CLASSIFICATION OF BOVINE REPRODUCTIVE CYCLE PHASES USING ULTRASOUND-DETECTED FEATURES

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Idalia Maldonado Castillo

©Idalia Maldonado Castillo, June/2007. All rights reserved.

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

With the combination of computer-assisted image analysis and ultrasonographic imaging technology, it has been possible to study and increase the knowledge in different areas of medicine. Studies of ovarian development in female mammals using ultrasonography have shown a relationship between the day in the estrous cycle and the main structures of the ovary.

Ultrasound images of bovine ovaries were used to determine whether ultrasound-detected features can automatically determine the phase in the estrous cycle based on a single day's ultrasound examination of the ovaries. Five ultrasound-detected features of the bovine ovaries were used to determine the phase in the estrous cycle: (1) size of the dominant follicle; (2) size of the first subordinate follicle; (3) size of the second subordinate follicle; (4) size of the corpus luteum and (5) number of subordinate follicles with size $\geq$ 2mm. The collection of ultrasound images used for this study was formed by a group of 45 pairs of ovaries (left and right) which were imaged on day 3, day $\simeq$10 and day $\geq$ 17 of the estrous cycle corresponding to the metestrus, diestrus and proestrus phases respectively.

Four different experiments were performed to test the hypothesis. For experiments 1, 2 and 3 the bovine ovaries were classified into three different classes: day 3 of wave 1 (D3W1), day 1 of wave 2 (D1W2) and day 17 or higher (D$\geq$17) that were related to the follicular development of the ovary and the estrous cycle phases as: metestrus, diestrus and proestrus respectively. For experiment 4 the bovine ovaries were classified into four classes: D3W1, D6W1, D1W2 and D$\geq$17. The additional class (D6W1: day 6 of wave 1) was incorporated to represent the early-diestrus phase in the estrous cycle.

Two classifiers were implemented for all experiments and their performances compared: a decision tree classifier and a naïve Bayes classifier. The decision tree classifier had the best performance with a classification rate of 100% for experiments 1, 2 and 3, giving a rather simple decision tree which used only two features to make a classification: size of the dominant follicle and size of the corpus luteum, suggesting these are key features in distinguishing between phases in the estrous cycle giving the most relevant information. The naïve Bayes had a classification rate of 86.36% for experiment 1, 95.55% for experiment 2 and 90% for experiment 3. The results of this study supported the hypothesis that by using ultrasound detected features of bovine ovaries we can determine automatically the stage in the estrous cycle based on a single day's examination.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Mark G. Eramian, for his advice and careful guidance and for his constant support in difficult times.

A very special thanks goes out to Dr. Gregg P. Adams and Dr. Jaswant Singh from the Department of Veterinary Biomedical Sciences, Western College of Veterinary Medicine and Dr. Roger A. Pierson from Obstetrics, Gynecology, and Reproductive Sciences at the University of Saskatchewan, for all their knowledge and information they shared with me; without all their guidance, data and images provided, this research project would not have been possible. I also would like to thank Dr. Michael Horsch and Dr. Edward Kendall for their invaluable suggestions and instructive comments for this project.

This thesis is dedicated to my parents. To my dear mother Josefina Castillo for being an inspiration and support in my life and whose support encouraged me to continue to the final line and in memory of my beloved father Carlos Bertoldo Maldonado who sadly departed this world before this manuscript was finished but will always live in my heart.

I wish to express my innermost thanks to my beloved fiancè Risto Rangel, who has been more than an inspiration to me, his love, understanding, patience, advice and support have helped me to follow and accomplish one of my dreams. I will always be grateful for that and I look forward to see the fulfillment of our dreams and for what the future holds for us.

Finally, I would like to express a very special thanks to my brothers: Juan Carlos, Marco Antonio and Edgar Maldonado who have supported and encouraged me through my entire life and always have believed in me.

