analysis. Chapter 3 presents the ASUWO method for addressing the imbalance problem in binary data classification. Chapter 4 provides details about the proposed CWOS-Ord method to address the imbalance dataset problem in ordinal regression. Chapter 5 presents the proposed automatic tracking, segmentation and analysis of multiple pelvic organs on dynamic MRI. Finally, Chapter 6 provides the summary and future work, summarizing the major finding and contribution of this dissertation.

Chapter 2

Literature Review

This chapter provides an overview of previous work in the areas of imbalance data classification, the imbalance dataset problem in ordinal regression, pelvic organ prolapse and its current diagnosis, and deformable object tracking and trajectory analysis.

2.1. Imbalanced Dataset Classification

Methods for addressing imbalanced dataset classification can be categorized into four main types of techniques: data preprocessing, algorithmic modification, cost-sensitive learning, and ensemble of classifier sampling methods [23, 24]. The data preprocessing techniques modify the data distribution in order to address the problem of the skewed class distribution in the learning phase [25-27]. The algorithmic modification approaches modify the existing algorithms, to give significance to minority instances [28-30]. Cost-sensitive methods combine both algorithm and data modification approaches to give different misclassification costs for each class in the learning process[31, 32]. Finally, ensemble of classifier sampling methods modify the ensemble learning algorithm to address the imbalance problem, however normally they do not change the base classifier [33-35].

Although there is no one single method that works well for all imbalanced dataset problems, sampling methods have shown great potential as they attempt to improve the dataset itself rather than the classifier [36-39]. Sampling methods change the distribution of each class observation by either oversampling the minority samples or undersampling the majority samples. In the case of oversampling, sampling methods generate new minority instances to balance the dataset and in the case of undersampling, they remove some majority instances from the dataset. Undersampling methods have shown to be less efficient than oversampling methods because the removal of majority instances may eliminate important information from the dataset, especially in cases where the dataset is small [40-42].

The simplest oversampling method is random sampling. It randomly selects a minority instance and duplicates it until the minority class reaches a desired size. Random oversampling generates new instances that are very similar to the original instances resulting in over-fitting. To overcome this problem, Chawla et al. developed Synthetic Minority Oversampling Technique (SMOTE) [37] where new synthetic instances are generated between randomly selected minority instances and their NN-nearest neighbors, where NN is a user-defined variable. However, this may cause over-generalization as the new instances are generated without considering the majority instances thus increasing the overlap between minority and majority classes [3, 39, 43]. Over-generalization can be exacerbated when the dataset has higher imbalance ratio as the instances of the minority class are very sparse and can become contained within the majority class after oversampling. This can further deteriorate subsequent classification performance [44].

Various approaches have been proposed to address over-generalization. Safe-level SMOTE [45] presents a method that calculates a “safe-level” value for each minority instance, then generates synthetic instances closer to the largest safe level. The safe-level value is defined as the number of other minority instances among its NN-nearest neighbors. Safe-level SMOTE can cause overfitting because synthetic instances are forced to be generated farther from the decision boundary. Borderline-SMOTE [38] presents a method to identify the borderline between the two classes, and oversamples only the minority samples on the borderline. ADASYN [41] assigns weights to minority instances so that those that have more majority instances in their neighborhood have higher chance to be oversampled. However, Borderline-SMOTE and ADASYN do not find all the minority instances close to the decision boundary [36]. MWMOTE [36] approaches this problem by presenting a two-step procedure to find candidate majority border instances and then candidate minority border instances. Then, weights are assigned to candidate minority instances based on their Euclidean distances to the candidate majority border instances so that those with higher weights have a higher chance to be oversampled. However, small concepts of minority instances that are far from the majority class are not detected even if they may contain important information as shown in Figure 2.1(a). In general, it is necessary to find hard-to-learn instances to be used for oversampling because they contain important information for the classifier. These instances are usually near the decision boundary or belong to small concepts [3, 40]. The presence of small concepts in the dataset is referred to as within-class imbalance.

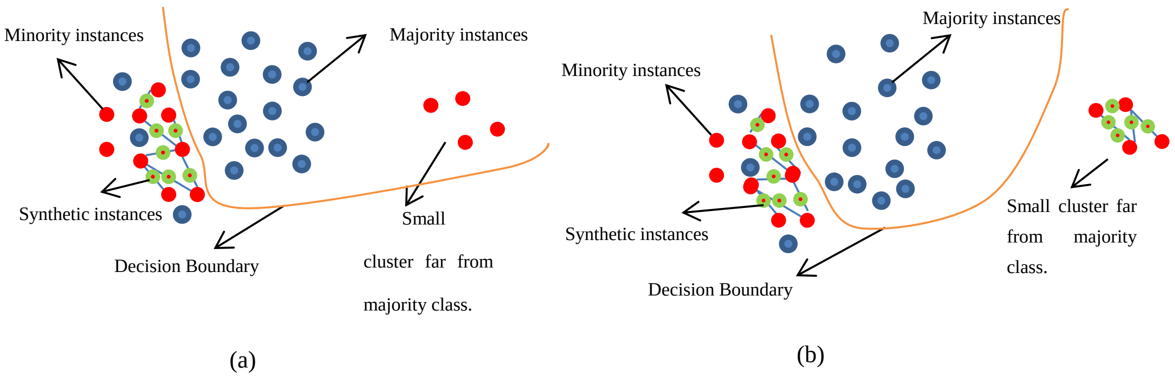


Figure 2.1 Isolated minority clusters. (a) Minority cluster that is far from the majority class is ignored and not oversampled. (b) All minority clusters are oversampled based on their misclassification complexity.

Recently, clustering-based methods [39, 46-49] have been presented to address within-class imbalance. Generally, these methods decompose the dataset into several smaller sub-clusters and use sampling methods to increase or decrease their size. In [46], each class is clustered separately so oversampling is performed on all the sub-clusters of the same class to make their size equal. Cluster-SMOTE [49] first clusters the minority class into m sub-clusters using k-means algorithm and then applies SMOTE to each of them. Under-sampling based on Clustering (SBC) [39] method first clusters the whole dataset into m sub-clusters, then for each of them, it computes the ratio of the number of majority instances to the number of minority instances. Finally, their method removes majority instances based on the ratio, i.e., they remove more majority instances from sub-clusters with lower ratio while they keep more majority instances from the ones with higher ratio. However, removing instances from the dataset may remove important information. In [47], the dataset is partitioned using the Hellinger distance and for each partition, the majority instances are undersampled while the minority instances are oversampled to reach a desired imbalance ratio. In [48], the minority class is clustered into several arbitrary shaped sub-clusters, and the synthetic instances are generated between the minority instances and their corresponding sub-cluster's pseudo-centroids. However, these methods do not identify instances that are close to the decision boundary and do not consider the classification complexity of the sub-clusters when determining the level to which each sub-cluster should be oversampled.

2.2. Ordinal Regression with Imbalanced Datasets

Ordinal regression is a supervised approach for learning ordering or ranking patterns, and has the properties of both multi-class classification and metric regression. It has properties of multi-class classification because the outcome is a finite set but it considers the ordinal relationship between classes. Ordinal regression also has properties of metric regression as it assumes the outcome variable is a latent continuous variable where the number of ranks is finite and the difference between ranks is not defined. Ordinal regression has applications in many areas such as information retrieval [50], credit rating [11], medical risk assessment [12], and preference learning [13] because very often, people represent their preferences via ranks and ordered classes. As an example, consider a clinical diagnosis where patients can be categorized to stages ranging from 0 to IV. Higher stages indicate higher severity of the condition so the misclassification error between different stages should be penalized differently. For instance, the misclassification error between stages 0 and IV should be much higher than the error between stages 0 and I. On the other hand, the stages are not continuous and the difference between adjacent stages is not equal making this problem different from regular regression.

Most of the research conducted to address the imbalanced dataset problem focus mainly on the two-class problem and some works have addressed the multi-class imbalanced problem [51-53]. Despite the increasing interest in ordinal regression problems, little research has focused on ordinal regression with imbalanced datasets. In [54], a new Ordinal Graph-based Oversampling (OGO) framework is proposed to generate synthetic instances by considering the ordering relationship between the classes. The framework consists of three versions: OGO-NI, OGO-SP, and OGO-ISP. OGO-NI first finds the instances on the border of the adjacent classes and then, it creates synthetic instances for the minority class between the minority class instances and the instances in the border of the adjacent classes. In OGO-ISP and OGO-SP, minority instances that are along the shortest path of their adjacent classes are identified and those that are not on the shortest path are removed from the dataset to avoid oversampling outliers. The difference is that in OGO-ISP, new instances are created only between the instances of the minority class and not the instances of adjacent classes. On the other hand, OGO-SP uses a probability weighting function to create synthetic instances between minority classes and also their adjacent classes. All three versions are computationally expensive due to the graph representation of the data and do not consider small clusters of data that may contain important information.

2.3. Pelvic Organ Prolapse and Current Diagnosis

There are three main types of prolapse depending on the part of the vagina being affected [55]: anterior (bladder), apical (uterus), and posterior (rectum). For each type of POP, its severity is graded into five levels: Stage 0, I, II, III, and IV, where Stage IV corresponds to the most severe level of POP. The distribution of stages for POP has been reported to be highly imbalanced [56, 57]. Overall, Swift et al. [57] reported the following stage distribution for POP: Stage 0, 6.4%; I, 43.3%; II, 47.7%; III, 2.6%; and IV, 0%. This highly imbalanced distribution results in insufficient data for certain stages of POP. Consequently, this has made it very difficult to identify risk factors that are directly related to the development of a certain stage of POP.

Various potential risk factors have been associated with POP such as age, and vaginal delivery [56, 58-61]. However, there is still very little evidence about risk factors that are directly related to the different types and stages of POP. Specifically, only weak to moderate correlations have been found between the presence of certain factors and the type and stage of POP [62-64]. For most cases of severe prolapse (stage II-IV), the preferred treatment is repair surgery. However, these surgeries are associated with high failure rates, with approximately 30% of women who undergo surgical repair requiring another surgery for recurrence of symptoms [65, 66]. Previous studies have shown the benefits of dynamic MRI for complementing clinical examination in the evaluation of POP [67, 68].

Dynamic MRI for pelvic area is a sequence of MRI images taken during straining maneuvers starting from minimal to maximal straining as can be seen in Figure 2.2 Dynamic MR images (DMRI) offers the advantages of providing a global assessment of the pelvic floor and it has been found especially important in the diagnosis of patients with multi-compartment prolapse or who have failed previous prolapse surgeries. It also provides a sequence of MR images to enable the observation of pelvic organ movement from rest to maximum strain during examination. DMRI is analyzed manually based on pelvic organ movement and reference lines to determine the stages of POP.

Various studies have been performed to correlate clinical examination and MRI data for POP diagnosis [17-19]. The main disadvantage of these studies is that they have not been completely tested on larger datasets given that the MRI data extraction remains manual, time-consuming and subjective. Also, there is currently no automated or quantitative approach to extract or measure MRI-based features or pelvic organ movement.

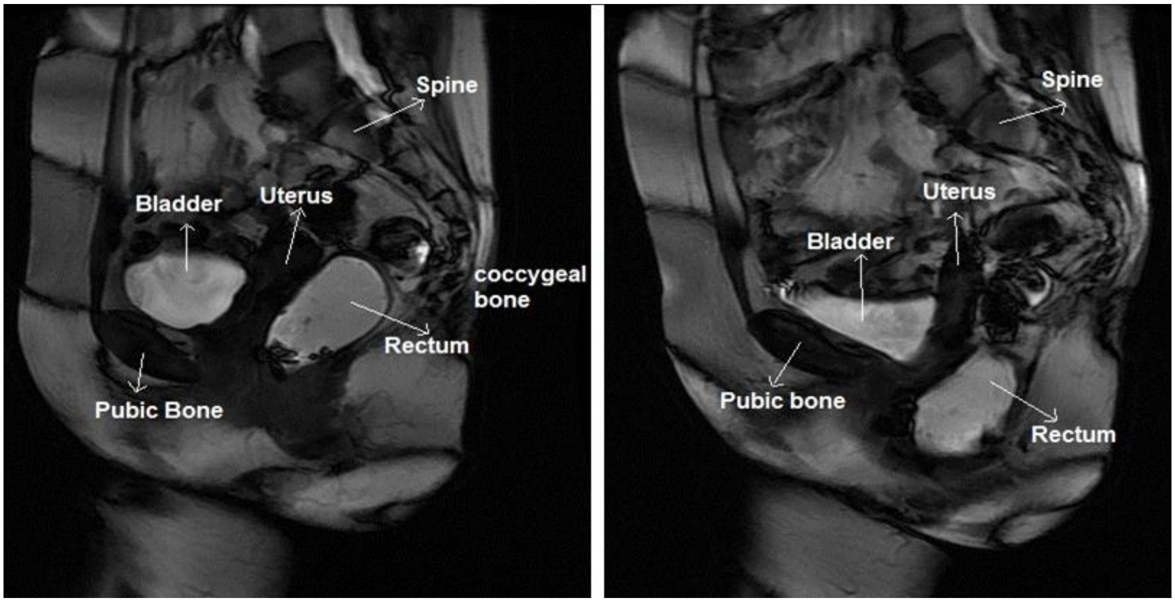


Figure 2.2 Dynamic MRI: Midsagittal dynamic MRI of pelvic floor at rest and at maximum strain.

Different methods based on DMRI have been suggested to assist in the diagnosis of POP [69]. The most common method is to manually find a reference line and determine the distances between the lowest point of the pelvic organ wall and the reference line. If the lowest point falls more than a specified distance below the reference line, the patient is considered to have prolapse. Three of the most widely used reference lines are depicted in Figure 2.3 and described as follows:

1) Pubococcygeal Line (PCL): Straight line between the inferior rim of the pubic bone and the last visible coccygeal line [69].
2) H-Line: Straight line between inferior rim of the pubic bone to the posterior anal canal [70].
3) Mid-pubic line (MPL): Line drawn through the longitudinal axis of the pubic bone and passing through its midequatorial point [69].

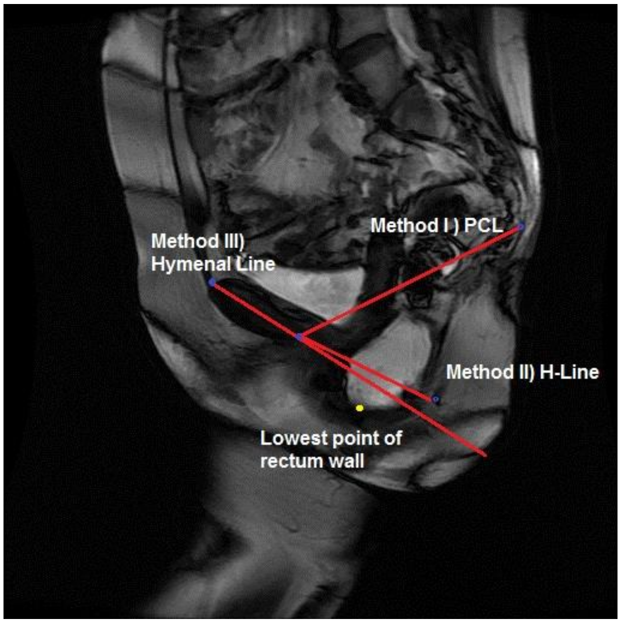


Figure 2.3 DMRI image of pelvic area after maximum strain and the three most widely used reference lines.

Even though 3D MRIs could possibly provide more comprehensive information regarding pelvic prolapse, it is normally not used clinically due to the high cost-benefit ratio. Moreover, using 3D MRI requires reconstruction of the organs, which is a tedious and time-consuming task due to highly irregular organ boundaries [71]. For these reasons, 3D MRI is not commonly used for supporting the diagnosis of POP so this research concentrates only on 2D MRIs.

2.4. Deformable Object Tracking, Segmentation and Trajectory Analysis

The proposed method in Chapter 5 consists of contour tracking followed by trajectory classification to improve clinical diagnosis of POP. In this section, an overview of related work on object tracking, segmentation, and trajectory classification is presented.

2.4.1. Tracking

Object tracking is an important topic in computer vision, particularly in applications such as teleconferencing, surveillance, human–computer interface, automation and robotics. Extensive surveys for object tracking can be found in [72-74]. Tracking deformable objects is more challenging because the object may go through changes in size, shape, color, and texture during the image sequence making it very difficult and sometimes impossible to track. Tracking algorithms can be grouped into three categories [72, 75]: point tracking, kernel tracking and silhouette tracking.

2.4.1.1. Point Tracking

In point tracking, the object of interest is represented by points and the problem is to find corresponding points during the video. A very well established method in this category is the Kalman Filter [76], which is used to estimate the state of a linear system that is assumed to follow a Gaussian distribution.

Another method in this category is particle filter, which has been shown to be very effective and fast for object tracking. A particle filter-based tracker maintains a probability distribution over the state (location, velocity) of the object being tracked. Particle filters represent this distribution as a set of weighted samples or particles. Each particle is a guess representing one possible location of the object being tracked and its corresponding velocity. The set of particles contains more weight at locations where the object being tracked has features that are more similar to a set of predefined features. These predefined features describe the intensity and texture of the object being tracked [77].

2.4.1.2. Kernel Tracking

Kernel tracking assume that an object can be distinguished from the background by a kernel probability density function (pdf). Among kernel based methods, particle filter [77] uses a cloud of weighted particles that propagates in time to represent the posterior pdf. Camshift [78, 79] uses the mean shift algorithm to find the centroid of the pdf that represents the object’s location. Silhouette tracking is used for complex shapes that cannot be easily described by simple geometric shapes.

2.4.1.3. Silhouette Tracking

In silhouette tracking, the object region is estimated in each frame and the information inside the object region like appearance density or shape models is used for tracking [72, 75]. Contour tracking is an example of silhouette tracking where an initial contour is evolved to its new position in the current frame by either using the state space models [80, 81] or direct minimization of some energy function [82-84]. In particular, the state space models update the state space at each frame to maximize the contour’s posteriori probability. The posterior probability depends on the prior state and the likelihood probability of the contour in the current frame. As an example, a novel and fast HMM framework is proposed in [80] in which a joint probability data association filter (JPDAF) is used to determine the HMM's transition probabilities. Contour evaluation methods define the contour as an energy function and minimize it using greedy methods or gradient descent. Current contour tracking methods require some part of the object in each frame to have overlap with the object region in the previous frame which is not the case in tracking pelvic organs.

The organs in dynamic MRI for POP, sometime move dramatically as the patient may move a lot during image acquisition.

2.4.2. Segmentation

Segmentation is the process of partitioning a digital image into multiple segments. Segmentation algorithms can be categorized in several categories including active contours (Snake) and graph-based algorithms [72]. Active contours are defined as deforming dynamic curves that self-adjust towards the object boundaries by an internal and external energy minimization. However, they tend to converge to the closest local minimum of its energy function. Therefore, they only provide an accurate segmentation if its initialization is close to the edges of the object. In particular, in the segmentation method presented by Chan and Vese [85], the snake starts from a rough estimate of the object and then evolves to a close approximation of the object. Each pixel on the snake is assigned a velocity that is determined based on the homogeneity of the image and shape of the snake at the pixel. This makes the snake move outwards faster in the pixels with higher velocity. Graph-based algorithms look for a set of optimum segment boundary lines that separate interior and exterior markers. However, they have an inherent bias towards shorter rather than better segment boundaries and are sensitive to marker location [86].

2.4.3. Trajectory Classification

Once the object has been tracked and segmented, its trajectory can be used for classification by building a model to predict the class of the moving object. Trajectory classification is defined as building a model to predict the class of moving objects using their trajectories. It has acquired interest recently due to the advance in both hardware and software technologies in extracting spatiotemporal data. Trajectory classification has applications in automatic video surveillance, activity analysis, sign language recognition, Global Positioning Systems (GPS), intelligent robots and autonomous vehicles. Review for trajectory classification and recognition can be found in Morris et al. [87] and Aggarwal et al. [88]. The current trajectory classification methods represent the trajectory as a set of 2-dimensional or 3-dimentional points and can be categorized into single and multiple trajectories classification.

2.4.3.1. Single Trajectory Classification

Trajectories for hand gestures are extracted in [89] and then classified using time-delay neural networks. The input for the neural network is the location, velocity and orientation of the points in the trajectories. The trajectory is compressed using Principle Component Analysis (PCA) and Gaussian Mixture Models (GMM) is used to model the low-dimensional trajectories in [90]. In [91], the trajectory is segmented at points of change in curvature, and then the sub-trajectories are represented by PCA. The PCA coefficients of the sub-trajectories are then modeled using GMM and Hidden Markov Models (HMM). In one other approach, trajectories are also segmented using a set of low level dynamic models [92]. Later HMM is used to describe the switching among the segments. Beta process HMM was also used [93], where in contrast to previous methods the segments are selectively shared among trajectories. In other words, in all previous methods, trajectories from different classes are separately modeled in which the segments cannot be shared among activities.

2.4.3.2. Multiple Trajectories Classification

Little research work has been conducted to model multiple trajectories for classification. A feature vector in terms of motion energy images (MEI's) and motion history images (MHI's) is determined for each instance in [94]. Then, Principle Component Analysis (PCA) is used to reduce the dimension of the features followed by a mixture of Gaussian models for classification. However, this model is static and does not consider the temporal ordering of the trajectories.

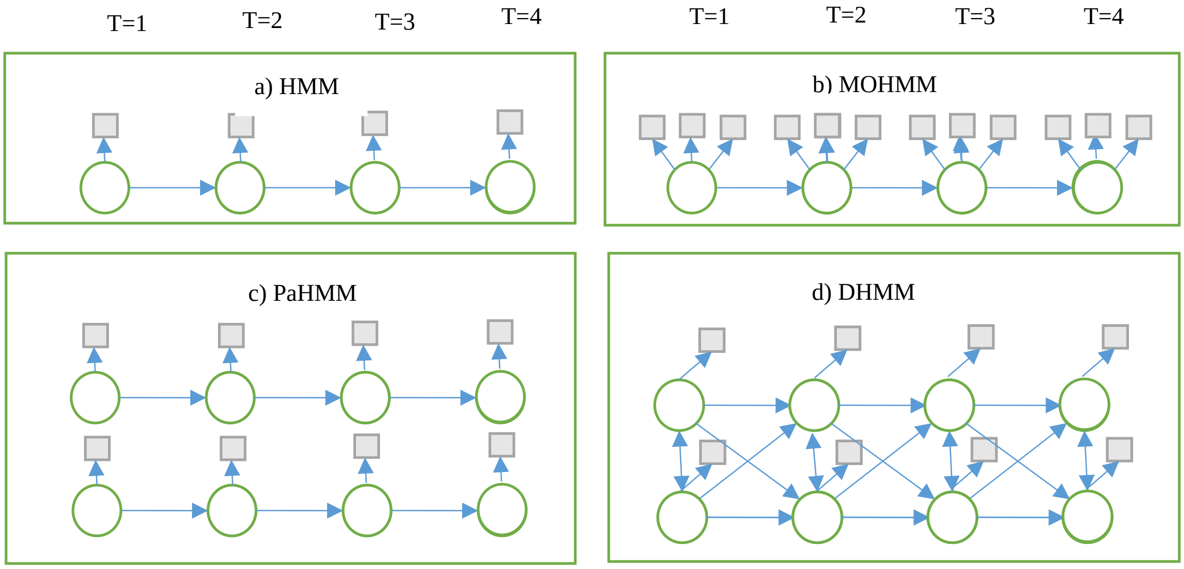


Figure 2.4 Schematic comparison of dynamic models for handling multiple

Dynamic models have been proposed to take into account the temporal ordering of the trajectories. A standard HMM (Figure 2.4.a) can be used to model multiple-objects trajectories which results in multidimensional state states and observation space [95]. However, using a simple HMM, the number of states increases exponentially with the number of objects and hence it is not computationally feasible for large number of objects and over long period of time. To address this problem, some topological extensions have been developed by factorizing either the state space or the observation space or both. In particular, Multi-Observation Hidden Markov Model (MOHMM) was suggested (Figure 2.4.b) in which the observation space is factorized by defining multiple observation variables in each time interval [96]. Parallel Hidden Markov Models (PaHMMs) [97] (Figure 2.4.c) was also proposed in which both space state and observation states are factorized. However, in their method the processes are assumed to be independent which is not always true when the trajectories of interacting objects are modeled. A novel distributed multi-dimensional hidden Markov model (DHMM) (Figure 2.4.d) was proposed in [98] proposed which first models the trajectories as a non-causal, multidimensional hidden Markov model. Then, it distributes the non-causal model into multiple distributed causal HMMs which are solvable. Finally, it approximates the simultaneous solution of multiple distributed HMMs in a sequential updating scheme. However, since DHMM has a fully connected state space, the factorization of the variables requires large computations.

While there has been extensive research on the dynamics of body organs like brain [99], heart [100] and lungs [101], few groups have conducted research to develop more efficient diagnostic processes using the movement and deformation of soft tissues in the pelvic area. In the work proposed in [102], landmarks are tracked over the boundary of pelvic organs during strain and the ones in the border having the most contribution to the diagnosis of prolapse are determined using statistical analysis. Other works concentrate on simulating the movement of pelvic organs (uterus, bladder and rectum) to generate biomechanical models to diagnose prolapse [103]. A finite-element-based numerical simulation was presented in [104] to study the effects of vaginal delivery on the pelvic floor. However, most of these models are built in 3-dimensional MRI, which is not very practical due to their high cost-benefit ratio.

Current contour tracking methods require manual localization of the objects to be segmented. They also rely on the assumption that the boundaries of the objects to be tracked and segmented are overlapping in consecutive frames, which is not the case in our MRI dataset. Then, the DMRI taken for POP contains multiple frames with very little changes that do not provide any additional information. Therefore, these frames need to be identified and removed to reduce computation time for segmentation. Finally, in Chapter 5, the concept of switched HMM [92] has been extended and solved for multiple trajectories using Coupled HMM.

Chapter 3

Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) for Imbalanced Datasets[1]

A new oversampling method called Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) is presented for imbalanced binary dataset classification. A-SUWO finds hard-to-learn instances by first clustering the minority instances and then assigning higher weights to those instances from each sub-cluster that are closer to the majority class. This approach enables the identification of all instances that are close to the decision boundary and also considers all sub-clusters, even small ones, for oversampling as shown in Figure 3.1(b). A-SUWO avoids over-generalization using two strategies. First, it clusters the minority instances by considering the majority class to reduce overlapping between the generated minority instances and majority instances. A semi-unsupervised hierarchical clustering approach is proposed that iteratively forms minority sub-clusters while avoiding majority sub-clusters in between. Second, it oversamples minority instances based on their average Euclidean distance to majority instances to further decrease the chance of generating overlapping instances between classes. In addition, A-SUWO determines sub-cluster sizes adaptively based on their misclassification error. In our method, misclassification error is an indication of sub-cluster complexity and is determined using a new measurement based on the standardized average error rate and cross validation. Sub-clusters with

[1] This chapter was published in Elsevier Journal of Expert Systems with Applications [105] I. Nekooeimehr and S. K. Lai-Yuen, "Adaptive semi-unsupervised weighted oversampling (A-SUWO) for Imbalanced Datasets," *Expert Systems with Applications,* 2015.. Permission is included in Appendix A.

higher misclassification error will be assigned a larger size while the ones with lower misclassification error will be assigned a smaller size.

