QATAR UNIVERSITY

COLLEGE OF ENGINEERING

CONCEPTUAL DATA SAMPLING FOR IMAGE SEGMENTATION- AN APPLICATION FOR BREAST CANCER IMAGES

BY

ZAINAB KHALID AWAN

A Thesis Submitted to the Faculty of the
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of
Master of Computing

June 2017

© 2017 Zainab Khalid Awan. All Rights Reserved.

COMMITTEE PAGE

The members of the committee approve the Thesis of Zainab Khalid Awan defended on May 23, 2017:

Dr. Ali Jaoua

Thesis Supervisor

Dr. Somaya Al-Madeed

Thesis Co-Supervisor

Dr. Farida Cheriet

Committee Member

Dr. Uvais Qidwai

Committee Member

Approved:

Dr. Khalifa Nasser Al-Khalifa, Dean, College of Engineering

Abstract

Awan, Khalid, Zainab, Masters:

June: 2017, Master of Computing

Title: Conceptual Data Sampling for Image Segmentation- An Application for Breast Cancer Images

Supervisor of Thesis: Dr. Ali Jaoua

At the present time data analytics have become a buzzword for the information technology sector. In an attempt to analyze data; one may follow various paths. Be it deploying sophisticated technologies to process big data or using commodity hardware while applying data reduction/sampling techniques to draw meaningful insights from a data. In this thesis, we aim to reduce data size in terms of th e number of tuples/objects for a given data. Our method has driven its roots from formal concept analysis (FCA); which is a mathematical framework for data analysis. The proposed transformation is preserving functional dependencies/implications in a database. Consequently, we can generate a much smaller data sample that is able to help in making decisions. In this study, we analyze a variety of reduction methods in order to recognize the best one(s), including randomized object selection procedures. The accuracy of the decision s made on generated sample is comparable to accuracy of the decision made of whole/original data. To illustrate the concept we have chosen data from medical image domain. The data used for experimentation contains microscopic images of breast cancer that need to be segmented into two categories; i.e. benign or malignant. Extensive set of experiments have been performed to show the strength of the proposed reduction method.

Dedication

To my son Mustafa, there is no greater honour in this world than being your mother

Acknowledgements

I would like to thank ALLAH *SWT* for blessing me with much more than I deserve. I owe a *lot* of gratitude for my brilliant supervisors Dr. Ali Jaoua and Dr. Somaya Al-Madeed for believing in me when I had lost faith in myself. A deep thanks for your *incredible* support throughout research journey. Thanks to the *wonderful* team led by Dr. Ali, they know who they are. I want to thank Fahad and Eman for always being there to clarify my confusions, whenever I had. I have to thank Dr. Nasir Rajpoot and Dhoha Abid for providing us with the data.

Thanks to my grandfather, *Ibrahim Awan*, who is no more with us, for always supporting and encouraging me to educate myself and be financially independent. He would have been so proud, if he were alive.

Thanks to my husband, *Zahoor*, for breaking the stereotypical gender roles and helping me in house chores and taking care of my kids. Thanks for driving me to the university and waiting for me while sitting in the car with a toddler. My son, *Mustafa* has been a part of this journey as much as myself. Where my daughter, *Fatimah*, had joined us when it was pretty much done.

Thanks to my *wonderful* parents and my brothers *Saifullah* and *Ubaidullah* for always being there through my thick and thin. This was not possible without all of you.

"This contribution was made possible by NPRP grant 07-794-1-145 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors."

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Data mining on complete datasets may take very long time. To prevent delays *data reduction* strategies are used to have a reduced representation of data yet analytically sound decisions. In order to do that one may apply dimensionality/attribute reduction or numerosity reduction. In former the number of features is reduced by applying for example **principal component analysis**. While in later, the volume of data gets reduced by applying for example, **sampling**.

Our work revolves around proposing a novel sampling method that could save computation burden for computers. Sampling refers to the process of taking a predetermined number of observations from the total population to have a smaller representative subset of a whole dataset.

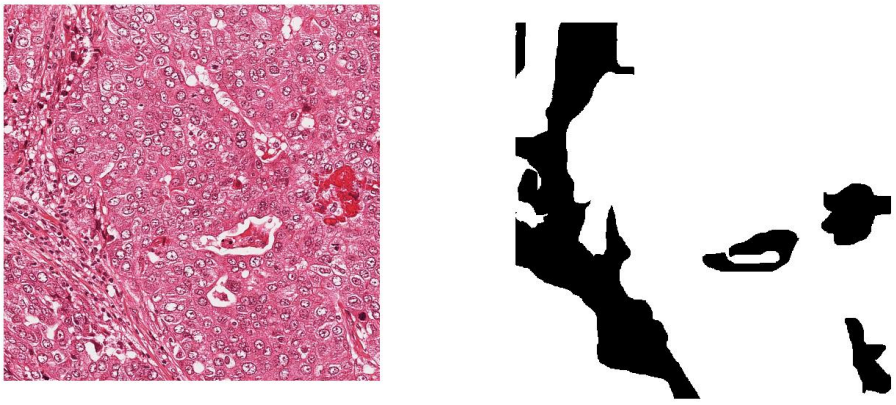
While one might argue that this is an era of *Big Data* with availability of *High Performance Computing*; we no longer need sampling methods. But the truth is contrary to this; simply throwing mountains of data to conventional machine learning algorithms will not help to build accurate models. With bigger data accurate models could be built. Accurate models add greater business value. But for that; we need to pay the price, in terms of high end and sophisticated big data technologies. Our sampling method shows the potential of not losing subtle patterns of a data in its generated sample. The proposed method is non-parametric, hence does not impose strict assumptions on data

distributions.

1.1 Image Segmentation

To provide experimental evidence we have chosen breast cancer microscopic images data. We opted for cancer images as *cancer* is reported [4] to be the *5th* cause of death in the world. The purpose here is to perform image segmentation on cancer images. Image segmentation is the process of assigning class labels to pixels in an image i.e. tumor (malignant) or non-tumor (benign) in our case. Image pixels that are annotated with same label, together form a region or segment. Within that region the physical characteristics are same. To illustrate this process, we have given Figure 1.1. Wherein, Figure 1.1a shows the initial microscopic image that is obtained from a patients' tissue. It serves as an *input* to the segmentation process. This image is then converted to a *mineable data* by extracting features from it. If necessary, feature space is reduced before applying machine learning techniques to annotate the pixels. Figure 1.1b is output of the segmentation process. Image segmentation should result into clear boundaries between different kinds of regions as in Figure 1.1b the *white* regions indicate cancer and *black* regions are non-cancer..
Breast cancer is increasingly becoming a commonly occurring disease. To determine the stage and severity of the disease, a histopathologist needs to sit in lab and examine microscopic image of breast tissue. The idea here is to automate this grunt work in order to save time and define treatment plan for a patient in timely manner.

In order to further speed-up the process data reduction strategies can be applied. This is why, we apply our sampling method on pre-processed im-



(a) Microscopic image of breast cancerous tissue (b) Ground truth

Figure 1.1: An instance of *image segmentation*; where first subfigure in the input and second is the output

ages data and compare it with two existing sampling methods. We apply this on a public *dataset*, made available by ICPR 2012[1]. The data has 50 H& E stained breast cancer tissue images from 5 patients. Each patient has contributed for 10 images. We elaborate more on the dataset in section 4.1.

1.2 Research Questions

In this thesis we pursue the following research questions:

- Is the proposed sampling method able to extract a subset of a given dataset that could generate sufficiently accurate classification that is as good as the classification done on complete dataset?
- How *applicable* is the proposed framework in the field of breast cancer image segmentation?

[1]http://ipal.cnrs.fr/ICPR2012

1.3 Contributions

Our contribution in this work is two-fold;

1. Our first contribution is developing a framework that enables us to extract representative subset/sample from datasets. Our framework is currently a working implementation in JAVA.

2. Secondly, extensive experiments have been performed on a dataset from medical image domain to provide experimental evidence of efficacy of the proposed framework. Experiments have been done using statistics and machine learning toolbox MATLAB[2].

1.4 Outline

We describe pertinent background literature in chapter 2. We dedicate chapter 3 to explain the proposed method. In chapter 4 we address the evaluation of our method. We finish with chapter 5, which gives concluding remarks and presents future directions.

[2]https://www.mathworks.com/products/statistics.html

Chapter 2: Background and Literature Review

In this chapter we provide related background concepts, theories and state-of-the-art that has been done in this domain. This will serve as a basis for what we will propose in the next chapter. We discuss formal concept analysis, sampling methods and image segmentation.

2.1 Formal Concept Analysis *FCA*

It is generally believed that origin of Formal Concept Analysis *FCA* can be dated back to 1982 with work of [35] with some previous attempts. *FCA* is a method of data analysis that has been applied across various domains such as information visualisation [29], feature reduction [21],image mining [36] and decision making [37]. FCA is mathematical tool that is used to build a concept lattice for a binary relation by using its attributes and objects. In FCA data is formally represented as a formal context (FC). The process of building FC has been very well explained in [26].

"A ***Formal Context (FC)*** is a triplet k=<G,M,I> where **G** is a finite set of elements called objects, **M** is a finite set of elements called attributes and I is a binary relation between G and M" [13]. Table 2.1 is an example of FC with G= {Lion, Finch, Eagle, Hare, Ostrich} and M= {Prey, Fly, Bird, Mammal} with I(Lion, Mammal)=1.

Table 2.1: Formal Context example

k	Prey	Fly	Bird	Mammal
Lion	1	0	0	1
Finch	0	1	1	0
Eagle	1	1	1	0
Hare	0	0	0	1
Ostrich	0	0	1	0

2.2 Functional Dependency

Attribute implications in FCA enable data analysis. Attribute implications convey a *dependency* that is valid in data. For example, if two employees of a company have same zip code in a database then they belong to the same city. *Dependencies* are a form of *constraint* that exists between attributes of database. A functional dependency can be formally defined as;

"If R is a relational schema and $X \subseteq R$ and tuples are represented by t then, *functional dependency*, represented by $X \rightarrow A$ holds in r if and only if $\forall\, t_i, t_j \in r$, $t_i[\mathrm{X}] = t_j[\mathrm{X}] \Rightarrow t_i[\mathrm{A}] = t_j[\mathrm{A}]$". To illustrate the concept see Table 2.2. Functional dependencies in a database are equivalent to attribute implications in FCA [8].

Table 2.2: Relation in which *functional dependency* holds from **w**→**z** and vice versa. When attribute **w** takes the same value attribute **z** also takes the same value

w	x	y	**z**
1	2	9	**3**
8	9	16	**5**
1	4	1	**3**
8	0	0	**5**

2.3 Incremental Lukasiewicz Data Reduction

A data reduction algorithm for formal contexts, based on Lukasiewicz implication is proposed in [12] and has been updated by the work of Eman et al. [26]. The original algorithm followed two steps;

- Apply Lukasiewicz implication to find any objects in a database that can be verified by other objects
- Remove the objects that could be verified by other objects

2.3.1 Definition

We are using standard definition from [26]. Let R be a fuzzy binary relation defined on universal set U. If we have two sets, $A \subseteq$G and B is a fuzzy set defined on M. Where, G is a finite set of objects and M is finite set of attributes and precision level, *delta* $\delta \in [0,1]$. The two operators *f(A)* and h_δ are defined as;

- *f(A)*=$\{$ d/α, α=min(μR(g,d)|g$\in$A)$\}$, d$\in$ P. Where, A$\subseteq$G and P$\subseteq$M
- h_δ(B)=$\{$g|d$\in$p$\Rightarrow$ (μB(d)$\rightarrow_L$ μR(g,d))$\geq$ δ. Where, $\rightarrow_L$ is known as *Lukasiewicz implication*
- a$\rightarrow_L$b is given as *min*(1,1-a+b) ; a,b $\in$ [0,1]

Eman et al. [26] have made this approach incremental. By incremental means, instead of waiting for whole FC to be built and then apply reduction rather reduction is applied *on the fly* while FC is being built. It serves two purposes, *firstly*, it minimizes *idle time* as reduction is applied on package sized FC. *Secondly,* it reduces memory overload by not building FC of size n^2. This approach allows us to reduce the number of objects in database. We started off our work with applying this method on our data and have discussed the results in Chapter 4.

Table 2.3: Formal Context on which Lukasiewicz Reduction will be applied

-	a	b	c	d
(t1,t2)	0	0	1	0
(t1,t3)	1	1	1	0
(t1,t4)	1	1	1	0
(t2,t3)	**0**	**0**	1	0
(t2,t4)	1	1	**1**	**0**
(t3,t4)	1	1	1	1

As an *example* of application of Lukasiewicz Reduction we consider a formal context in Table 2.3. The precision level, δ is '1'. Let X be first tuple (t1,t2) and A is the second tuple (t2,t3) from Table 2.3. We want to know if X can be verified by other objects. For this we apply *Lukasiewicz implication.*

- min(1,1-X(a)+A(a))=min(1,1-0+1)=1 $\geq$ 1
- min(1,1-X(b)+A(b))=min(1,1-0+1)=1 $\geq$ 1
- min(1,1-X(c)+A(c))=min(1,1-1+1)=1 $\geq$ 1
- min(1,1-X(d)+A(d))=min(1,1-0+0)=1 $\geq$ 1

A is verifying X, in this way we check for all the remaining objects if they verify X. We found that all the objects verify X. As a last step we compute the difference between X and the minimum. Minimum values are bold faced in Table 2.3. Minimum is found by taking minimum along each attribute. X can be removed if its values are $\geq$ to the *minimum.* In the same way all objects are checked for the possibility of being verified by other objects. The reduced F is given in Table 2.4.