CONTENTS

Permission to Use i

Abstract ii

Acknowledgements iii

Contents iv

List of Tables vi

List of Figures vii

List of Abbreviations viii

1 Introduction 1
- 1.1 Goal of the thesis 1
- 1.2 Motivation 2
- 1.3 Structure of the thesis 2

2 Background and Literature Review 4
- 2.1 Reproductive biology 4
- 2.2 Classification methods 9
  - 2.2.1 Nearest neighbor classification 10
  - 2.2.2 Decision tree classifier 12
  - 2.2.3 Bayesian decision classifier 15
  - 2.2.4 Support vector machine 17
  - 2.2.5 Clustering 19

3 Materials and Methods 22
- 3.1 Image data set 22
- 3.2 Feature selection and extraction 23
- 3.3 Classifier design 26
  - 3.3.1 Naïve Bayes classifier 27
  - 3.3.2 Decision tree classifier 33
- 3.4 Validation methods 44

4 Estrous Phase Classification (Experiments and Results) 48
- 4.1 Experiment 1: 3-class classification using hold–out estimate methodology 48
  - 4.1.1 Decision tree classifier 49
  - 4.1.2 Naïve Bayes classifier 50
- 4.2 Experiment 2: 3-class classification using cross-validation methodology 50
  - 4.2.1 Decision tree classifier 51
  - 4.2.2 Naïve Bayes classifier 55
- 4.3 Experiment 3: 3-class classification for animals with 2-wave follicular patterns 57
  - 4.3.1 Decision tree classifier 57
  - 4.3.2 Naïve Bayes classifier 59
- 4.4 Experiment 4: 4-class classification using hold–out estimate methodology 60
  - 4.4.1 Decision tree classifier 61
  - 4.4.2 Naïve Bayes classifier 62

5 Discussion and Conclusions **64**
5.1 Experiment 1: 3-class classification using hold–out estimate methodology 64
5.2 Experiment 2: 3-class classification using cross-validation methodology 65
5.3 Experiment 3: 3-class classification for animals with 2-wave patterns 65
5.4 Experiment 4: 4-class classification using hold–out estimate methodology 66

References **71**

A Bovine Ultrasound Image Data Set **72**
A.1 Data set A 72
A.2 Data set B 74

B Experiment Results **76**
B.1 Cross validation results 76

List of Tables

3.1 Training data set A for class D3W1 with summary statistics. 30
3.2 Training data set A for class D1W2 with summary statistics. 31
3.3 Training data set A for class D≥17 with summary statistics. 32
3.4 Decision tree learning algorithm 36
3.5 Dominant feature values from training data set A. 39
3.6 Corpus luteum feature values from training data set A. 42
3.7 Example of a confusion matrix. 46

4.1 Results from experiment 1. Confusion matrix resulting from the classification of data set B by the decision tree classifier. 49
4.2 Results from experiment 1. Confusion matrix resulting from the classification of data set B by the naïve Bayes classifier. 50
4.3 Results from experiment 2. Confusion matrix resulting from run 1 for the cross validation by the decision tree classifier. 52
4.4 Results from experiment 2. Confusion matrix resulting from run 2 for the cross validation by the decision tree classifier. 53
4.5 Results from experiment 2. Confusion matrix resulting from run 3 for the cross validation by the decision tree classifier. 53
4.6 Results from experiment 2. Confusion matrix resulting from run 4 for the cross validation by the decision tree classifier. 54
4.7 Results from experiment 2. Confusion matrix resulting from run 5 for the cross validation by the decision tree classifier. 55
4.8 Results from experiment 2. Confusion matrix resulting from run 2 for the cross validation by the naïve Bayes classifier. 56
4.9 Results from experiment 2. Confusion matrix resulting from run 4 for the cross validation by the naïve Bayes classifier. 56
4.10 Results from experiment 3. Confusion matrix resulting from training with data set A' and testing with data set B' for the decision tree classifier. 59
4.11 Results from experiment 3. Confusion matrix resulting from training with data set A' and testing with data set B' for the naïve Bayes classifier. 59
4.12 Results from experiment 4. Confusion matrix resulting from the decision tree classifier. 63
4.13 Results from experiment 4. Confusion matrix resulting from the naïve Bayes classifier. 63

A.1 Feature Values for data set A. 72
A.2 Summary Statistics for data set A. 73
A.3 Feature values for data set B. 74
A.4 Summary statistics for data set B. 75

B.1 Feature values for data set $A \cup B$. 76
B.2 Results from experiment 2. Confusion matrix resulting from the classification using the cross validation technique by the decision tree classifier. 77
B.3 Results from experiment 2. Confusion matrix resulting from the classification using the cross validation technique by the naïve Bayes classifier. 77
B.4 Summary statistics for data set A and data set B used for cross validation. 78