A-SUWO consists of three main steps: (1) Semi-Unsupervised Clustering, (2) Adaptive Sub-cluster Sizing, and (3) Synthetic Instance Generation. In the first step, the minority instances are clustered using a semi-unsupervised hierarchical clustering approach that iteratively forms minority sub-clusters while avoiding majority sub-clusters in between. In the Adaptive Sub-cluster Sizing step, the size to which each minority sub-cluster will be oversampled is determined based on its complexity in being classified (misclassification error). A new measurement based on the standardized average error rate is proposed to determine the sub-cluster complexity and is calculated using cross validation. Finally, in the Synthetic Instance Generation step, a new weighting system is proposed to assign weights to minority instances based on their average Euclidean distance to their NN-nearest majority class neighbors so that synthetic instances are generated based on these weights.

3.1. Semi-Unsupervised Clustering

In general, there are two approaches for generating synthetic instances. The first one is to generate a new instance between a candidate instance and one of its NN-nearest neighbors [37, 38, 41]. The second approach is to generate a new instance between a candidate instance and one of its neighbors from the same sub-cluster [36]. As can be seen in Figure 3.1(a) and 3.1(b), both approaches can lead to the generation of synthetic instances that overlap with the instances of the other class. In the first approach, some of the NN-nearest neighbors may be far from the candidate instance whereas in the second approach, sub-clusters from different classes may overlap. Overlapping synthetic instances can deteriorate the performance of the classifiers significantly [36, 106].

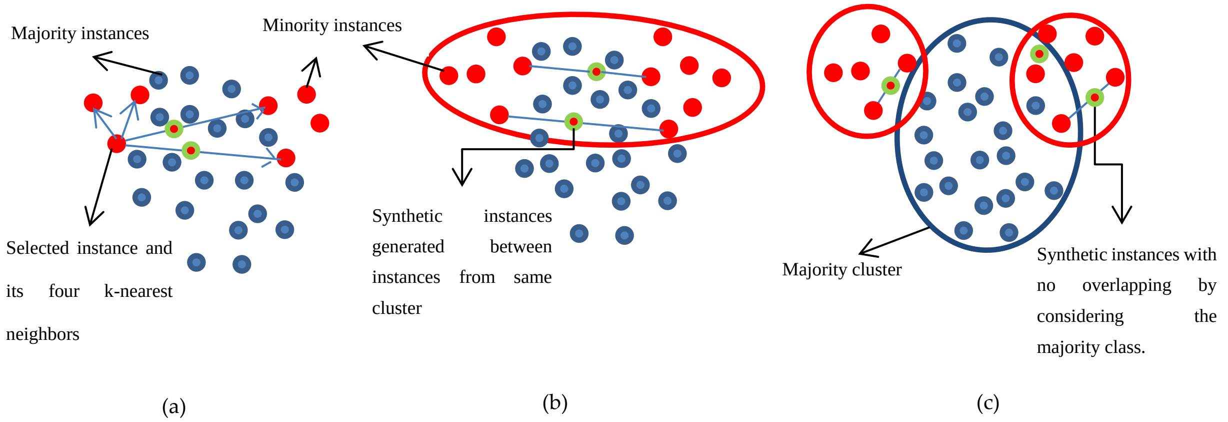

Majority instances
Minority instances
Selected instance and its four k-nearest neighbors
Synthetic instances generated between instances from same cluster
Majority cluster
Synthetic instances with no overlapping by considering the majority class.
(a)
(b)
(c)

Figure 3.1 Approaches for new instance (red dots with green outline) generation. (a) between selected instances and two of its 4-nearest neighbors (red dots) where the generated instances overlap with majority instances (blue dots); (b) between instances of the same cluster where the generated instances overlap with majority instances; and (c) between selected instances and its 4-nearest neighbors provided that they belong to the same cluster. The majority class was also considered while clustering the minority instances.

Our proposed semi-unsupervised hierarchical clustering algorithm is designed to reduce the generation of overlapping synthetic instances. The algorithm is based on the Agglomerative Complete-Linkage Hierarchical Clustering [107] in which overlapping is checked in each iteration for the two minority sub-clusters that are nominated to be merged. If a majority sub-cluster exists between them, the algorithm will not merge the minority sub-clusters. Otherwise, the two nominated sub-clusters are merged if their distance is less than a pre-defined threshold. In contrast with the algorithm presented in [107], our hierarchical clustering method uses information about the majority instances to merge the nominated minority sub-clusters and avoid generating overlapping synthetic instances as shown in Figure 3.1(c). Given that information about the majority instances is used in our clustering approach, the algorithm is not fully unsupervised as in conventional clustering approaches but semi-unsupervised.

Before clustering, noisy instances are identified for both classes using the method suggested by [38] and removed from the dataset. For each instance, NS-nearest neighbors are found. If all the NS-nearest neighbors belong to the other class, then the instance is considered as noise and removed from the dataset because it indicates that it is surrounded by instances of the other class. The Semi-Unsupervised clustering algorithm starts by first clustering the majority class using hierarchical clustering, which results in m majority sub-clusters $Cmaj_{i=1,\dots,m}$. For the minority class, a modification of the hierarchical clustering approach was used because hierarchical clustering enables the detection and avoidance of majority sub-clusters between the generated minority ones. The steps of our proposed semi-unsupervised hierarchical clustering algorithm are as follows, assuming that we have a dataset with N instances as input:

1) Assign each minority instance to a separate sub-cluster. This will result in N sub-clusters of size one $B = \{Cmin_{\tau=1,\dots,N}\}$.
2) Identify the two sub-clusters say $Cmin_a$ and $Cmin_b$ with the lowest Euclidean distance between them. Let their distance be represented by π.
3) Find majority sub-clusters, say $Cmaj_{i\in A}$ with the Euclidean distance to $Cmin_a$ and $Cmin_b$ less than π. A is the set of majority class indices with such property.
4) If $A \neq \emptyset$, then, there exists a majority sub-cluster between $Cmin_a$ and $Cmin_b$ and hence they should not be merged. The distance between $Cmin_a$ and $Cmin_b$ will be set to a large number to avoid being considered for merging again.
5) Else, $Cmin_a$ and $Cmin_b$ are merged into one sub-cluster $Cmin_c$. This will result in one less member in B.
6) Finally, the Euclidean distance between the newly formed minority sub-cluster $Cmin_c$ and existing sub-clusters is recalculated. Steps 2 to 6 are repeated until the Euclidean distance between the closest sub-clusters is less than a threshold T. This will result in n minority sub-clusters.

In contrast with the clustering algorithm from [107], our proposed semi-unsupervised hierarchical clustering algorithm checks whether the two sub-clusters $Cmin_a$ and $Cmin_b$ contain part of a majority sub-cluster (steps 3 through 5). If so, they will not be merged.

In order to obtain a better estimate of T for both minority and majority classes, the median Euclidean distance $d_{med,h}$ of each minority (majority) instance h to all other minority (majority) instances is determined. The median Euclidean distance is used rather than the average distance because the former is more robust to noisy minority instances. Then, we define d_{avg} as the average $d_{med,h}$ over all minority (majority) instances. Therefore, T can be estimated as follows:

$$T = d_{avg} * c_{thres} \qquad (3.1)$$

where c_{thres} is a user-defined constant parameter and its optimum value depends on the dataset. Further suggestion regarding the selection of reasonable values for this parameter can be found in the "Results and Discussion" section.

3.2. Adaptive Sub-cluster Sizing

In current cluster-based oversampling techniques, all sub-clusters have similar sizes after oversampling. However, there might be some sub-clusters with higher misclassification error rate that need more oversampling. Similarly, there might be some with lower misclassification error rate that do not need much oversampling. In the proposed A-SUWO method, the size of the sub-clusters depends on the misclassification rate of the instances in the sub-cluster. The misclassification error for each sub-cluster is calculated using cross validation. This has two main goals. The first goal is to balance the dataset with a 1:1 ratio so that both classes are of the same size. The second goal is to assign a larger size to sub-clusters with higher misclassification error to provide more importance to the ones whose instances are harder to classify.

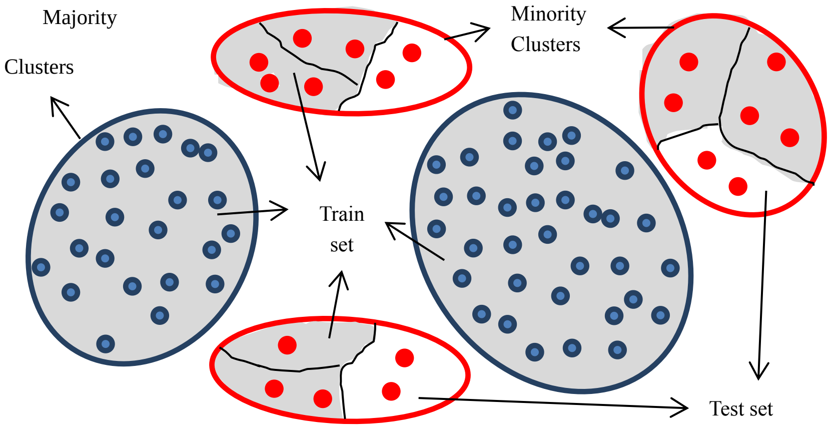


Figure 3.2 Adaptive minority cluster size identification for oversampling based on misclassification error and cross validation. Majority samples (blue dots) and minority samples (red dots).

As shown in Figure 3.2, our method first randomly splits each of the n minority sub-clusters into K similar size partitions (K = 3 in the figure). Then, the classification method (Linear Discriminant Analysis) runs K times and in each fold, $K - 1$ partitions from each minority sub-cluster and all majority instances (in gray background) are used as the training set. Linear Discriminant Analysis was used as our classifier because it is simple and does not require any parameters to tune. Moreover, it was selected over other methods because the purpose is not to get high measures, but rather an estimate of the complexity of the sub-clusters. The remaining one partition from each minority sub-cluster (in white background) is used as the testing set. The misclassification error $\varepsilon_{j\kappa}$ for each minority sub-cluster j in fold κ is determined as the number of minority instances in the testing set incorrectly classified as majority. The error rate $\varepsilon^*{}_{j\kappa}$ is obtained by dividing $\varepsilon_{j\kappa}$ by the number of instances in each sub-cluster R_j. The average error rate $\bar{\varepsilon}_j^*$ is then obtained by averaging the error rate over all folds.

The next step is to standardize $\bar{\varepsilon}_j^*$ to obtain standardized average error rate $\hat{\varepsilon}_j^*$ using the following equation.

$$\hat{\varepsilon}_j^* = \frac{\bar{\varepsilon}_j^*}{\sum_{j=1}^{n} \bar{\varepsilon}_j^*} \qquad (3.2)$$

Following our second goal, the final sizes of any two minority sub-clusters, say $L1$ and $L2$ should have similar ratio to their average error rates $\hat{\varepsilon}_{L1}^*$ and $\hat{\varepsilon}_{L2}^*$. That means,

$$\frac{S_{L1}}{S_{L2}} = \frac{\hat{\varepsilon}_{L1}^*}{\hat{\varepsilon}_{L2}^*} \; \forall \, L1, L2 \in \{1, \dots, n\} \qquad (3.3)$$

where S_{L1} and S_{L2} are the final sizes of $L1$ and $L2$ after oversampling, respectively. $\hat{\varepsilon}_{L1}^*$ and $\hat{\varepsilon}_{L2}^*$ are the standardized average error rate for $L1$ and $L2$, respectively.

The proposed method does not undersample any sub-cluster even if their size calculated using cross-validation is less than the initial sub-cluster size to avoid losing any information. After determining the required number of instances for each minority sub-cluster ($S_{j=1,..,n}$), they should be oversampled to have the corresponding sizes.

3.3. Synthetic Instance Generation

In A-SUWO, we propose to generate synthetic instances between the original instances and their NN-nearest neighbors provided that they belong to the same sub-cluster (Figure 3.1(c)). This is to avoid selecting a NN-nearest neighbor that is far from the instance and that belongs to another sub-cluster thus reducing the generation of overlapping synthetic instances. At the same time, A-SUWO assigns weights to the instances of all sub-clusters separately, which will guarantee that all sub-clusters are oversampled and no isolated small ones are ignored. This is in contrast with the work in [36], where there might be some sub-clusters that are not oversampled at all. It is important to oversample all sub-clusters in order to overcome *within-class* imbalance because ignoring some of them will bias the classifier toward oversampled ones.

Following is the description of the A-SUWO oversampling approach. The first step in oversampling each minority sub-cluster is to assign weights to each minority instance in the sub-cluster based on its average Euclidean distance to NN-nearest majority class neighbors. The reason for giving weights to minority instances lies on the fact that the minority instances closer to the majority instances are more prone to be misclassified and thus more important for classification. This is in contrast with [36], where the weights are assigned based on their average Euclidean distance to all candidate majority border instances even if they are far to some of the minority instances.

For the hth minority instance x_{jh} in minority sub-cluster $Cmin_j$, we find its k nearest neighbors according to its Euclidean distance among majority instances $y_{jh(v)}$ and record the distance $d\left(x_{jh}, y_{jh(v)}\right)$, where $v = 1, \dots, k$ represents the indices of the NN-nearest neighbors. We normalize the distance $d\left(x_{jh}, y_{jh(v)}\right)$ by dividing by the number of features D to make it robust to datasets with different number of features. Therefore, we have:

$$\hat{d}\left(x_{jh}, y_{jh(v)}\right) = \frac{d\left(x_{jh}, y_{jh(v)}\right)}{D} \qquad (3.4)$$

Now, let's define $\Gamma\left(x_{jh}, y_{jh(v)}\right)$ as the closeness factor between x_{jh} and $y_{jh(v)}$.

$$\Gamma\left(x_{jh}, y_{jh(v)}\right) = f_i\left(\frac{1}{\hat{d}\left(x_{jh}, y_{jh(v)}\right)}\right) \qquad (3.5)$$

where f_j is a cutoff function for sub-cluster C_j that prevents $\frac{1}{\hat{d}\left(x_{jh}, y_{jh(v)}\right)}$ from becoming extremely large in the case when the two instances x_{jh} and $y_{jh(v)}$ become too close to each other. Therefore, f_j is defined as:

$$f_j\,(x) = \begin{cases} x & if\ x \leq TH_i \\ TH_j & otherwise \end{cases} \qquad (3.6)$$

TH_j is the largest value $f_j\,(x)$ can reach. In our method, TH_j is determined for each sub-cluster C_j automatically. This is achieved by finding the Euclidean distance of all minority instances x_{jh} in each sub-cluster to their closest majority instance $y_{jh(1)}$ and then determining $f(\frac{1}{\hat{d}(x_{jh},y_{jh(1)})})$. TH_j is then set as the average of $f(\frac{1}{\hat{d}(x_{jh},y_{jh(1)})})$.

$$TH_j = \sum_{j=1}^{R_j} f(\frac{1}{\hat{d}(x_{jh},y_{jh(1)})}) \qquad (3.7)$$

where R_j is the number of instances in C_j. Determining TH_j automatically is a critical step in our method as our weighting algorithm runs for each minority sub-cluster separately and each sub-cluster requires a specific threshold.

In equation 3.5, we have taken the reciprocal of $\hat{d}(x_{jh}, y_{jh(v)})$ because the minority instances closer to the majority instances should have higher weights, while the ones farther from majority instances should have lower weights. Finally, the weights $W(x_{jh})$ are determined based on the Euclidean distance of x_{ij} from all NN-nearest neighbors as follows:

$$W(x_{jh}) = \sum_{v=1}^{k} \Gamma(x_{jh}, y_{jh(v)}) \qquad (3.8)$$

The weights are converted into a probability distribution $P(x_i)$ by dividing each weight by the summation of all weights as follows:

$$P(x_{jh}) = \frac{W(x_{jh})}{\sum_{h=1}^{R_j} W(x_{jh})} \qquad (3.9)$$

In the last step, each $C_j, j = 1, \dots, n$ will be oversampled so that they will have size S_j. In order to oversample them, we first select an instance a in the sub-cluster by sampling from the probability distribution $P(x_{jh})$. Then, one of its NN-nearest neighbors b is randomly selected provided that they belong to the same sub-cluster and a new instance c is generated between a and b as follows:

$$c = \beta a + (1 - \beta) b \qquad (3.10)$$

where β is a random number between 0 and 1. At the end, as can be seen in Figure 3.3, each minority sub-cluster will have some synthetic instances that are generated between original minority instances and are closer to the majority instances. The proposed Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) procedure is described as follows:

```
Algorithm 3.1 – Adaptive Cluster-based Borderline Oversampling (A-CBOS):
Inputs:
- $I$: Original dataset to be oversampled.
- $c_{thres}$: Coefficient to tune the threshold for clustering.
- $NN$: Number of nearest neighbors to be found for each minority instance to determine the weights.
- $NS$: Number of nearest neighbors used to identify noisy instances.
- $K$: Number of folds in the $K$-fold Cross Validation.
Outputs:
  $O$: Oversampled dataset.
Procedure:
  Semi-Unsupervised Clustering
    1. Remove noisy instances from the dataset.
    2. Determine $T$ using equation 3.1.
    3. Cluster majority class, which will result in $m$ sub-clusters $Cmaj_{i=1,\dots,m}$.
    4. Assign each minority instance to a separate sub-cluster.
    5. Find the two closest sub-clusters $Cmin_a$ and $Cmin_b$.
    6. Check if there is any overlapping majority sub-cluster between $Cmin_a$ and $Cmin_b$.
    7. If yes, set their distance to infinity and return back to step 5. Else, merge $Cmin_a$ and $Cmin_b$ into one sub-cluster $Cmin_c$.
    8. Repeat steps 5 to 7 until the Euclidean distance between the closest sub-clusters is less than a threshold $T$.
  Adaptive Sub-cluster Sizing
    1. Randomly split each minority sub-cluster into $K$ folds.
    2. Build a model using $K-1$ folds from each minority sub-cluster in addition to all majority instances as the training set.
    3. Test the model using the remaining one fold from each minority sub-cluster.
    4. Determine Standardized Average Minority Error Rate $\hat{\varepsilon}_j^*$.
    5. Repeat steps 2 to 4 $K$ times.
    6. Determine final sizes $S_j$ for all sub-clusters $Cmin_{j=1,\dots,n}$ using equations 3.2 and 3.3.
  Synthetic Instance Generation
  Determine the probability distribution for instances within each minority sub-cluster:
    - For each sub-cluster $j = 1, 2, \dots, n$
        1. For all minority instances $x_{jh}$ in sub-cluster $Cmin_j$, find $NN$-nearest neighbors among majority instances.
        2. Determine $W(x_{jh})$ for each minority instance in sub-cluster $Cmin_j$ using equations 3.4 – 3.8 and by estimating $TH_j$.
        3. Transform the weights to a probability distribution $P(x_{jh})$ using equation 3.9.
  Oversample minority instances:
    - Initialize $O = I$.
    - For each sub-cluster $j = 1, 2, \dots, n$
        1. Select a minority instance $a$ in sub-cluster $j$ by sampling from the probability distribution $P(x_{jh})$.
        2. Select randomly one of its $NN$-nearest neighbors $b$ that belongs to the same sub-cluster.
        3. Generate a new synthetic instance $c$ between $a$ and $b$ using equation 3.10.
        4. Add $c$ to set $O$.
        5. Repeat steps 1 to 4 until the sub-cluster size reaches $S_j$.
```

3.4. Results and Discussions

The proposed A-SUWO method was evaluated on 16 publicly available datasets and compared with eight other oversampling techniques: 1) Random Oversampling, 2) SMOTE [37], 3) Borderline SMOTE [38], 4) Safe-Level SMOTE [45], 5) Cluster SMOTE [49], 6) SBC [46], 7)

Clustering-Based Oversampling (CBOS) [46], and 8) MWMOTE [36]. The techniques were implemented using Matlab on a workstation with 64-bit Operating System, 4.00 GB RAM, and 2.67 GHz CPU.

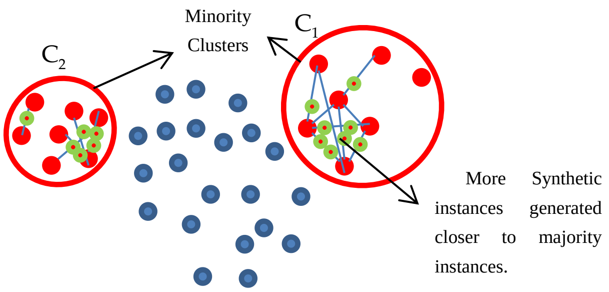

Minority Clusters
C_2
C_1
More Synthetic instances generated closer to majority instances.

Figure 3.3 Generation of synthetic instances. Synthetic instances (red dots with green outline) are generated between original minority instances (red dots) where the generated instances are closer to majority instances (blue dots).

In this study, the performance measures used to compare the different methods are: F-measure, G-mean, and Area under Receiving Operator Characteristics Graph (AUC). F-measure can be calculated using equations 3.11, 3.12 and 3.13 in which minority instances are referred to as positive (P) and majority instances are referred to as negative(N) in the confusion matrix.

$$Precision = \frac{TP}{TP+FP} \qquad (3.11)$$

$$Recall = \frac{TP}{TP+FN} \qquad (3.12)$$

$$F_{measure} = \frac{(1+\beta^2)*Recall*Precision}{\beta^2*Recall+Precision} \qquad (3.13)$$

Precision measures the exactness of the classifier that is, the number of instances labeled as positive (minority) that are actually positive. Recall measures the completeness of the classifier as the number of positive examples that were classified correctly as positive. The parameter β for $F_{measure}$ adjusts the relative importance between precision and recall.

The G-mean is determined as follows:

$$G_{mean} = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \qquad (3.14)$$

G-mean considers the accuracy for both classes. As a result, if the minority class is ignored by the classifier and the majority class is favored, the classifier will obtain a low G-mean.

AUC is the area under ROC graph and is not sensitive to the distribution of the two classes which makes it suitable as a performance measure for the imbalanced problem. The ROC graph is obtained by plotting the True Positive Rate (TPR) over the False Positive Rate (FPR) as defined as follows:

$$TPR = \frac{TP}{N_p} \qquad FPR = \frac{FP}{N_n} \tag{3.15}$$

where N_p is the number of positive (minority) instances and N_n is the number of negative (majority) instances.

Table 3.1 Description of the datasets

#	Dataset	Minority Class	Majority Class	# of features	# of instances	# of minority instances	# of majority instances	Imbalanced Ratio
1	Vehicle	Class "van"	All other	17	846	199	647	1:3.25
2	Ecoli	Class "pp"	All other	7	336	52	259	1:4.98
3	Pima	Class "1"	Class "0"	8	768	268	500	1:1.87
4	Balance	Class "2”	All other	4	625	49	576	1:11.76
5	Liver	Class "1"	Class "2"	6	345	145	200	1:1.38
6	Wine	Class "2"	All other	13	178	71	130	1:1.83
7	Breast Tissue	Class "car" and “fad	All other	9	106	36	70	1:1.94
8	Libra	Class "1", “2”, “3”	All other	90	360	72	288	1:4:00
9	LEV	Class “1”	All. other	4	1000	93	907	1:9.75
10	Iris	Class "2"	All other	4	150	50	100	1:2.00
11	Heart	Class "1"	Class "-1"	13	270	120	150	1:1.25
12	Glass	Class "1"	All other	9	214	70	138	1:1.97
13	Haberman	Class "2"	Class "1"	3	306	81	225	1:2.78
14	Eucalyptus	Class "5"	All other	91	736	105	631	1:6.01
15	Heating	Class “6”, “7”, “8”	All other	8	768	201	567	1:2.82
16	Segment	Class of “WINDOW”	All other	18	2310	330	1980	1:6.00

More detailed information about the 16 datasets is shown in Table 3.1. For those datasets with more than two classes, they were converted into two-class datasets. In order to determine the mean and standard deviation of the performance measures for the oversampling methods, 4-fold stratified cross validation was used. Each experiment was repeated 3 times to report the average in order to alleviate the randomness effects on the results.

Four classifiers were used to evaluate the oversampling methods: Support Vector Machine with radial basis function (SVM) [108], Logistic Regression [109], Nearest Neighbors (KNN) [110], and Linear Discriminant Analysis (LDA) [111]. The parameters of the four classifiers and the nine sampling methods are optimized over a small set of values using stratified cross-validation and considering only the training set.

The cross-validation criterion is G-mean because it is the only criteria that accounts for all values in the confusion matrix and provides the more reliable measure. In particular, for SVM, the parameters for both cost C and gamma γ were selected among the values (2^{-1} , 2^{0} , 2^{1}). For KNN, the number of nearest neighbors was selected among the values (4, 5, 6). Logistic Regression and LDA do not require any parameters to be tuned. For ASUWO, c_{thres} was selected among (1, 2) , NN was selected among (3, 5). NS was selected among (4, 6) and k was set to 3.

Tables 3.2, 3.3, 3.4 and 3.5 show the results of the mean and standard deviation for our proposed A-SUWO method and the other eight sampling methods on the 16 datasets classified using the four classifiers. The best measures are shown in bold. A-SUWO obtains the best results according to at least one of the measures in 13 out of the 16 datasets when SVM and Logistic Regression were used and in 10 out of the 16 datasets when KNN and LDA were used. Additionally, the performance variability for A-SUWO does not vary significantly over the four-fold cross validation and 3 iterations.

The results are further summarized in Table 3.6, which shows each method’s mean rankings in terms of F-measure, G-mean and AUC for all the tested datasets. For each dataset, the best performing method receives a ranking of 1 while the method with the worst performance receives a ranking of 9. Friedman’s test followed by Holm’s test were performed to verify the statistical significance of our method compared to the other sampling methods. Friedman test is a non-parametric equivalent of the repeated-measures ANOVA.

The null hypothesis in Friedman test is whether all classifiers are performing similarly in mean rankings. The results for the Friedman test are shown in Table 3.7. As can be observed from the results, for all four classifiers and all three measures, there exists enough evidence at $\alpha = 0.05$ to reject the null hypothesis, which means that classifiers are not performing similarly.

Since the null hypothesis is rejected for all three performance measures, a post-hoc test is applied. The Holm's test was used where our method was considered as the control method. Holm's test is the non-parametric analog of multiple t-test that adjusts α to compensate for multiple comparisons in a step-down procedure. The null hypothesis is whether ASUWO performs better than other methods as the control algorithm. Table 3.8 shows the adjusted α and the corresponding p-value for each method.

As can be seen from the table, the proposed A-SUWO method outperforms all other methods based on all three measures when SVM was used as the classifier. When KNN, Logistic Regression and LDA was used as the classifier, A-SUWO is significantly better than all other methods in terms of G-mean and F-measure.

We can also observe that the cluster based undersampling method [39] was the method that performs the worst based on all three measures for all classifiers. On the other side, SMOTE and Cluster SMOTE perform well according to AUC, while MWMOTE and Safe-Level SMOTE perform satisfactory according to F-measure and G-mean. Moreover, it can be observed that methods that perform well in terms of G-mean also perform well in terms of F-measure while they do not perform well in terms of AUC.