Table 2.4: Reduced Formal Context after applying Lukasiewicz Reduction

-	a	b	c	d
(t2,t4)	1	1	1	0
(t3,t4)	1	1	1	1

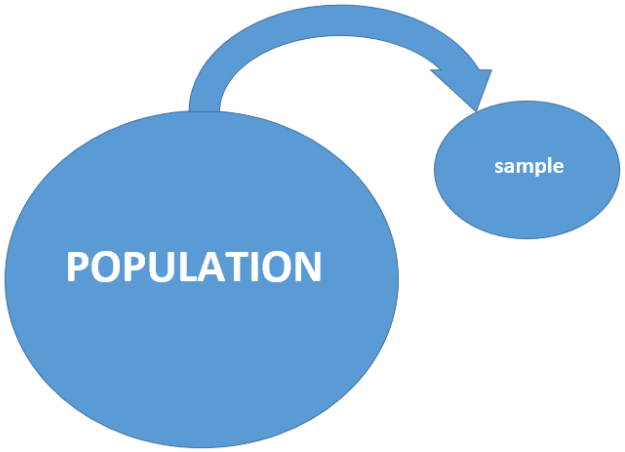


Figure 2.1: Sampling Process

2.4 Sampling Methods

Data analytics involve examining data in order to draw meaningful information from it. Since, the data available for analysis from different domains such as social media, medicine and web is increasing at an unprecedented pace. If the goal is not to analyze whole available data the solution is to perform sampling. Sampling refers to the process of drawing representative data points from any data that helps in making inferences as shown in Figure 2.1.

The accuracy of decision making is primarily dependant on the quality of sample generated. Therefore the sampling process should be reliable, robust and representative.

Sampling designs are categorized to two main types; *probabilistic sampling* and *non-probabilistic sampling*. In former type, the points are chosen randomly and the probability of a point being chosen is known beforehand. In later type, probabilities are not assigned rather samples are drawn based on the purpose of analysis, this is why this is referred to as *purposive sampling*. Non-probabilistic methods are purely subjective as to it depends on the reason why one wants to extract a sample. Therefore, we have narrowed down our

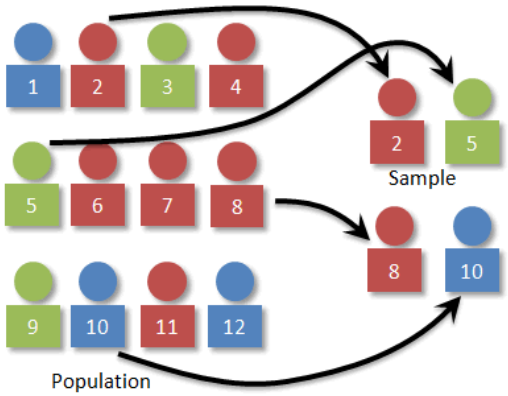


Figure 2.2: Random Sampling [1]

focus to probabilistic methods. Widely used probabilistic sampling methods include simple random sampling, stratified sampling, cluster sampling and systematic sampling. Each one of them has been discussed in upcoming sections.

2.4.1 Simple Random Sampling (SRS)

In this straightforward method each point in the population stands equal chance of being chosen. This is easy to implement but it is not efficient [32]. As the points in the sample might not be discriminative and representative of data. Figure 2.2 shows a population of twelve data points. When a sample of size 4 was to be chosen from population; data point 2, 5, 8 and 10 was chosen randomly.

For mass spectrometry data [7], a sampling technique based on *simple random sampling* is proposed. Mass spectrometry data is obtained in chemical labs by ionizing chemicals and is useful for many biological applications. The data obtained is quite big and need big data technologies. In this work, the authors were able to reduce data by 20% by systematically doing random sampling.

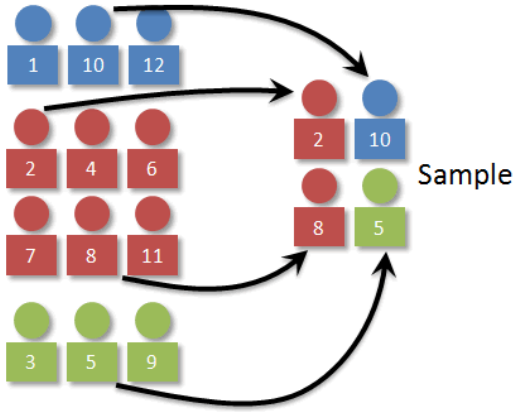


Figure 2.3: Stratified Sampling [1]

2.4.2 Stratified Sampling (SS)

In this method whole population is first divided into groups or stratum based on certain demographics such as age or gender. Afterwards, random data points are chosen from each stratum based on the proportions [9]. It is more efficient as compared to SRS. Figure 2.3 shows an instance of stratified sampling. In which one-fourth of original data is blue points so the sample should also contain one-fourth blue points. This applies to rest of the strata as well.

Most *naive* way to do image segmentation is by doing *thresholding* of a grayscale image. If intensity of image pixel is greater than a certain threshold than the binary image contains **1** otherwise **0**. Yunzhi et. al [19] have proposed automatic multilevel thresholding based image segmentation by doing stratified sampling as their first step.

2.4.3 Cluster Sampling

In this type of sampling the population is divided in to clusters/groups and then randomly some clusters (*whole*) are picked to be included in sample. The

Figure 2.4: Cluster Sampling [1]

samples generated are not representative of population. This must not be confused with stratified sampling. It is applied only when a population is already divided in the form of clusters or groups. Figure 2.4 shows an example of cluster sampling. Wherein, we have six clusters in total and we want a sample of 4 data points. So we will choose two clusters randomly.

2.4.4 Systematic Sampling

In systematic sampling, samples are chosen based on a parameter called *sampling interval* which is represented by $\boldsymbol{k}$. Sampling interval is calculated by dividing size of total population with the size of desired sample. In figure 2.5 where the size of k is 2; every third data point will be included in the sample. It gives best results when the data points depict linear ordering [6].

2.5 Image Segmentation

Image segmentation refers to the process of assigning labels to each pixel of an image to a particular category. Appropriate example for this is its application to a microscopic image of a cancerous tissue. Pathologist is a person who

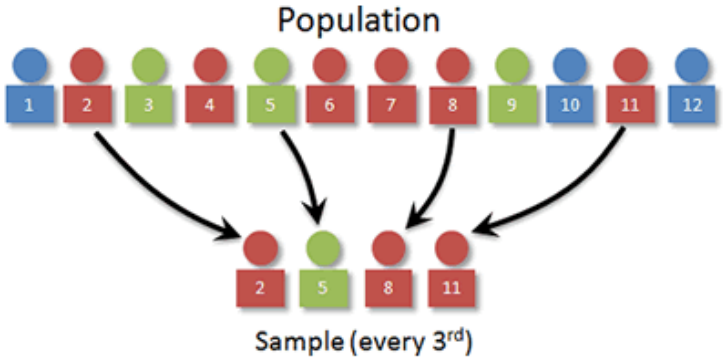


Figure 2.5: Systematic Sampling [1]

studies the cell structure in microscopic image of tissues and decides on the category of cell to be benign or malignant. There have been attempts in the literature to make this process automatic for colon cancer [15],lung cancer [31], pancreatic cancer [30] and breast cancer [24], [25]. For our purposes we have investigated breast cancer images. There is no universal method that exists for this process and it is an area of ongoing research. Different methods have been adopted to perform image segmentation which includes classification and clustering. The main work-flow includes following steps

1. **Pre-processing** is a preliminary step for any image analysis task [17] as the image data is inconsistent and noisy. It involves removal of noise and distortions. In [20] RGB images were converted to grey scale and noise was removed by using median filter. Histogram based noise removal method was adopted in [38]. In which only those pixels were chosen for further processing that have passed a certain threshold in gray scale. In our work the pre-processing is done in three steps; images are first subjected to *stain normalization* then *conversion to gray scale* and lastly *smoothing* is applied.

2. **Feature Extraction** The preprocessed images are then used for *feature extraction* task. For image segmentation purpose three main kinds of features can be used; spectral, contextual and textural [16]. Spectral

features signify the tonal variations in infra-red and visible parts of electromagnetic spectrum. Contextual features contain information about the surroundings of an image segment that is being analysed. Textural features contain information that is much more meaningful than spectral or contextual features. Texture contains information about structural arrangement of tumor regions. The feature vector used in our work contains 8 features in total called MR8 features which are textural features. Those features have been extracted by applying MR8 filter bank [2]. The feature extraction task has been done and kindly provided to us by [3]. These features are used for learning process.

3. **Feature Reduction/Selection** This step involves reducing the dimensionality of feature vector if it is very large. In the past *FCA* has been applied to the reduce the feature dimensionality in the image data [21]. Maximum relevance and minimum redundancy techniques have also been applied to select most relevant features [11]. However, for our work we have not implemented this step as the feature vector is already small.

4. **Classification/Clustering** In this step machine learning frameworks are used to classify the data that we get from previous step. There have been attempts in the literature that employ neural networks [33], [27] and support vector machines [34], [28], [14].

Breast cancer grading is another important area relevant to classification of tumor to benign and malignant. This is used to determine the severity of the malignancy. For that glandular segmentation has been reported in [23] , [22]. For glandular segmentation morphology features are used in addition to texture features [23].

2.6 Summary

In this chapter we have explained the relevant background that includes FCA, sampling methods and image segmentation. We have also pointed out some closely related research attempts in those areas. On the basis of the concepts discussed here we detail our methodology in the next Chapter.

Chapter 3: Approach/Methodology

This chapter introduces the methodology that we have adopted to pursue our goals. On the top level it involves; acquiring images, preprocessing and extracting features from images, applying our sampling method, testing on machine learning classifiers. The first two steps were done by [3]. We have applied our method on the given data and test it on machine learning algorithms.

Our goal here is to explain in depth the proposed technique. In upcoming sections sampling method is presented along with a concrete example to materialize our method. The method encompasses four steps to generate subset of a given data. Those steps include; *building formal context of data*, *mapping formal context to pattern table*, *estimating pattern proportions* and *linking back to original data to get a subset/sample.*

3.1 Nomenclature

Here we explain some of the terms that are used in the upcoming sections; to prevent a reader from confusion.

Pattern Table (PT) It is a table that contains binary combinations for a given feature vector. The number of combinations is equal to $2^{no.of features.}$. For a feature vector of size two, Table 3.1, shows its corresponding pattern table.

Table 3.1: Pattern Table for two features

0	0
0	1
1	0
1	1

Counter Table (CT) It contains the count/frequencies of occurrence each pattern.

Multiplier It is a number that is multiplied with proportions obtained from the counter table. The resultant is used to extract sample from the original dataset. Larger databases need bigger multiplier values as compare to smaller ones.

Terms that are used interchangeably include;

- Relation/database/dataset
- subset/sample
- attributes/features

3.2 Input Data

As explained in the feature extraction step of the section 2.5 the input data has 8 features. To illustrate, we have randomly chosen 8 data points from whole data. The Table 3.2 has 9 columns, wherein first 8 of them from x1-x8 represent the feature vector and the 9th column in class label. If the class label is **0** then it is *benign* otherwise *malignant*. After we get our data in the form shown in Table 3.2 we split the data into its respective classes. In our case we have only two classes i.e. **1** and **0**, so we have two datasets. Idea of splitting

Table 3.2: Input Data

-	x1	x2	x3	x4	x5	x6	x7	x8	label
t1	3.4148	1.015	1.2909	1.1316	1	1.1461	1.0805	1.0239	0
t2	2.21	1.0052	1.1696	1.1191	1.0517	1.0602	1.0135	1.0002	0
t3	2.1354	1.0124	1.0151	1	1.0123	1.1693	1.071	1.0172	0
t4	1.3433	1.0144	1.0662	1.0613	1.0397	1.1184	1.0547	1.0121	0
t5	1.4644	1.0068	1.032	1.0262	1.0228	1.0638	1.0236	1.0115	1
t6	1.8885	1.0085	1.0988	1.0599	1.0199	1.0289	1.0192	1.0122	1
t7	1.7668	1.0095	1.0611	1.0927	1.0828	1.0544	1.0336	1.0176	1
t8	1.0265	1.0056	1.0454	1.0252	1	1.0557	1.0259	1.0098	1

is influenced by the concept of stratified sampling. As we want to take sample size of each class which is proportionate to its size in whole dataset.