LIST OF FIGURES

2.1 Schematic representation of an ovary showing the main ovarian structures: dominant and subordinate follicles and corpus luteum. 5
2.2 Schematic representation of the 2-wave and 3-wave pattern. 6
2.3 Schematic representation of follicle wave growth in a 2-wave cycle. 7
2.4 Image representing the development of ovulatory ovarian follicles during the follicular, ovulatory and luteal phases of the estrous cycle. 8
2.5 Schematic representation of corpus luteum growth during the estrous cycle 8
2.6 Ultrasound image of a bovine ovary with major structures identified. 9
2.7 Schematic example of a decision tree classifier. 13
2.8 A structure of a multi-class pattern classifier using discriminant functions. 16
2.9 Principle of the support vector machine (SVM). 18

3.1 Mean feature values for training data set A after feature extraction. 25
3.2 Mean feature values for testing data set B after feature extraction. 26
3.3 Matlab GUI application for the manual feature extraction displaying the horizontal diameter of the follicle. 27
3.4 Matlab GUI application for the manual feature extraction displaying the vertical diameter of the follicle. 28
3.5 Diagram showing the process in a decision tree creation. 35
3.6 Decision tree generated by training data set A. 44
3.7 Cross validation methodology. 47

4.1 Mean and standard deviation feature values for the complete data set formed by data set A and data set B ($A \cup B$). 51
4.2 Decision tree for experiment 2 using cross validation methodology. 52
4.3 Decision tree for experiment 2 using cross validation methodology. 54
4.4 Mean feature values for training data set A′. 58
4.5 Mean feature values for testing data set B′. 58
4.6 Mean feature values for data set A with 4 classes. 60
4.7 Mean feature values for data set B with 4 classes. 61
4.8 Decision tree for experiment 4 using data set A as the training set including 4 classes. 62

List of Abbreviations

AI	Artificial Intelligence
CL	Corpus luteum
D	Dominant follicle
S1	First subordinate follicle
GUI	Graphical User Interface
IOI	Interovulatory interval
k-NN	k-nearest neighbor
MRI	Magnetic Resonance Imaging
NF	Number of follicles
pdf	probability density function
S2	Second subordinate follicle
SVM	Support Vector Machines

CHAPTER 1

INTRODUCTION

Ovaries are an important part of the reproductive system in female mammals. The reproductive cycles of female mammals can be monitored by imaging the ovaries using ultrasonography [4, 6, 7, 19, 20, 27, 28, 30, 32, 42, 43]. Studies of ovarian development in cattle have shown a relationship between the day in the estrous cycle and the characteristics of ovarian structures (dominant and subordinate follicles and corpora lutea) [21, 22, 23, 29]. The main structures inside the ovary are visible in ultrasound images.

The characteristics of ovarian structures provide clues to the current reproductive status or phase of the animal. The reproductive cycle, or *estrous* cycle, of domestic animals consists of four phases: metestrus, diestrus, proestrus and estrus. In non-pregnant cows, ovulation occurs at approximately 19 to 23 day intervals and usually happens near the end of (or shortly after) the estrus phase.

1.1 Goal of the thesis

The objective of this project was to test the hypothesis that by using ultrasound detected features of bovine ovaries we can determine automatically the stage in the estrous cycle based on a single day's examination. The bovine ovaries were classified into three different classes which corresponded roughly to the stages in the estrous cycle in cattle known as: metestrus, diestrus and proestrus respectively. A set of features derived from the characteristics of the ovarian structures in the ultrasound images were used to perform the classification. The features were measured over both left and right ovaries and used to construct a classifier to determine the stage in the estrous cycle.

The collection of ultrasound images used for this study was formed by a group of 45 pairs of ovaries which were collected and imaged on day 3, day $\simeq$10 and day $\geq$ 17 of the estrous cycle corresponding to the metestrus, diestrus and proestrus phases respectively. For this study two pattern recognition methods were used to classify the bovine ovaries into temporal categories using the features extracted from ultrasound images: a decision tree classifier and a naïve Bayes classifier. Both classifiers were fully implemented and their performance compared. The implementations were compared with results from an existing machine learning and data mining application called Weka

[50] to gain confidence in the correctness of the implementations.

1.2 Motivation

Ultrasonography has become an essential tool for monitoring ovarian maturation and ovulation of female mammals during the estrous cycle. Advancements in the ability to monitor ovaries help with the determination of follicle growth patterns and detection of impending ovulation.

At present the determination of the stage of the estrous cycle can only be made after serial examination of the ovaries over several days. To date, there is no method to identify the physiological status of the ovaries on the basis of a single ultrasound examination.