Table 3.2 Results for the sampling methods on the 16 datasets classified using SVM

Dataset	Meas.	Random	SMOTE	Borderline SMOTE	Safe-level SMOTE	SBC	Cluster SMOTE	CBOS	MWMOTE	A-SUWO
Vehicle	F_M	0.955±0.014	0.953±0.019	0.959±0.010	0.954±0.014	0.917±0.026	0.858±0.099	0.955±0.015	0.948±0.010	**0.961±0.012**
	G-M	0.969±0.009	0.969±0.013	0.972±0.009	0.970±0.009	0.962±0.014	0.931±0.064	0.973±0.007	0.970±0.006	**0.976±0.007**
	AUC	0.995±0.002	0.994±0.003	0.995±0.003	**0.996±0.001**	0.993±0.003	0.989±0.013	**0.996±0.002**	0.995±0.002	**0.996±0.002**
Ecoli	F_M	0.844±0.055	0.860±0.023	0.756±0.069	**0.867±0.033**	0.586±0.158	0.671±0.230	0.796±0.081	0.851±0.025	0.860±0.032
	G-M	0.933±0.039	0.940±0.032	0.905±0.035	0.938±0.030	0.818±0.114	0.835±0.144	0.913±0.030	0.934±0.032	**0.940±0.034**
	AUC	0.954±0.036	0.958±0.028	0.947±0.028	0.960±0.034	0.946±0.038	0.929±0.040	**0.961±0.031**	0.950±0.035	0.959±0.031
Pima	F_M	0.593±0.086	0.589±0.080	0.595±0.088	0.607±0.065	0.652±0.020	0.660±0.037	0.649±0.028	**0.669±0.022**	0.658±0.018
	G-M	0.682±0.071	0.678±0.066	0.681±0.071	0.692±0.053	0.726±0.016	0.736±0.031	0.726±0.022	**0.743±0.018**	0.734±0.015
	AUC	0.769±0.080	0.757±0.069	0.754±0.069	0.767±0.063	0.809±0.022	0.822±0.036	0.811±0.016	0.813±0.021	**0.825±0.022**
Balance	F_M	NaN	0.113±0.066	0.129±0.051	NaN	0.183±0.060	0.213±0.032	**0.250±0.054**	0.221±0.030	0.212±0.036
	G-M	0.115±0.126	0.358±0.116	0.398±0.086	0.079±0.122	0.574±0.113	0.634±0.048	**0.654±0.089**	0.596±0.050	0.548±0.086
	AUC	0.666±0.012	0.703±0.030	0.715±0.022	0.679±0.077	0.648±0.096	0.695±0.021	**0.767±0.045**	0.727±0.026	0.758±0.026
Liver	F_M	0.623±0.032	0.607±0.055	0.628±0.042	**0.637±0.044**	0.590±0.033	0.592±0.030	0.596±0.057	0.595±0.053	0.604±0.044
	G-M	0.668±0.026	0.655±0.041	0.670±0.033	**0.683±0.036**	0.565±0.067	0.546±0.051	0.640±0.037	0.643±0.040	0.659±0.033
	AUC	0.726±0.027	0.727±0.034	0.734±0.038	**0.735±0.034**	0.672±0.062	0.671±0.049	0.697±0.046	0.712±0.047	0.719±0.032
Wine	F_M	0.976±0.020	0.976±0.020	0.976±0.020	0.976±0.020	0.874±0.113	0.974±0.023	0.981±0.017	**0.983±0.021**	0.979±0.018
	G-M	0.978±0.019	0.978±0.019	0.978±0.019	0.978±0.019	0.874±0.137	0.978±0.020	0.983±0.015	**0.985±0.020**	0.981±0.017
	AUC	**0.999±0.001**	**0.999±0.001**	**0.999±0.001**	**0.999±0.001**	0.990±0.011	**0.999±0.002**	**0.999±0.002**	**0.999±0.001**	**0.999±0.001**
Breast	F_M	0.634±0.085	0.654±0.085	0.677±0.082	0.663±0.060	**0.695±0.074**	0.679±0.063	0.664±0.087	0.672±0.087	0.685±0.088
	G-M	0.704±0.077	0.722±0.078	0.741±0.081	0.729±0.057	**0.749±0.085**	0.718±0.082	0.734±0.077	0.739±0.086	0.748±0.088
	AUC	0.815±0.063	0.829±0.052	0.834±0.071	0.833±0.054	0.806±0.076	0.834±0.077	**0.860±0.059**	0.849±0.065	**0.860±0.066**
Libra	F_M	0.509±0.085	0.625±0.140	0.540±0.098	0.610±0.117	0.731±0.142	**0.803±0.099**	0.719±0.101	0.670±0.080	0.684±0.098
	G-M	0.725±0.072	0.756±0.115	0.750±0.082	0.750±0.093	0.762±0.115	**0.828±0.092**	0.751±0.082	0.711±0.065	0.723±0.079
	AUC	0.995±0.007	0.994±0.007	0.995±0.007	0.995±0.007	0.989±0.013	0.988±0.012	**0.996±0.006**	**0.996±0.006**	**0.996±0.006**
LEV	F_M	0.477±0.048	0.510±0.043	0.471±0.039	**0.568±0.049**	0.358±0.056	0.459±0.103	0.415±0.053	0.513±0.045	0.553±0.064
	G-M	0.747±0.047	0.746±0.049	0.746±0.041	0.762±0.048	0.750±0.055	0.773±0.066	0.792±0.040	0.797±0.057	**0.804±0.076**
	AUC	0.748±0.055	0.750±0.053	0.755±0.053	0.787±0.062	0.864±0.034	0.884±0.041	0.869±0.038	**0.889±0.029**	0.870±0.073
Iris	F_M	0.947±0.035	**0.956±0.025**	0.938±0.042	**0.956±0.025**	0.803±0.141	0.828±0.120	0.951±0.031	0.947±0.034	**0.956±0.025**
	G-M	0.967±0.024	**0.972±0.018**	0.962±0.028	**0.972±0.018**	0.832±0.136	0.868±0.100	0.967±0.022	0.970±0.021	**0.972±0.018**
	AUC	0.993±0.004	**0.994±0.004**	0.986±0.014	0.993±0.004	0.990±0.012	0.983±0.016	0.993±0.005	0.993±0.004	**0.994±0.004**
Heart	F_M	0.797±0.028	0.796±0.034	0.794±0.022	0.793±0.026	0.790±0.059	**0.812±0.027**	0.794±0.030	0.801±0.037	0.810±0.036
	G-M	0.817±0.023	0.816±0.029	0.813±0.019	0.814±0.022	0.776±0.130	0.828±0.021	0.814±0.025	0.821±0.031	**0.829±0.030**
	AUC	0.865±0.027	0.867±0.027	0.858±0.020	0.864±0.026	0.864±0.043	0.870±0.031	0.860±0.022	0.872±0.028	**0.873±0.027**
Glass	F_M	**0.755±0.028**	0.740±0.042	0.746±0.047	0.745±0.035	0.650±0.040	0.662±0.042	0.741±0.031	0.750±0.033	**0.755±0.027**
	G-M	**0.828±0.028**	0.813±0.036	0.819±0.041	0.818±0.031	0.699±0.050	0.709±0.067	0.812±0.032	0.821±0.027	**0.828±0.025**
	AUC	0.873±0.020	0.873±0.022	0.867±0.030	**0.880±0.023**	0.821±0.063	0.856±0.037	0.862±0.030	0.861±0.035	0.870±0.028
Haber	F_M	0.435±0.036	0.410±0.042	**0.445±0.067**	0.401±0.035	0.442±0.049	0.395±0.059	0.389±0.034	0.395±0.069	0.412±0.050
	G-M	0.591±0.031	0.571±0.037	**0.604±0.059**	0.566±0.031	0.593±0.042	0.553±0.045	0.552±0.028	0.559±0.062	0.571±0.046
	AUC	0.651±0.031	0.645±0.040	0.649±0.052	0.632±0.027	0.628±0.032	0.636±0.049	0.621±0.025	0.633±0.042	**0.659±0.018**
Eucal.	F_M	NaN	0.189±0.118	0.412±0.082	0.162±0.097	0.327±0.130	0.379±0.124	0.410±0.127	**0.421±0.059**	0.417±0.061
	G-M	0.097±0.112	0.335±0.145	0.567±0.063	0.303±0.116	0.594±0.192	**0.648±0.182**	0.615±0.132	0.604±0.077	0.617±0.061
	AUC	0.707±0.053	0.733±0.036	0.778±0.031	0.724±0.045	0.746±0.072	0.752±0.073	0.753±0.049	**0.797±0.053**	0.785±0.045
Heating	F_M	NaN	0.591±0.021	0.572±0.022	0.614±0.027	0.741±0.071	0.698±0.031	0.746±0.022	0.756±0.039	**0.759±0.052**
	G-M	0.431±0.374	0.690±0.007	0.674±0.012	0.706±0.025	0.844±0.052	0.823±0.013	0.846±0.025	0.859±0.025	**0.865±0.034**
	AUC	0.844±0.038	0.875±0.027	0.853±0.011	0.887±0.015	0.891±0.044	0.880±0.037	0.919±0.020	0.919±0.014	**0.923±0.021**
Seg.	F_M	0.874±0.017	**0.886±0.031**	0.852±0.024	0.883±0.033	0.650±0.158	0.696±0.062	0.855±0.015	0.826±0.056	0.868±0.006
	G-M	0.956±0.009	0.956±0.014	0.940±0.027	0.950±0.012	0.884±0.081	0.915±0.021	**0.957±0.011**	0.946±0.011	0.948±0.008
	AUC	0.986±0.006	0.986±0.006	0.986±0.004	0.983±0.002	0.969±0.017	0.976±0.006	**0.987±0.004**	0.983±0.008	0.984±0.005

Table 3.3 Results for the sampling methods on the 16 datasets classified using KNN

Dataset	Meas.	Random	SMOTE	Borderline SMOTE	Safe-level SMOTE	SBC	Cluster SMOTE	CBOS	MWMOTE	A-SUWO
Vehicle	F_M	0.840±0.025	0.835±0.018	0.864±0.027	0.840±0.022	0.743±0.102	0.833±0.022	0.869±0.032	0.842±0.023	**0.903±0.020**
	G-M	0.930±0.014	0.928±0.009	0.940±0.018	0.929±0.010	0.854±0.074	0.926±0.011	0.942±0.014	0.930±0.012	**0.954±0.010**
	AUC	0.981±0.006	0.984±0.007	0.982±0.004	**0.985±0.005**	0.962±0.014	0.985±0.004	0.983±0.004	0.982±0.005	0.983±0.008
Ecoli	F_M	0.735±0.042	0.731±0.072	0.692±0.055	0.828±0.045	0.601±0.141	0.766±0.025	0.780±0.068	0.795±0.057	**0.840±0.049**
	G-M	0.906±0.027	0.902±0.031	0.890±0.029	0.932±0.035	0.822±0.107	0.915±0.026	0.913±0.029	0.924±0.035	**0.935±0.034**
	AUC	0.939±0.035	0.939±0.035	0.924±0.026	**0.947±0.040**	0.932±0.033	0.944±0.037	0.935±0.034	0.944±0.039	0.945±0.037
Pima	F_M	0.617±0.029	0.616±0.044	0.609±0.026	0.627±0.034	0.590±0.050	**0.639±0.022**	0.579±0.034	0.618±0.035	0.612±0.023
	G-M	0.690±0.023	0.687±0.039	0.673±0.024	0.701±0.028	0.663±0.036	**0.707±0.020**	0.663±0.029	0.687±0.031	0.683±0.020
	AUC	0.742±0.042	0.741±0.045	0.733±0.036	0.754±0.036	0.736±0.032	**0.768±0.026**	0.715±0.037	0.747±0.033	0.741±0.026
Balance	F_M	0.000±0.000	0.094±0.038	0.093±0.052	0.000±0.000	0.208±0.021	**0.243±0.045**	0.124±0.051	0.134±0.061	0.126±0.075
	G-M	0.076±0.118	0.362±0.085	0.357±0.107	0.170±0.137	0.612±0.041	**0.649±0.081**	0.417±0.109	0.433±0.125	0.417±0.146
	AUC	0.445±0.020	0.525±0.042	0.528±0.048	0.454±0.040	0.729±0.039	**0.717±0.064**	0.564±0.049	0.557±0.064	0.592±0.056
Liver	F_M	0.592±0.029	0.576±0.039	**0.596±0.031**	0.574±0.049	0.557±0.054	0.581±0.037	0.587±0.031	0.584±0.036	0.553±0.064
	G-M	0.567±0.025	0.551±0.044	0.569±0.019	0.554±0.054	0.481±0.049	0.550±0.040	**0.577±0.039**	0.562±0.021	0.567±0.033
	AUC	0.608±0.044	0.602±0.035	**0.629±0.044**	0.607±0.054	0.575±0.058	0.611±0.037	0.612±0.036	0.616±0.048	0.603±0.030
Wine	F_M	0.950±0.030	0.953±0.027	0.958±0.031	0.964±0.028	0.910±0.118	0.957±0.024	0.960±0.022	0.964±0.030	**0.969±0.031**
	G-M	0.956±0.026	0.960±0.023	0.965±0.026	0.968±0.025	0.908±0.132	0.964±0.022	0.964±0.020	0.968±0.027	**0.971±0.029**
	AUC	0.990±0.012	0.990±0.012	**0.992±0.012**	0.991±0.012	0.979±0.035	0.990±0.013	0.981±0.020	0.991±0.012	0.991±0.013
Breast	F_M	0.682±0.076	0.697±0.080	0.700±0.083	0.706±0.066	0.689±0.075	**0.738±0.095**	0.698±0.075	0.700±0.067	0.710±0.064
	G-M	0.752±0.065	0.763±0.075	0.763±0.080	0.771±0.064	0.734±0.093	**0.795±0.090**	0.760±0.072	0.766±0.069	0.779±0.053
	AUC	0.845±0.035	0.849±0.039	0.851±0.046	0.846±0.039	0.825±0.052	**0.859±0.054**	0.844±0.056	0.856±0.038	0.845±0.045
Libra	F_M	0.974±0.018	0.979±0.021	0.966±0.025	0.974±0.026	0.811±0.140	0.949±0.039	0.979±0.018	0.977±0.019	**0.981±0.020**
	G-M	0.983±0.014	0.984±0.015	0.979±0.015	0.978±0.024	0.918±0.073	0.978±0.022	0.984±0.015	0.984±0.015	**0.985±0.016**
	AUC	0.986±0.015	0.988±0.015	0.988±0.015	0.986±0.015	0.986±0.014	**0.992±0.012**	0.986±0.015	0.986±0.015	0.988±0.015
LEV	F_M	0.446±0.044	0.451±0.033	0.436±0.038	0.568±0.056	0.324±0.036	0.474±0.072	0.465±0.042	0.477±0.042	**0.581±0.066**
	G-M	0.760±0.044	0.755±0.034	0.759±0.038	0.768±0.038	0.735±0.046	0.755±0.064	0.741±0.036	**0.771±0.043**	0.759±0.038
	AUC	0.791±0.046	0.799±0.042	0.787±0.039	0.810±0.050	**0.833±0.049**	0.814±0.056	0.795±0.038	0.801±0.046	0.807±0.043
Iris	F_M	0.937±0.035	0.942±0.040	0.916±0.041	0.937±0.043	0.826±0.114	0.937±0.039	0.929±0.047	0.933±0.034	**0.942±0.025**
	G-M	0.959±0.025	0.962±0.026	0.946±0.030	0.957±0.031	0.873±0.099	0.959±0.029	0.954±0.033	0.957±0.024	**0.965±0.016**
	AUC	0.975±0.023	**0.980±0.019**	0.973±0.026	0.975±0.022	0.966±0.024	0.979±0.020	0.979±0.019	0.972±0.019	0.977±0.020
Heart	F_M	0.825±0.024	0.826±0.021	0.809±0.014	0.824±0.024	0.823±0.017	0.823±0.027	0.787±0.026	0.820±0.024	**0.832±0.021**
	G-M	0.836±0.021	0.838±0.019	0.812±0.019	0.836±0.021	0.833±0.016	0.835±0.023	0.801±0.025	0.832±0.022	**0.845±0.018**
	AUC	0.893±0.019	0.890±0.020	0.876±0.015	0.892±0.020	0.888±0.024	**0.894±0.019**	0.861±0.021	0.891±0.020	0.886±0.027
Glass	F_M	0.688±0.027	0.707±0.041	0.696±0.029	0.696±0.034	0.629±0.038	0.702±0.031	0.709±0.034	0.700±0.034	**0.720±0.023**
	G-M	0.766±0.024	0.783±0.037	0.773±0.024	0.773±0.029	0.669±0.062	0.778±0.027	0.786±0.030	0.779±0.029	**0.796±0.022**
	AUC	0.839±0.039	0.855±0.036	0.841±0.048	0.850±0.030	0.819±0.043	0.853±0.026	**0.859±0.034**	0.846±0.033	0.856±0.030
Haber	F_M	0.401±0.043	0.393±0.068	0.403±0.071	0.367±0.057	0.391±0.052	**0.447±0.047**	0.383±0.052	0.401±0.087	0.392±0.048
	G-M	0.560±0.038	0.552±0.062	0.559±0.068	0.535±0.051	0.520±0.036	**0.593±0.046**	0.548±0.045	0.560±0.078	0.558±0.042
	AUC	0.574±0.035	0.566±0.054	0.569±0.051	0.571±0.039	0.562±0.062	**0.609±0.043**	0.578±0.040	0.576±0.062	0.576±0.044
Eucal.	F_M	**0.395±0.027**	0.343±0.025	0.342±0.018	0.354±0.024	0.340±0.020	0.388±0.061	0.361±0.021	0.329±0.031	0.368±0.020
	G-M	0.679±0.028	0.647±0.032	0.644±0.021	0.650±0.029	0.620±0.036	**0.687±0.064**	0.666±0.020	0.632±0.035	0.674±0.020
	AUC	0.724±0.049	0.730±0.026	0.719±0.028	0.728±0.018	0.730±0.041	**0.734±0.069**	0.708±0.032	0.701±0.040	0.720±0.025
Heating	F_M	0.697±0.022	0.711±0.030	0.707±0.044	0.724±0.019	0.697±0.014	0.716±0.021	0.665±0.098	0.704±0.014	**0.752±0.048**
	G-M	0.821±0.011	0.835±0.021	0.834±0.035	0.845±0.015	0.827±0.011	0.841±0.013	0.781±0.094	0.829±0.006	**0.860±0.027**
	AUC	0.874±0.014	0.875±0.016	0.866±0.023	0.886±0.019	0.896±0.024	**0.897±0.013**	0.865±0.033	0.885±0.021	0.886±0.016
Seg.	F_M	0.832±0.028	0.827±0.033	0.829±0.024	0.848±0.027	0.593±0.048	0.833±0.041	0.824±0.027	0.838±0.031	**0.855±0.025**
	G-M	0.948±0.012	0.946±0.013	0.945±0.015	0.950±0.009	0.876±0.027	**0.953±0.013**	0.947±0.010	0.950±0.011	0.947±0.013
	AUC	0.967±0.012	0.973±0.012	0.972±0.010	0.971±0.012	0.954±0.017	**0.978±0.009**	0.969±0.009	0.969±0.011	0.964±0.006

Table 3.4 Results for the sampling methods on the 16 datasets classified using Logistic Regression

Dataset	Meas.	Random	SMOTE	Borderline SMOTE	Safe-level SMOTE	SBC	Cluster SMOTE	CBOS	MWMOTE	A-SUWO
Vehicle	F_M	0.932±0.019	**0.934±0.020**	0.923±0.028	0.932±0.019	0.903±0.025	0.931±0.021	0.930±0.021	0.934±0.020	0.921±0.028
	G-M	**0.963±0.012**	0.961±0.014	0.951±0.025	0.960±0.016	0.950±0.009	0.960±0.016	0.958±0.016	0.961±0.014	0.951±0.024
	AUC	0.991±0.006	0.991±0.005	0.989±0.007	0.991±0.006	0.987±0.004	0.991±0.005	0.991±0.006	0.990±0.005	**0.991±0.006**
Ecoli	F_M	0.703±0.039	0.696±0.034	0.601±0.046	0.700±0.020	0.546±0.156	0.692±0.035	0.685±0.085	0.698±0.051	**0.716±0.033**
	G-M	**0.878±0.021**	0.871±0.029	0.839±0.015	0.857±0.039	0.780±0.118	0.863±0.024	0.860±0.031	0.873±0.030	0.863±0.015
	AUC	**0.933±0.026**	0.933±0.027	0.913±0.023	0.933±0.026	0.875±0.088	0.927±0.025	0.924±0.025	0.931±0.026	0.933±0.027
Pima	F_M	0.593±0.086	0.589±0.080	0.595±0.088	0.607±0.065	0.652±0.020	0.660±0.037	0.649±0.028	0.669±0.022	**0.658±0.018**
	G-M	0.682±0.071	0.678±0.066	0.681±0.071	0.692±0.053	0.726±0.016	0.736±0.031	0.726±0.022	0.743±0.018	**0.734±0.015**
	AUC	0.769±0.080	0.757±0.069	0.754±0.069	0.767±0.063	0.809±0.022	**0.822±0.036**	0.811±0.016	0.813±0.021	0.825±0.022
Balance	F_M	0.110±0.032	0.111±0.025	0.116±0.030	0.115±0.060	0.089±0.028	0.133±0.039	0.102±0.023	0.111±0.019	**0.149±0.027**
	G-M	0.442±0.068	0.447±0.053	0.456±0.061	0.425±0.093	0.393±0.067	0.489±0.083	0.427±0.052	0.448±0.036	**0.517±0.048**
	AUC	0.419±0.042	0.432±0.033	0.428±0.037	0.449±0.089	0.448±0.052	0.483±0.070	0.436±0.050	0.417±0.034	**0.530±0.065**
Liver	F_M	0.606±0.052	0.627±0.043	0.611±0.068	0.625±0.037	0.596±0.015	0.617±0.033	0.579±0.064	0.628±0.040	**0.637±0.044**
	G-M	0.641±0.048	0.658±0.034	0.646±0.054	0.661±0.029	0.570±0.063	0.651±0.030	0.600±0.044	0.662±0.031	**0.676±0.034**
	AUC	0.714±0.031	0.718±0.029	0.715±0.033	**0.720±0.024**	0.694±0.026	0.718±0.025	0.690±0.045	0.720±0.023	0.720±0.032
Wine	F_M	0.947±0.034	0.945±0.030	0.945±0.030	0.942±0.030	0.942±0.025	0.945±0.030	0.947±0.034	0.945±0.030	**0.952±0.036**
	G-M	0.954±0.032	0.952±0.030	0.952±0.030	0.950±0.029	0.954±0.022	0.952±0.030	0.954±0.032	0.952±0.030	**0.959±0.031**
	AUC	0.995±0.005	0.994±0.007	0.994±0.008	0.995±0.006	0.992±0.010	0.995±0.006	0.995±0.005	0.994±0.008	**0.996±0.004**
Breast	F_M	0.724±0.085	0.733±0.096	0.696±0.078	0.738±0.093	0.697±0.064	0.759±0.066	0.739±0.093	0.725±0.074	**0.764±0.091**
	G-M	0.792±0.070	0.796±0.076	0.770±0.065	0.803±0.075	0.763±0.062	0.821±0.054	0.806±0.080	0.794±0.062	**0.824±0.073**
	AUC	0.880±0.079	0.882±0.080	**0.896±0.061**	0.883±0.080	0.854±0.073	0.880±0.083	0.892±0.072	0.885±0.069	0.890±0.073
Libra	F_M	0.485±0.109	0.504±0.101	0.505±0.092	0.503±0.098	0.320±0.077	0.498±0.115	0.529±0.128	0.508±0.112	**0.541±0.099**
	G-M	0.647±0.090	0.662±0.088	0.659±0.084	0.658±0.082	0.536±0.079	0.656±0.104	0.678±0.114	0.665±0.097	**0.683±0.088**
	AUC	0.696±0.091	**0.710±0.106**	0.705±0.099	0.707±0.101	0.549±0.095	0.708±0.101	0.708±0.104	0.703±0.102	0.707±0.096
LEV	F_M	0.445±0.024	0.469±0.030	0.387±0.025	0.565±0.058	0.428±0.033	0.448±0.045	0.428±0.055	0.512±0.032	**0.586±0.082**
	G-M	0.813±0.032	0.822±0.033	0.795±0.027	0.815±0.063	0.810±0.026	0.811±0.042	0.815±0.037	**0.824±0.042**	0.813±0.071
	AUC	0.893±0.034	0.894±0.034	0.888±0.034	0.896±0.034	0.892±0.037	0.893±0.034	0.883±0.050	0.893±0.034	**0.897±0.039**
Iris	F_M	0.936±0.029	0.931±0.031	**0.941±0.018**	**0.941±0.018**	0.893±0.110	0.941±0.019	0.931±0.031	0.936±0.029	**0.941±0.018**
	G-M	0.955±0.023	0.950±0.026	**0.960±0.012**	**0.960±0.012**	0.918±0.101	0.957±0.017	0.950±0.026	0.955±0.023	**0.960±0.012**
	AUC	0.991±0.006	0.992±0.006	0.991±0.006	0.992±0.006	0.970±0.057	**0.993±0.005**	0.992±0.006	0.992±0.006	0.992±0.005
Heart	F_M	**0.853±0.027**	0.853±0.028	0.852±0.027	0.849±0.029	0.845±0.028	0.852±0.028	0.829±0.025	0.851±0.027	0.853±0.028
	G-M	0.866±0.025	0.865±0.026	0.864±0.026	0.862±0.027	0.858±0.026	0.865±0.026	0.844±0.023	0.864±0.025	**0.866±0.026**
	AUC	0.930±0.016	**0.930±0.016**	0.923±0.015	0.930±0.015	0.923±0.015	0.930±0.015	0.915±0.015	0.929±0.015	0.929±0.017
Glass	F_M	0.649±0.038	0.637±0.040	0.635±0.050	0.629±0.062	0.642±0.034	0.640±0.058	**0.670±0.058**	0.641±0.046	0.663±0.040
	G-M	0.735±0.030	0.725±0.033	0.721±0.044	0.718±0.049	0.699±0.049	0.727±0.049	**0.753±0.049**	0.728±0.037	0.749±0.035
	AUC	0.827±0.037	**0.830±0.035**	0.820±0.038	0.827±0.034	0.803±0.044	0.824±0.036	0.822±0.037	0.825±0.034	0.818±0.033
Haber	F_M	0.477±0.049	0.465±0.032	0.467±0.041	0.462±0.022	0.458±0.075	0.459±0.067	0.469±0.080	0.453±0.056	**0.508±0.074**
	G-M	0.626±0.041	0.617±0.022	0.622±0.034	0.612±0.022	0.601±0.074	0.605±0.052	0.614±0.079	0.606±0.046	**0.649±0.059**
	AUC	0.673±0.049	0.648±0.029	0.654±0.039	0.653±0.036	0.629±0.082	0.645±0.062	0.634±0.104	0.638±0.038	**0.695±0.074**
Eucal.	F_M	0.499±0.037	0.498±0.041	0.512±0.048	0.496±0.015	0.366±0.024	0.502±0.077	**0.515±0.023**	0.498±0.044	0.511±0.029
	G-M	0.731±0.038	0.730±0.039	0.746±0.046	0.720±0.014	0.672±0.036	0.727±0.071	**0.747±0.018**	0.724±0.034	0.728±0.013
	AUC	0.846±0.016	0.845±0.017	0.843±0.015	0.846±0.018	0.738±0.016	0.842±0.012	0.845±0.023	**0.848±0.017**	0.834±0.029
Heating	F_M	0.720±0.059	0.726±0.041	0.720±0.053	0.723±0.058	0.726±0.055	**0.732±0.042**	0.720±0.059	0.730±0.048	0.728±0.059
	G-M	0.839±0.045	0.842±0.029	**0.845±0.043**	0.837±0.044	0.840±0.042	0.844±0.029	0.835±0.043	0.845±0.033	0.841±0.047
	AUC	0.916±0.027	0.919±0.028	0.907±0.030	0.918±0.028	0.915±0.028	0.919±0.026	0.915±0.030	0.917±0.028	**0.921±0.026**
Seg.	F_M	0.641±0.039	0.647±0.030	0.602±0.026	0.657±0.025	0.591±0.075	**0.667±0.008**	0.630±0.023	0.657±0.034	0.661±0.018
	G-M	0.878±0.012	0.877±0.017	0.854±0.017	0.879±0.016	0.867±0.042	0.872±0.002	**0.881±0.019**	0.875±0.004	0.879±0.011
	AUC	0.942±0.008	0.942±0.008	0.908±0.021	0.942±0.008	0.901±0.051	**0.944±0.010**	0.937±0.006	0.944±0.010	0.942±0.008