3.3 Formal Context Generation

To build a binary FC of data in Table 3.2 a pairwise comparison is performed among all the tuples. If a data has n rows, the FC of that data would be of the size, Equation 3.1,

$$FC \quad size = Min(2^{no.of features}, n * \frac{(n-1)}{2}) \tag{3.1}$$

An FC object corresponds to a pairwise comparison of two objects in a given data. Binary FC object is combination of 1's and 0's based on the similarity between the two objects that together constitute it. If two data objects are similar we put 1 in FC and 0 otherwise. We can decide on the similarity of pairwise comparison. It could be either pure equality '==' or based on the similarity measure [10]. The similarity measure algebraically is in Equation 3.2;

$$SimilarityMeasure = 1 - \frac{|n_1 - n_2|}{max(n_1, n_2)} \tag{3.2}$$

If the similarity is greater than a specific threshold, for example 90%, then FC value would be 1 otherwise 0. As we can see from the table 3.2 that none of the two values across the tuples are exactly equal to each other. By virtue of this, FC for 100% equality contains all rows as *'0'*.

Therefore, we consider the similarity index 90%. While building FC, we do not consider the last column in the Tables 3.3 and 3.5 as it contains class labels only. The given databases in Tables 3.3 and 3.5 have been converted to their respective formal contexts in Tables 3.4 and 3.6.

Table 3.3: DBI Malignant

ID	x1	x2	x3	x4	x5	x6	x7	x8	label
t1	1.4644	1.0068	1.032	1.0262	1.0228	1.0638	1.0236	1.0115	1
t2	1.8885	1.0085	1.0988	1.0599	1.0199	1.0289	1.0192	1.0122	1
t3	1.7668	1.0095	1.0611	1.0927	1.0828	1.0544	1.0336	1.0176	1
t4	1.0265	1.0056	1.0454	1.0252	1	1.0557	1.0259	1.0098	1

Table 3.4: FC for DBI Malignant

–	x1	x2	x3	x4	x5	x6	x7	x8
(t1,t2)	0	1	1	1	1	1	1	1
(t1,t3)	0	1	1	1	1	1	1	1
(t1,t4)	0	1	1	1	1	1	1	1
(t2,t3)	1	1	1	1	1	1	1	1
(t2,t4)	0	1	1	1	1	1	1	1
(t3,t4)	0	1	1	1	1	1	1	1

Table 3.5: DBI Benign

ID	x1	x2	x3	x4	x5	x6	x7	x8	label
t1	3.4148	1.015	1.2909	1.1316	1	1.1461	1.0805	1.0239	0
t2	2.21	1.0052	1.1696	1.1191	1.0517	1.0602	1.0135	1.0002	0
t3	2.1354	1.0124	1.0151	1	1.0123	1.1693	1.071	1.0172	0
t4	1.3433	1.0144	1.0662	1.0613	1.0397	1.1184	1.0547	1.0121	0

Table 3.6: FC for DBI Benign

–	x1	x2	x3	x4	x5	x6	x7	x8
(t1,t2)	0	1	1	1	1	1	1	1
(t1,t3)	0	1	0	0	1	1	1	1
(t1,t4)	0	1	0	1	1	1	1	1
(t2,t3)	1	1	0	0	1	1	1	1
(t2,t4)	0	1	1	1	1	1	1	1
(t3,t4)	0	1	1	1	1	1	1	1

3.4 Mapping Formal Context to Pattern Table

In this step a set of binary patterns is generated, which is called pattern table. Total number of binary patterns is less than or equal to $2^{no.of features}$.

For the current DBI as we have 8 features so the PT is going to have 2^8=256 patterns. Now, each FC object is linked to its corresponding binary combination in pattern table. Table 3.7 shows the pattern table, whose first column indicates nothing but decimal equivalent of a binary combination. For the sake of brevity only those patterns are shown which actually correspond to FCs (tables 3.6 and 3.4) of the two databases (tables 3.3 and 3.5). Each time when a FC object matches a binary pattern in PT, its associated counter is increased in the CT. Last two columns of Table 3.7 are the counter tables for data. For example the 127th row in Table 3.7 which is 01111111 occurs 5 times in Malignant DBI 3.4 and 3 times in Benign DBI 3.6, so these are the counts in counter tables.

3.5 Calculation of Pattern Proportions

After mapping each FC object to its corresponding binary pattern, now we calculate the proportion of occurrence of each binary pattern. Last two column of Table 3.7 give the frequency of each pattern in FC. We calculate proportions

Table 3.7: Mapping FC to PT

-	-	-	-	-	-	-	-	-	Count for DBI Malignant	Count for DBI Benign
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	1	1	0	0
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0	1	0	0
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
79	0	1	0	0	1	1	1	1	0	1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
95	0	1	0	1	1	1	1	1	0	1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
127	0	1	1	1	1	1	1	1	5	3
.	.	.	.	.	.	.	.	.	.	.
207	1	1	0	0	1	1	1	1	0	1
.	.	.	.	.	.	.	.	.	.	.
255	1	1	1	1	1	1	1	1	1	0

Table 3.8: Calculation of Proportions

(a) Malignant DBI

Pattern No.	Proportion
127	5/6=0.83
255	1/6=0.16

(b) Benign DBI

Pattern No.	Proportion
79	1/6=0.16
95	1/6=0.16
127	3/6=0.5
207	1/6=0.16

based on the following formula, Equation 3.3;

$$Pattern \quad Proportion = \frac{Count \quad of \quad Pattern \quad from \quad CT}{No. \quad of \quad FC \quad objects} \tag{3.3}$$

The proportions for the PT 3.7 are given in Table 3.8 ;

Table 3.9: Sample Size

(a) Malignant DBI Sample Size

Pattern No.	Sample Count
127	1
255	1

(b) Benign DBI Sample Size

Pattern No.	Sample Count
79	1
95	1
127	1
207	1

3.6 Generating Sample based on Pattern Proportions

In this last step of our method we link back PT to FC and from FC to DBI. From the Table 3.8 we have the proportions of each pattern. We need to multiply those values with a number called *multiplier*. It could be 7, 20 or 100 even bigger. If the proportions values are already big enough we might not even need to multiply. But if proportions are very small as in this case we may multiply if we need a bigger sample or we can just round those proportions to 1. The size of the input data tells us what a multiplier should be. In this example, since the data is already very small, only 10 rows, still we multiply proportions with a number then as a result our sample will cover whole dataset. Then there is no point in applying this method. So, we round the proportions to 1 and link back to FCs.

Table 3.9 is rounded off version of Table 3.8. So now we take one sample from 127th and 255th row of pattern table for Malignant DBI. Similarly, 1 sample from 79th, 95th, 127th and 207th row of Benign DBI.

From the Table 3.4 we take first row (t1,t2),which corresponds to 127th row in PT, and fourth row (t2,t3), which corresponds to 255th row in PT.

From the Table 3.6 we take (t1,t3) for 79th row in PT, (t1,t4) for 95th row in PT, (t3,t4) for 127th in PT and (t2,t3) for 207th row in PT. Our sam-

pled database for Malignant class contains tuples t1,t2 and t3 and t4 is being removed. Whereas for benign class all the tuples are required to generate a sample. Hence, we are not loosing too much information. This signifies that benign class needs more instances to represent its distribution. At the end we combine the samples from both classes and the final output. Our sampled database is in Table 3.10 which has 7 rows in total as compared to whole dataset which was 8 rows. This might look insignificant, because the example is quite small. However, in the evaluation chapter we show the real strength of our method where, for example, in some cases the whole data had 30,000 rows it could be represented as 1300 rows only by using our method.

Table 3.10: Sampled Data

-	x1	x2	x3	x4	x5	x6	x7	x8	label
t1	3.4148	1.015	1.2909	1.1316	1	1.1461	1.0805	1.0239	0
t2	2.21	1.0052	1.1696	1.1191	1.0517	1.0602	1.0135	1.0002	0
t3	2.1354	1.0124	1.0151	1	1.0123	1.1693	1.071	1.0172	0
t4	1.3433	1.0144	1.0662	1.0613	1.0397	1.1184	1.0547	1.0121	0
t5	1.4644	1.0068	1.032	1.0262	1.0228	1.0638	1.0236	1.0115	1
t6	1.8885	1.0085	1.0988	1.0599	1.0199	1.0289	1.0192	1.0122	1
t7	1.7668	1.0095	1.0611	1.0927	1.0828	1.0544	1.0336	1.0176	1

3.7 Pseudocode

Algorithm 1: Pattern based Proportional Sampling

1: $\boldsymbol{X}$=Input Dataset
2: Transform $\boldsymbol{X}$ to ***Formal Context***
3: Map ***Formal Context*** to ***Pattern Table***
4: Estimate ***Proportion*** of each pattern
5: Link back ***Pattern Table*** to $\boldsymbol{X}$
6: Generate ***Sample***

3.8 Efficient Variants of Baseline PPS Method based on Objects

We have two variants of the basic algorithm. These two tweaks are done at the stage where we link pattern table back to FC and choose objects that are to be included in the final sample.

3.8.1 Maximize Overlap

After we link back PT to FC we choose objects from FC by maximizing overlap between objects. For example in Table 3.4 (01111111) from PT is linked back to (t1,t2), (t1,t3), (t1,t4),(t2,t4) and (t3,t4). If we had to choose 2 objects, for the baseline method we will choose randomly any two out of the five objects. But, to maximize overlap we will choose objects that correspond to same rows in the DBI. So, if we choose (t1,t2) then the other object must be (t1, t3) , (t1,t4) or (t2,t4) but it cannot be (t3,t4).

3.8.2 Minimize Overlap

As opposed to the concept of maximizing overlap, continuing with the same example as in last section 3.8.1, the two objects must be (t1,t2) and (t3,t4).

3.9 Efficient Variants of Baseline PPS Method based on Proportions

In an attempt to improve over the basic method we did some tweaks to the algorithm, mainly in the last step, where we choose samples. In the section 3.6 we have explained how we pick our samples based on the calculated proportions.

3.9.1 Pattern Based Proportional Sampling with Minority Bias MB-PPS

In this variant of the algorithm, we take into consideration all the patterns with proportion >=0. For the patterns with proportion <1 we take one sample only.

3.9.2 Pattern Based Sampling- Without Proportions PS

Here we force the algorithm, to choose only one sample corresponding to each pattern. We neglect the proportions here. This way the sample size gets reduced even more than previous methods.

3.10 Supervised Learning Frameworks

In this section we explain the tools and metrics that have been used to evaluate our method. *Supervised learning*, also referred to as *classification* means that machine learning algorithms learn from labelled data. The data contains fea-

ture vectors and categories/labels across each instance of feature vector. This data is used for training a model. At the end, the *trained model* is tested on completely unseen data. Testing gives us clear idea about how well the model has generalized with the given set of examples/data. The problem at hand is a *binary classification problem*. We have tested our samples on five different classifiers. Those are artificial neural networks (patternet PN, cascadeforwardnet CFN and feedforwardnet FFN), support vector machine SVM, and naive bayes NB. We discuss them one by one in next subsections.

3.10.1 Artificial Neural Network

Artificial Neural Networks have been inspired from human brain. ANN tries to mimic the way in which human brain works. There are 60 trillion neurons in the human brain and they are massively interconnected with each other. There are three layers in ANN; *(1)input layer, (2)hidden layer and (3)output layer*. Input layer has the number of neurons which is equal to the number of features. There can be more than one hidden layer and number of neurons can be adjusted. Output layer has neurons equal to number of classes. Weights of neurons are adjusted during the training by iteratively propagating back the error and reaching a stable state. We have used three types of *ANN*; patternnet PN in Figure 3.1, cascadeforwardnet CFN Figure 3.2 and feedforwardnet FFN Figure 3.3. In FFN each layer is connected with its subsequent layer and PN is a FFN for which target data is represented differently than FFN. Whereas, CFN are similar to FFN but each hidden layer and even output layer is connected with input layer.