The determination of the phase in the estrous cycle of an animal automatically from a single day's examination would enable rapid decisions to be made whether to begin monitoring the animal daily in order to determine specific follicle selection, facilitate the division of a livestock herd into reproductively active or unresponsive groups, and aid in the determination of optimal timing for insemination. In addition, excised bovine ovaries obtained from abbatoirs are routinely used for *in vitro* fertilization and embryo production. Automated classification of these ovaries into different estrous cycle phases will help in obtaining uniform groups of oocytes for commercial and scientific purposes.

This study can also be a precedent for future studies applied to humans, where the determination of the stage in the reproductive cycle can help women undergoing controlled ovarian hyperstimulation or ovulation induction prior to insemination or oocyte/egg retrieval for the treatment of fertility.

1.3 Structure of the thesis

The material presented in this thesis is organized into five chapters. Chapter 1 gives a general introduction presenting the objective, importance and motivation of this research. A general background and literature review is provided in Chapter 2. This includes an introduction of the reproductive biology, focusing on the main ovarian structures as well as ovarian follicular development. It continues with a survey of pattern recognition techniques.

Chapter 3 begins with a detailed explanation of the ultrasound image data set used in this research followed by a discussion of the selection and extraction of the image features used for the classification. Then a detailed description of the classification implementation is given for both classifiers: the decision tree classifier and the naïve Bayes classifer.

The different experiments and results for both classifiers are presented in Chapter 4. The presentation of the results gives a detailed explanation of four different experiments made for this study. An evaluation of the four experiments is included in this chapter.

Chapter 5 includes a summary of the research and methods, including a discussion and conclusions made from the experimental results. Finally suggestions for future research are proposed.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

For the purpose of this thesis it is important to recognize the main structures of the ovary and the way the structures develop, moreover we are interested in the different stages that the ovaries undergo during the estrous cycle. In this chapter a review of structures found within the ovary and their development patterns over the course of an estrous cycle is conducted in Section 2.1, common pattern recognition techniques are reviewed in Section 2.2.

2.1 Reproductive biology

Ovaries are part of the reproductive system in female mammals. Female mammals have two ovaries (left and right) whose main function is to produce and release eggs. The major ovarian structures are follicles and corpora lutea. Ovarian follicles are roughly spherical, fluid-filled structures which contain developing oocytes (eggs). The reproductive cycle or estrous cycle culminates in the rupture of a follicle and release of its egg. The corpus lutem (CL) is a gland that is formed from the remains of the ruptured follicle following ovulation. These structures are illustrated schematically in Figure 2.1.

Ovarian follicular development is a dynamic process which occurs in a wave-like pattern during the estrous cycle [4, 6, 7, 30, 31]. A group of follicles begin growing simultaneously as a cohort at a diameter of 2 to 4 mm (wave emergence). The group of follicles continues growing for 2 to 3 days. At this time, all follicles in this cohort, except one (the dominant follicle), begin to regress and degenerate while the dominant follicle continues preferential development (continues growing).

It has been shown that cattle may have either two or three waves of follicular activity per estrous cycle. These studies have shown that around 80% of heifers have two waves (heifers are young cows who have not yet given birth to a calf) and 20% have three waves of follicular development [22, 23]. In both 2- and 3-wave growth patterns, the dominant follicle of the final wave ovulates, while dominant follicles of earlier waves ultimately regress and degenerate in a process known as atresia. All subordinate follicles in each follicular wave ultimately become atretic (degenerate). Figure 2.2 gives a schematic representation of the 2-wave and 3-wave pattern. Follicles within a wave either ovulate or degenerate; follicles do not regress at the end of one wave, and then re-emerge in a