Table 3.5 Results for the sampling methods on the 16 datasets classified using LDA

Dataset	Meas.	Random	SMOTE	Borderline SMOTE	Safe-level SMOTE	SBC	Cluster SMOTE	CBOS	MWMOTE	A-SUWO
Vehicle	F_M	0.923±0.014	0.923±0.020	0.921±0.025	0.925±0.016	0.909±0.020	0.926±0.014	0.932±0.012	0.917±0.019	**0.935±0.013**
	G-M	0.963±0.005	0.963±0.007	0.950±0.021	0.963±0.005	0.959±0.010	0.963±0.006	**0.965±0.007**	0.961±0.009	0.964±0.010
	AUC	0.990±0.004	0.991±0.004	0.990±0.007	0.990±0.004	0.986±0.011	**0.991±0.004**	0.989±0.006	0.991±0.004	0.990±0.005
Ecoli	F_M	0.729±0.029	0.724±0.046	0.618±0.039	**0.743±0.038**	0.583±0.142	0.718±0.034	0.713±0.089	0.722±0.038	0.735±0.021
	G-M	0.907±0.017	0.901±0.008	0.847±0.014	**0.911±0.017**	0.818±0.116	0.890±0.018	0.899±0.041	0.901±0.010	0.901±0.018
	AUC	0.937±0.027	0.936±0.027	0.919±0.017	0.937±0.028	0.920±0.059	0.939±0.027	0.938±0.027	0.938±0.026	**0.939±0.028**
Pima	F_M	0.663±0.039	0.669±0.038	**0.675±0.027**	0.670±0.038	0.666±0.034	0.661±0.028	0.652±0.024	0.666±0.025	0.671±0.038
	G-M	0.739±0.033	0.743±0.031	**0.747±0.023**	0.744±0.031	0.740±0.028	0.737±0.023	0.728±0.020	0.741±0.021	0.745±0.032
	AUC	0.828±0.030	0.829±0.029	0.828±0.031	**0.831±0.027**	0.825±0.028	0.830±0.025	0.814±0.023	0.828±0.026	0.826±0.028
Balance	F_M	0.110±0.031	0.111±0.025	0.116±0.030	0.119±0.066	0.115±0.017	0.126±0.028	0.107±0.035	0.113±0.021	**0.149±0.029**
	G-M	0.442±0.067	0.447±0.053	0.456±0.061	0.435±0.105	0.454±0.030	0.476±0.059	0.437±0.079	0.451±0.045	**0.517±0.049**
	AUC	0.419±0.042	0.432±0.033	0.428±0.037	0.449±0.089	0.451±0.032	0.472±0.066	0.454±0.074	0.429±0.043	**0.533±0.051**
Liver	F_M	0.604±0.051	0.601±0.063	0.600±0.064	0.610±0.057	0.599±0.013	0.602±0.062	0.592±0.029	0.603±0.057	**0.613±0.063**
	G-M	0.636±0.048	0.632±0.051	0.631±0.054	0.640±0.048	0.555±0.057	0.632±0.051	0.621±0.033	0.633±0.050	**0.654±0.048**
	AUC	0.708±0.040	0.711±0.040	0.710±0.039	**0.713±0.039**	0.676±0.031	0.710±0.040	0.678±0.038	0.710±0.037	0.708±0.047
Wine	F_M	0.965±0.032	0.959±0.029	0.976±0.021	**0.961±0.031**	0.929±0.068	0.973±0.018	0.966±0.019	0.966±0.022	0.968±0.019
	G-M	0.967±0.031	0.964±0.026	0.977±0.019	**0.964±0.030**	0.936±0.069	0.974±0.018	0.970±0.017	0.970±0.021	0.970±0.018
	AUC	0.999±0.001	**0.999±0.002**	0.999±0.002	0.998±0.002	0.990±0.012	0.999±0.001	0.999±0.002	0.999±0.001	0.999±0.001
Breast	F_M	0.707±0.066	0.696±0.060	0.698±0.083	0.706±0.076	0.677±0.057	0.704±0.080	0.703±0.078	0.719±0.080	**0.719±0.093**
	G-M	0.762±0.087	0.754±0.078	0.752±0.094	0.765±0.091	0.720±0.079	0.760±0.092	0.763±0.091	0.769±0.097	**0.773±0.102**
	AUC	**0.899±0.031**	0.897±0.042	0.882±0.026	0.891±0.028	0.873±0.044	0.887±0.028	0.887±0.033	0.892±0.034	0.897±0.028
Libra	F_M	0.511±0.092	0.521±0.111	0.517±0.106	0.502±0.149	0.300±0.079	0.511±0.100	0.500±0.110	0.506±0.128	**0.541±0.098**
	G-M	0.676±0.079	0.684±0.096	0.675±0.091	0.667±0.124	0.519±0.083	0.674±0.091	0.662±0.095	0.670±0.112	**0.691±0.081**
	AUC	0.696±0.089	0.702±0.096	0.705±0.097	0.693±0.094	0.546±0.082	0.701±0.090	0.695±0.081	0.686±0.093	**0.715±0.101**
LEV	F_M	0.498±0.050	0.518±0.055	0.459±0.042	0.533±0.064	0.386±0.028	0.493±0.051	0.489±0.056	0.541±0.071	**0.564±0.075**
	G-M	0.675±0.034	0.745±0.052	0.712±0.042	0.698±0.045	**0.779±0.030**	0.736±0.041	0.718±0.052	0.721±0.046	0.712±0.053
	AUC	0.861±0.037	0.854±0.036	0.837±0.034	**0.873±0.045**	0.869±0.039	0.858±0.017	0.847±0.033	0.853±0.040	0.847±0.058
Iris	F_M	0.854±0.027	0.858±0.050	0.820±0.032	0.858±0.038	0.772±0.083	**0.879±0.047**	0.844±0.056	0.856±0.024	0.855±0.033
	G-M	0.906±0.022	0.910±0.037	0.878±0.028	0.910±0.029	0.829±0.080	**0.926±0.032**	0.899±0.041	0.909±0.017	0.906±0.024
	AUC	0.984±0.012	0.981±0.014	0.980±0.014	0.982±0.014	**0.986±0.014**	0.981±0.014	0.983±0.014	0.981±0.014	0.980±0.014
Heart	F_M	0.854±0.027	0.854±0.024	**0.857±0.029**	0.854±0.025	0.849±0.027	0.854±0.025	0.838±0.036	0.854±0.026	0.854±0.026
	G-M	0.864±0.026	0.864±0.023	**0.867±0.028**	0.864±0.024	0.858±0.026	0.864±0.025	0.849±0.035	0.864±0.024	0.864±0.025
	AUC	0.928±0.017	**0.929±0.017**	0.923±0.016	0.929±0.016	0.920±0.018	0.928±0.016	0.916±0.027	0.928±0.017	0.927±0.017
Glass	F_M	0.640±0.048	0.636±0.034	0.633±0.052	0.634±0.044	0.611±0.037	0.634±0.040	0.644±0.038	0.631±0.059	**0.658±0.045**
	G-M	0.715±0.050	0.708±0.039	0.703±0.062	0.706±0.053	0.647±0.072	0.710±0.041	0.709±0.045	0.707±0.060	**0.739±0.045**
	AUC	0.821±0.035	0.821±0.032	0.814±0.035	**0.824±0.035**	0.803±0.039	0.821±0.034	0.809±0.040	0.818±0.036	0.804±0.047
Haber	F_M	0.470±0.060	0.460±0.035	0.462±0.048	0.453±0.025	0.428±0.062	0.419±0.070	0.418±0.104	0.441±0.040	**0.483±0.060**
	G-M	0.618±0.049	0.611±0.025	0.617±0.039	0.604±0.020	0.581±0.048	0.569±0.055	0.573±0.095	0.597±0.032	**0.631±0.054**
	AUC	0.664±0.053	0.637±0.036	0.647±0.039	0.643±0.038	0.625±0.061	0.622±0.079	0.643±0.104	0.627±0.043	**0.684±0.069**
Eucal.	F_M	0.268±0.107	0.385±0.069	0.293±0.151	0.406±0.110	0.399±0.122	0.394±0.133	**0.432±0.070**	0.314±0.041	0.318±0.073
	G-M	0.411±0.105	0.537±0.056	0.436±0.126	0.550±0.096	**0.628±0.224**	0.534±0.101	0.578±0.053	0.468±0.033	0.460±0.055
	AUC	0.838±0.012	0.860±0.015	0.847±0.029	0.849±0.025	0.782±0.090	0.849±0.029	**0.880±0.008**	0.865±0.019	0.841±0.018
Heating	F_M	0.754±0.060	0.758±0.046	0.726±0.036	**0.761±0.030**	0.746±0.063	0.741±0.029	0.719±0.062	0.744±0.025	0.751±0.023
	G-M	0.845±0.035	0.851±0.027	0.833±0.018	0.850±0.017	**0.852±0.034**	0.843±0.015	0.807±0.048	0.838±0.008	0.844±0.009
	AUC	0.920±0.022	0.921±0.017	0.907±0.029	0.925±0.014	0.913±0.043	0.919±0.021	0.917±0.017	0.917±0.018	**0.931±0.015**
Seg.	F_M	0.614±0.016	0.618±0.014	0.613±0.031	0.621±0.014	0.583±0.031	**0.625±0.019**	0.617±0.012	0.613±0.021	0.621±0.013
	G-M	0.881±0.009	0.881±0.011	0.877±0.030	0.883±0.010	0.868±0.015	0.874±0.024	**0.885±0.008**	0.878±0.014	0.884±0.010
	AUC	0.931±0.013	0.932±0.012	0.877±0.027	0.932±0.012	0.908±0.033	**0.937±0.013**	0.924±0.017	0.932±0.014	0.927±0.017

Our results also indicate that compared to other methods, our method works better for datasets with higher imbalance ratio like Balance and LEV datasets. This is because in such datasets, minority instances are highly sparse meaning that there exists small minority clusters in the dataset. In other words, such datasets have high *within-class* imbalance. Therefore, it is very important to identify these small sub-clusters and emphasize them through oversampling as in cluster-based methods. Results also show that our method outperforms other cluster-based methods in most datasets. This is because, unlike the cluster-based methods, we adaptively determine sub-cluster sizes and oversample minority instances based on their distance to the majority class.

Table 3.6 Results for mean ranking of the 9 methods averaged over the 16 datasets

Classification method: SVM									
Measure	**Random**	**SMOTE**	**Borderline SMOTE**	**Safe-level SMOTE**	**SBC**	**Cluster SMOTE**	**CBOS**	**MWMOTE**	**A-SUWO**
F-measure	5.500	4.969	5.406	4.781	7.438	6.375	4.125	3.906	**2.625**
G-mean	5.688	5.406	5.219	4.938	6.500	5.750	4.875	3.938	**2.688**
AUC	5.906	5.781	5.844	5.344	6.125	5.438	4.313	3.750	**2.500**
Classification method: KNN									
Measure	**Random**	**SMOTE**	**Borderline SMOTE**	**Safe-level SMOTE**	**SBC**	**Cluster SMOTE**	**CBOS**	**MWMOTE**	**A-SUWO**
F-measure	5.594	5.125	5.750	4.156	8.000	4.250	5.500	4.438	**2.188**
G-mean	5.500	5.125	5.750	4.375	8.188	4.250	5.313	4.125	**2.375**
AUC	5.906	4.250	5.875	4.094	6.813	**2.500**	6.250	4.938	4.375
Classification method: Logistic Regression									
Measure	**Random**	**SMOTE**	**Borderline SMOTE**	**Safe-level SMOTE**	**SBC**	**Cluster SMOTE**	**CBOS**	**MWMOTE**	**A-SUWO**
F-measure	5.000	4.688	5.594	5.000	7.938	4.406	5.750	4.688	**1.938**
G-mean	4.500	4.250	5.281	5.750	8.250	5.031	5.000	4.375	**2.563**
AUC	5.031	3.500	6.594	3.563	8.188	3.938	5.875	4.813	**3.500**
Classification method: Linear Discriminant Analysis									
Measure	**Random**	**SMOTE**	**Borderline SMOTE**	**Safe-level SMOTE**	**SBC**	**Cluster SMOTE**	**CBOS**	**MWMOTE**	**A-SUWO**
F-measure	4.750	4.313	5.875	3.250	8.188	5.000	6.563	5.188	**1.875**
G-mean	4.313	4.313	5.875	3.688	8.188	5.625	6.125	4.688	**2.188**
AUC	4.750	**3.094**	6.375	3.188	7.875	4.313	6.500	4.500	4.406

Table 3.7 Results for Friedman's test

F-measure		G-mean		AUC	
Classification Method	P-Value	Classification Method	P-value	Classification Method	P-value
SVM	0.005956**	SVM	0.001138**	SVM	3.69E-05**
KNN	1.35E-06**	KNN	1.34E-06**	KNN	0.000148**
Logistic Regression	1.41E-06**	Logistic Regression	3.98E-06**	Logistic Regression	3.31E-07**
LDA	1.72E-09**	LDA	4.84E-08**	LDA	6.55E-07**

Table 3.8 Holm's test P-value - Control algorithm: A-SUWO

Classification model: SVM							
i	$\alpha_{0.10}$	F-measure		G-mean		AUC	
		Method	P-value	Method	P-value	Method	P-value
1	0.0125	SBC	4.12E-05**	SBC	9.06E-05**	SBC	3.34E-07**
2	0.0143	Cluster SMOTE	0.000781**	Random	0.000217**	Cluster SMOTE	5.38E-05**
3	0.0167	Random	0.000973**	Border SMOTE	0.000277**	Random	0.001492**
4	0.0200	SMOTE	0.002493**	SMOTE	0.000351**	Border SMOTE	0.002036**
5	0.0250	Border SMOTE	0.004471**	Cluster SMOTE	0.001207**	SMOTE	0.007747**
6	0.0333	Safe-Level SMOTE	0.010068*	Safe-Level SMOTE	0.001657**	Safe-Level SMOTE	0.012975*
7	0.0500	CBOS	0.011934*	CBOS	0.030607*	CBOS	0.060668*
8	0.1000	MWMOTE	0.098353*	MWMOTE	0.098353*	MWMOTE	0.092873*
Classification model: KNN							
i	$\alpha_{0.10}$	F-measure		G-mean		AUC	
		Method	P-value	Method	P-value	Method	P-value
1	0.0125	SBC	9.68E-10**	SBC	9.68E-10**	SBC	0.005911**
2	0.0143	Border SMOTE	0.000117**	Border SMOTE	0.000245**	CBOS	0.026404
3	0.0167	Random	0.000217**	Random	0.000624**	Random	0.056886
4	0.0200	CBOS	0.000312**	CBOS	0.001207**	Border SMOTE	0.060668
5	0.0250	SMOTE	0.001207**	SMOTE	0.002254**	MWMOTE	0.280638
6	0.0333	MWMOTE	0.010068*	Safe-Level SMOTE	0.019434*	SMOTE	0.551361
7	0.0500	Cluster-SMOTE	0.01658*	Cluster SMOTE	0.026404*	Safe-Level SMOTE	0.614273
8	0.1000	Safe-level SMOTE	0.02101*	MWMOTE	0.035351*	Cluster SMOTE	0.973596
Classification model: Logistic Regression							
i	$\alpha_{0.10}$	F-measure		G-mean		AUC	
		Method	P-value	Method	P-value	Method	P-value
1	0.0125	SBC	2.88E-10	SBC	2.13E-09**	SBC	6.45E-07**
2	0.0143	CBOS	4.12E-05	Safe-Level SMOTE	0.000497**	Border SMOTE	0.000699**
3	0.0167	Border SMOTE	7.96E-05	Border SMOTE	0.002493**	CBOS	0.007086**
4	0.0200	Random	0.000781	Cluster SMOTE	0.005391**	Random	0.056886
5	0.0250	Safe-Level SMOTE	0.000781	CBOS	0.005911**	MWMOTE	0.087622
6	0.0333	SMOTE	0.002254	Random	0.022694*	Cluster SMOTE	0.325689
7	0.0500	MWMOTE	0.002254	MWMOTE	0.030607*	Safe-Level SMOTE	0.474266
8	0.1000	Cluster SMOTE	0.005391	SMOTE	0.040681*	SMOTE	0.500000
Classification model: LDA							
i	$\alpha_{0.10}$	F-measure		G-mean		AUC	
		Method	P-value	Method	P-value	Method	P-value
1	0.0125	SBC	3.53E-11**	SBC	2.88E-10**	SBC	0.00017**
2	0.0143	CBOS	6.45E-07**	CBOS	2.38E-05**	CBOS	0.015293
3	0.0167	Border SMOTE	1.80E-05**	Border SMOTE	6.99E-05**	Border SMOTE	0.02101
4	0.0200	MWMOTE	0.000312**	Cluster SMOTE	0.000192**	Random	0.361286
5	0.0250	Cluster SMOTE	0.000624**	MWMOTE	0.004912**	MWMOTE	0.461433
6	0.0333	Random	0.001492**	Random	0.014093*	Cluster SMOTE	0.538567
7	0.0500	SMOTE	0.005911*	SMOTE	0.014093*	Safe-Level SMOTE	0.895934
8	0.1000	Safe-Level SMOTE	0.077790*	Safe-Level SMOTE	0.060668*	SMOTE	0.912378

3.4.1. Choosing Parameters for A-SUWO

A-SUWO requires four parameters to be defined: c_{thres}, NN, NS and k. In this section, we briefly explain how to choose appropriate values for these parameters. We also perform sensitivity analysis by running A-SUWO with different set of values for each parameter. The results are shown in Table 3.9.

- c_{thres}: This parameter was used to adjust the threshold for agglomerative clustering in Section 2.1. Larger values of c_{thres} will result in smaller clusters with larger sizes while smaller values of c_{thres} will result in larger clusters with smaller sizes. Its optimum value depends on the dataset. Generating large sized clusters as a result of large c_{thres} will increase the chance of over-generalization or generation of overlapping instances. On the other hand, generating small sized clusters will result in over-fitting or generation of less diverse synthetic instances. As can be seen from Table 3.9, a good range for c_{thres} is between 0.7 and 2. Actually, the G-mean for all values of c_{thres} larger than 3 is similar because all clusters are merged into one cluster.
- NN: This parameter determines the number of nearest neighbors used to assign weights to each minority instance. The weight for each minority instance depends on the average closeness factor to all NN-nearest neighbors from the majority class. If NN is selected as a large value, the algorithm assigns almost similar weights to all minority instances even if they are far away from the majority class. This is because the closeness factors are averaged over a large number of nearest neighbors. On the other hand, if NN is selected as a small value, then the weights could be very sensitive to noisy majority instances. As can be seen from Table 3.9, a reasonable value for NN could be selected between 3 and 7.
- NS: This parameter is used to find noisy instances. If all NS nearest neighbors of an instance are from a different class, then the instance is considered as noise in our method. If NS is selected as a large value, then the method is not able to find noisy instances whereas if NS is selected as a small value, the method will consider many of the valid instances as noise. As can be seen from Table 3.9, a reasonable value for NS can be between 3 and 7.

- k: This parameter determines the number of folds in our adaptive cluster sizing. The larger this parameter gets the more expensive the computation becomes as the classification method used in A-SUWO to determine the complexity of each cluster is required to run more times. As can be seen from Table 3.9, k can be selected between 2 and 5.

Table 3.9 Sensitivity analysis on A-SUWO parameters using SVM

Dataset	**G-mean measure for different values of c_{thres}**		**G-mean measure for different values of *NN***		**G-mean measure for different values of *NS***		**G-mean measure for different values of *k***	
	c_{thres}	**G-mean**	NN	**G-mean**	NS	**G-mean**	k	**G-mean**
Haberman	0.3	0.516±0.047	1	0.516±0.047	1	0.550±0.054	1	0.582±0.041
	0.7	0.577±0.017	2	0.577±0.017	2	0.584±0.012	2	**0.594±0.046**
	1.0	0.574±0.039	3	0.574±0.039	3	0.587±0.028	3	0.552±0.066
	1.5	0.541±0.019	4	0.541±0.019	4	0.602±0.018	4	0.559±0.067
	2	**0.611±0.040**	5	**0.611±0.040**	5	**0.604±0.016**	5	0.557±0.085
	2.5	0.584±0.063	7	0.584±0.063	7	0.585±0.015	6	0.550±0.062
	3	0.557±0.031	10	0.557±0.031	10	0.584±0.016	8	0.588±0.069
	8	0.557±0.031	15	0.557±0.031	15	0.585±0.019	10	0.583±0.019
Ecoli	0.3	0.936±0.026	1	0.941±0.018	1	0.939±0.015	1	0.941±0.024
	0.7	0.935±0.025	2	0.940±0.010	2	0.943±0.011	2	**0.942±0.029**
	1.0	**0.937±0.028**	3	0.944±0.012	3	0.941±0.010	3	0.938±0.027
	1.5	0.936±0.030	4	0.944±0.012	4	0.941±0.010	4	0.940±0.027
	2	0.933±0.026	5	**0.946±0.010**	5	0.942±0.011	5	0.936±0.026
	2.5	0.933±0.026	7	0.942±0.011	7	**0.944±0.012**	6	0.940±0.027
	3	0.933±0.026	10	0.939±0.009	10	0.942±0.010	8	0.938±0.026
	8	0.933±0.026	15	0.939±0.009	15	0.942±0.010	10	0.938±0.026
Wine	0.3	0.972±0.021	1	0.967±0.028	1	0.976±0.019	1	**0.985±0.015**
	0.7	0.970±0.021	2	0.986±0.021	2	0.976±0.019	2	**0.985±0.027**
	1.0	**0.975±0.017**	3	0.978±0.018	3	**0.979±0.015**	3	**0.985±0.015**
	1.5	0.967±0.015	4	0.978±0.018	4	**0.979±0.015**	4	0.981±0.012
	2	0.967±0.015	5	0.986±0.011	5	**0.979±0.015**	5	**0.985±0.015**
	2.5	0.969±0.015	7	**0.993±0.011**	7	0.976±0.019	6	0.978±0.026
	3	0.969±0.015	10	**0.993±0.011**	10	0.976±0.019	8	0.981±0.012
	8	0.969±0.015	15	**0.993±0.011**	15	0.976±0.019	10	0.978±0.026
Breast	0.3	0.695±0.034	1	0.690±0.058	1	0.723±0.031	1	0.736±0.114
	0.7	0.705±0.068	2	0.709±0.062	2	0.731±0.074	2	**0.747±0.034**
	1.0	**0.732±0.054**	3	0.722±0.056	3	**0.750±0.060**	3	0.706±0.072
	1.5	0.704±0.029	4	0.730±0.056	4	**0.750±0.060**	4	0.710±0.074
	2	0.692±0.038	5	**0.739±0.067**	5	0.742±0.049	5	0.725±0.065
	2.5	0.692±0.038	7	0.653±0.032	7	0.742±0.049	6	0.716±0.069
	3	0.680±0.048	10	0.654±0.034	10	0.742±0.049	8	0.726±0.062
	8	0.680±0.048	15	0.665±0.021	15	0.727±0.064	10	0.706±0.053
Libra	0.3	0.722±0.045	1	0.726±0.015	1	0.697±0.029	1	0.781±0.036
	0.7	0.751±0.028	2	0.790±0.024	2	0.714±0.063	2	0.782±0.013
	1.0	0.778±0.041	3	**0.799±0.013**	3	**0.763±0.024**	3	0.781±0.036
	1.5	**0.787±0.045**	4	**0.799±0.013**	4	0.754±0.037	4	0.781±0.036
	2	0.772±0.042	5	**0.799±0.013**	5	0.763±0.042	5	**0.799±0.013**
	2.5	0.772±0.042	7	0.781±0.037	7	0.731±0.089	6	0.735±0.049
	3	0.772±0.042	10	0.781±0.037	10	0.742±0.073	8	0.781±0.026
	8	0.772±0.042	15	0.781±0.037	15	0.742±0.073	10	0.790±0.023

3.5. Conclusions

In this paper, a new oversampling algorithm called Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) has been presented for imbalanced binary dataset classification. The advantages of A-SUWO are that it avoids generating overlapping synthetic instances by considering the majority instances when clustering minority instances; it determines the sub-cluster sizes adaptively using the standardized average error rate and cross-validation; it oversamples the sub-clusters by assigning weights to their instances to avoid over-generalization; and it does not ignore isolated sub-clusters. A-SUWO was tested on 16 publicly available datasets with different imbalance ratios and compared with other sampling techniques using different types of classifiers. Results show that our method performs significantly better compared to other sampling methods in most datasets and in larger datasets with higher imbalance ratio. As future work, the application of A-SUWO to multi-class classification problems will be studied.

Chapter 4

Cluster-based Weighted Oversampling for Ordinal Regression (CWOS-Ord)

In this paper, we propose a new oversampling method called Cluster-based Weighted Oversampling for Ordinal Regression (CWOS-Ord) to address the imbalanced dataset problem in ordinal regression. CWOS-Ord identifies clusters of data by first clustering all classes except the largest class using hierarchical clustering to ensure that all clusters are considered for oversampling. The set of all classes except the largest class will be referred to as the smaller classes. The largest class is not considered for oversampling. A modification of the traditional hierarchical clustering is presented that clusters the instances of smaller classes by considering other class instances to reduce overlapping between the generated instances and instances of other classes. Then, the final size to oversample the clusters depends on their complexity and initial size so that more synthetic instances are generated for more complex and smaller clusters while fewer instances are generated for less complex and larger clusters. Consequently, the clusters will not necessary have the same size after oversampling but in general, all the classes will be of equal size. This is particularly practical for ordinal regression as it contains multiple classes and oversampling the clusters of each class to the size of the largest majority cluster can result in a very large dataset. CWOS-Ord avoids over-generalization and mislabeling errors in terms of the ordinal label scale by oversampling instances of smaller classes based on their average Euclidean distance and rank differences to other class instances. Finally, well-known oversampling methods designed for two-class classification have been extended to the ordinal regression problem for performance comparison.

The contribution of this paper is three-fold. First, a modified agglomerative hierarchical clustering is introduced to reduce the generation of overlapping synthetic instances during oversampling. This is achieved by iteratively merging clusters of the same class while considering clusters of instances of other classes. Second, a new measure is proposed that quantifies the trade-off between cluster complexity and the initial size of the cluster. The new measure is used to determine the number of oversampled instances for each cluster. Finally, a new probability distribution is proposed that incorporates the distance as well as rank distance to other-class instances so that instances closer to the non-adjacent classes are oversampled more. As an additional contribution, existing oversampling methods for binary classification have been extended to ordinal regression.

In order to assess CWOS-Ord, extensive experiments have been conducted. The proposed CWOS-Ord method is tested on 11 publicly available datasets, and compared with five other techniques. Average Mean Absolute Error (AMAE), and Maximum Mean Absolute Error (MMAE) are used as the performance measures. The mean and standard deviation of the performance measures for each of the methods are determined using 3-fold stratified cross validation and repeated three times.

The remainder of this chapter is organized as follows. In the next section, a description of our extension of well-known oversampling methods to ordinal regression is presented for subsequent method comparison. In section 4.2, the proposed CWOS-Ord methodology is described. Section 4.3 presents the results and discussion while Section 4.4 provides the conclusions.