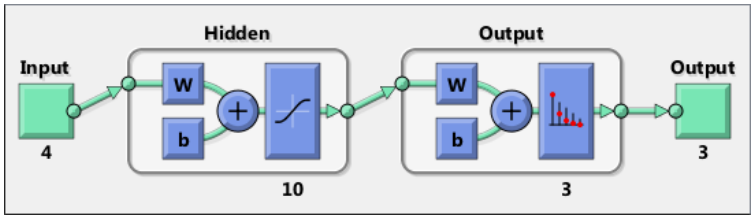


Figure 3.1: ANN Patternnet - MATLAB

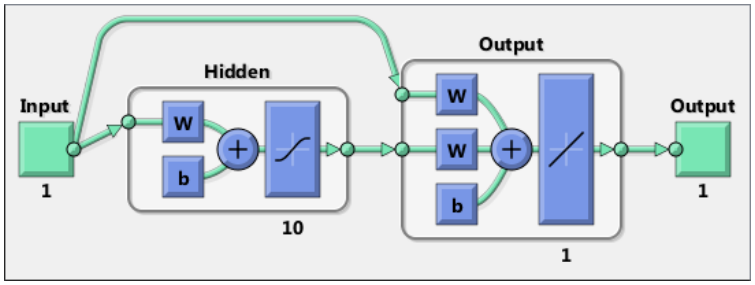


Figure 3.2: ANN Cascadeforwardnet - MATLAB

3.10.2 Support vector Machines

This is a kind of classifier that maximises the margin between itself and nearest data points of two distinct categories to find optimal hyperplane. Figure 3.4 shows *hyperplane* which not only linearly classifies data but also maximise the margin between itself and two support vectors. Linear SVM can be represented in the form of Equation 3.4. Where x is actually the feature vector, w is called weights and b is a *bias*. In our case we have used linear SVM.

$$f(x) = w(x) + b \tag{3.4}$$

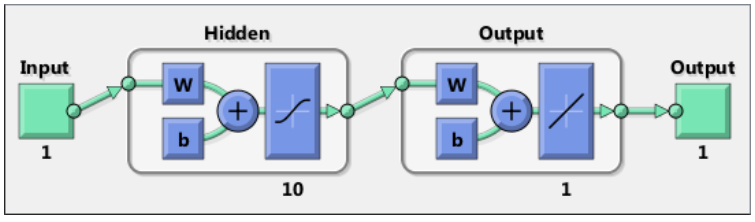


Figure 3.3: ANN Feedforwardnet - MATLAB

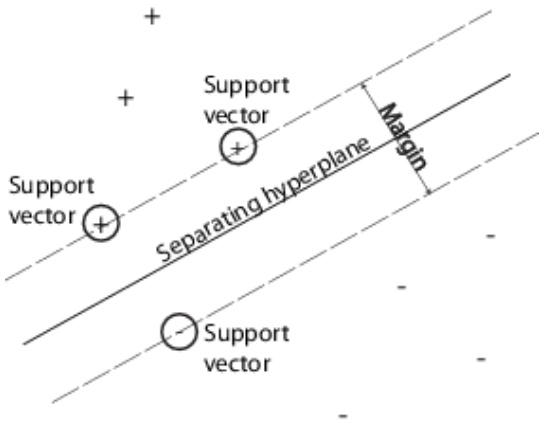


Figure 3.4: SVM Hyperplane

3.10.3 Naive Bayes

Naive Bayes is a *'simple'* classifier as its name suggests, based on Bayesian rule. It assumes that all the attributes are conditionally independent. Each feature and class label is treated as a random variable. If class is represented by *c* and features by *(x1,x2,x3,...,xn)*, then our goal is to predict class *c*. That is done by finding value of c that maximizes P(c| $x1x2...xn$).

$$\mathrm{P(c} \mid x1x2...xn) = \frac{P(x1x2...xn|c)P(c)}{P(x1x2...xn)} \tag{3.5}$$

3.11 Evaluation Metrics

We have used two evaluation metrics to quantify performances over different machine learning algorithms. Those metrics are accuracy and f1-measure. In order to understand these metrics, we must know meaning of true positive **TP**, false positive **FP**, true negative **TN** and false negative **FN**.

True Positive (TP): This includes the cases where predicted *yes* is actually a *yes*.

False Positive (FP): These are the cases where predicted *yes* is actually a

no.

True Negative (TN): It represents the cases where predicted *no* is in reality a *no*.

False Negative (FN): It includes the cases where predicted *no* was actually a *yes*.

3.11.1 Accuracy

Accuracy is defined as the proportion of observations that are correctly classified over total number of instances in a dataset. Higher accuracy means that most of the instances were classified correctly. Accuracy is defined algebraically in Equation 3.6;

$$Accuracy = \frac{TP + TN}{FP + TP + FN + TN} \tag{3.6}$$

3.11.2 F1-Measure

F1 measure is defined on the basis of two measures precision P and recall R.

Precision P can be defined as in Equation 3.7;

$$Precision = \frac{TP}{TP + FP} \tag{3.7}$$

Whereas **Recall R** is defined as in Equation 3.8;

$$Recall = \frac{TP}{TP + FN} \tag{3.8}$$

Finally, the F1 measure is a harmonic mean of P and R, given in Equation 3.9;

$$F1 = 2 * \frac{P * R}{P + R} \tag{3.9}$$

3.12 Summary

In this chapter we have explained the methodology in depth along with an example from the dataset used for experiments purposes. We have also explained the learning frameworks that have been used to evaluate the method. We have also discussed some variants to the baseline algorithm. This chapter provides sound understanding of the experiments that are given in the next chapter.

Chapter 4: Evaluation/Validation

The content of this chapter describes all the experiments that have been done to assess the quality of the proposed sampling method. Moreover,it gives information about the dataset and machine learning frameworks that were used to classify the dataset.

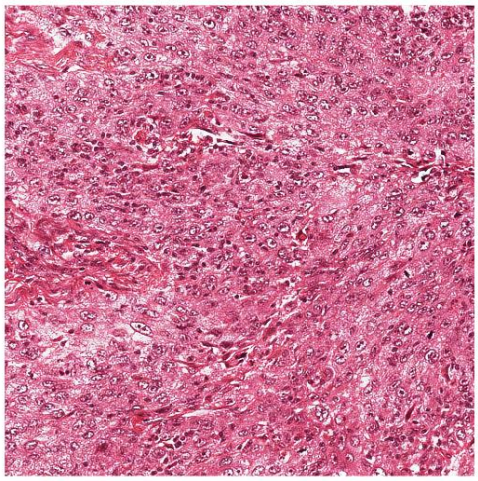
4.1 Dataset-MITOSIS 2012-ICPR

The dataset used is publicly available and is called MITOSIS-2012 given by International Conference on Pattern Recognition-ICPR[1] for automatically detecting and grading mitotic cells. This was originally prepared by the team of professor Fr ed erique Capron at the pathology department at Piti e-Salpˆetri‘ere Hospital in Paris, France. However, for the purpose of our study we have employed the dataset for tumor segmentation. This dataset comprises of 5 patients and 50 images in which each patient has contributed 10 images.Size of each image is relatively small 512*512 [5]. The images are *Haematoxylin* & *Eosin* (H & E) stained. The breast cancer tissues are stained [18] in order to get a detailed view of cancerous tissue which otherwise will appear transparent.

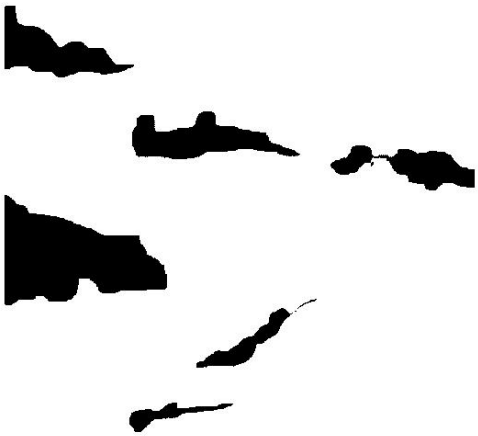
4.1.1 Ground Truth (GT)

This dataset comes with three sets of ground truth. Of which, two GT sets are prepared manually by senior pathologists Dr. Asha Rupani and Dr. Hesham

[1]http://ipal.cnrs.fr/ICPR2012



(a) Microscopic Cancerous Tissue Image



(b) Fused Ground Truth

Figure 4.1: Microscopic image of breast cancer tissue with corresponding ground truth; the white patches are cancerous and black patches are non-cancerous

El. Daly at Addenbrookes Hospital, Cambridge, UK. Since, the two GT sets were prepared individually there stood good chance of difference between them. To overcome that, a *fused* GT was prepared. In which a pixel is assigned as tumor only when both of the pathologists had agreed , otherwise, non-tumor. We have fused GT only for the purpose of our experiments.

Figures 4.1a and 4.1b show the breast cancer tissue image and its corresponding ground truth. White area in the Figure 4.1b indicates cancerous/malignant segments whereas black segments are non-cancerous/benign.

4.2 Experiment with Lukasiewicz Reduction

As detailed in Section 2.3, we had started our work with the existing data reduction algorithm. For this purpose we split our dataset to 50%/50% for training/testing. We had input data of size **137,500** tuples, out of which, **87,500** belonged to cancer class and **50,000** belonged to benign class. We had

applied this method at four similarity levels , 70, 80, 90 and 100%. The concept of similarity level has been well explained in Section 3.3. The number of tuples after applying reduction is given in Table 4.1. The number of instances have been *reduced* big time. From **137,500** tuples in the input data the reduced data could be represented with as less as **22** tuples only.

Table 4.1: Data Size after applying Lukasiewicz Reduction

Similarity Level	Total Instances After Reduction
70%	22
80%	23
90%	30
100%	46

For this set of experiments we had used SVM, NB, sequential minimal optimization SMO, random forest RF and multilayer perceptron from WEKA toolkit. Evaluation results are shown in Table 4.2. The best results are obtained with 100% precision level with 46 tuples only, with the *accuracy* and *f1 measure* of **79.49** and **79.95**.

Table 4.2: Evaluation of Lukasiewicz Reduction

-	Accuracy	F1 100%	Accuracy	F1 90%	Accuracy	F1 80%	Accuracy	F1 70%
NB	66.73	70.11	68.02	70.87	56.37	59.95	59.02	62.68
SVM	**79.49**	**79.95**	**79.36**	**79.18**	**79.01**	**78.08**	**78.3**	**76.6**
SMO	78.5	77.8	74.2	75.3	69.7	71.05	68.4	79.5
RF	60.59	64.28	55.47	59.64	70.30	70.00	69.3	72.5
MLP	62.9	66.7	68.68	72.25	63.33	66.9	61.50	65.04

4.3 Evaluation of the Proposed Sampling Method-PPS

In this section we show the performances of the obtained samples from our own method on different classifiers;namely, Support Vector Machine, Naive Bayes, Artificial Neural Network (Patternnet, Cascadeforwardnet and Feedforwardnet).

4.3.1 Train and Test Split

For the purpose of experiments the dataset has been split to 50%/50% for training/testing. The training dataset has 25 images of 512*512 pixels, which makes it 25*512*512=6553600 pixels. For each pixel we have different number of features. In case of MR8 features, we have 8 features so 8*6553600 is a huge number. In order to reduce the computation time, class based (stratified) random sampling has been done on the training data. In which 200 cancerous and 200 non-cancerous pixels from each image have been chosen randomly.So, now we have 10000 pixels only for training. Out of which 5000 belong to cancerous and other 5000 belong to non-cancerous.

4.3.2 Evaluation Configurations

We have generated samples from three variants of our method under three levels of similarity index i.e. 70%, 80% and 90% where the multiplier used was 1000. All of the three samples have been tested through 5 classifiers. We provide the details of evaluation for those samples in terms of Accuracy and F1 Measure in the upcoming sections.

4.3.3 Baseline PPS Results

This is the basic algorithm wherein we consider only those patterns that have proportion >=1. Table 4.3 shows the number of instances after applying PPS to 3 similarity levels. From **10,000** tuples we have managed to reduce to as low as **1149** tuples. The evaluation of the three samples is given in Table 4.4; where SVM outperforms all other classifiers with highest accuracy and f1 measure of **79.1** and **84.9**.

Table 4.3: Data Size after applying PPS

Similarity Index	Benign Instances	Malignant Instances	Total Instances
70%	570	579	1149
80%	664	681	1345
90%	791	773	1564

Table 4.4: Results for PPS

Classifier	Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
NB	71.9	79.6	71.9	79.4	72.9	82.5
SVM	**79.1**	**84.9**	**78.9**	**84.7**	**78.7**	**84.5**
PN	76.2	82.8	76.9	83.2	78.02	84.1
CFN	75.7	82.5	76	82.6	77.8	83.9
FFN	76	82.4	75.7	82.3	78	84.1

4.3.4 Minority Biased PPS

Here, we consider all the patterns with proportion >=0. If the proportion is less than 1 we take one sample from that pattern. Table 4.5 gives the number of instances after applying MB-PPS. Sample sizes here are slightly bigger than in Table 4.3 as we consider even those patterns that have very small proportions with the exception of 70% similarity level for which sample size here is slightly smaller. Table 4.6 shows the evaluation. SVM is again outperforming rest of the classifiers.

Table 4.5: Data Size after applying MB-PPS

Similarity Index	Benign Instances	Malignant Instances	Total Instances
70%	557	568	1125
80%	669	637	1336
90%	815	787	1602

Table 4.6: Results for MB-PPS

Classifier	Accuracy and	F1 70%	Accuracy and	F1 80%	Accuracy and	F1 90%
NB	72.5	80.1	72.6	80.1	73.6	81.3
SVM	**78.9**	**84.7**	**78.3**	**84.1**	**78.4**	**84.3**
PN	76.8	83.3	76.3	82.7	77.8	83.8
CFN	74.7	81.5	76.5	82.9	78.04	84.1
FFN	76.01	82.5	76.9	83.2	77.6	83.8

4.3.5 Without Proportions

In this scenario, we take one sample for each pattern. Sample sizes are given in Table 4.7 and the smallest is just **25** tuples. Evaluations are given in Table 4.8. SVM, as previously performs best with accuracy of **79%** and f1 measure **85.2**. This value of accuracy is by far the highest value obtained ever in our work.