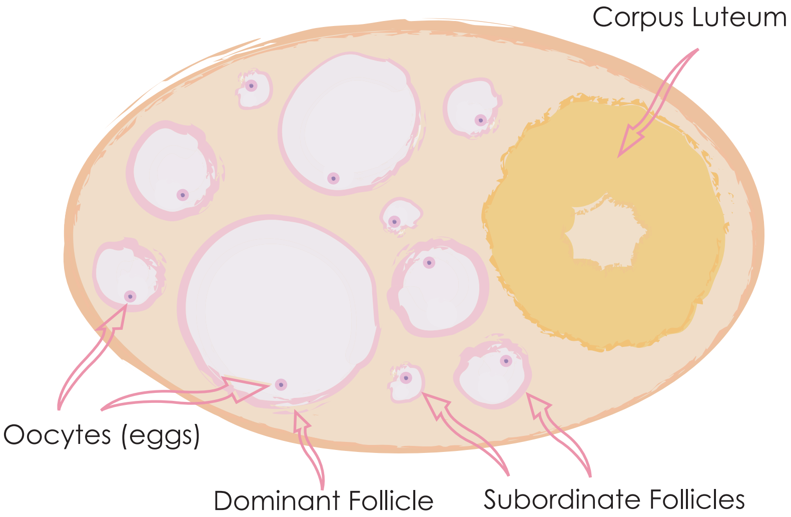


Figure 2.1: Schematic representation of an ovary. The major structures are follicles, in which oocytes (eggs) develop, and the corpus luteum (CL). The largest follicle is termed the "dominant" follicle and the remaining follicles are termed "subordinate" follicles.

subsequent wave. [4, 6, 7, 27, 31].

In 2-wave cycles, the first wave emerges on the day of ovulation, denoted as day 0. During the following days, all follicles inside both ovaries start growing as a group. At around day 3 of wave 1 a follicle is preferentially selected as the dominant follicle [1, 2]. The dominant follicle continues to develop and grow, while the remaining subordinate follicles in the cohort regress and degenerate. The dominant follicle reaches its maximal diameter around day 6 of wave 1, remains static for a time, and begins to degenerate around day 9 or 10. At this time, a second wave of completely new follicles emerges (wave 2). The dominant follicle from this second cohort ovulates at the end of the cycle at around day 21 and the subordinate follicles regress and degenerate, as in the first wave [21, 22]. For both waves, subordinate follicles cease growing approximately 3 days after the cohort of follicles is first visible using ultrasonography. The dominant follicle of wave 1 is characterized for having a growing phase during day 3 of wave 1 (D3W1), a static phase during day 1 of wave 2 (D1W2) and a regressing phase on day 17 or higher (D$\geq$17) during the estrous cycle. A new estrous cycle begins if pregnancy is not established. This follicle wave growth process is illustrated in Figure 2.3 and is described in more detail in [21, 22]. In cattle with 2-wave pattern the estrous cycle lasts 19-20 days in average with wave emergences in day 0 and day 10. Similarly for 3-wave pattern, the wave emergences are on days 0, 9 and 16, with an estrous cycle of 23 days [4, 23].

For cattle the estrous cycle can be divided into four basic and important phases: estrus, metestrus, diestrus and proestrus. The *estrus* phase is characterized by high sexual activity, also called "heat", because it describes the degree of activity and excitement associated with this phase.

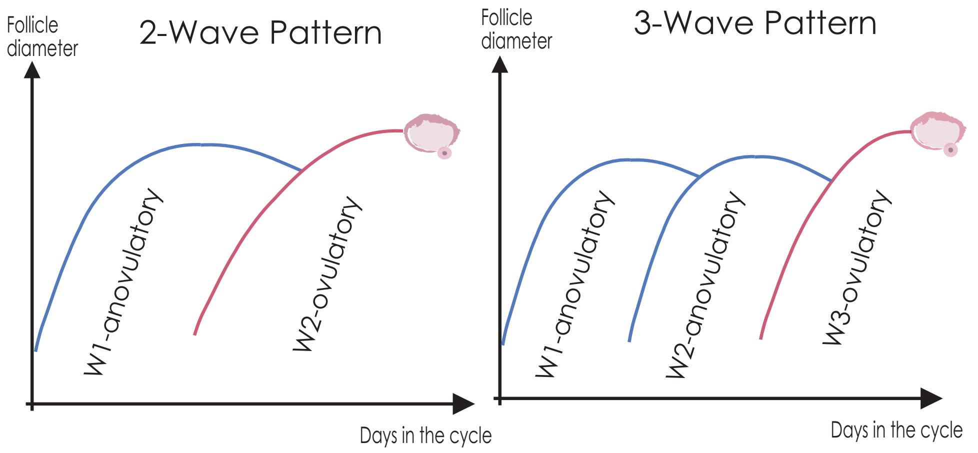


Figure 2.2: Schematic representation of the 2-wave and 3-wave pattern. In 2- and 3-wave like pattern the final wave is ovulatory (ovulation occurs), while all preceding waves are anovulatory (no ovulation occurs). This image is a schematic illustration that shows the waves formed by the growing and regressing stages of the dominant follicles of each wave.