4.1. Oversampling for Ordinal Regression

In this section, we describe our extension of well-known oversampling methods for ordinal regression to enable subsequent comparison. Consider the ordinal regression problem where the outcome variable is a set of finite ordered ranks $r_{j=1,\ldots,m}$ with ordered relation $r_1 \prec r_2 \prec \cdots \prec r_m$. In ordinal regression, it is more important to distinguish classes with larger rank differences than classes closer to each other. The methods extended for ordinal regression include random oversampling, SMOTE [26], MWMOTE [27], and ADASYN [112]. The extension of random oversampling and SMOTE [26] for ordinal regression, which we will refer to as E-OR and E-SMOTE, respectively, consisted of applying the corresponding methods to ensure that all classes have the same number of instances. For E-OR, for each class in the dataset, instances were selected and duplicated until the class size is equal to the size of the largest class. For E-SMOTE, for each class, an instance was selected randomly. Then, one of its *k*-nearest neighbors in the same class was selected randomly and a synthetic instance was generated between them. The process is repeated until the class size is equal to the size of the largest class. Instances of the largest class are not oversampled.

In order to extend MWMOTE [27] and ADASYN [112] for ordinal regression, the ordering relationship among the classes was considered when assigning weights to minority instances. Details about the specific methods are out of the scope of this paper and can be obtained from the corresponding references. For the extension of MWMOTE, which we will refer as E-MWMOTE, for each class j, Class Borderline Instances (CBIs) and Other-classes Borderline Instances (OBIs) are found. The other-class instances are all instances except the instances that belong to the class j**.** In order to find the OBIs, for each instance in class j, their k_1-nearest neighbors among the instances from all other classes are found. Then, for each OBI, its k_2-nearest neighbors among the instances of class j are found to obtain the CBIs. After finding OBIs and CBIs for class j, the next step is to assign weights to CBIs based on their average Euclidean distance and their rank differences to OBIs. For the ith CBI in class j with the feature vector x_{ij}, its Euclidean distance $d(x_{ij}, y_{lj})$ to the lth OBI with feature vector y_{lj} is determined. The distance $d(x_{ij}, y_{lj})$ is normalized by dividing it over the number of features D to make it robust to datasets with different number of features. We call the normalized distance as $d_D(x_{ij}, y_{lj})$. Then, the closeness factor $C_f(x_{ij}, y_{lj})$ between x_{ij} and y_{lj} is defined. The ordering relationship is considered by multiplying the rank difference in the original equation of MWMOTE for closeness factor. As can be seen from Figure 4.1, the instances closer to the non-adjacent classes are assigned higher weights and hence have a higher chance to be oversampled.

The new equation for the closeness factor $C_f(x_{ij}, y_{lj})$ for E-MWMOTE is as follows:

$$C^{(1)}{}_f(x_{ij}, y_{lj}) = \frac{f\left(\frac{1}{d_D(x_{ij}, y_{lj})}\right)}{C_f(max)} * \left|r_{x_{ij}} - r_{y_{lj}}\right| \qquad (4.1)$$

where f is a cutoff function that prevents $\frac{1}{d_D(x_{ij}, y_{lj})}$ from becoming extremely large in the case when the two instances x_{ij} and y_{lj} become too close to each other, $C_f(max)$ is the largest value $f(x)$ can reach, and the term $\left|r_{x_{ij}} - r_{y_{lj}}\right|$ is the rank difference among x_{ij} and y_{lj}. The first term in equation 4.1 forces higher weights to the instances in class j that are closer to the instances of other classes whereas the second term gives higher weights to the instances with larger rank difference. In other words, the instances in class j that are closer to instances of non-adjacent classes will have higher weights.

Using equation 4.1, more synthetic instances are generated for instances of class j that overlap with non-adjacent classes. The advantage of this new equation is that it can help move the decision boundary towards the class j and hence avoid overfitting. Using the closeness factor, a weight will be assigned to each instance in class j and then the weights are converted into a probability distribution $P(x_{ij})$ by dividing each weight by the summation of all weights. The probability distribution is used to take samples from instances of class j. Therefore, more synthetic instances are generated using the instances with larger closeness factor.

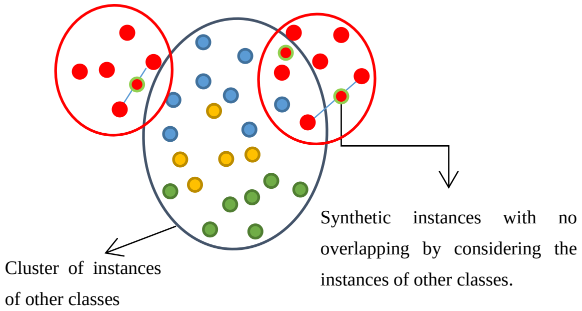

Synthetic instances with no overlapping by considering the instances of other classes.
Cluster of instances of other classes

Figure 4.1 Clustering of class with red points. The instances of other classes (blue, yellow, green) were also considered while clustering the instances of class with red points.

For the extension of ADASYN, E-ADASYN, for each class j, for the ith example with the feature vector x_{ij}, its k-nearest neighbors among all instances are found. Then, a weight ratio $\rho_{ij}$is assigned to x_{ij} using (2), which has been modified to reflect the ordinal relationship among classes. The weight ratio ρ_{ij} is:

$$\rho^{(1)}{}_{ij} = \frac{\sum_{h=1}^{m}\left(\Delta_{ijh} * \left|r_h - r_{x_{ij}}\right|\right)}{K} \qquad (4.2)$$

where Δ_{ijh} is the number of instances in the k-nearest neighbors of x_{ij} that belong to the hth class. r_h is the rank of the hth class and $r_{x_{ij}}$is the rank of the instance x_{ij}. K is a constant that can be deleted from (2) as ρ_{ij} will be normalized in the following steps. The modified formula will give higher weights to the instances that have more non-adjacent instances in their neighbors. Later ρ_{ij} is converted to probability distribution to take samples from instances of class j.

4.2. Cluster-based Weighted Oversampling for Ordinal Regression (CWOS-Ord)

In this section, a new oversampling method CWOS-Ord that is specifically designed for the imbalanced dataset problem in ordinal regression is presented. The objective is to balance all the classes by making all the classes of equal size. To achieve this, we consider all classes except the largest class as the smaller classes and the largest class as the only majority class. CWOS-Ord identifies clusters of data by clustering the instances of the smaller classes using a One-Versus-All (OVA) semi-unsupervised hierarchical clustering approach. The new clustering approach iteratively forms clusters for each class while avoiding clusters of other classes in between. Then, the size to which each cluster will be oversampled is determined using a new measurement based on cluster's complexity and initial size. In order to avoid over-generalization and mislabeling errors caused by ordinal label scale, CWOS-Ord assigns weights to instances of smaller classes based on their closeness to instances of other classes and their rank differences. CWOS-Ord consists of two main steps: (1) OVA Semi-Unsupervised Hierarchical Clustering, and (2) Synthetic Instance Generation. In the first step, the smaller classes are individually clustered following a variation of the Agglomerative Complete-Linkage Hierarchical Clustering method [113]. The final size of each cluster is determined based on its complexity and initial size. In the Synthetic Instance Generation step, a new weighting system is proposed to assign weights to minority instances for the generation of synthetic instances. The following sub-sections provide the details of CWOS-Ord.

4.2.1. One-versus-All Semi-Unsupervised Hierarchical Clustering for Ordinal Classes

In general, there are two approaches for generating synthetic instances. The first one is to generate a new instance between a candidate instance and one of its NN-nearest neighbors [26, 38, 112]. The second approach is to generate a new instance between a candidate instance and one of its neighbors from the same cluster [27]. Both approaches can lead to the generation of synthetic instances that overlap with other class instances. In the first approach, some of the NN-nearest neighbors may be far from the candidate instance whereas in the second approach, clusters from different classes may overlap. Overlapping synthetic instances can deteriorate the performance of the classifiers significantly [27, 106].

To reduce the generation of overlapping synthetic instances, we previously designed a semi-unsupervised hierarchical clustering algorithm as presented in [25] for binary classification. In this algorithm, any two minority clusters that are nominated to be merged are checked in each iteration. If a majority cluster exists between them, the minority clusters are not merged. Otherwise, the two nominated clusters are merged if their distance is less than a pre-defined threshold. For ordinal regression, the semi-unsupervised hierarchical clustering algorithm has been designed in a One-Versus-All (OVA) framework to check overlapping of instances of each class with instances of other classes. In other words, for each class, the algorithm checks whether a cluster from any of the other classes exists between the nominated clusters.

Before clustering, noisy instances are identified for both classes using the method suggested by [38] and removed from the dataset. For each instance, NS-nearest neighbors are found. If all the NS-nearest neighbors belong to the other non-adjacent classes, then the instance is considered as noise and removed from the dataset because it indicates that it is surrounded by instances of the other classes.

In our algorithm, instances of all classes are clustered except for the largest-sized class. For each class j to be oversampled, the OVA Semi-Unsupervised clustering algorithm starts by first clustering the instances of all other classes except the instances of class j using hierarchical clustering. This results in m_j clusters $Cother_{i=1,\dots,m_j}$. $Cother_{i=1,\dots,m_j}$ is the set of clusters for the instances of classes other than class j. Then, for each class j, the proposed OVA semi-unsupervised hierarchical clustering algorithm is applied as follows.

Assuming that the class has N_j instances, for each desired class j to be oversampled:

1) Assign each instance to a separate cluster. This will result in N_j clusters of size one $B_j = \{Cdes_{\tau=1,\dots,N_j}\}$.
2) Identify the two clusters say $Cdes_a$ and $Cdes_b$ with the lowest Euclidean distance between them. Let their distance be represented by δ.
3) Find other-class clusters, say $Cother_{i\in A_j}$ with Euclidean distance to $Cdes_a$ and $Cdes_b$ less than δ. A_j is the set of other-class indices with such property.
4) If $A_j \neq \emptyset$, then, there exists an other-class cluster between $Cdes_a$ and $Cdes_b$ and hence, they should not be merged. The distance between $Cdes_a$ and $Cdes_b$ will be set to a large number to avoid being considered for merging again.
5) Else, $Cdes_a$ and $Cdes_b$ are merged into one cluster $Cdes_c$. This will result in one less member in B_j.
6) Finally, the Euclidean distance between the newly formed cluster of the desired class $Cdes_c$ and existing cluster is recalculated. Steps 2 to 6 are repeated until the Euclidean distance between the closest clusters is larger than a threshold T_j. At the end, we will have n_j minority clusters for class j.

In contrast with the clustering algorithm developed in our previous work [25], the proposed OVA semi-unsupervised hierarchical clustering algorithm checks whether the two clusters of the desired class $Cdes_a$ and $Cdes_b$ contain part of other-class clusters. In order to have a good estimate of T_j for each class, the median Euclidean distance $d_{med,ij}$ of each instance i in class j to all other instances of class j is determined. Then, $d_{avg,j}$ is defined as the average $d_{med,ij}$ over all instances in class j. Therefore, T_j can be estimated as follows:

$$T_j = d_{avg,j} * C_{thresh} \qquad (4.3)$$

where C_{thresh} is a user-defined constant parameter used for all classes. Larger values of C_{thresh} will result in smaller number of clusters with larger sizes whereas smaller values of C_{thresh} will lead to larger number of clusters with smaller sizes. Large-sized clusters will increase the chance of over-generalization or generation of overlapping instances while small-sized clusters will result in over-fitting or generation of less diverse synthetic instances.

In the next step, g_{hj} synthetic instances will be generated for each cluster *h* in each class *j* with the initial size of q_{hj} based on the cluster's complexity and initial size. Therefore, each cluster *h* in each class *j* will have $S_{hj} = g_{hj} + q_{hj}$ instances after oversampling. Let's assume the largest-sized class has L instances and that all classes will have similar size at the end of oversampling. Then, for each class j, $G_j = L - Q_j$ new instances should be generated where $Q_j$is the initial size of the class *j*.

In this paper, a new measurement is proposed to determine the final size of each cluster based on its complexity and initial size. In order to determine cluster complexity, for each instance *i* in each cluster *h* of class *j*, its *k*-nearest neighbors among all instances are found. Then $\rho^{(i)}{}_{hj}$, the average rank difference of instance *i* to all its *k*-neighbors is calculated. $\rho^{(i)}{}_{hj}$ is an indicator of complexity for instance *i* because higher $\rho^{(i)}{}_{hj}$ means instance *i* is surrounded by many non-adjacent instances. $\rho^{(i)}{}_{hj}$ is then averaged over all instances of cluster *h* to denote the average *k*-nearest neighbors' label differences as an indicator of cluster complexity using the following formula:

$$\bar{\rho}_{hj} = \frac{\sum_{i=1}^{q_{hj}} \rho^{(i)}{}_{hj}}{q_{hj}} \qquad (4.4)$$

where q_{hj} is the initial size of the hth cluster of the jth class. Using this equation, the clusters that are surrounded by instances of non-adjacent classes are considered as more complex while clusters surrounded by instances of the same class or adjacent classes are considered as less complex.

Finally, we can determine g_{hj}, the number of synthetic instances to be generated for each cluster, as a factor of both cluster complexity and initial size:

$$g_{hj} = G_j * \frac{\bar{\rho}_{hj} * \frac{1}{q_{hj}{}^{\alpha}}}{\sum_{h=1}^{n_j} \bar{\rho}_{hj} * \frac{1}{q_{hj}{}^{\alpha}}} \qquad (4.5)$$

where α defines a trade-off between the complexity and initial size of each cluster. As α increases, the smaller clusters are oversampled more, while as α decreases, more complex clusters are oversampled more. Equation 4.5 indicates that more instances are generated for clusters with higher complexity and smaller initial size. In other words, more complex and smaller clusters are emphasized so that they are not ignored for oversampling.

4.2.2. Synthetic Instance Generation

In this stage, weights are assigned to instances of smaller classes for subsequent oversampling. These weights are assigned by considering the other-class instances to reduce over-generalization. In CWOS-Ord, new synthetic instances are generated between the original instances and their NN-nearest neighbors in the same class given that they are also from the same cluster. The reason to restrict the NN-nearest neighbors to be in the same cluster is to avoid selecting a NN-nearest neighbor that is far away from the selected instance and that belongs to another cluster. This way, the chance of generating synthetic instances that overlap with instances from the other class is reduced. The synthetic instance generation approach of CWOS-Ord is repeated for instances of all classes except from the largest-sized class.

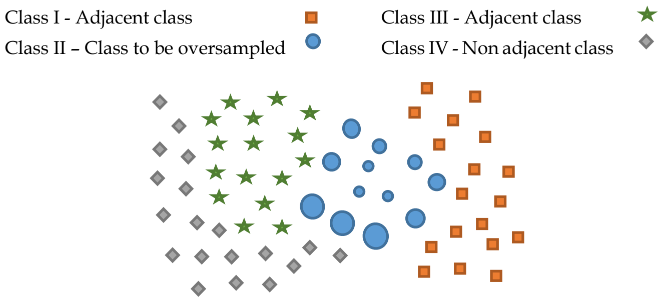

Class I - Adjacent class ■
Class II – Class to be oversampled ●
Class III - Adjacent class ★
Class IV - Non adjacent class ◆

Figure 4.2 Assigning weights for oversampling. Larger blue circles indicates larger weights to be assigned to them. Instances closer to non-adjacent classes are assigned higher weights

For each cluster h of class j, we first assign weights to each instance in the cluster based on its Euclidean distance and rank difference to other-class instances. For the ith instance x_{ihj} in the hth cluster of class j, we find its k nearest neighbors using the Euclidean distance to all other instances $y_{ihj(v)}$ and record the distance $d\left(x_{ihj}, y_{ihj(v)}\right)$, where $v = 1, \ldots, k$ are the indices of the k nearest neighbors. We divide the distance $d\left(x_{ihj}, y_{ihj(v)}\right)$ by the number of features D. Therefore, we have:

$$d_D\left(x_{ihj}, y_{ihj(v)}\right) = \frac{d\left(x_{ihj}, y_{ihj(v)}\right)}{D} \qquad (4.6)$$

$d_D\left(x_{ihj}, y_{ihj(v)}\right)$ is more robust to datasets with different number of features. Later, we define $\Gamma\left(x_{ihj}, y_{ihj(v)}\right)$ as the closeness factor between x_{ihj} and $y_{ihj(v)}$.

$$\Gamma\left(x_{ihj}, y_{ihj(v)}\right) = f_{hj}\left(\frac{1}{d_D\left(x_{ihj}, y_{ihj(v)}\right)}\right) \qquad (4.7)$$

where f_{hj} is a cutoff function for cluster h that prevents $\frac{1}{d_D\left(x_{ihj}, y_{ihj(v)}\right)}$ from becoming extremely large in the case when the two instances x_{ihj} and $y_{ihj(v)}$ become too close to each other. Therefore, f_{hj} is defined as:

$$f_{hj}\,(x) = \begin{cases} x & if\ x \leq TH_{hj} \\ TH_{hj} & otherwise \end{cases} \qquad (4.8)$$

TH_{hj} is the largest value $f_{hj}\,(x)$ can reach. In our method, TH_{hj} is determined for each cluster C_{hj} automatically. This is achieved by finding the Euclidean distance of all instances x_{ihj} in each cluster to their closest other-class instance $y_{ihj(1)}$ and then determining $f\left(\frac{1}{d_D(x_{ihj},y_{ihj(v)})}\right)$. TH_{hj} is then set as the average of $f\left(\frac{1}{d_D(x_{ihj},y_{ihj(v)})}\right)$.

$$TH_{hj} = \sum_{j=1}^{R_{hj}} f\left(\frac{1}{d_D(x_{ihj},y_{ihj(1)})}\right) \qquad (4.9)$$

where R_{hj} is the number of instances in cluster C_{hj}.

Determining TH_{hj} automatically is a critical step in our method as our weighting algorithm runs for each cluster separately and each cluster requires a specific threshold. Then, the weights $W(x_{ihj})$ are determined based on the Euclidean distance of x_{ihj} from all k nearest neighbors. In this step, we impose the ordering relationship among instances of different classes.

$$W(x_{ihj}) = \sum_{v=1}^{k}\left(\Gamma\left(x_{ihj}, y_{ihj(v)}\right) * \left|r_{x_{ihj}} - r_{ihj(v)}\right|\right) \qquad (4.10)$$

In this equation, the instances in cluster h of class j that are closer to instances of non-adjacent classes will have higher weights as can be seen in Figure 4.2.

Finally, the weights are converted into a probability distribution $P(x_{ihj})$ by dividing each weight by the summation of all weights as follows:

$$P(x_{ihj}) = \frac{W(x_{ihj})}{\sum_{i=1}^{R_h} W(x_{ihj})} \qquad (4.11)$$

In the last step, each cluster $C_{hj}, h = 1, \dots, n_j$ will be oversampled so that they will have size S_{hj}. For oversampling, an instance *a* in each class is selected by sampling from the probability distribution $P(x_{ihj})$. Then, one of its NN-nearest neighbors b is randomly selected and a new instance c is generated between a and b as follows:

$$c = \beta a + (1 - \beta)b \qquad (4.12)$$

where β is a random number between 0 and 1. In terms of complexity, the most time-consuming part of the CWOS-Ord algorithm is the OVA semi-unsupervised hierarchical clustering, which has a complexity of *O(JN^3)*, where *J* is the number of classes and *N* is the size of the dataset. By implementing the hierarchical clustering using an optimally efficient method [39], the complexity can be reduced to *O(JN^2)*. The proposed CWOS-Ord algorithm is described through the following algorithm:

```
Algorithm 1 – Cluster-based Weighted Oversampling for Ordinal Regression (CWOS-Ord)
Inputs:
    - Original features: The features of original dataset that should be oversampled.
    - Original labels: The labels of original dataset that should be oversampled.
    - $C_{thresh}$: The coefficient to tune the threshold for the hierarchical clustering.
    - $NN$: Number of nearest neighbors to be found for each instance to determine the weights and cluster complexity.
    - $NS$: Number of nearest neighbors used to identify noisy instances.
    - $\alpha$: Parameter to tune the trade-off between complexity and initial size to determine cluster size.

Outputs:
    - Final features: The features of the oversampled dataset.
    - Final labels: The labels of the oversampled dataset.

Procedure:
  For each class $j$ except the largest-size class:
  i. Hierarchical Clustering
    1. Remove noisy instances from the dataset.
    2. Determine $T_j$.
    3. Cluster all instances of other classes, which will result in $m$ clusters $Cother_{i=1,\dots,m_j}$.
    4. Assign each instance of class $j$ to a separate cluster.
    5. Find the two closest clusters $Cdes_a$ and $Cdes_b$.
    6. Check if there is any overlapping other-class cluster between $Cdes_a$ and $Cdes_b$.
    7. If yes, set their distance to infinity and return to step 5. Else, merge $Cdes_a$ and $Cdes_b$ into one cluster $Cdes_c$.
    8. Repeat steps 5 to 7 until the Euclidean distance between the closest sub-clusters is less than a threshold $T_j$.
    9. Determine cluster sizes $S_{hj}$ for all clusters of class $j$ using equation 4.5.
  ii. Synthetic Instance Generation
    a) Determine the probability distribution for instances within each cluster of class $j$:
    - For each cluster $h = 1, 2, \dots, n_j$
        1. For all instances $x_{ihj}$ in cluster $C_j$, find the $NN$-nearest neighbors among other-class instances.
        2. Determine $W(x_{ihj})$ for each instance in cluster $C_h$ using equation 4.6 – 4.10 and by estimating $TH_{hj}$.
        3. Transform the weights to a probability distribution $P(x_{ihj})$ using equation 4.11.
    b) Oversample instances of class $j$:
    - For each cluster $h = 1, 2, \dots, n_j$
        1. Select an instance $a$ in cluster $h$ by sampling from probability distribution $P(x_{ihj})$.
        2. Select one of its $NN$ nearest neighbors $b$ randomly given that they belong to the same cluster.
        3. Generate a new synthetic instance between $a$ and $b$ using equation 4.12.
        4. Repeat steps 1 to 3 until the cluster size reaches $S_{hj}$.
```

4.3. Results

The performance of CWOS-Ord was tested on 11 publicly available datasets and compared with five other oversampling methods: 1) Extension of random oversampling (E-RO), 2) Extension of SMOTE (E-SMOTE) [26], 3) Extension of MWMOTE (E-MWMOTE) [27], 4) Extension of ADASYN (E-ADASYN) [112], and 5) Graph-based Oversampling for Ordinal regression via Shortest Path (OGO-SP) [114]. OGO-SP was selected among the three versions of OGO due to its superior results compared to the other two versions as demonstrated in [114]. The following performance measures were used: Average Mean Absolute Error (AMAE), and Maximum Mean Absolute Error (MMAE) [115, 116], which are suitable for imbalanced dataset problems in ordinal regression.

Table 4.1 Description of the datasets

#	Dataset	# of features	# of instances	# of classes	# of instances in each class	Imbalanced Ratio
1	Stock	9	950	10	48/110/108/119/168/104/104/103/64/22	2.2:5.0:4.9:5.2:**7.6**:4.9:4.9:4.8:2.9:1.0
2	Auto	7	392	5	91/131/101/59/10	9.1:**13.1**:10.1:5.9:1.0
3	Machine	6	209	4	152/27/13/17	**11.7**:2.1:1.0:1.3
4	Balance	4	625	3	288/49/288	**5.9**:1:**5.9**
5	ESL	4	488	5	14/38/351/62/23	1:2.**7:25**.1:4.4:1.4
6	Heating	8	768	8	20/265/112/51/119/85/82/34	1.0:**13.3**:5.6:2.6:6.0:4.3:4.1:1.7
7	ERA	4	1000	9	92/142/181/172/158/118/88/31/18	5.1:7.9:**10.1**:9.6:8.8:6.6:4.9:1.7:1.0
8	Wisconsin	32	194	5	67/41/43/24/19	**3.5**:2.2:2.3:1.3:1.0
9	Triazines	60	186	4	17/26/86/57	1.0:1.5:5.1:**3.4**
10	Wine Quality Red	12	1599	6	10/53/681/638/199/18	1.0:5.3:**68.1**:63.8:19.9:1.8
11	New Thyroid	5	215	3	30/150/35	1.0:**5.0**:1.2

AMAE measures the average of Mean Absolute Error (MAE) independently across classes and is particularly suitable for imbalanced datasets. MAE is the average deviation of the prediction from the observed values.

$$MAE_j = \frac{1}{N_j} \sum_{i=1}^{n} |p_i - o_i| \qquad (4.13)$$

where p_i is the predicted value and o_i is the observed value. MAE is not used directly as it is not suitable for imbalanced datasets. Instead, we use AMAE that is described as follows:

Table 4.2 MMAE results for the oversampling methods on the 11 datasets using OR-EBC

Dataset	NO	E-OR	E-SMOTE	E-ADASYN	E-MWMOTE	OGOSP	CWOS-ORD
Auto	1.000±0.000	0.603±0.168	**0.550±0.200**	0.639±0.244	0.499±0.124	0.585±0.115	0.533±0.196
ERA	1.998±0.094	1.980±0.018	1.896±0.171	1.943±0.072	1.883±0.076	1.927±0.053	**1.822±0.096**
Balance	**0.100±0.053**	0.179±0.020	0.179±0.022	0.177±0.015	0.181±0.021	0.212±0.016	0.161±0.019
ESL	1.000±0.125	0.481±0.130	0.433±0.147	0.428±0.153	0.430±0.031	0.435±0.091	**0.417±0.072**
Stock	0.992±0.065	0.740±0.075	0.714±0.084	**0.666±0.078**	0.736±0.071	0.813±0.070	0.726±0.107
Wisconsin	2.177±0.216	**1.509±0.148**	1.593±0.126	1.587±0.145	1.563±0.165	1.611±0.135	1.548±0.105
triazines	2.000±0.000	1.789±0.141	1.733±0.205	**1.706±0.200**	1.794±0.142	1.739±0.211	1.728±0.203
Wine-Red	2.100±0.141	1.319±0.398	1.237±0.230	1.301±0.424	1.219±0.257	**1.212±0.283**	1.293±0.152
Machine	0.770±0.196	0.581±0.151	0.576±0.176	0.581±0.151	0.688±0.231	0.634±0.167	**0.548±0.279**
Heating	0.880±0.435	0.654±0.339	0.685±0.348	0.647±0.332	0.689±0.356	0.566±0.446	**0.520±0.406**
NewThyroid	0.399±0.031	0.161±0.053	0.161±0.053	0.147±0.092	0.140±0.040	**0.127±0.046**	**0.127±0.046**

Table 4.3 AMAE results for the oversampling methods on the 11 datasets using OR-EBC

Dataset	NO	E-OR	E-SMOTE	E-ADASYN	E-MWMOTE	OGOSP	CWOS-ORD
Auto	0.389±0.031	0.304±0.056	0.289±0.045	0.321±0.058	**0.290±0.049**	0.312±0.042	0.313±0.052
ERA	1.449±0.136	1.371±0.063	1.302±0.079	1.301±0.089	1.316±0.091	1.366±0.039	**1.267±0.096**
Balance	**0.057±0.023**	0.103±0.011	0.102±0.007	0.101±0.007	0.104±0.011	0.126±0.009	0.093±0.008
ESL	0.606±0.086	0.269±0.053	0.255±0.043	0.256±0.035	0.297±0.059	0.286±0.082	**0.253±0.052**
Stock	0.465±0.025	0.353±0.044	0.349±0.045	**0.349±0.042**	0.360±0.039	0.379±0.045	0.351±0.034
Wisconsin	1.205±0.066	1.110±0.079	1.138±0.069	1.129±0.089	1.115±0.087	1.118±0.088	**1.106±0.062**
triazines	0.995±0.012	0.951±0.071	0.940±0.116	0.950±0.128	0.985±0.107	0.941±0.070	**0.914±0.030**
Wine-Red	1.098±0.044	0.816±0.025	0.805±0.045	0.810±0.016	0.788±0.041	0.786±0.035	**0.747±0.019**
Machine	0.446±0.138	0.314±0.056	0.310±0.066	0.313±0.055	0.326±0.068	0.342±0.058	**0.291±0.120**
Heating	0.482±0.240	0.260±0.133	0.259±0.132	0.258±0.132	0.272±0.140	**0.218±0.174**	0.235±0.187
NewThyroid	0.233±0.042	0.074±0.034	0.076±0.030	0.074±0.060	0.073±0.024	**0.067±0.024**	0.075±0.042

$$AMAE = \frac{1}{J}\sum_{j=1}^{J} MAE_j \qquad (4.14)$$

MMAE is the maximum MAE among all classes and is a suitable measure for both ordinal regression and imbalanced dataset problems. This is because it represents the individual performance for the worst ordered class in such a way that a low MMAE represents a low error for all classes of the problem (including minority ones):

$$MMAE = \max\left\{MAE_j; j \in \{1, \dots, J\}\right\} \qquad (4.15)$$

Table 4.4 Results for mean ranking of the 7 methods averaged over the 11 datasets

Measure	NO	E-OR	E-SMOTE	E-ADASYN	E-MWMOTE	OGOSP	CWOS-ORD
MMAE	6.455	4.682	3.636	3.409	3.818	4.136	**1.864**
AMAE	6.455	4.182	3.273	3.636	4.364	4.000	**2.091**

Table 4.5 Results for Friedman’s Test

P-Value for MMAE	P-value for AMAE
0.0001172***	0.0003814***

Table 4.6 Holm’s test P-value - Control algorithm: CWOS-ORD

i	$\alpha_{0.10}$	MMAE		AMAE	
		Method	P-value	Method	P-value
1	0.0167	No	3.11E-07**	No	1.08E-06**
2	0.0200	E-OR	0.001108625**	E-MWMOTE	0.006806**
3	0.0250	OGOSP	0.006806454**	E-OR	0.011606*
4	0.0333	E-MWMOTE	0.016923311*	OGOSP	0.019107*
5	0.0500	E-SMOTE	0.027145425*	E-ADASYN	0.046695*
6	0.1000	E-ADASYN	0.046695338*	E-SMOTE	0.099745*

Measures like MAE or accuracy were not considered in our experiments. MAE is not suitable for imbalanced datasets because datasets with high MAE values for the minority classes may have very low MAE as a whole. On the other side, Accuracy is not a good performance measure for ordinal regression because it does not consider the difference of errors in the ranks.