Table 4.7: Data Size after applying PPS-no proportions

Similarity Index	Benign Instances	Malignant Instances	Total Instances
70%	12	13	25
80%	41	23	64
90%	89	80	169

Table 4.8: Results for PBS

Classifier	Accuracy and	F1 70%	Accuracy and	F1 80%	Accuracy and	F1 90%
NB	66.8	76.6	74.8	**82.7**	73.7	82.4
SVM	**78.5**	**84.5**	**74.7**	80.6	**79**	**85.2**
PN	65.4	71.5	65.2	72.2	78.3	84.6
CFN	62.7	69.3	53.9	56.7	70.6	77.2
FFN	63.5	70.3	61.8	67.6	69.9	74.3

4.4 Cross Validation

In order to prevent over-fitting and let our models generalize well, we have performed 10 fold cross validation. For this purpose the dataset has been randomly split to 10 disjoint subsets.

- In total, we have 50 images.Where each subset contains 5 images.
- For each image we do stratified random sampling by taking 450 malign pixels and 300 benign pixels. In total we take 750 pixels from each image for training purpose.

- For 10 fold cross validation, we train/test the data for 10 iterations.
- For each iteration we give 750*45=33750 pixels to PPS algorithm.
- However, for testing purpose, we test on whole image(512*512 pixels).
- In first iteration; first 9 subsets are used for training and the 10th(last) subset is used for test.
- For the second iteration; first 8 subsets and 10th subset is used for training and this time 9th subset is used for testing.
- This process is repeated 10 times until each subset has been used for testing exactly one.
- Each subset of data has been fed as an input to PPS algorithm to generate samples. We have applied PPS 10 times for 10 subsets.
- We have applied this for all the three variants of PPS.

For the comparison we have compared accuracy and F1 Measure before and after sampling for all the subsets. We have also shown the size of training data before and after applying for each training set. Multiplier used was 10000. We have performed cross-validation for three levels of similarity; 90%, 80% and 70% and to all the three variants discussed in 3.9.1,3.9.2 .

4.4.1 Evaluation of subsets before sampling

We have tested all the ten training sets with five classifiers before applying PPS on them. Table 4.9 for patternnet, Table 4.10 for cascadeforwardnet, Table 4.11 for feedforwardnet, Table 4.12 for SVM and Table 4.13 for NB. Highest average accuracy and f1 measure have achieved by SVM which is **78.3** and **84.3**. Whereas, lowest accuracy and f1 is gained by NB, with the values of

71.9 and **79.7**. PN, CFN and FFN are not far from SVM and have achieved accuracy and f1 measure with a difference of very small fractions.

Table 4.9: Before sampling results-Patternnet

Train Sets	Accuracy	F1 Measure
1	73.3	79.4
2	74.2	81.03
3	87.1	90
4	76.1	82.4
5	78.1	85.2
6	73.9	80.1
7	74.8	83.3
8	83.2	88
9	78.8	85.1
10	82.8	86.4
Average	**78.23**	**84.093**

Table 4.10: Before Sampling results-Cascadeforwardnet

Training Sets	Accuracy	F1 Measure
1	73.3	79.1
2	74.4	81.1
3	87.2	90
4	76.4	82.7
5	78	85.1
6	74.1	80.3
7	74	82.7
8	83.3	88.1
9	78.7	85.1
10	83.01	86.5
Average	**78.241**	**84.07**

4.4.2 Cross-validation PPS

Table 4.14 shows the number of instances before and after applying PPS for three similarity levels. Higher the similarity measure, bigger is the sample. This is why, we have sample for 90% in the range of **3000** as opposed to 70% similarity, for which the sample is of the order of **1100**.

Table 4.11: Before Sampling results-Feedforwardnet

Training Sets	Accuracy	F1 Measure
1	73.2	79
2	74.4	81.1
3	87.2	90.1
4	76.4	82.7
5	77.7	84.9
6	74.1	80.1
7	74.4	83.07
8	83.2	88.09
9	78.6	84.9
10	83.04	86.5
Average	**78.224**	**84.046**

Table 4.12: Before Sampling results-SVM

Training Sets	Accuracy	F1 Measure
1	73.2	79.4
2	74.6	81.3
3	87.7	89.8
4	75.3	82.1
5	78.8	86.01
6	74.1	80.4
7	75.6	84.1
8	82.9	88
9	79.1	85.5
10	82.7	86.5
Average	**78.3**	**84.311**

Table 4.13: Before Sampling results-Naive Bayes

Training Sets	Accuracy	F1 Measure
1	69.1	76.3
2	68.5	77
3	79.9	85.2
4	67.5	76.3
5	71.9	81
6	69.4	76.9
7	72.7	81.9
8	75.5	82.6
9	71.9	80.1
10	73.5	79.7
Average	**71.99**	**79.7**

Here we have applied the basic PPS algorithm and have tested with five classifiers and have given results in Tables 4.15, 4.16, 4.17, 4.18 and 4.19. Of all the three similarity levels and 5 classifiers, SVM for 90% PPS have given the best performance with accuracy and f1 measure of **77.99** and **83.93**.

Table 4.14: Data size of each training set; After sampling PPS (90% , 80% and 70%)

Training Sets	Before sampling	PPS 90%	PPS 80%	PPS 70 %
1	33750	3738	1312	1151
2	33750	3661	1323	1158
3	33750	8561	1364	1176
4	33750	3821	1353	1176
5	33750	3535	1322	1142
6	33750	3644	1311	1174
7	33750	3705	1356	1137
8	33750	3791	1344	1158
9	33750	3661	1347	1179
10	33750	3726	1337	1153

4.4.3 Cross-validation Minority Biased PPS

We have performed cross-validation for three levels of similarity; 90%, 80% and 70%. The number of instances before and after applying MB-PPS for three similarity levels are given in Table 4.20. Sample sizes are consistent with the sizes of prevois. Tables 4.21, 4.22, 4.23, 4.24 and 4.25 show the results we have obtained. SVM for 90% similarity and 80% similarity has achieved

Table 4.15: Results for PPS on Patternnet; After Sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.4	77.9	72.3	79	70.4	78.4
74.5	81.2	71.8	79.3	71.8	79.2
86.1	89.3	83.35	87.6	83.1	87.4
74.9	81.2	76.2	81.2	68.2	75.9
80.5	87.4	76.3	83.6	71.5	79.3
72.9	79.1	72.9	79.8	72.5	79.2
73.3	82.1	71.8	81.06	71.6	80.9
82.5	87.8	82.9	87.9	80.9	85.8
76.5	83.1	75.7	81.8	68.06	73.6
80	83.5	81.9	85.3	71.7	74.4
77.36	**83.26**	**76.5**	**82.8**	**73.03**	**79.4**

Table 4.16: Results for PPS Cascadeforwardnet; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.5	78.3	72.5	80.2	71.2	79.1
70.4	77.8	66.7	74.47	69.2	77.9
84.2	87.7	73.4	79.2	73.1	79.04
74.6	80.9	76.28	82.99	74.01	79.9
80.9	87.8	75.5	83.14	67.1	74.9
70.5	76.8	72.6	79.23	69.6	76.7
71.4	80.6	72.8	81.8	73.1	82.4
82.5	87.9	82.5	87.7	81.07	86.2
76.4	83.1	75.8	81.8	72.7	78.8
78.1	81.9	79.9	83.2	71.9	75.05
76.15	**82.28**	**74.8**	**81.3**	**72.3**	**79.05**

Table 4.17: Results for PPS Feedforwardnet; After Sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.3	77.9	71.4	77.7	67.8	76.3
73.6	80.5	66.04	73.9	69.04	77.13
84.4	87.8	74.8	80.6	72.8	79.1
74.4	80.7	75.6	82.2	75.07	81.16
79.6	86.7	73.5	81.2	79.2	86.8
71.3	77.7	72.2	78.4	68.8	74.6
72.01	81.07	71.6	80.8	73.1	82.4
81.2	86.5	82.6	87.7	78.7	84.2
75.7	82.4	74.5	80.6	67.6	73.09
78.7	82.3	80.4	83.7	68.8	78.8
76.76	**82.85**	**74.3**	**80.7**	**72.1**	**78.5**

Table 4.18: Results for PPS SVM; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.9	78.9	72.1	80.9	73.08	80.8
74.3	81.1	73.1	80.5	75.4	83.16
86.2	89.5	84.5	88.8	83.6	88.3
75.3	81.8	75.1	82.09	75.2	81.9
81.2	88.08	78.6	85.5	78.6	85.4
73.7	79.7	73.4	80.1	73.8	80.7
74.5	83.2	74.4	83.2	76.6	85.09
82.5	87.7	83.05	88.12	82.6	87.5
77	83.6	75.7	81.8	74.4	80.4
82.3	85.8	82.2	85.4	81.6	84.6
77.99	**83.93**	**77.3**	**83.5**	**77.5**	**83.8**

Table 4.19: Results for PPS Naive Bayes; After sampling)

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
71.7	79.4	69.6	78	66.8	77.9
73.7	81.4	66.03	75.36	68.3	78.1
77.9	84.6	77.7	84.9	77.6	84.7
71.1	80	68.01	77.3	67.5	76.5
78.6	86.8	71.9	81.1	70.7	80.05
71.2	79.5	68.9	77.8	69.2	77.9
76.5	85.07	72.6	82.01	73.4	82.6
75.9	84.1	75.1	82.9	74.6	82.5
74.02	82.2	73.1	81.4	70.2	78.3
71.5	79.5	72.5	79.5	74.02	79.08
74.21	**82.25**	**71.5**	**80.06**	**71.46**	**79.82**

highest accuracy **77.76** and f1 measure **84.1**. This variation to the baseline PPS has achieved higher f1 measure than basic method by **0.17**. However, the accuracy has gone down by **0.22**.

Table 4.20: Data size of each training set;MB-PPS

Training Sets	Before sampling	MB-PPS 90%	MB-PPS 80%	MB-PPS 70%
1	33750	3699	1396	1180
2	33750	3803	1396	1171
3	33750	3608	1398	1180
4	33750	3704	1397	1169
5	33750	3586	1360	1164
6	33750	3759	1417	1192
7	33750	3773	1412	1166
8	33750	3572	1407	1171
9	33750	3726	1393	1187
10	33750	3733	1415	1169

Table 4.21: Results for MB-PPS Patternnet; After Sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.7	77.9	72.5	79.5	71.5	79.9
74.4	81.2	63.6	72.9	70.8	78.5
86.01	89.3	80.4	85.2	78.9	84.1
75.6	82.03	76.2	82.8	73.8	80.4
80.4	87.3	76.4	83.6	75.6	82.9
72.5	78.8	73.9	80.7	72.2	79.01
72.9	81.8	72.8	81.8	72.05	81.4
68.8	80.9	82.4	87.8	82.8	87.6
75.6	82.4	76.6	82.6	70.7	76.3
81.5	85.09	81.1	84.3	73.5	77.2
76.04	**82.67**	**75.6**	**82.1**	**74.23**	**80.7**

Table 4.22: Results for MB PPS cascadeforwardnet; After Sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.5	77.8	72.1	79.5	68.7	76.7
73.2	80.2	71.09	78.35	66.7	74.4
82.2	85.9	76.24	81.2	68.3	73.5
74.9	81.4	75.7	83.1	74.4	80.9
80.4	87.4	76.8	84.6	71.7	79.8
70.6	76.9	73.8	80.5	72.5	78.6
73.2	82.1	77.6	85.9	75.5	84.4
69.1	81.1	82.4	87.3	81.9	86.9
75.3	82.1	75.4	81.4	67.8	73.5
79.06	82.6	80.3	83.4	71.06	74.31
75.03	**81.75**	**76.1**	**82.5**	**71.8**	**78.3**

Table 4.23: Results for MB-PPS feedforwardnet

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.6	77.8	71.3	78.4	67.4	76.2
70.9	78.3	70.6	78.3	72.5	80.8
80.9	85.5	73.08	79.08	75.9	82.2
74.9	81.6	75.77	82.90	74.7	81.4
78.5	85.8	76.9	84.1	75.4	82.8
70.9	77.3	73.5	80.6	69.2	75.4
73.2	82.08	74.6	83.3	74.7	83.8
68.8	81.07	81.6	87.02	81.2	86.4
75.2	82.08	75.2	81.2	70	75.7
79.2	82.07	79.2	82.7	70.22	73.08
74.51	**81.4**	**75.2**	**81.8**	**73.1**	**79.8**

Table 4.24: Results for MB-PPS SVM; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.8	78.7	72.9	80.3	73.06	80.8
74.4	81.2	74.6	82.1	75.1	82.7
86.4	89.6	84.9	89.1	83.5	88.3
75.2	82.1	75.02	82.3	75.1	81.8
81.2	88	78.9	85.8	79.8	86.5
73.7	79.8	73.8	81.1	73.9	80.9
74.7	83.3	75.2	83.8	76.5	84.9
79.8	85.5	83.08	88.06	82.4	87.5
77.1	83.7	76.8	82.9	74.5	80.4
82.3	85.8	82.05	85.2	81.3	84.3
77.76	**83.74**	**77.7**	**84.1**	**77.5**	**83.8**

Table 4.25: Results for MB-PPS Naive Bayes; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
70.7	79	70.18	78.9	69.3	78.5
73.4	81	68.8	78.2	68.4	78.1
86.4	89.6	76.1	84.07	77.4	84.5
78.2	84.8	69.05	78.7	67.8	76.9
71.2	80.2	73.7	82.6	71.5	80.6
78.6	86.6	69.8	79.2	68.8	77.7
77	85.4	75.1	84.1	73.9	83.07
70.1	81.6	75.5	83.4	74.4	82.4
74.3	82.4	73.6	82.3	70.5	78.8
72.7	79.7	73.4	80.36	73.8	78.7
73.82	**82.07**	**72.5**	**81.2**	**71.6**	**79.9**

4.4.4 Cross Validation -Pattern Based Sampling

We have performed cross-validation for three levels of similarity; 90%, 80% and 70%. Table 4.26 The number of instances before and after applying PBS for three similarity levels are given.The samples generated here are of much smaller size as compared to previous two methods. However, this method gives the best performance with highest accuracy **78.6** and f1 measure **85.6** with SVM for 80% similarity. Tables 4.27, 4.28, 4.29, 4.30 and 4.31 show the evaluation results for all the classifiers.

Table 4.26: Data size of each training set; PBS 90%, PBS 80% and 70%

Training Sets	Before sampling	PBS 90%	PBS 80%	PBS 70%
1	33750	180	81	33
2	33750	177	84	30
3	33750	187	89	36
4	33750	179	85	30
5	33750	179	87	30
6	33750	194	77	31
7	33750	176	81	31
8	33750	176	76	29
9	33750	183	92	33
10	33750	187	85	31

Table 4.27: Results for PBS Patternnet; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
71.4	80.04	70.1	79.14	71.58	78.9
76.13	84.2	65.2	73.4	71.1	81.4
82.7	86.6	74.7	79.8	33.6	31.8
74.57	80.9	72.7	80.9	32.4	0.017
81.3	87.9	80.7	88.5	71.6	79.6
74.51	82.28	71.2	78.2	63.4	72.9
79.8	87.7	76.6	85.19	65.05	76.11
78.84	85.39	49.01	52.3	35.4	28.7
79.9	86.8	79.2	86.4	66.04	72.9
80.74	85.14	59.08	55.8	78.5	82.7
78.01	**84.7**	**69.8**	**75.9**	**58.8**	**57.6**

Table 4.28: Results for PBS cascadeforwardnet; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
71.2	79.07	62.15	70.74	68.6	76.1
72.7	81.9	75.2	85.4	51.4	57.8
84.5	88.7	74.7	83.19	60.55	67.01
74.4	82.7	34.2	0.12	65.9	75.4
67.8	76	79.5	87.8	60.4	77.9
73.1	80.19	60.04	61.08	69.7	77.2
77.2	85.9	69.13	80.36	51.8	63.9
71.91	77.2	72.6	82.7	61.4	63.6
75.2	82.8	75.2	82	72.5	82.1
79.6	84.2	60.87	67.18	69.1	72.8
74.8	**81.8**	**66.39**	**71.27**	**64.07**	**71.4**

Table 4.29: Results for PBS feedforwardnet; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
71.2	80.5	68.1	77.05	37.9	0.19
73.6	82.6	32.8	0.192	38.4	38.3
82.6	87.8	50.28	46.8	81.2	86.5
71.8	81.05	65.9	75.14	68.8	78.4
80.8	88.01	74.17	82.9	60.7	69.9
70.81	77.6	65.7	77.3	63.3	73.6
72.2	81.4	64.7	75.3	73.2	83.2
72.18	82.4	80.39	80.06	51.7	62.1
59.3	64.5	74.3	81.4	50.7	52.3
82.1	86.04	79.06	84.19	47.99	42.2
73.6	**81.1**	**65.5**	**70.5**	**57.4**	**60.6**

Table 4.30: Results for PBS SVM; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
72.5	81.6	72.76	81.6	72.9	81.09
76.4	84.7	76	84.1	76.6	84.8
2.5	87.7	85.04	89.15	83.2	88.1
73.7	82.6	74.08	82.4	74.6	81.7
82.5	89.2	82.01	88.7	81.4	88
73.9	82.5	73.9	81.9	74.03	81.4
80.6	88.3	78.9	86.9	80.1	87.8
80.6	87.1	82.8	88.15	81.3	87.5
80.1	86.9	79.9	86.9	78.9	85.2
82.2	86.6	81.3	86.14	82.2	86.5
78.5	**85.7**	**78.6**	**85.6**	**78.5**	**85.2**

Table 4.31: Results for Naive Bayes PBS; After sampling

Accuracy and	F1 90%	Accuracy and	F1 80%	Accuracy and	F1 70%
69.4	80.75	69.1	81.02	68.5	80.8
69.7	80.8	69.01	80.9	68.9	80.6
69.6	80.7	71.4	79.1	68.6	80.1
70.15	80.97	69.5	80.8	67.7	78.3
70.17	81.02	69.8	80.8	68.3	78.5
69.8	80.9	72.1	80.9	69.7	79.5
69.9	80.7	69.1	80.6	67.4	77.8
69.8	80.6	69.5	80.9	68.4	79.8
69.6	80.8	68.9	80.8	68.7	80.2
71.7	81.1	70.1	80.4	66.7	77.3
70.03	**80.08**	**69.8**	**80.6**	**68.3**	**79.3**

4.4.5 Minimizing and Maximizing Overlap- Cross Validation

As explained in Sections 3.8.1 and 3.8.2 we have performed cross validation for these two variant with 90% similarity and MB-PPS only. As from our previous experiments, we have found out that 90% similarity gives best results. Data sizes after applying sampling are given in Table 4.32. Tables 4.33, 4.34, 4.35, 4.36 and 4.37 show the evaluation results. Clearly, minimum overlapping obtains smaller sample and better performance. As it tries to retrieve distinct object, resulting into a better and compact representation of data. SVM yields best results in this experiment as well.

Table 4.32: Data size of each training set; Minimum and Maximum Overlap

Training Sets	Before sampling	Minimum Overlap	Maximum Overlap
1	33750	857	4173
2	33750	837	3514
3	33750	839	6355
4	33750	857	527
5	33750	824	719
6	33750	891	3598
7	33750	861	3533
8	33750	872	6304
9	33750	847	6326
10	33750	812	6354

Table 4.33: Results for PPS-Overlap Patternnet; After sampling

Train sets	Accuracy and	F1 (Min Overlap)	Accuracy and	F1(Max Overlap)
1	73.4	80.3	38.5	20.1
2	68.9	76.6	27.9	55.6
3	85.5	88.6	85.6	88.7
4	74	79.8	48.1	44.9
5	73.5	80.9	48.1	44.9
6	70.6	75.9	32.2	2.03
7	70.8	80.03	0.17	0
8	80.7	85.9	83.1	87.7
9	77.8	84.2	74.5	80.6
10	77.08	81	81.8	84.9
Average	**75.2**	**81.3**	**55.5**	**53.96**

Table 4.34: Results for PPS-Overlap Cascadeforwardnnet; After sampling

Train sets	Accuracy and	F1 (Min Overlap)	Accuracy and	F1(Max Overlap)
1	72.6	78.9	37.7	20.41
2	71.04	78.28	28.7	7.9
3	82.4	85.6	84.3	87.5
4	72.8	78.8	61.5	65.3
5	70.5	78.2	59.3	70.3
6	70.6	76.1	32.1	1.7
7	71.8	81.3	18	0.6
8	81.1	86.6	81.3	86.03
9	77.01	83.34	74.16	80.23
10	79.7	84.01	79.2	82.5
Average	**74.9**	**81.1**	**55.6**	**50.2**

Table 4.35: Results for PPS-Overlap Feedforwardnet; After sampling

Train sets	Accuracy and	F1 (Min Overlap)	Accuracy and	F1(Max Overlap)
1	71.1	76.6	37.7	20.9
2	75.4	83.4	29.1	9.1
3	81.9	85.2	84.3	87.7
4	70.5	75.2	67.3	73.2
5	72.7	82.4	72.3	82.01
6	71.5	77.5	32.4	2.6
7	69.5	79.1	18.01	0.0008
8	79.8	85.3	82.05	86.8
9	76.5	82.8	74.4	80.47
10	80.28	84.1	78.2	81.3
Average	**74.9**	**81**	**57.6**	**52.4**

Table 4.36: Results for PPS-Overlap SVM; After sampling

Train sets	Accuracy and	F1 (Min Overlap)	Accuracy and	F1(Max Overlap)
1	72.9	80.06	32.7	0
2	71.8	79.1	25.9	0
3	86.2	89.1	86.6	89.6
4	74.4	80.2	73.1	80.6
5	73.8	81.1	81.2	88.2
6	72.5	78.2	31.7	0
7	73.9	82.6	0.17	0
8	82.7	87.6	82.6	87.4
9	78.2	84.4	75.7	81.7
10	82.4	85.7	82.4	85.9
Average	**76.9**	**82.8**	**59.02**	**59.34**

Table 4.37: Results for PPS-Overlap NB; After sampling

Train sets	Accuracy and	F1 (Min Overlap)	Accuracy and	F1(Max Overlap)
1	70.32	79.8	33.16	73.9
2	70.8	79.6	32.6	0.91
3	71.2	80.16	70.54	80.05
4	71.2	80.2	68.4	78.5
5	70.6	79.6	69.9	80.6
6	71.2	79.7	32.45	4.02
7	70.12	79.6	32.6	0.58
8	70.73	80.08	70.07	80.2
9	71.19	80.26	69.5	78.05
10	70.4	79.4	69.9	79.5
Average	**70.8**	**79.8**	**54.9**	**48.9**

4.4.6 Averaged performance of PPS and its variants in bar plots

In this section we have put all the results collectively in the form of bar plots to have a better visualisation. Figure 4.2, 4.3 and 4.4 show PPS accuracy and f1 measure for 70%, 80% and 90% similarity index.