The techniques were implemented using Matlab on a workstation with 64-bit Operating System, 16.00 GB RAM, and 3.60 GHz CPU. Table 4.1 contains detailed information regarding all 11 datasets from the University of California at Irvine (UCI) repository with different imbalance ratios as high as 1:68. Imbalance ratio is defined as the proportion of instances in the majority classes with respect to instances of minority classes. In Table 4.1, the imbalance ratio for all classes with regard to the smallest class is shown in the last column and the largest imbalance ratio for each dataset is shown in bold. Most of the datasets in Table 4.1 are specific for ordinal regression. However, some of them (Wisconsin, Stock, Machine, Triazines and Auto) are not originally for ordinal regression and were converted into ordinal classification by discretizing the outcome variable into equal-sized bins [13]. The mean and standard deviation of MMAE and AMAE for each method on the 11 datasets are determined by using stratified 3-fold cross validation and repeating the experiment 3 times. Repeating the experiments several times was performed to address the randomness effects on the results.

Ordinal Regression by Extended Binary Classification (OR-EBC) [117] is used to evaluate the oversampling methods because of its fast training speed and good generalization performance. OR-EBC has a decomposition framework that first converts the ordinal regression problem into a set of binary problems. Then, it solves all the binary problems jointly by proposing a new formulation for SVM to obtain a single binary classifier. Finally, it converts the binary outputs to ranks. The radial kernel is used for SVM and the parameters for OR-EBC and the oversampling methods are optimized over a small set of values using cross-validation. In particular, the parameters for both cost C and gamma γ are selected among the values (10^{-1} ,10^{0} ,10^{1}). k = 5 nearest neighbors is used for all methods that require a number of neighbors to be selected as suggested by other works [27, 114]. For the graph based method [114] a = 2, b = 0.15 where a and b are the parameters for the gamma distribution used to generate β in equation 4.10 as suggested by the paper. For our method (CWOS-ORD), C_{thresh} was selected among (1, 2, 3) and α was selected among the values (0.1, 0.5, 1, 1.5)

The MMAE and AMAE results for CWOS-Ord and the other five methods on 11 real datasets and classified using OR-EBC are shown in Tables 4.2 and 4.3, respectively. The best measures are shown in bold. It can be observed from these two tables that when no oversampling is performed (the first column in Tables 4.2 and 4.3), the results are clearly inferior to all oversampling methods for all datasets. Random oversampling also clearly does not provide good results because, as mentioned earlier, it leads to overfitting.

The mean ranking for each method in terms of MMAE and AMAE for all tested datasets are shown in Table 4.4. The method that performed the best is assigned a ranking of 1 while the method that performed the worst is assigned a ranking of 7. As can be seen from the table, our method has the lowest ranking in terms of both measurements. In order to verify whether the results obtained by our method are statistically significant to other methods, the Friedman's test followed by Holm's test were applied. Friedman test is a non-parametric statistical test and is very similar to the repeated-measures ANOVA. The null hypothesis is that all oversampling methods are performing similarly in mean rankings. The results for the Friedman test are shown in Table 4.5. It can be observed that, for both measures, there exists enough evidence at α = 0.05 to reject the null hypothesis. This means that based on the current datasets, the oversampling methods are not performing similarly.

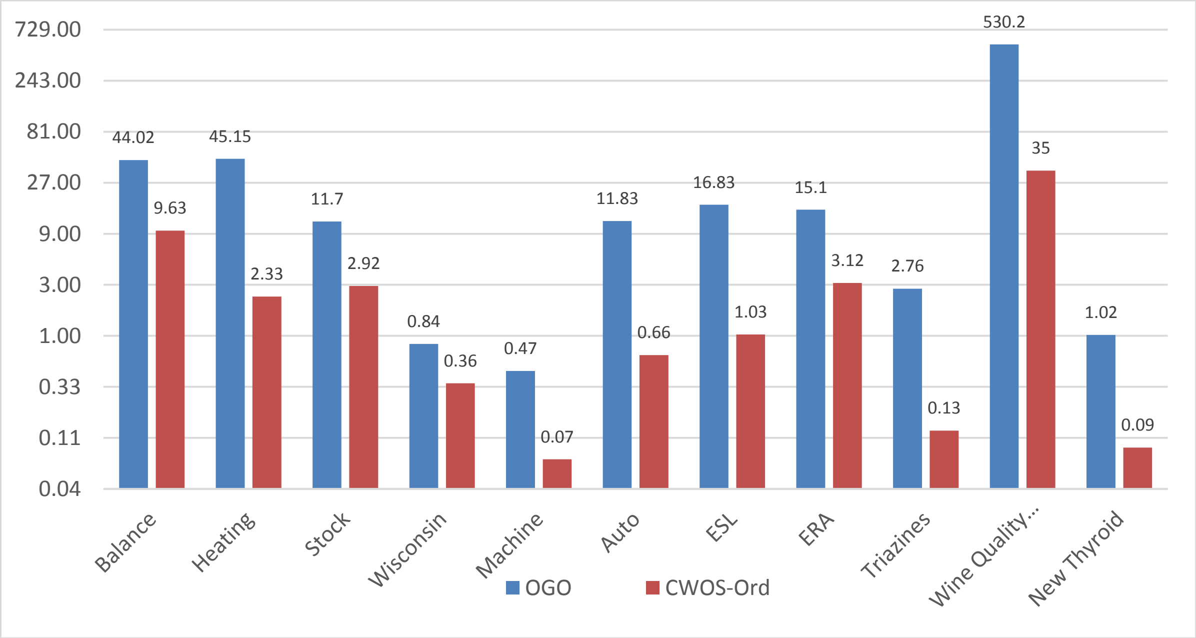


Figure 4.3 Timewise comparison of CWOS-ORD and OGO in logarithmic scale (in seconds).

Rejection of the null hypothesis for both performance measures means a post-hoc test can be followed. As the post-hoc test, Holm's test was used where our method was considered as the control method. Holm's test is the non-parametric equivalent to multiple t-test in which α is adjusted in a step-down procedure to compensate for multiple comparisons. Table 4.6 shows the adjusted α and the corresponding p-value for each method. The largest α is equal to 0.1 in our experiments.

As can be seen from Table 4.6, in terms of both MMAE and AMAE, the proposed CWOS-Ord method is significantly better than all other methods. Both E-ADASYN and E-SMOTE have higher p-values than no oversampling, E-OR and OGO-SP which indicates that both of them performed satisfactorily according to MMAE and AMAE. On the other hand, OGO-SP has a lower p-value indicating that it did not perform satisfactory. Finally, from Tables 4.2 and 4.3, we can also observe that our method has lower variance compared to other methods.

Table 4.7 Time comparison between OGO and CWOS-Ord in seconds

Datasets	**OGO**	**CWOS-Ord**	**Ratio**
Balance	44.02	9.63	4.57
Heating	45.15	2.33	19.36
Stock	11.70	2.92	4.01
Wisconsin	0.84	0.36	2.32
Machine	0.47	0.07	6.63
Auto	11.83	0.66	17.93
ESL	16.83	1.03	16.32
ERA	15.1	3.12	4.83
Triazines	2.76	0.13	20.97
Wine Quality Red	530.20	35.00	15.15
New Thyroid	1.02	0.09	11.39

We also determined the computation time of our method versus the OGO-SP method, which is the only method designed specifically for ordinal regression. The results are shown in Table 4.7 and Figure 4.3 in logarithmic scale as the computational time ranges from 0.07 sec to 530.20 sec so this allowed the real values to be shown in a single graph. It can be observed that the computational time depends on the number of instances in the datasets as well as the number of features. The larger the dataset and the number of features, the more computation time is needed for the OGO method and the more prominent the time difference is between OGO and our method. For example, for small datasets like Wisconsin and Machine, our method is two and six times faster, respectively, whereas for large datasets like Heating and Triazines, our method is almost 20 times faster. Therefore, the proposed CWOS-Ord method is shown to perform better than other methods in terms of performance measures and computational time.

4.3.3. Choosing Parameters for CWOS-Ord

CWOS-Ord requires four parameters to be selected: c_{thres}, NN, NS and α. In this section, some suggestions are given to better select these parameters. Sensitivity analysis is also performed on few datasets by running CWOS-Ord with different set of values for each parameter. The results are shown in Table 4.8.

- c_{thres}: The threshold for agglomerative clustering is adjusted by this parameter. If c_{thres} is selected as a large value, fewer clusters with larger size will be generated, while if it is selected as a small value, more clusters with smaller sizes will be generated. Therefore, the best value for c_{thres} depends on the dataset. Large sized clusters normally increase the chance of over-generalization due to generation of overlapping instances. On the other hand, small sized clusters normally lead to over-fitting. As can be seen from Table 4.8 a good range for c_{thres} is between 1 and 2.5.
- NN: The number of nearest neighbors used to assign weights to the instances and determine cluster complexity is determined by this parameter. For large values of NN, almost similar weights are assigned to all instances and all clusters will have similar complexity. On the other hand, for small values of NN, both the weights and cluster complexity could be very sensitive to noisy instances. As can be seen from Table 4.8, a reasonable value for NN could be selected between 3 and 7.
- NS: Noisy instances are found using this parameter. If for an instance, all NS nearest neighbors are from non-adjacent classes, then the instance is considered as noise in our

method. A large value for NS makes the method to not be able to find noisy instances whereas a small value for NS makes the method to consider many of the valid instances as noise. As can be seen from Table 4.8, a reasonable value for NS can be between 3 and 7.

- α: This parameter determines the trade-off between complexity of each cluster and the initial size of each cluster as the leading factor in finding the final size of each cluster. The larger the α is, the more the smaller clusters are emphasized, while more complex clusters are ignored and the smaller the α is, the less the smaller clusters are emphasized, while more complex clusters are emphasized more. As can be seen from Table 4.8, α can be selected between 0.4 and 1.

Table 4.8 Sensitivity analysis on CWOS-Ord parameters using OR-EBC

Dataset	**AMAE measure for different values of c_{thres}**		**AMAE measure for different values of *NN***		**AMAE measure for different values of *NS***		**AMAE measure for different values of α**	
	c_{thres}	**AMAE**	***NN***	**AMAE**	***NS***	**AMAE**	α	**AMAE**
Stock	0.5	0.305±0.023	1	0.310±0.038	1	0.411±0.014	0.1	0.300±0.039
	1.0	0.299±0.026	2	0.305±0.029	2	0.291±0.030	0.4	0.302±0.027
	1.5	**0.289±0.023**	4	**0.300±0.037**	4	**0.281±0.031**	0.7	**0.288±0.027**
	2.0	0.304±0.029	6	0.303±0.037	6	0.288±0.031	1.0	0.305±0.034
	2.5	0.304±0.028	8	0.309±0.028	8	0.287±0.024	1.5	0.306±0.043
	3.0	0.302±0.028	10	0.303±0.042	10	0.290±0.025	2.0	0.312±0.024
	3.5	0.307±0.027	15	0.307±0.026	15	0.282±0.022	3.0	0.310±0.030
New Thyroid	0.5	0.038±0.017	1	0.069±0.031	1	0.147±0.086	0.1	0.067±0.040
	1.0	0.037±0.016	2	0.063±0.033	2	0.055±0.030	0.4	**0.066±0.037**
	1.5	**0.032±0.018**	4	**0.056±0.027**	4	0.063±0.036	0.7	0.067±0.053
	2.0	0.045±0.025	6	0.058±0.032	6	**0.052±0.032**	1.0	0.067±0.028
	2.5	0.047±0.024	8	0.061±0.027	8	0.055±0.035	1.5	0.069±0.042
	3.0	0.052±0.020	10	0.064±0.038	10	0.056±0.037	2.0	0.072±0.050
	3.5	0.050±0.025	15	0.071±0.032	15	0.057±0.043	3.0	0.073±0.036
Wisconsin	0.5	1.199±0.115	1	1.174±0.094	1	1.311±0.107	0.1	1.141±0.100
	1.0	1.246±0.094	2	1.180±0.082	2	1.256±0.058	0.4	**1.134±0.099**
	1.5	1.199±0.101	4	**1.174±0.073**	4	**1.159±0.045**	0.7	1.144±0.107
	2.0	**1.195±0.128**	6	1.183±0.064	6	1.181±0.016	1.0	1.152±0.098
	2.5	1.238±0.088	8	1.186±0.071	8	1.218±0.058	1.5	1.153±0.081
	3.0	1.208±0.087	10	1.188±0.054	10	1.185±0.057	2.0	1.154±0.117
	3.5	1.215±0.113	15	1.175±0.084	15	1.221±0.059	3.0	1.145±0.079
Triazines	0.5	0.982±0.047	1	0.966±0.035	1	1.000±0.000	0.1	0.960±0.036
	1.0	0.976±0.049	2	0.979±0.036	2	0.977±0.034	0.4	0.967±0.035
	1.5	0.966±0.041	4	0.978±0.040	4	**0.964±0.033**	0.7	**0.947±0.036**
	2.0	0.964±0.054	6	**0.965±0.051**	6	0.971±0.033	1.0	0.954±0.034
	2.5	**0.949±0.052**	8	0.974±0.037	8	0.970±0.034	1.5	0.953±0.034
	3.0	0.965±0.051	10	0.983±0.039	10	0.966±0.035	2.0	0.954±0.039
	3.5	0.964±0.053	15	0.973±0.055	15	0.969±0.031	3.0	0.956±0.036

4.4. Applying Oversampling Methods for Ordinal Regression to Predict Stages of POP

In this section, clinical and demographical information along with MRI measurements are modeled for predicting the stages of POP. The input variables used for the models are shown in Table 4.9. Some of the MRI measurements are suggested by [118] and are distinguished from other variables by an asterisk next to the name of the variable. These features obtained from MRI were identified that, together with the patient’s background information, were found to be significant in differentiating between low and high stages of POP. This work [118] only considered the binary problem where low stage represents stages 0 and I whereas high stage corresponds to stages II, III, and IV.

Table 4.9 List of demographic, clinical and MRI-based variables

Category	Variable Name
Demographic information	*Age*
	BMI(kg/m2)
	Parity
	Gravida
	Vaginal delivery
	Caesarean Delivery
clinical history	*Hysterectomy*
	Uterosacral colpexy
	Sacrospinous ligament fixation
	Sacrocolpopexy
	Cystocele (anterior) repair
	Rectocele (posterior) repair
	Incontinence Surgery
MRI-based features:	*H-Line*
	PCL
	*Distance Ratio(PCL/MPL)**
	*Distance Ratio(TCL/MPL)**
	*Distance Ratio(OCL/MPL)**
	*Distance Ratio(DCL/MPL)**
	*Angle(between TCL and MPL)**
	*Angle(between DCL and PCL)**

Logistic Regression for ordinal regression and Ordinal Regression by Extended Binary Classification (OR-EBC) were used as the prediction model to investigate if a combination of variables correlates to the outcome variable. Prior to building the prediction model, the dataset was pre-processed through the following stages: 1) Dataset normalization; 2) Feature selection; and 3) Dataset balancing. Following is a description of the pre-processing stages.

1) *Dataset normalization*. The dataset is normalized to transform the range of all variables to [0-1]. Normalization is required to transform the variables to the same scale and allow comparison.
2) *Feature selection*. In order to select relevant features, the greedy algorithm proposed in [] was used. In their method, the feature selection problem for ordinal regression is formulated as an optimization problem with the purpose of finding the features with the maximum total importance scores and minimum total similarity scores. Feature selection enhance generalization by reducing overfitting.
3) *Dataset balancing*. Given that the dataset contains different number of instances for each class, the dataset needs to be balanced. In order to balance the dataset, all the 7 methods explained in Chapter 3 were examined.

After data pre-processing, the prediction models were built. In order to evaluate the performance of the prediction models, Weighted Accuracy, AMAE and MMAE were used as explained in Chapter 4. 3-fold cross-validation was used to measure the performance of the prediction models in terms of measurements. The experiments were repeated three times to report the average in order to alleviate the randomness effects on the results.

The results are shown in Tables 4.10 and 4.11. As can be seen from the results, the accuracy is low and AMAE and MMAE are high for all three types of POP. This indicates that currently used image and clinical features are not sufficient to discriminate among the five POP stages complicating the development of more robust prediction models in the presence of imbalanced datasets.

Table 4.10 Results for the sampling methods on the POP datasets classified using OR-EBC

Dataset	Meas.	NO	E-OR	E-SMOTE	E-ADASYN	E-MWMOTE	OGOSP	CWOS-ORD
Anterior	WAcc	0.4770.058	0.5040.063	0.5020.069	0.4890.053	0.4970.059	0.4990.053	0.5230.068
	MMAE	0.921±0.264	0.837±0.179	0.845±0.182	0.860±0.184	0.834±0.107	0.858±0.157	0.784±0.109
	AMAE	0.642±0.096	0.566±0.058	0.567±0.056	0.587±0.044	0.571±0.058	0.567±0.050	0.539±0.048
Apical	WAcc	0.3390.017	0.4600.054	0.4190.052	0.4450.049	0.3600.081	0.3830.086	0.4050.087
	MMAE	1.929±0.120	0.802±0.072	0.827±0.064	**0.793±0.061**	0.947±0.083	0.930±0.085	0.911±0.092
	AMAE	0.981±0.034	**0.599±0.062**	0.638±0.058	0.622±0.065	0.655±0.086	0.643±0.087	0.642±0.078
Posterior	WAcc	0.3390.040	0.3490.019	0.3630.040	0.3570.037	0.3690.060	0.3670.040	0.3480.075
	MMAE	0.980±0.055	0.932±0.059	0.949±0.050	0.956±0.042	0.929±0.104	**0.914±0.080**	0.930±0.123
	AMAE	0.725±0.048	0.726±0.040	0.707±0.062	0.717±0.047	**0.705±0.050**	0.707±0.046	0.711±0.073

Table 4.11 Results for the sampling methods on the POP datasets classified using Logistic Regression

Dataset	Meas.	NO	E-OR	E-SMOTE	E-ADASYN	E-MWMOTE	OGOSP	CWOS-ORD
Anterior	WAcc	0.4670.043	0.4670.043	0.4670.043	0.5180.045	0.5350.034	0.5440.044	0.5410.042
	MMAE	0.995±0.094	0.995±0.094	0.995±0.094	0.717±0.061	0.719±0.050	0.672±0.077	**0.661±0.068**
	AMAE	0.735±0.085	0.735±0.085	0.735±0.085	0.595±0.021	0.584±0.030	0.572±0.054	0.594±0.048
Apical	WAcc	0.4200.036	0.4200.036	0.4200.036	0.4400.075	0.4360.065	0.4250.050	0.4400.071
	MMAE	1.305±0.208	1.305±0.208	1.305±0.208	0.947±0.131	0.853±0.100	0.943±0.136	**0.851±0.123**
	AMAE	0.826±0.072	0.826±0.072	0.826±0.072	0.711±0.090	0.744±0.067	0.754±0.024	0.734±0.093
Posterior	WAcc	0.4020.071	0.4020.071	0.4020.071	0.4380.067	0.3970.040	0.4180.052	**0.4420.020**
	MMAE	1.007±0.095	1.007±0.095	1.007±0.095	0.881±0.105	0.928±0.103	0.945±0.098	**0.860±0.158**
	AMAE	**0.706±0.041**	**0.706±0.041**	**0.706±0.041**	0.729±0.095	0.774±0.053	0.768±0.045	0.709±0.046

4.5. Conclusions

In this chapter, a new oversampling algorithm called cluster-based weighted oversampling for Ordinal Regression (CWOS-Ord) was presented for ordinal regression with imbalanced datasets. The advantages of CWOS-Ord are as follows: it avoids generating overlapping synthetic instances by considering other-class instances when clustering instances of smaller classes; it determines the cluster sizes using a new measurement based on the cluster complexity and initial size; and it avoids over-generalization and mislabeling errors in the rank scale by assigning weights to instances based on their distance to other-class instances and their rank differences. In addition, well-known oversampling algorithms designed for the imbalanced two-class classification were extended for imbalanced dataset ordinal regression. CWOS-Ord was compared with five other methods. All methods were tested on 11 publicly available datasets with different imbalance ratios and compared using two performance measures. Results show that the proposed CWOS-Ord method performs significantly better to all other methods based on both of the performance measures. This indicates that identifying small clusters of data for subsequent oversampling consideration, and incorporating information on instances' rank differences and cluster size can be important in addressing imbalanced datasets for ordinal regression.

Chapter 5

Automatic Tracking, Segmentation and Analysis of Pelvic Organs Movement in Dynamic MRI to Improve Multi-stage POP Diagnosis

A new contour tracking method is presented to automatically track and segment pelvic organs on DMRI followed by a multiple-object trajectory classification method to improve the diagnosis of pelvic organ prolapse. Organs are first tracked using particle filters and K-means clustering with prior information. Then, they are segmented using the convex hull of the cluster of particles. Finally, the trajectories of the pelvic organs are modeled using a new Coupled Switched Hidden Markov Model (CSHMM) to classify the severity of pelvic organ prolapse. The tracking and segmentation results have been validated using Dice Similarity Index (DSI) whereas the classification results have been compared with two manual clinical measurements. Results demonstrate that the presented method is able to automatically track and segment pelvic organs with a DSI above 82% for 94 tested cases. The accuracy of the trajectory classification is better than current manual measurements for all three types of prolapse. In terms of f-measure, the proposed method was shown to be better than the manual measurements for anterior and apical prolapse but not for posterior prolapse. This work aims to automatically extract and analyze image data to improve the prediction of disorders such as pelvic organ prolapse.

5.1. Methodology

The proposed method to automatically track, segment and analyze the movement of pelvic organs is described in this section. Figure 5.1 gives an overview of the proposed method. The process starts with the data collection, followed by a contour tracking method for automated tracking and segmentation of pelvic organs using prior information. Finally, the pelvic organ trajectories are analyzed using a proposed coupled switched hidden Markov model.

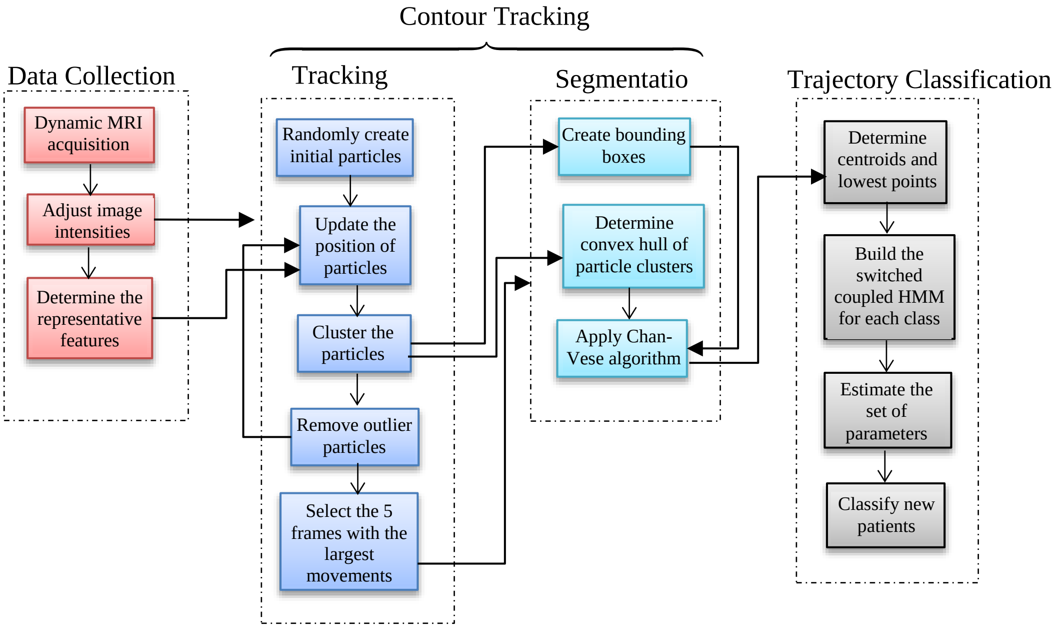

Contour Tracking
Data Collection
Dynamic MRI acquisition
Adjust image intensities
Determine the representative features
Tracking
Randomly create initial particles
Update the position of particles
Cluster the particles
Remove outlier particles
Select the 5 frames with the largest movements
Segmentatio
Create bounding boxes
Determine convex hull of particle clusters
Apply Chan-Vese algorithm
Trajectory Classification
Determine centroids and lowest points
Build the switched coupled HMM for each class
Estimate the set of parameters
Classify new patients

Figure 5.1 Overview of the proposed predictive model

5.1.1. Data Acquisition

A representative clinical dataset of 94 cases with dynamic MRI was used in this study. The Institutional Review Board at the University of South Florida considered the study exempt since all protected health information was previously removed from the clinical and MRI data before collected from a database for this study. MR imaging was taken on a 3-Tesla GE system (General Electric Company, GE Healthcare, Buckinghamshire, UK) using an 8-channel torso phased-array coil with the patient in a modified dorsal lithotomy position. Prior to imaging, 60ml of ultrasound gel was placed in the rectum for improved visualization. Dynamic imaging was taken in a multiphase, single-slice sequence. The images were acquired in the midsagittal plane for 23-27 seconds, using a T2-weighted single-shot fast-spin echo sequence. Patients were coached, prior to imaging, on performance of an adequate valsalva maneuver. Each patient has 20 frames showing the pelvic floor structures from rest to maximum strain.

The image data has been preprocessed and de-identified. Each patient has been manually examined through POP-Q and a stage has been assigned based on the POP-Q measurements for each type of prolapse (anterior, apical and posterior). The stages are from stage 0 through stage 4 and the patients in this study have different stages of POP. The purpose of this study is to classify the patients into two stages: high prolapse and low prolapse. Patients with prolapse stages of 0, 1 and 2 are considered as low severity of prolapse whereas patients with stages of 3 and 4 are considered as high severity of prolapse.

Before analyzing the MRI data, the images are normalized to improve the contrast of the input images by stretching the range of intensity values. Then, a training set is selected from the dataset to analyze and extract a representative set of intensity and texture features *R* for the bladder and rectum. The texture features include the range, standard deviation and entropy. The uterus, although also a pelvic organ, is not considered in this work as many cases in our dataset belong to patients whose uterus has been surgically removed (hysterectomy).

5.1.2. Automated Tracking and Segmentation of Pelvic Organs Using Prior Information

In the first stage of the proposed contour tracking method, the bladder and rectum are tracked using an adapted particle filter approach with prior information. This information consists of the relative locations and common movement directions of the pelvic organs. The following prior information has been incorporated in the particle filter tracking and are explained in more details throughout this section:

1) No part of the bladder and rectum is located on the top quartile of the images.
2) The pelvic organs tend to move down or to the right during dynamic MRI.
3) The bladder is always on the left side of the image while the rectum is on the right side.

This prior information is used to improve the generation, updating, and resampling of the particles. For example, since no part of bladder and rectum is located in the top quartile of the images, particles are not generated on this quartile to improve particle tracking (Figure 5.2).

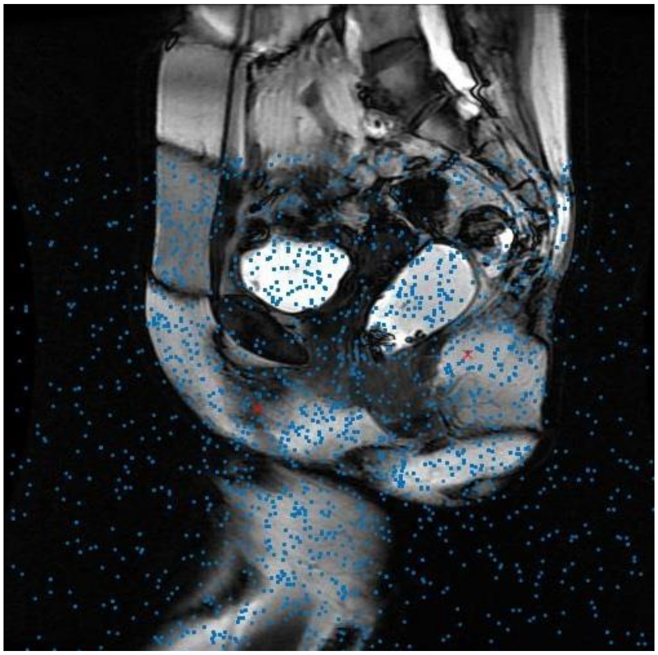


Figure 5.2 Random particles generated using information on common organ location

1) Update the position of the particles by assuming a proper velocity. We assume uniform linear motion for the bladder and rectum, and use prior information on their common movement directions to improve the tracking results. This is achieved by updating the particles using the linear velocity and imposing higher chances that a particle moves down or to the right.

2) Calculate the likelihood of particles $L^{(k)}$. For each particle *k*, we measure how close its features q_k are from *R* where σ is the standard deviation of $(q_k - R)$.

$$L^{(k)} = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(q_k-R)^2}{2\sigma^2}\right) \tag{5.1}$$

3) Resample the particles with replacement according to their likelihood, where $P^{(k)}$ is the likelihood of the *k*th particle and *N* is the number of particles:

$$P^{(k)} = \frac{L^{(k)}}{\sum_{l=1}^{N} L^{(l)}} \tag{5.2}$$

After resampling, for each frame, we use *k*-means to cluster the particles into two groups corresponding to each pelvic organ (bladder and rectum). Prior information on the relative location of the bladder and rectum in the image is incorporated to provide a better initialization for the *k*-means clustering. In particular, it is known that the bladder is always on the left side of the image while the rectum is on the right side. Therefore, the initial placement for the centers in *k*-means is based on this information to improve clustering of the two organs. Outlier particles are removed from each cluster using the Grubbs test (Figure 5.3) [119], because during the resampling there is a chance that some particles with low likelihood are selected. The Grubbs's test statistics of all particles to their corresponding center is measured based on their distance assuming they have normal distribution. Then, the ones that are statistically farther from the center at $\alpha = 0.05$ are identified as outliers and are eliminated.

It was observed that the majority of the pelvic DMRI frames did not provide any significant information on the pelvic organ movement. Given that segmentation is a computationally expensive process, frames that do not contribute with information are removed and segmentation is performed only on a representative set of frames to reduce computation time without losing information. In this work, the movement of particles' centroids for bladder and rectum are measured and the five frames with the largest movement are selected as the representative frames. We chose five frames based on our analysis of the image dataset.

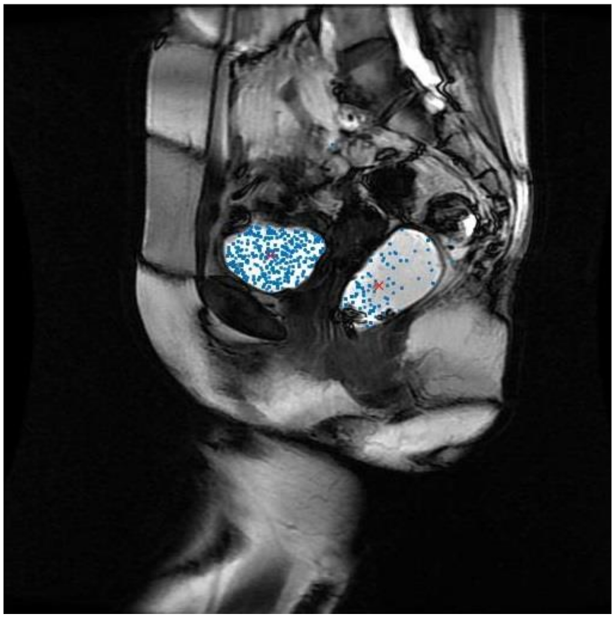


Figure 5.3 Updated particles used for tracking after removing outliers.

The resulting two clusters of particles are used to define a bounding box for each pelvic organ to constrain the search space during segmentation and significantly reduce the computational time. An initial adaptive contour is proposed for segmentation that is generated from the convex hull of each particle cluster. This provides a good initial contour to initialize the Chan-Vese contour segmentation algorithm [85] and automate the process. In contrast with the original Chan-Vese algorithm that requires an initial contour to be manually defined for each frame, our approach determines the initial contour for each frame automatically and adaptively using the convex hull of particles to identify the boundaries of the bladder and rectum. The generated bounding box and convex hull are depicted in Figure 5.4.

5.1.3. Multiple Pelvic Organs Trajectory Analysis

Using data from the segmented and tracked frames, the trajectory of the bladder's and rectum's centroids and lowest points are analyzed. The lowest points are considered because they are of clinical interest and currently used to determine the stage (or severity) of prolapse. This leads to four trajectories that will be obtained for each patient (Figure 5.5) where the individual organ movement and their interactions are important. A new method called Coupled Switched Hidden Markov Model (CSHMM) is proposed to capture the interactions among the four trajectories to classify the severity of pelvic prolapse.

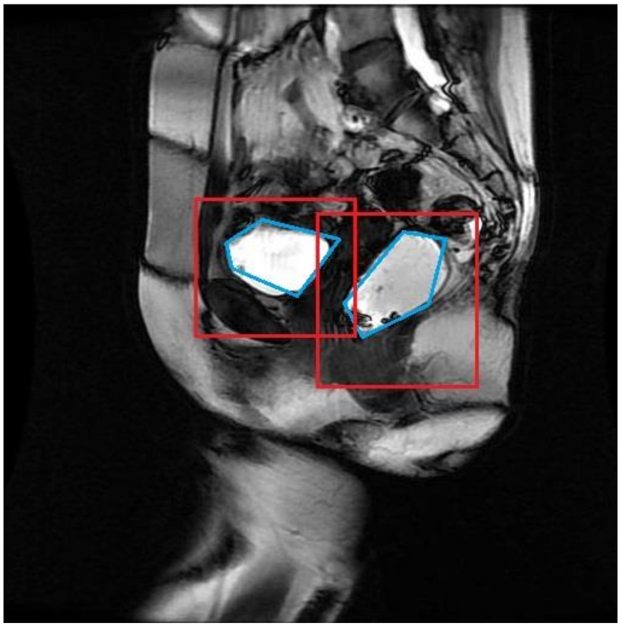


Figure 5.4 Generating bounding box (red) and initial curve (blue) for bladder (left side) and rectum (right side) using their corresponding particles.

In this work, patients are to be classified into two classes: high severity of prolapse (class +1) and low severity of prolapse (class -1), so the set of output variable is $c \in \{+1,-1\}$. For each patient *i* in class *c*, there exists four trajectories $l \in \{1,2,3,4\}$ with the sequence of positions $x_l = (x_{1l},..,x_{5l})$ where $x_{lt} \in \Re^2$. CSHMM is a generative model, hence a separate model should be made for the examples of each class. As can be seen in Figure 5.6, in our model, the state of each trajectory at time *t* depends on its own state at time *t-1*, its observation at time *t* and on the states of other trajectories at time *t-1*.

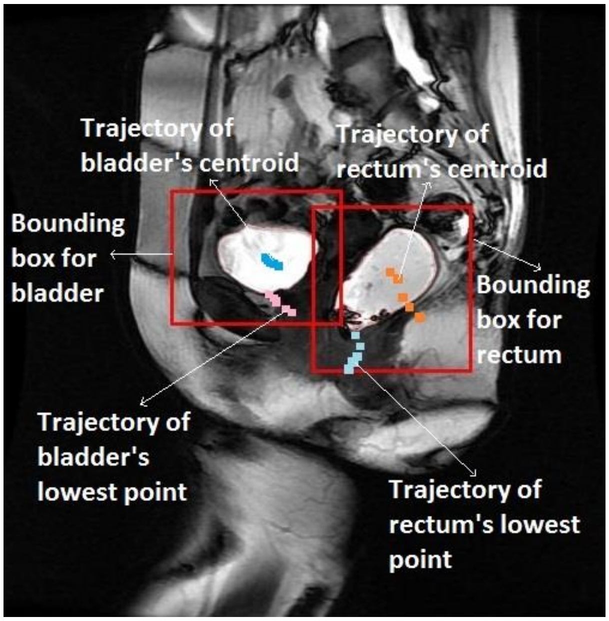

Trajectory of bladder's centroid
Trajectory of rectum's centroid
Bounding box for bladder
Bounding box for rectum
Trajectory of bladder's lowest point
Trajectory of rectum's lowest point

Figure 5.5 Four trajectories to be analyzed for each patient

We consider the observed variables as the set of the relative movement $\{d_{lt} = (x_{l,t+1} - x_{l,t}), l = 1,2,3,4, t = 1,2,3,4\}$ of the four trajectories rather than their absolute positions $X = \{x_l, l = 1,2,3,4\}$ because we want to study the movements of the organs. As shown in Figure 5.6, the hidden states in our model are "stopped", "moving up", "moving down-right", "moving down" and "moving down-left".

Given the observed feature vector $\{d_l = (d_{l1},..., d_{l4}), l = 1,2,3,4\}$ and the corresponding set of hidden state $\{h_l = (h_{l1},..., h_{l4}), l = 1,2,3,4\}$, the task is to estimate the set of parameters $\gamma_c = (\Pi_c, \theta_c, A_c)$ for each class c. $\Pi_c = \{\pi_c(s_1,..., s_4), s_l = 1,..., N, l = 1,2,3,4\}$ are the initial probabilities for the states, given that each state can take N different values. θ_c is the set of parameters for the Gaussian distribution including the mean $\mu_{(s_1,...,s_4)}$ and the variance $\Sigma_{(s_1,...,s_4)}$ and A_c is the state transition probabilities. In contrast to [92], in which first the Gaussian parameters θ_c are estimated and then (Π_c, A_c) are estimated separately, in our method all the parameters are determined simultaneously resulting in better estimation of the parameters at the expense of higher computational time. This approach can be justified when the dataset is small.

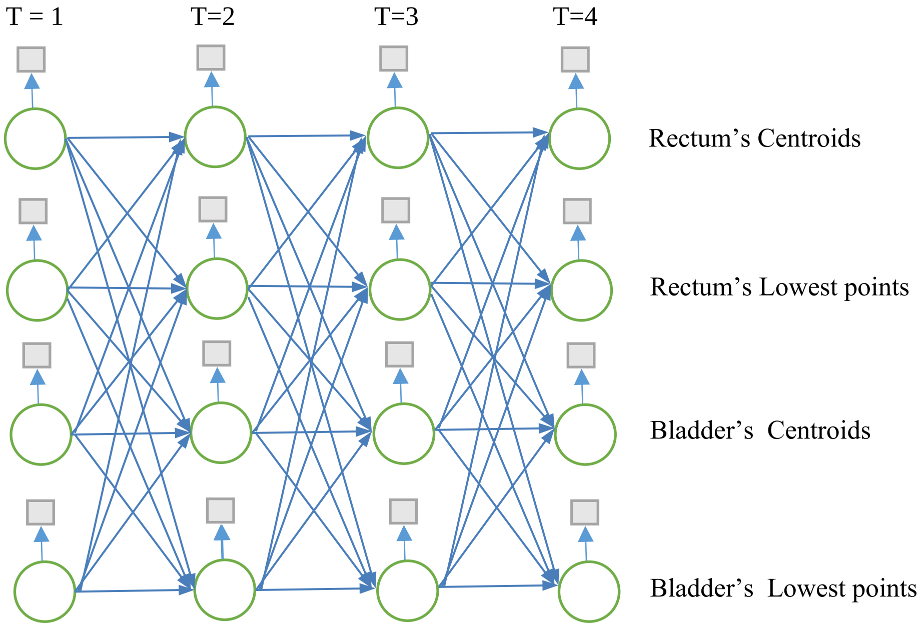


Figure 5.6 CSHMM model

After building a model for each class of prolapse, the next step is to classify new cases with a set of four observed trajectories into different classes. The *maximum a posteriori* rule is being used for this purpose:

$$c = \arg\max_c \{p(x|c)p(c)\} = \arg\max_c \{p(x|\hat{\Pi}_c,\hat{\theta}_c,\hat{A}_c)p(c)\} \qquad (5.3)$$

in which the $p(x|\hat{\Pi}_c,\hat{\theta}_c,\hat{A}_c)$ is the log likelihood of the most probable explanation (mpe) of example *x* using the model for class *c* and *p(c)* is the *a priory* probability of the class *c*. In our experiments, we set *p(c)* equal to the proportion of each class in the dataset. Hence, we set *p(c=+1)* = 0.34 and *p(c=-1)* = 0.66. We also used the Viterbi algorithm [120] to find the mpe and likelihood of each new patient for each model.

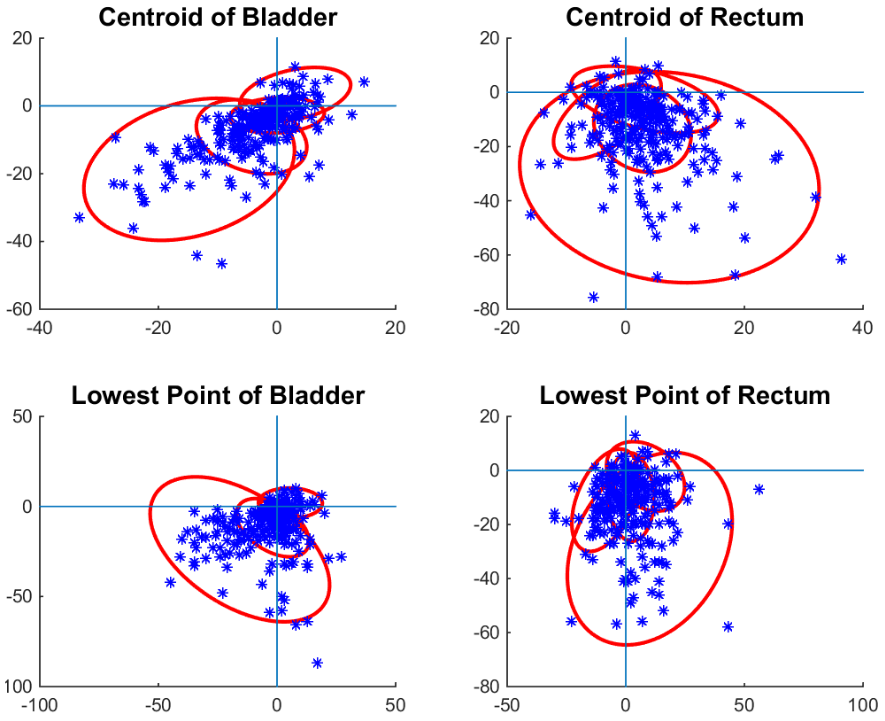


Figure 5.7 The scatterplot of trajectories' displacement. Each eclipse refers to the Gaussian distribution of the hidden states.

5.2. Results and Discussions

The proposed contour tracking method for tracking and segmentation of multiple pelvic organs has been tested on 94 cases, which were manually segmented by an expert as the ground truth. The composition of the dataset based on POP-Q, i.e. the number of patients that are diagnosed as high prolapse and low prolapse for the three types of prolapse based on manual examination are shown in Table 5.2. As can be seen from Table 5.2, for the anterior and posterior prolapse, more patients are suffering from high severity of prolapse, while in the case of apical prolapse, less patients are suffering from high severity of prolapse. Tracking and segmentation results are validated using Dice Similarity Index (DSI). DSI is a common measure to quantify the degree of overlap among objects in binary images [121] and it is defined as follows:

$$DSI = \frac{2a}{(2a+b+c)} \qquad (5.4)$$

where *a* is the number of pixels with a value of “1” in both binary images, *b* is the number of pixels with a value of “1” just in the first image and *c* is the number of pixels with value of “1” just in the second image. DSI was used as a quantitative measure of the similarity between our method’s segmentations and the ground truth. For each patient, the DSI for the five frames were calculated and averaged. Then the averaged DSI for all the patients were averaged over the 94 patients. Results indicate that the proposed method is able to automatically track and segment pelvic organs with a DSI of 0.8249±0.0399 for the tested cases. The contour tracking results for a patient are shown in Figure 5.8.

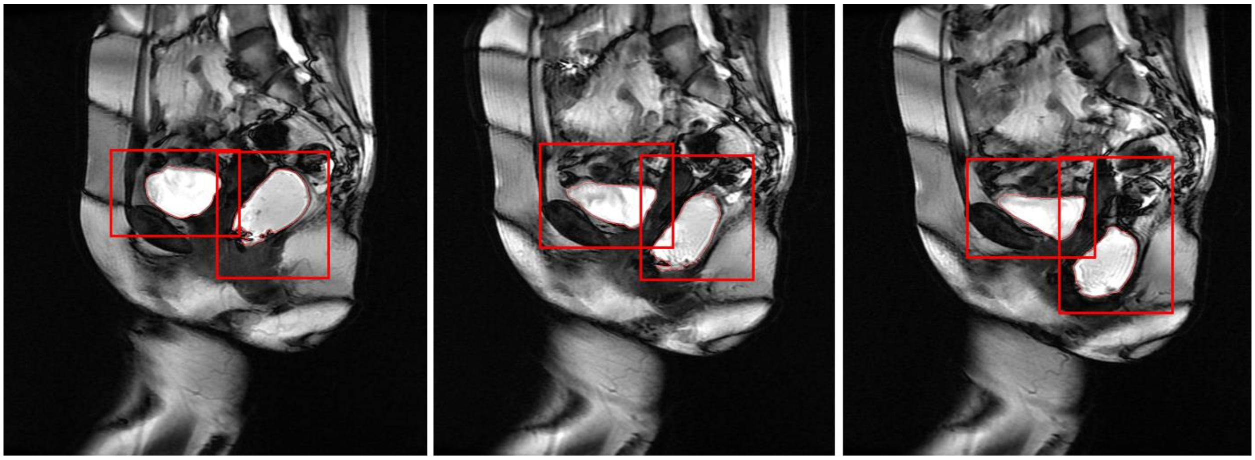


Figure 5.8 Results for the tracked and segmented organs.

In order to determine whether there is a relationship between pelvic organs movement on dynamic MRI with the severity of prolapse, the maximum displacement of the organs’ lowest point from rest to maximum strain was analyzed. The mean and standard deviation of this displacement were determined for the total study population and compared with the two classes of prolapse (low severity and high severity) as shown in Table 5.1. The statistical significance of the maximum displacement difference between the two classes was measured using a two-sided t-test. Alpha = 0.05 was used to accept or reject whether there exists a difference between the two classes for each organ or not. As can be seen from Table 5.1, at alpha = 0.05 the difference is significant for anterior prolapse. On the other hand, although on average the rectum was shown to move more for the case of high severity of prolapse, the difference in displacement from rest to maximum strain was not found to be significant for posterior prolapse. Therefore, we can conclude that for the case of anterior prolapse, large bladder displacement observed on MRI from rest to maximum strain is related to high severity of prolapse. However, a similar conclusion cannot be made for the case of posterior prolapse.

Table 5.1 Summary statistics for the total displacement of bladder and rectum

POP Type	Total (n = 46)	Low prolapse	High prolapse	Pvalue
Anterior	37.049±23.375	27.446±15.429	42.491±25.394	0.0023*
Posterior	39.749±23.375	34.983±24.016	41.5718±19.557	0.1741

In addition, it was studied whether there exists any correlation between the lowest point's largest displacement of the bladder and rectum on MRI from rest to maximum strain. Figure 12 shows the displacement of the bladder on the y-axis and the displacement of the rectum on the x-axis.

Kendall's tau for the correlation was 0.3636 and the p-value was 2.2113e-07. The Pearson correlation coefficient was also 0.3905 and the p-value was 9.9720e-05. Therefore, at alpha = 0.05, based on both Kendall's tau and Pearson coefficient, we can conclude that there exists enough evidence that the maximum displacement of the bladder and rectum are correlated. This indicates that a large bladder displacement tends to also present with high rectum displacement and vice versa. These results confirm the importance of considering the interactions of pelvic organs in dynamic MRI to improve understanding of the condition.

The proposed CSHMM used for classification of the severity of posterior pelvic prolapse was compared with two commonly-used manual measurements: 1) Pubococcygeal Line (PCL) and

2) Mid-pubic line (MPL). These measurements were measured by an expert radiologist and were converted to stages of prolapse using the standard criteria described in [122].

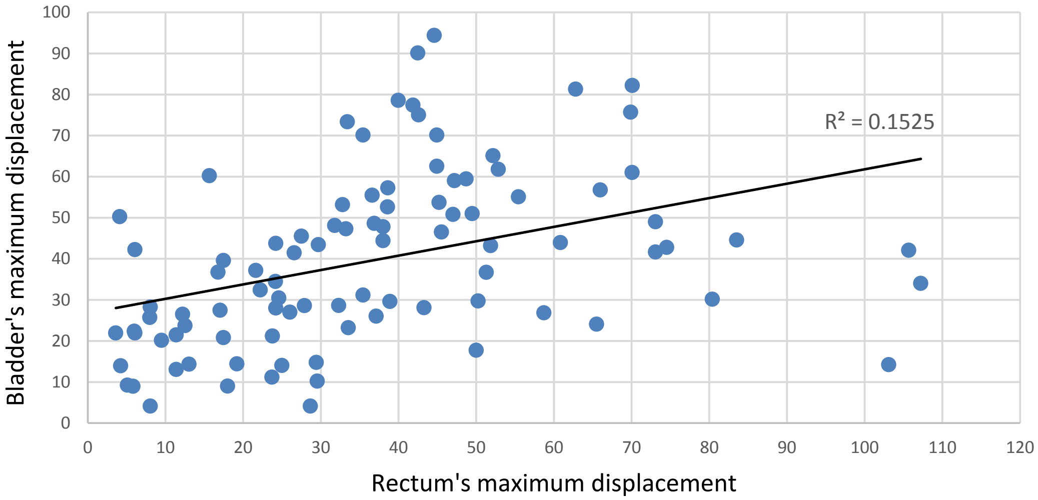


Figure 5.9 Correlation of rectum’s maximum displacement and bladder’s maximum displacement

Leave-one-out cross-validation was used to measure the performance of the prediction model in terms of accuracy and F-measure. In leave-one-out cross-validation, all but one of the examples from the dataset are used for training the model, and the remaining example is used for testing the model. This process is repeated for each of the examples in the dataset to predict if the example has high severity of prolapse or not.

Table 5.2 Composition of the dataset based on POPQ

POP Type	Low Prolapse	High Prolapse
Anterior	34	60
Apical	78	16
Posterior	26	68

The prediction for each example is compared with the POP-Q measurement of each example to obtain the accuracy and the F-measure of all 94 examples. F-measure is the weighted average of recall and precision. Precision measures the exactness of our prediction model that is, the number of patients labeled as low severity that are actually high severity. Recall measures the completeness of our prediction model as the number of patients with high prolapse that were predicted correctly. The experiments were repeated three times to report the average in order to alleviate the randomness effects on the results. The comparison between MPL, PCL and our method for the three types of prolapse is shown in Table 5.3.

The results show that the proposed model provides greater accuracy compared to the manual measurements for all types of prolapse. In terms of f-measure, the proposed method is showing better results for both MPL and PCL for anterior and apical prolapse, but not for posterior prolapse. Also, in agreement with the results in [122] and as a secondary conclusion, MPL measurements work better than PCL for the three types of prolapse in our 94 patients. As future work, experiments will be performed on the dataset of 207 cases to obtain more robust results, determine a patient-specific feature set, and generalize the method to a larger and more diverse dataset.

Table 5.3 Results comparing our proposed CSHMM with commonly-used manual measurements to predict severity of posterior prolapse

POP Type	**Measurement**	***Proposed method***	***MPL***	***PCL***
Anterior	Accuracy	**0.6277**	0.5598	0.3589
	Fmeasure	**0.6317**	0.6309	0.3524
Apical	Accuracy	**0.8191**	0.7608	0.7321
	Fmeasure	**0.8957**	0.8512	0.8427
Posterior	Accuracy	**0.6702**	0.5502	0.4354
	Fmeasure	0.4364	**0.4891**	0.4327

Chapter 6

Summary and Future Work

The main outcome of this research is the development of two oversampling methods to address the imbalance problem in binary data classification and ordinal regression. These techniques were tested on public datasets and then were examined in a gynecological diagnosis application to predict the risk of development of multi-stage pelvic organ prolapse with imbalanced datasets using image data from pelvic organ movement.

For the first objective of this research, a new oversampling algorithm called Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) has been presented for imbalanced binary dataset classification. The advantages of A-SUWO are that it avoids generating overlapping synthetic instances by considering the majority instances when clustering minority instances; it determines the sub-cluster sizes adaptively using the standardized average error rate and cross-validation; it oversamples the sub-clusters by assigning weights to their instances to avoid over-generalization; and it does not ignore isolated sub-clusters. A-SUWO was tested on 16 publicly available datasets with different imbalance ratios and compared with other sampling techniques using different types of classifiers. Results show that our method performs significantly better compared to other sampling methods in most datasets and in larger datasets with higher imbalance ratio.