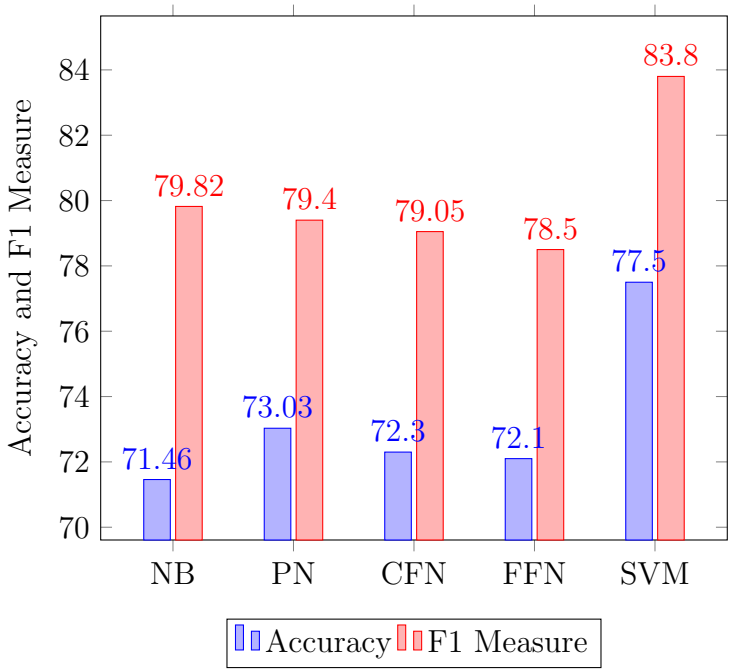


Figure 4.2: PPS (70%); After sampling

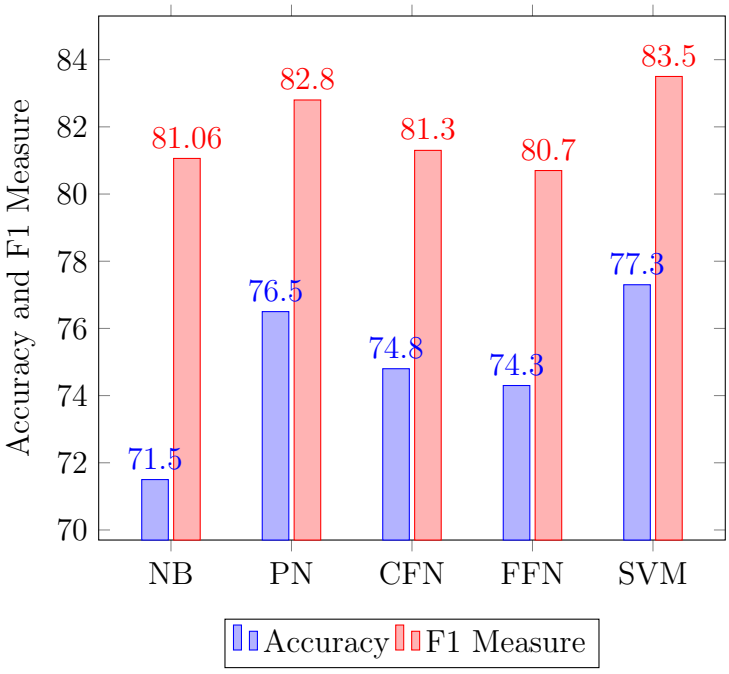


Figure 4.3: PPS (80%); After sampling

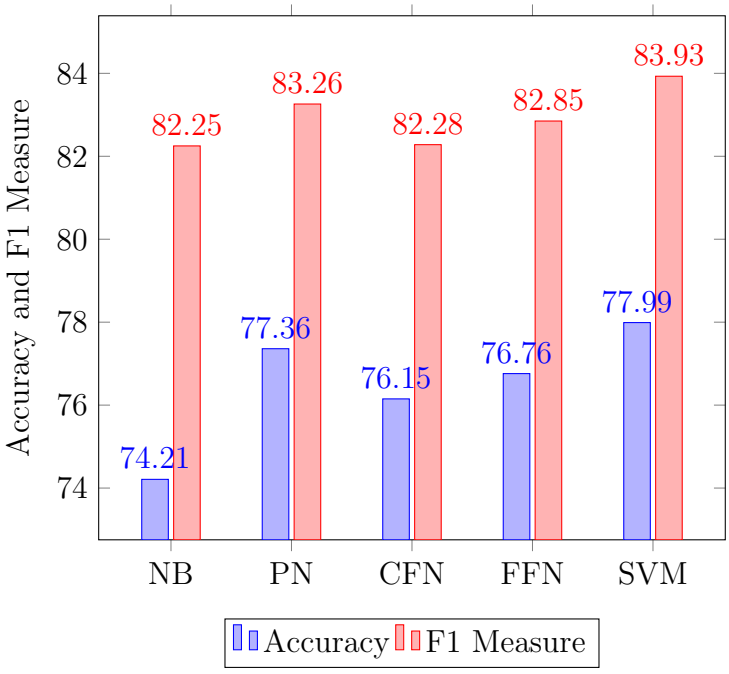


Figure 4.4: PPS (90%); After sampling

Figure 4.5, 4.6 and 4.7 show MB-PPS accuracy and f1 measure for 70%, 80% and 90% similarity index.

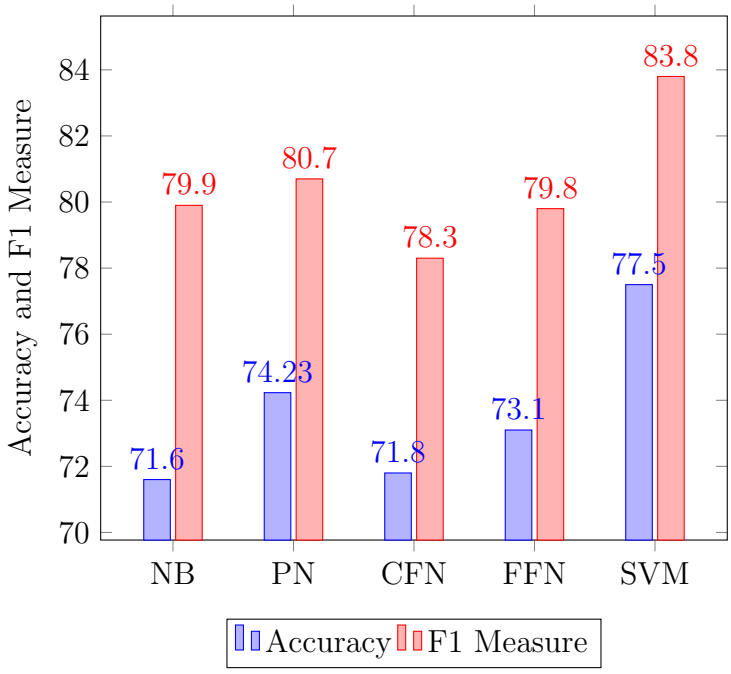


Figure 4.5: MB-PPS (70%); After sampling

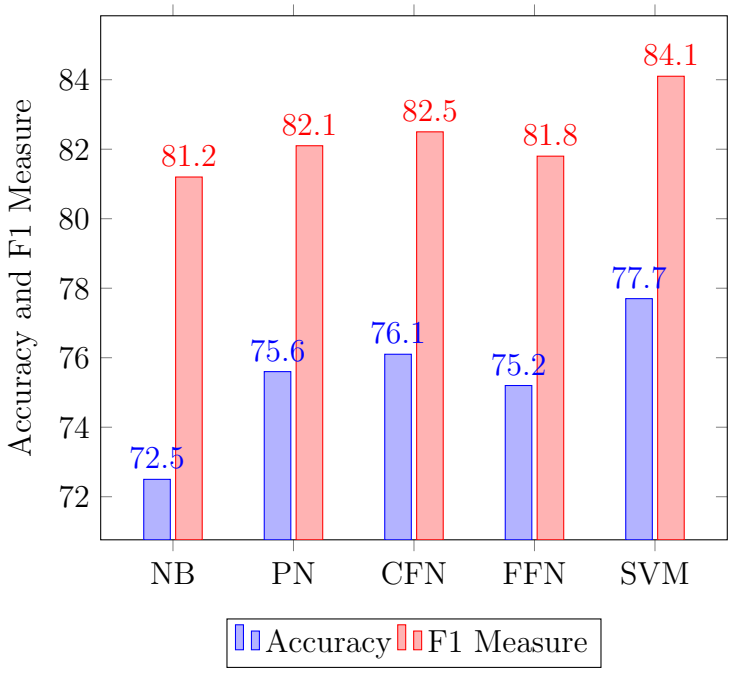


Figure 4.6: MB-PPS (80%); After sampling

Figure 4.8, 4.9 and 4.10 show PBS accuracy and f1 measure for 70%, 80% and 90% similarity index.

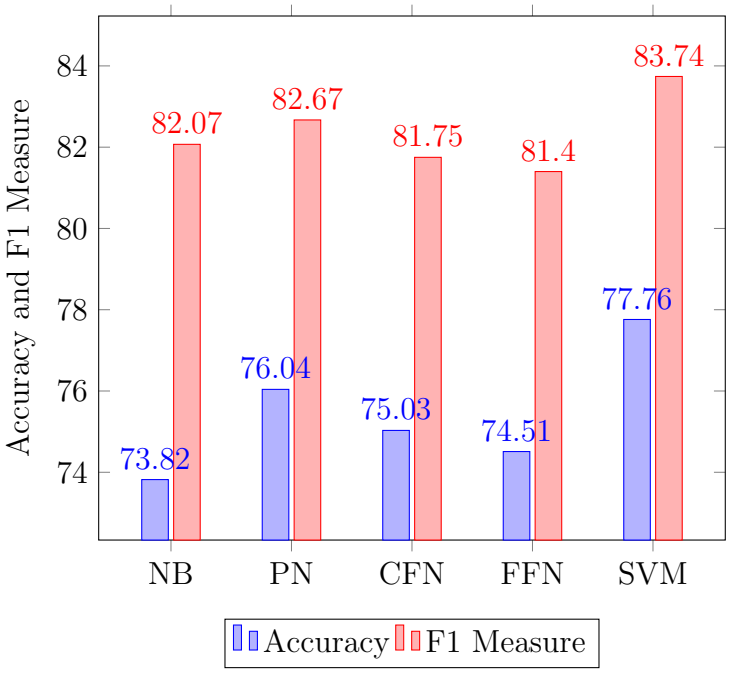


Figure 4.7: MB-PPS (90%); After sampling

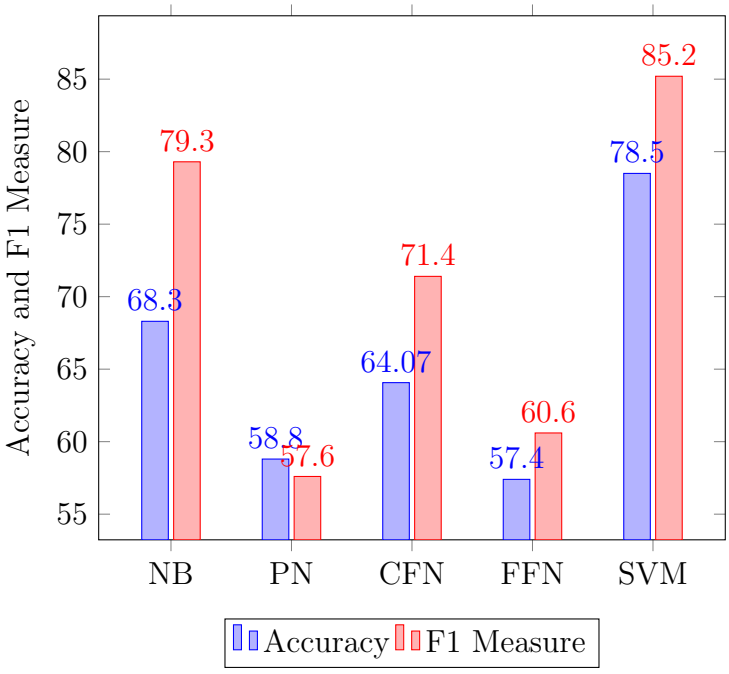


Figure 4.8: PBS (70%); After sampling

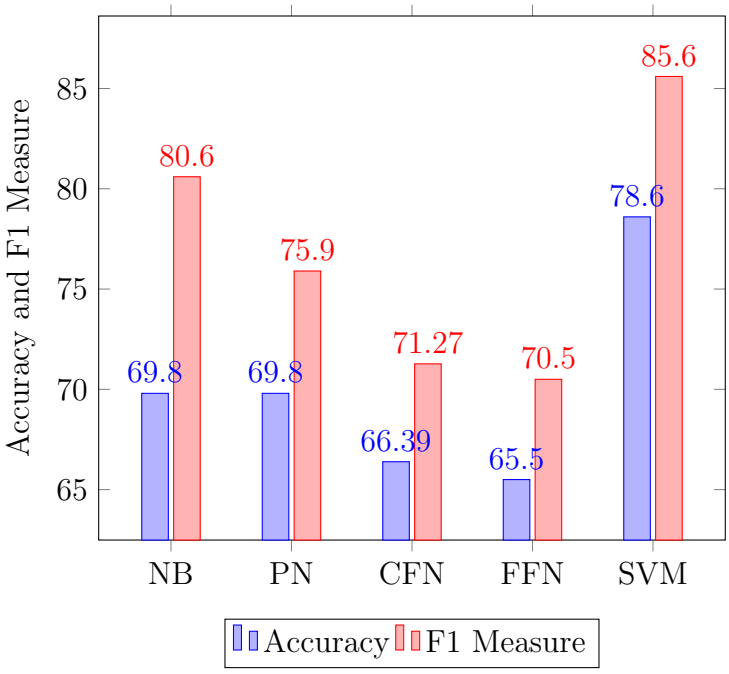


Figure 4.9: PBS (80%); After sampling

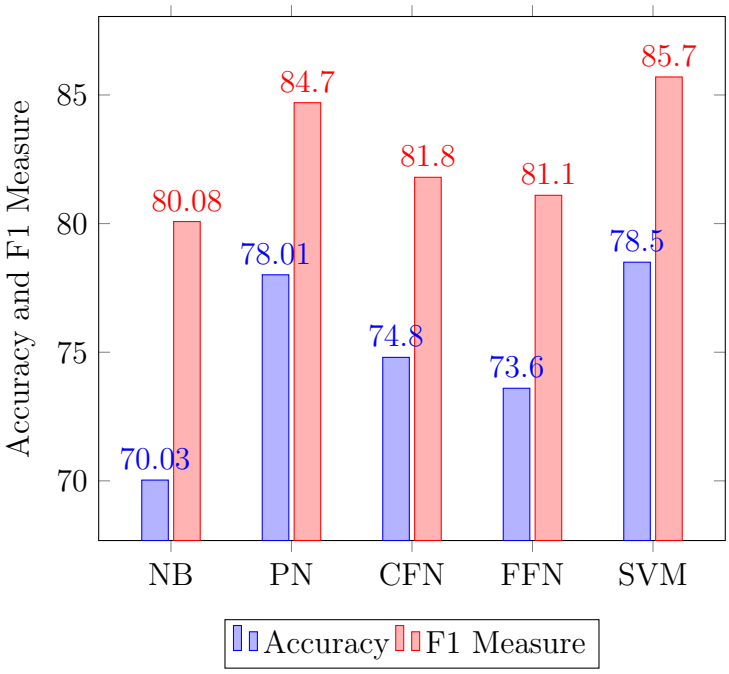


Figure 4.10: PBS (90%); After sampling

4.5 Evaluation of SRS and SS

For the sake of comparison we have performed cross-validation by applying **simple random sampling** and **stratified sampling** as explained in Sections

2.4.1 and 2.4.2. We have chosen **1200** tuples for SRS and **1215** tuples for SS for each of 10 training sets. The averaged accuracies and F1 measures of 10 training sets for all the classifiers are shown in Table 4.38. We have found that, the performance is not very far from what our method has achieved. However, these two methods are random and will not be able to reproduce the exact same results. It might do well sometimes and it may give poor results also.

Table 4.38: Evaluation of SRS and SS

Classifiers	Accuracy and	F1 (SRS)	Accuracy and	F1 (SS)
NB	72.09	80.09	72.1	80.2
SVM	**78.2**	**84.5**	**78.1**	**84.3**
PN	77.1	83.5	77.2	83.4
CFN	77.8	84.1	77.3	83.6
FFN	77.9	84.1	77.4	83.7

4.6 Summary

By applying multiple variations of PPS we have observed that there exists positive correlation between similarity index and sample size. Higher similarity index means the bigger sample. Out of the three methods, the last method which is *simple* as compared to the rest of the two as it does not take into account the proportions, yields the highest accuracy and f1 measure. But the selection of objects here is purely random. We may not be able to get the same result each time we run the experiment.

Of all the experiments performed; the best results obtained before and after sampling in the cross-validation can be summarized in Table 4.39. The **pattern based sampling** method, explained in Section 3.9.2 is giving us the best results.

To *summarize*, we have performed mainly three sets of experiments

Table 4.39: Best performance among all the cross-validation experiments

-	Accuracy	F1	Classifier	Similarity Level	Sample Size
Before sampling	78.3	84.3	SVM	N/A	33750
After pattern based sampling	**78.6**	**85.6**	SVM	80%	84

- Incremental Lukasiewicz reduction by doing 50%/50% split (no cross-validation) at four similarity levels 70%, 80% ,90% and 100%

- Pattern based proportional sampling method along with its variants by doing 10 fold cross validation for three *similarity levels* of 70%, 80% and 90%

- Simple random sampling and stratified sampling with 10 fold cross validation

We have put the results of Dhoha Abid's work [3] in Table 4.40. Dhoha had used 137,500 tuples to get **77.5** accuracy and **83.5** f1 measure. This result is not directly comparable to our work as we have performed cross validation. Wherein [3] the evaluation was done by doing 50%/50% split of data. If we compare this work with our experiment of Lukasiewicz Reduction in Section 4.2, the classification accuracy is higher than [3]. As in Lukasiewicz reduction we were able to get accuracy of **79.1** with 46 tuples only as opposed to only 77.5% accuracy with 137,500 tuples.

Table 4.40: Summarized results from [3]

-	Accuracy	F1	Classifier	Train and Test split	Sample Size
Dhoha's Work	77.5	83.5	SVM	50%/50%	137,500

Chapter 5: Conclusions and Future Work

We conclude this thesis by summarizing our objectives, achievements,limitations and some future directions on our proposal.

5.1 Conclusions

We have presented a sampling method based on patterns. The patterns are obtained by building formal context of a given database. The strength of the method arises from the fact that it preserves data characteristics in form of functional dependencies and discards outliers. Thus helping us to perform classification tasks with smaller datasets and reduced computation time.

To test the efficacy of proposed method we have chosen the domain of *image segmentation* particularly for *breast cancer images*. We were motivated to work with this particular topic due to the increasing prevalence of breast cancer around the globe. To automatically segment the images of cancer will save time and tireless efforts required from the pathologists. It will ensure the timely delivery of right treatment to patients.

Image segmentation is a task that involves four main subtasks. Those are, *pre-processing, feature extraction, feature reduction* and *decision making/classification*. Our work revolves around the third task which is *feature reduction.* We are not trying to reduce the features rather we are trying to reduce the *number of instances* of given data.

We have performed extensive sets of experiments with multiple variations

of the baseline PPS algorithm and have reached to the conclusion that it is important to take into account each pattern even if it occurs in a very small proportion.

Currently, our method works well with a small number of features 8 or 10. However, if we asked were to apply PPS on a larger number of features then one plausible route is to perform feature ranking at first to choose top 8 features and then apply our sampling method.

For the comparison purpose, we have compared our method with two conventional sampling methods, SRS and SS. We are not able to compare our results with the participants of ICPR conference MITOTIS 2012 dataset was used for automatically detecting and grading mitotic cells. However, we have used that dataset for tumor segmentation.

5.2 Future Work

As stated earlier in the last section, our immediate goal is to work with a much larger feature vector, by simply taking the most discriminative of them. We also intend to improve on the way the objects are chosen in the last step to be considered for inclusion in the output.

Our next target is to work with *multi-class classification* problems. Our method does not impose any restrictions on the kind of data we use. We will also apply it on the text data for *sentence classification*.

5.3 Publication

During the course of this research, following publications were done;

- Eman Rezk, **Zainab Awan**, Fahad Islam, Somaya Al Madeed, Ali Jaoua, Nan Zhang, Gautam Das, *Proportional Sampling using Binary Patterns:*

Microscopic Images Tumor Classification Application in Machine Learning and Data Analytics Symposium MLDAS 2017, Doha, Qatar.

- Eman Rezk, **Zainab Awan**, Fahad Islam, Somaya Al Madeed, Ali Jaoua, Nasir Rajpoot, Nan Zhang, Gautam Das, "Conceptual Data Sampling Applied in Microscopic Images Tumour Classification". (To be submitted to Journal of Information Science)

Bibliography

[1] Effective sampling methods. `https://faculty.elgin.edu/dkernler/statistics/ch01/1-4.html`. Accessed: 2017-2-07.

[2] Texture classification. `http://www.robots.ox.ac.uk/~vgg/research/texclass/`. Accessed: 2015-12-07.

[3] Dhoha Abid. Segmentation of tumour regions in microscopic images of breast cancer tissue. Master's thesis, Qatar University, 2016.

[4] PR Anisha, C Kishor Kumar Reddy, and LV Narasimha Prasad. A pragmatic approach for detecting liver cancer using image processing and data mining techniques. In *Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on*, pages 352–357. IEEE, 2015.

[5] Erchan Aptoula, Nicolas Courty, and Sébastien Lefèvre. Mitosis detection in breast cancer histological images with mathematical morphology. In *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pages 1–4. IEEE, 2013.

[6] Linda Aune-Lundberg and Geir-Harald Strand. Comparison of variance estimation methods for use with two-dimensional systematic sampling of land use/land cover data. *Environmental Modelling & Software*, 61:87–97, 2014.

[7] Muaaz Gul Awan and Fahad Saeed. On the sampling of big mass spectrometry data. In *7th International Conference on Bioinformatics and Computational Biology, BICOB 2015*. The International Society for Computers and Their Applications (ISCA), 2015.

[8] Jaume Baixeries, Mehdi Kaytoue, and Amedeo Napoli. Characterizing functional dependencies in formal concept analysis with pattern structures. *Annals of mathematics and artificial intelligence*, 72:129–149, 2014.

[9] Roberto Benedetti, Federica Piersimoni, and Paolo Postiglione. *Sampling Spatial Units for Agricultural Surveys*. Springer, 2015.

[10] Ethan D Bolker and Maura M Mast. Relative and absolute change percentages. *non-final Office action issued in connection with US Appl*, (14/069,257), 2007.

[11] Andrei Chekkoury, Parmeshwar Khurd, Jie Ni, Claus Bahlmann, Ali Kamen, Amar Patel, Leo Grady, Maneesh Singh, Martin Groher, Nassir Navab, et al. Automated malignancy detection in breast histopathological images. In *SPIE Medical Imaging*, pages 831515–831515. International Society for Optics and Photonics, 2012.

[12] Samir Elloumi, Jihad Jaam, Ahmed Hasnah, Ali Jaoua, and Ibtissem Nafkha. A multi-level conceptual data reduction approach based on the lukasiewicz implication. *Information Sciences*, 163(4):253–262, 2004.

[13] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.

[14] H Gomez-Moreno, P Gil-Jimenez, S Lafuente-Arroyo, R Vicen-Bueno, and R Sanchez-Montero. Color images segmentation using the support vector machines. *Recent Advances in Intelligent Systems and Signal Processing*, pages 151–155, 2003.

[15] Cigdem Gunduz-Demir, Melih Kandemir, Akif Burak Tosun, and Cenk Sokmensuer. Automatic segmentation of colon glands using object-graphs. *Medical image analysis*, 14(1):1–12, 2010.

[16] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, 3(6):610–621, 1973.

[17] D Jude Hemanth and J Anitha. Image pre-processing and feature extraction techniques for magnetic resonance brain image analysis. In *Computer Applications for Communication, Networking, and Digital Contents*, pages 349–356. Springer, 2012.

[18] Humayun Irshad, Antoine Veillard, Ludovic Roux, and Daniel Racoceanu. Methods for nuclei detection, segmentation, and classification in digital histopathology: a review—current status and future potential. *IEEE reviews in biomedical engineering*, 7:97–114, 2014.

[19] Yunzhi Jiang, Pohsiang Tsai, Zhifeng Hao, and Longbing Cao. Automatic multilevel thresholding for image segmentation using stratified sampling and tabu search. *Soft Computing*, 19(9):2605–2617, 2015.

[20] Rohini Paul Joseph, C Senthil Singh, and M Manikandan. Brain tumor mri image segmentation and detection in image processing. *International Journal of Research in Engineering and Technology*, 3(1):1–5, 2014.

[21] Wang Li and Luo Wei. Data dimension reduction based on concept lattices in image mining. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, volume 5, pages 369–373. IEEE, 2009.

[22] Shivang Naik, Scott Doyle, Shannon Agner, Anant Madabhushi, Michael Feldman, and John Tomaszewski. Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 284–287. IEEE, 2008.

[23] Kien Nguyen, Michael Barnes, Chukka Srinivas, and Christophe Chefd'hotel. Automatic glandular and tubule region segmentation in histological grading of breast cancer. In *SPIE Medical Imaging*, pages 94200G–94200G. International Society for Optics and Photonics, 2015.

[24] Mohammad Peikari, Mehrdad J Gangeh, Judit Zubovits, Gina Clarke, and Anne L Martel. Triaging diagnostically relevant regions from pathology whole slides of breast cancer: A texture based approach. *IEEE transactions on medical imaging*, 35(1):307–315, 2016.

[25] AiPing Qu, JiaMei Chen, LinWei Wang, JingPing Yuan, Fang Yang, QingMing Xiang, Ninu Maskey, GuiFang Yang, Juan Liu, and Yan Li. Segmentation of hematoxylin-eosin stained breast cancer histopathological images based on pixel-wise svm classifier. *Science China Information Sciences*, 58(9):1–13, 2015.

[26] Eman Rezk, Syrinne Babi, Fahad Islam, and Ali Jaoua. Uncertain training data set conceptual reduction: A machine learning perspective. In *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on*, pages 1842–1849. IEEE, 2016.

[27] Rahimeh Rouhi, Mehdi Jafari, Shohreh Kasaei, and Peiman Keshavarzian. Benign and malignant breast tumors classification based on region growing and cnn segmentation. *Expert Systems with Applications*, 42(3):990–1002, 2015.

[28] K Sakthivel, R Nallusamy, and C Kavitha. Color image segmentation using svm pixel classification image. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 8(10):1919–1925, 2015.

[29] Kazuhito Sawase and Hajime Nobuhara. A lattice visualization method by formal concept analysis and its application to huge image database. In *Soft Computing in Industrial Applications, 2008. SMCia'08. IEEE Conference on*, pages 149–152. IEEE, 2008.

[30] Akinobu Shimizu, Tatsuya Kimoto, Hidefumi Kobatake, Shigeru Nawano, and Kenji Shinozaki. Automated pancreas segmentation from three-dimensional contrast-enhanced computed tomography. *International journal of computer assisted radiology and surgery*, 5(1):85–98, 2010.

[31] Shanhui Sun, Christian Bauer, and Reinhard Beichel. Automated 3-d segmentation of lungs with lung cancer in ct data using a novel robust

active shape model approach. *IEEE transactions on medical imaging*, 31(2):449–460, 2012.

[32] Yves Tillé. *Sampling algorithms*. Springer, 2011.

[33] Nima Torbati, Ahmad Ayatollahi, and Ali Kermani. An efficient neural network based method for medical image segmentation. *Computers in biology and medicine*, 44:76–87, 2014.

[34] Xiang-Yang Wang, Xian-Jin Zhang, Hong-Ying Yang, and Juan Bu. A pixel-based color image segmentation using support vector machine and fuzzy c-means. *Neural Networks*, 33:148–159, 2012.

[35] Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In *Ordered sets*, pages 445–470. Springer, 1982.

[36] Qizhi Xiao, Kun Qin, Zequn Guan, and Tao Wu. Image mining for robot vision based on concept analysis. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 207–212. IEEE, 2007.

[37] Li Yang and Yang Xu. Decision making with uncertainty information based on lattice-valued fuzzy concept lattice. *J. UCS*, 16(1):159–177, 2010.

[38] Xiaodong Zhang, Fucang Jia, Suhuai Luo, Guiying Liu, and Qingmao Hu. A marker-based watershed method for x-ray image segmentation. *Computer methods and programs in biomedicine*, 113(3):894–903, 2014.