For the second objective, a new oversampling algorithm called cluster-based weighted oversampling for Ordinal Regression (CWOS-Ord) was presented for imbalanced dataset ordinal regression. The advantages of CWOS-Ord are that it finds small clusters of data and considers them for oversampling; and avoids over-generalization and mislabeling errors in the rank scale by assigning weights to instances based on their distance to other-class instances and their rank differences. In addition, well-known oversampling algorithms designed for the imbalanced two-class classification were extended for imbalanced dataset ordinal regression. Results show that the proposed CWOS-Ord method performs significantly better to all other methods based on the performance measures. This indicates that identifying small clusters of data for subsequent oversampling consideration, and incorporating information on instances' rank differences and cluster size can be important in addressing imbalanced datasets for ordinal regression.

For the third and last objective, an automatic method was presented to track, segment, and analyze the trajectories of pelvic organs on dynamic MRI. A modified particle filter approach was designed by incorporating prior information and clustering to track the pelvic organs automatically. An adaptive initial curve for segmentation using the convex hull of the particle clusters was proposed to automate and reduce computation time for segmentation. Later, the trajectories of centroids and lowest points of the segmented pelvic organs were modeled using a new Coupled Switched Hidden Markov Model (CSHMM) to classify the severity of pelvic organ prolapse. Results demonstrate that the proposed method can accurately track and segment the pelvic organs, and improve the classification of the severity of pelvic prolapse by modeling the resulted trajectories. The proposed method can be used to analyze the movement of pelvic organs to improve the diagnosis of pelvic organ prolapse. It can also be used for the automatic tracking, segmentation and classification of deformable structures from a sequence of images. As future work, we would like to extend this work for the case of classifying all the 5 stages of POP and incorporating other patient data such as medical and demographical data. Finally, we would like to perform our experiments on a larger dataset.

References

[1] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web,* vol. 16, pp. 449-475, 2013.

[2] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Machine Learning: ECML 2004*, ed: Springer, 2004, pp. 39-50.

[3] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 21, pp. 1263-1284, 2009.

[4] P. Li, K. L. Chan, S. Fu, and S. M. Krishnan, "Kernel Machines for Imbalanced Data Problem in Biomedical Applications," in *Support Vector Machines Applications*, ed: Springer, 2014, pp. 221-268.

[5] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data," *ACM Sigkdd Explorations Newsletter,* vol. 6, pp. 80-89, 2004.

[6] L. Piras and G. Giacinto, "Synthetic pattern generation for imbalanced learning in image retrieval," *Pattern Recognition Letters,* vol. 33, pp. 2198-2205, 2012.

[7] D. Thammasiri, D. Delen, P. Meesad, and N. Kasap, "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition," *Expert Systems with Applications,* vol. 41, pp. 321-330, 2014.

[8] R. C. Prati, G. E. Batista, and D. F. Silva, "Class imbalance revisited: a new experimental setup to assess the performance of treatment methods," *Knowledge and Information Systems*, pp. 1-24, 2014.

[9] S. Alshomrani, A. Bawakid, S.-O. Shim, A. Fernández, and F. Herrera, "A proposal for evolutionary fuzzy systems using feature weighting: Dealing with overlapping in imbalanced datasets," *Knowledge-Based Systems,* vol. 73, pp. 1-17, 2015.

[10] X.-Y. Liu, Wu, J., Zhou, Z.-H., "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Knowledge and Data Engineering, Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 39, pp. 539-550, 2009.

[11] K.-j. Kim and H. Ahn, "A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach," *Computers & Operations Research,* vol. 39, pp. 1800-1811, 2012.

[12] R. Caruana, Baluja, S., Mitchell, T. , "Using the future to" sort out" the present: Rankprop and multitask learning for medical risk evaluation," *Advances in neural information processing systems,* 1996.

[13] W. Chu, Ghahramani, Z. , "Preference learning with Gaussian processes," in *22nd international conference on Machine learning*, 2005, pp. 137-144.

[14] P. Dallenbach, Kaelin-Gambirasio, I., Jacob, S., Dubuisson, J.B., Boulvain, M., "Incidence rate and risk factors for vaginal vault prolapse repair after hysterectomy," *IntUrogynecol J,* vol. 19, pp. 1623-1629, 2008.

[15] L. L. Subak, Waetjen, L. E., van den Eeden, S., Thom, D. H., Vittinghoff, E., Brown, J. S., "Cost of pelvic organ prolapse surgery in the United States," *Obstet Gynecol, 2001,* vol. 98, pp. 646-651, 2001.

[16] J. R. Popovic, Kozak, L. J., "National hospital discharge survey: annual summary," *Vital Health Stat,* vol. 2000, p. 194.

[17] E. Cortes, Reid, W.,M., N., Singh, K., Berger, L., "Clinical Examination and Dynamic Magnetic Resonance Imaging in Vaginal Vault Prolapse," *Obstetrics & Gynecology,* vol. 103, pp. 41-46, 2004.

[18] A. Lienemann, Sprenger, D., Janssen, U., Grosch, E., Pellengahr, C., Anthuber, C., "Assessment of pelvic organ descent by use of functional cine-MRI: which reference line should be used? ," *Neurourol Urodyn* vol. 23, pp. 33-37, 2004.

[19] C. J. Robinson, Swift, S., Johnson, D. D., Almeida, J. S., "Prediction of pelvic organ prolapse using an artificial neural network," *American Journal of Obstetrics and Gynecology,* pp. 193.e1 – 193.e6, 2008.

[20] S. Onal, Lai-Yuen, S., Bao, P., Weitzenfeld, A., Hart, S., "MRI based Segmentation of Pubic Bone for Evaluation of Pelvic Organ Prolapse," *IEEE Journal of Biomedical and Health Informatics,* vol. 18, pp. 1370-1378, 2013.

[21] M. Rahim, M.-E. Bellemare, R. Bulot, and N. Pirró, "Pelvic organs dynamic feature analysis for MRI sequence discrimination," in *Pattern Recognition (ICPR), 2010 20th International Conference on,* 2010, pp. 2496-2499.

[22] Z.-W. Chen, P. Joli, Z.-Q. Feng, M. Rahim, N. Pirró, and M.-E. Bellemare, "Female patient-specific finite element modeling of pelvic organ prolapse (POP)," *Journal of biomechanics,* vol. 48, pp. 238-245, 2015.

[23] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 42, pp. 463-484, 2012.

[24] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, "Diversity techniques improve the performance of the best imbalance learning ensembles," *Information Sciences,* vol. 325, pp. 98-117, 2015.

[25] I. Nekooeimehr and S. K. Lai-Yuen, "Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets," *Expert Systems with Applications,* vol. 46, pp. 405-416, 2016.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[27] S. Barua, Islam, M., Yao, X., Murase, K., "MWMOTE--Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, pp. 405-425, 2014.

[28] G. Wu and E. Y. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, pp. 786-795, 2005.

[29] F. Verhein and S. Chawla, "Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets," in *Seventh IEEE International Conference on Data Mining, 2007. ICDM 2007.,* 2007, pp. 679-684.

[30] L. Gonzalez-Abril, H. Nuñez, C. Angulo, and F. Velasco, "GSVM: An SVM for handling imbalanced accuracy between classes inbi-classification problems," *Applied Soft Computing*, vol. 17, pp. 23-31, 2014.

[31] Y. Sun, Kamel, M. S., Wong, A. K., Wang, Y., "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, pp. 3358-3378, 2007.

[32] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Mining and Knowledge Discovery*, vol. 17, pp. 225-252, 2008.

[33] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, pp. 185-197, 2010.

[34] S. Wang, L. L. Minku, and X. Yao, "Resampling-Based Ensemble Methods for Online Class Imbalance Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 1356-1368, 2015.

[35] Y. Qian, Y. Liang, M. Li, G. Feng, and X. Shi, "A resampling ensemble algorithm for classification of imbalance problems," *Neurocomputing*, vol. 143, pp. 57-67, 2014.

[36] S. Barua, Islam, M., Yao, X., Murase, K., "MWMOTE--Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning," *IEEE Transactions on Knowledge and Data Engineering,*, vol. 26, pp. 405-425, 2014.

[37] N. V. Chawla, Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P., "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Resear*, vol. 16, pp. 321-357, 2002.

[38] H. Han, Wang, W. Y., Mao, B. H. , "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," *Advances in intelligent computing, ed: Springer*, pp. 878-887, 2005.

[39] S.-J. Yen, Lee, Y.-S., "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, pp. 5718-5727, 2009.

[40] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, pp. 429-449, 2002.

[41] H. He, Y. Bai, E. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, 2008, pp. 1322-1328.

[42] L. Zhou, "Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods," *Knowledge-Based Systems*, vol. 41, pp. 16-25, 2013.

[43] V. López, Fernández, A., García, S., Palade, V.,Herrera, F., "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113-141, 2013.

[44] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, pp. 25-36, 2006.

[45] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Advances in Knowledge Discovery and Data Mining*, ed: Springer, 2009, pp. 475-482.

[46] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explorations Newsletter*, vol. 6, pp. 40-49, 2004.

[47] D. A. Cieslak and N. V. Chawla, "Start globally, optimize locally, predict globally: Improving performance on imbalanced data," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 2008, pp. 143-152.

[48] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "DBSMOTE: density-based synthetic minority over-sampling technique," *Applied Intelligence,* vol. 36, pp. 664-684, 2012.

[49] D. A. Cieslak, N. V. Chawla, and A. Striegel, "Combating imbalance in network intrusion datasets," in *GrC*, 2006, pp. 732-737.

[50] J. Lin, "The curse of zipf and limits to parallelization: A look at the stragglers problem in mapreduce," in *7th Workshop on Large-Scale Distributed Systems for Information Retrieval*, 2009.

[51] S. Wang, Yao, X. , "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 42, pp. 1119-1130, 2012.

[52] A. Fernández, López, V., Galar,M., Del Jesus, M. J., Herrera, F. , "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowledge-Based Systems,* vol. 42, pp. 97-110, 2013.

[53] T. W. Liao, "Classification of weld flaws with imbalanced class data," *Expert Systems with Applications,* p. 35, 2008.

[54] M. Perez-Ortiz, P. A. Gutierrez, C. Hervas-Martinez, and X. Yao, "Graph-Based Approaches for Over-sampling in the context of Ordinal Regression," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 27, pp. 1233-1245, 2015.

[55] (September 28,). *The Harvard Medical School Family Health Guide*. Available: http://www.health.harvard.edu/fhg/updates/update0805c.shtml

[56] R. M. Ellerkman, Cundiff, G. W., Melick, C. F., Nihira, M. A., Leffler, K., Bent, A. E., "Correlation of symptoms with location and severity of pelvic organ prolapse," *American Journal of Obstetrics and Gynecology, ,* vol. 185, pp. 1332-1338, 2001.

[57] S. E. Swift, "The distribution of pelvic organ support in a population of female subjects seen for routine gynecologic health care," *Am J Obstet Gynecol,* vol. 183, pp. 277-85, 2000.

[58] J. S. Tan, Lukacz, E. S., Menefee, S. A., Powel, C. R., Nager, C. W., "San Diego Pelvic Floor Consortium, Predictive value of prolapse symptoms: a large database study," *Int Urogynecol J,* vol. 16, pp. 203-209, 2005.

[59] C. Ghetti, Gregory, W., Edwards, S.R., Otto, L. N., Clark, A. L., "Pelvic organ descent and symptoms of pelvic floor disorders," *American Journal of Obstetrics and Gynecology,* vol. 193, pp. 53-57, 2005.

[60] T. A. Smith, Poteat, T. A, Shobeiri, A., "Pelvic organ prolapse: an overview," *JAAPA,* vol. 27, pp. 20-24, 2014.

[61] B. I. Kudish, Iglesia, C. B., Gutman, R. E., "Risk factors for prolapse development in white, black, and Hispanic women," *Female Pelvic Med Reconstr Surg,* vol. 17, pp. 80-90, 2011.

[62] E. C. Samuelsson, Arne Victor, F. T., Tibblin, G., Svardsudd, K. F., "Signs of genital prolapse in a Swedish population of women 20 to 59 years of age and possible related factors," *Am J Obstet Gynecol,* vol. 180, pp. 299-305, 1999.

[63] M. D. Barber, Neubauer, N. L, Klein-Olarte, V., "Can we screen for pelvic organ prolapse without a physical examination in epidemiologic studies?," *Am J Obstet Gynecol,* vol. 195, pp. 942–948, 2006.

[64] L. J. Burrows, Meyn, L. A., Walters, M. D., Weber, A. M. , "Pelvic symptoms in women with pelvic organ prolapse," *Obstet Gynecol,* vol. 104, pp. 982-88, 2004.

[65] A. M. Weber, Walters, M. D., Piedmonte, M. R., Ballard, L. A., "Anterior colporrhaphy: A randomized trial of three surgical techniques," *Am. J. Obstet. Gynecol.,* vol. 185, pp. 1299-1306, 2001.

[66] B. Vakili, Zheng, Y.T., Loesch, H., Echols, K. T., Franco, N., Chesson, R. R., "Levator contraction strength and genital hiatus as risk factors for recurrent pelvic organ prolapse," *Am. J. Obstet. Gynecol.*, vol. 192, pp. 1592-1598, 2005.

[67] A. E. Gousse, Barbaric, Z. L, Safir, M. H., "Dynamic "HASTE" MRI sequence in the evaluation of all female pelvic pathology," *Urology,* vol. 159, pp. 328-334, 1998.

[68] H. K. Pannu, "MRI of pelvic organ prolapse," *Eur. Radiol.*, vol. 14, pp. 1456-1464, 2004.

[69] S. R. Broekhuis, Futterer, J. J., Barentsz, J. O., Vierhout, M. E., "A systematic review of clinical studies on dynamic magnetic resonance imaging of pelvic organ prolapse: the use of reference lines and anatomical landmarks," *Int Urogynecol J Pelvic Floor Dysfunct,* vol. 20, pp. 721-729, 2009.

[70] C. V. Comiter, Vasavada, S.P., Barbaric, Z.L., Gousse, A.E., Raz, S. , "Grading pelvic prolapse and pelvic floor relaxation using magnetic resonance imaging," *Urology,* vol. 54, 1999.

[71] M. Borse, S. Patil, and B. Patil, "LITERATURE SURVEY FOR 3D RECONSTRUCTION OF BRAIN MRI IMAGES," *International Journal of Research in Engineering and Technology,* vol. 2, 2013.

[72] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR),* vol. 38, p. 13, 2006.

[73] F. Porikli and A. Yilmaz, "Object detection and tracking," in *Video Analytics for Business Intelligence*, ed: Springer, 2012, pp. 3-41.

[74] J. Wang and M. F. Cohen, *Image and video matting: a survey*: Now Publishers Inc, 2008.

[75] K. A. Joshi and D. G. Thakore, "A survey on moving object detection and tracking in video surveillance system," *IJSCE, ISSN,* pp. 2231-2307, 2012.

[76] T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* pp. 90-99, 1986.

[77] B. Ristic, S. Arulampalam, and N. J. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*: Artech house, 2004.

[78] D. Exner, E. Bruns, D. Kurz, A. Grundhöfer, and O. Bimber, "Fast and robust CAMShift tracking," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on,* 2010, pp. 9-16.

[79] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 25, pp. 564-577, 2003.

[80] Y. Chen, Y. Rui, and T. S. Huang, "JPDAF based HMM for real-time contour tracking," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on,* 2001, pp. I-543-I-550 vol. 1.

[81] J. MacCormick and A. Blake, "Probabilistic exclusion and partitioned sampling for multiple object tracking," *International Journal of Computer Vision,* vol. 39, pp. 57-71, 2000.

[82] X. Sun, H. Yao, S. Zhang, and D. Li, "Non-Rigid Object Contour Tracking via a Novel Supervised Level Set Model," 2015.

[83] A.-R. Mansouri, "Region tracking via level set PDEs without motion computation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 947-961, 2002.

[84] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, pp. 1531-1536, 2004.

[85] T. F. Chan, Vese, L. A., "Active Contours Without Edges," *IEEE Transaction on Image Processing,* vol. 10, pp. 266-277, 2001.

[86] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision,* vol. 59, pp. 167-181, 2004.

[87] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 18, pp. 1114-1127, 2008.

[88] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR),* vol. 43, p. 16, 2011.

[89] M.-H. Yang and N. Ahuja, "Recognizing hand gestures using motion trajectories," in *Face Detection and Gesture Recognition for Human-Computer Interaction*, ed: Springer, 2001, pp. 53-81.

[90] Z. Li, Y. Fu, T. Huang, and S. Yan, "Real-time human action recognition by luminance field trajectory analysis," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 671-676.

[91] F. Bashir, A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden Markov models," *Image Processing, IEEE Transactions on,* vol. 16, pp. 1912-1919, 2007.

[92] J. C. Nascimento, M. A. Figueiredo, and J. S. Marques, "Trajectory classification using switched dynamical hidden Markov models," *Image Processing, IEEE Transactions on,* vol. 19, pp. 1338-1348, 2010.

[93] S. Sun, J. Zhao, and Q. Gao, "Modeling and recognizing human trajectories with beta process hidden Markov models," *Pattern Recognition,* vol. 48, pp. 2407-2417, 2015.

[94] R. Rosales and S. Sclaroff, "3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999.

[95] H. Ardö, K. Åström, and R. Berthilsson, "Online Viterbi Optimisation for Simple Event Detection in Video," in *International Computer Vision Summer School 2007*, 2007.

[96] S. Gong and T. Xiang, "Recognition of group activities using dynamic probabilistic networks," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 742-749.

[97] C. Vogler and D. Metaxas, "A framework for recognizing the simultaneous aspects of american sign language," *Computer Vision and Image Understanding,* vol. 81, pp. 358-384, 2001.

[98] X. Ma, D. Schonfeld, and A. Khokhar, "Distributed multi-dimensional hidden Markov model: theory and application in multiple-object trajectory classification and recognition," in *Electronic Imaging 2008*, 2008, pp. 68200O-68200O-12.

[99] S. Bauer, R. Wiest, L.-P. Nolte, and M. Reyes, "A survey of MRI-based medical image analysis for brain tumor studies," *Physics in medicine and biology,* vol. 58, p. R97, 2013.

[100] H. Wang and A. Amini, "Cardiac motion and deformation recovery from MRI: a review," *Medical Imaging, IEEE Transactions on*, vol. 31, pp. 487-503, 2012.

[101] Y. Zhang, Y. Jing, X. Liang, G. Xu, and L. Dong, "Dynamic lung modeling and tumor tracking using deformable image registration and geometric smoothing," *Computational Modelling of Objects Represented in Images III: Fundamentals, Methods and Applications*, p. 215, 2012.

[102] M. Rahim, M.-E. Bellemare, R. Bulot, and N. Pirró, "A diffeomorphic mapping based characterization of temporal sequences: Application to the pelvic organ dynamics assessment," *Journal of mathematical imaging and vision*, vol. 47, pp. 151-164, 2013.

[103] M. Cosson, C. Rubod, A. Vallet, J. Witz, P. Dubois, and M. Brieu, "Simulation of normal pelvic mobilities in building an MRI-validated biomechanical model," *International urogynecology journal*, vol. 24, pp. 105-112, 2013.

[104] M. Parente, R. N. Jorge, T. Mascarenhas, A. Fernandes, and J. Martins, "Deformation of the pelvic floor muscles during a vaginal delivery," *International Urogynecology Journal*, vol. 19, pp. 65-71, 2008.

[105] I. Nekooeimehr and S. K. Lai-Yuen, "Adaptive semi-unsupervised weighted oversampling (A-SUWO) for Imbalanced Datasets," *Expert Systems with Applications*, 2015.

[106] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," *Pattern Recognition*, pp. 1653-1672, 2014.

[107] E. M. Voorhees, "Implementing agglomerative hierarchic clustering algorithms for use in document retrieval," *Information Processing & Management*, vol. 22, pp. 465-476, 1986.

[108] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, p. 27, 2011.

[109] P. McCullagh, "Generalized linear models," *European Journal of Operational Research*, vol. 16, pp. 285-292, 1984.

[110] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, pp. 21-27, 1967.

[111] Y. Guo, T. Hastie, and R. Tibshirani, "Regularized linear discriminant analysis and its application in microarrays," *Biostatistics*, vol. 8, pp. 86-100, 2007.

[112] H. He, Y. Bai, E. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence).* , 2008, pp. 1322-1328.

[113] H. Schütze and C. Silverstein, "Projections for efficient document clustering," in *ACM SIGIR Forum*, 1997, pp. 74-81.

[114] M. Perez-Ortiz, P. A. Gutierrez, C. Hervas-Martinez, and X. Yao, "Graph-Based Approaches for Over-sampling in the context of Ordinal Regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 1233-1245, 2015.

[115] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal regression," in *Ninth International Conference on Intelligent Systems Design and Applications, 2009. ISDA 2009.* , 2009, pp. 283-287.

[116] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21-31, 2014.

[117] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Advances in neural information processing systems*, 2006, pp. 865-872.

[118] S. Onal, S. Lai-Yuen, P. Bao, A. Weitzenfeld, D. Hogue, and S. Hart, "Quantitative assessment of new MRI-based measurements to differentiate low and high stages of pelvic organ prolapse using support vector machines," *International urogynecology journal,* vol. 26, pp. 707-713, 2015.

[119] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics,* vol. 11, pp. 1-21, 1969.

[120] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of the IEEE,* vol. 61, pp. 268-278, 1973.

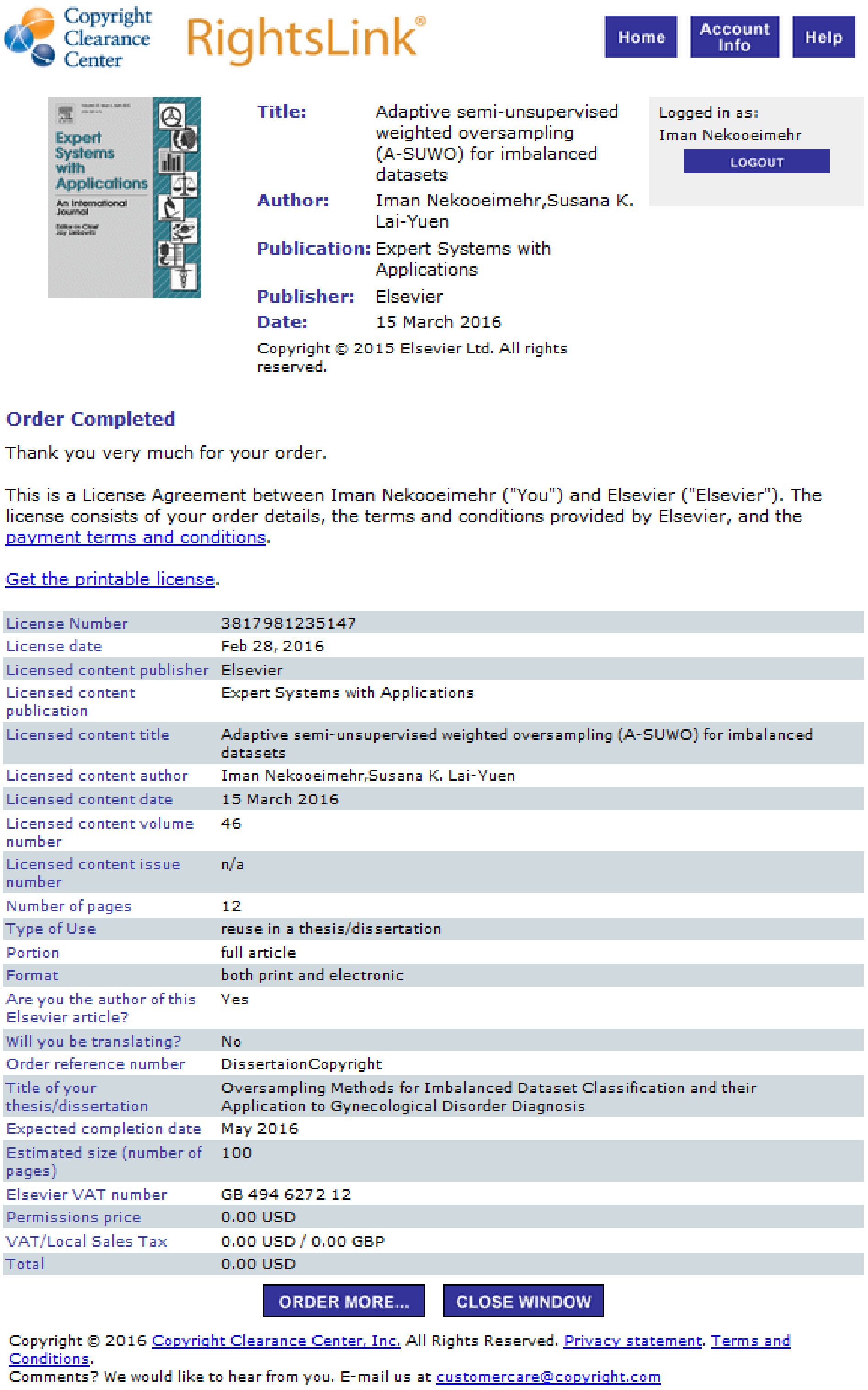
[121] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey, "Complex wavelet structural similarity: A new image similarity index," *Image Processing, IEEE Transactions on,* vol. 18, pp. 2385-2401, 2009.

[122] C. A. Woodfield, Hampton, B. S., Sung, V., Brody, J. M., "Magnetic resonance imaging of pelvic organ prolapse: comparing pubococcygeal and midpubic lines with clinical staging," vol. 20, pp. 695-701, 2009.

Appendices

Appendix A Copyright Permissions

Below is permission for the use of material in Chapter 3.



Title:	Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets
Author:	Iman Nekooeimehr,Susana K. Lai-Yuen
Publication:	Expert Systems with Applications
Publisher:	Elsevier
Date:	15 March 2016

Copyright © 2015 Elsevier Ltd. All rights reserved.

Logged in as:
Iman Nekooeimehr

Order Completed

Thank you very much for your order.

This is a License Agreement between Iman Nekooeimehr ("You") and Elsevier ("Elsevier"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

Get the printable license.

License Number	3817981235147
License date	Feb 28, 2016
Licensed content publisher	Elsevier
Licensed content publication	Expert Systems with Applications
Licensed content title	Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets
Licensed content author	Iman Nekooeimehr,Susana K. Lai-Yuen
Licensed content date	15 March 2016
Licensed content volume number	46
Licensed content issue number	n/a
Number of pages	12
Type of Use	reuse in a thesis/dissertation
Portion	full article
Format	both print and electronic
Are you the author of this Elsevier article?	Yes
Will you be translating?	No
Order reference number	DissertaionCopyright
Title of your thesis/dissertation	Oversampling Methods for Imbalanced Dataset Classification and their Application to Gynecological Disorder Diagnosis
Expected completion date	May 2016
Estimated size (number of pages)	100
Elsevier VAT number	GB 494 6272 12
Permissions price	0.00 USD
VAT/Local Sales Tax	0.00 USD / 0.00 GBP
Total	0.00 USD

Comments? We would like to hear from you. E-mail us at customercare@copyright.com

About the Author

Iman Nekooeimehr received his BSc degree in Industrial Engineering from Sharif University of Technology, Tehran, Iran, in 2011 and his MSc in Industrial Engineering from University of South Florida, Tampa, Florida in 2013. He is currently a PhD candidate and graduate assistant at the University of South Florida. His research interests include Machine Learning, Data Mining for Medical Diagnosis and Medical Image Processing.