The next period is a transition out of the high sexual activity period (the post-ovulatory period) called *metestrus* (after estrus). The period of *diestrus* is of low sexual activity, and finally the phase prior to estrus called *proestrus* [26, 45].

Following ovulation, the wall of the ruptured dominant follicle collapses forming a structure called the corpus luteum CL (see Figure 2.4). The CL passes through a period of initial growth during *metestrus*, followed by a period of maximal size and function during *diestrus* and ending with a period of regression *proestrus* and ultimate demise (*proestrus/estrus*) preceding the next ovulation [28, 41]. Figure 2.5 depicts the mean values for the CL's diameter during the estrous cycle obtained from a previous study by Tom et. al. [47]. In non-pregnant cows, ovulation occurs at 19 to 23 day intervals and usually occurs near the end of (or shortly after) estrus. Ovulation can occur either in the left ovary or the right ovary, at present the selection of the ovary in which the ovulation will occur seems to be random and no pattern has been found [4, 8]. The dominant follicle that will ovulate can emerge from either of the two ovaries; similarly the corpus luteum (generated from the previous ovulation) can be located either in the same ovary as the current dominant follicle or the opposite ovary. The group of subordinate follicles appear in both left and right ovary during the estrous cycle. Therefore the main structures inside the ovaries (follicles and corpora lutea) can be present in any of the two ovaries; the left and right ovaries work as one.

Due to the wave-like follicular growth, it was conjectured that the size of the dominant follicle, the size of the two largest subordinate follicles, the total number of subordinate follicles, and the size of the CL would be useful features for distinguishing between the metestrus, diestrus and proestrus

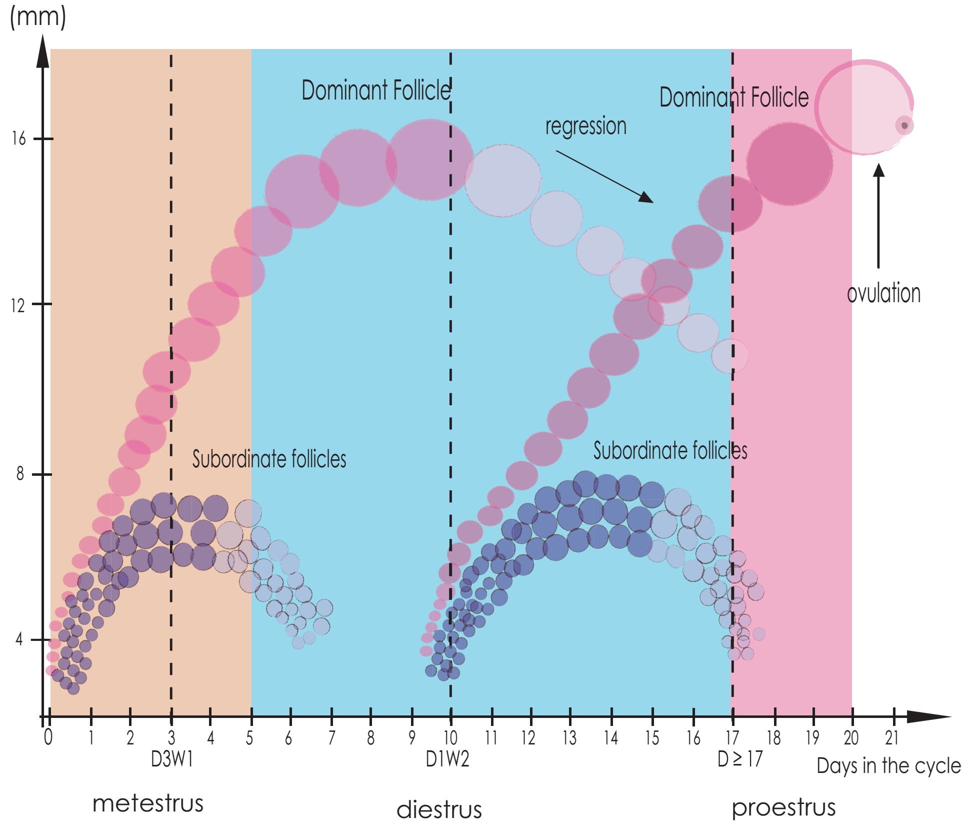


Figure 2.3: Schematic representation of follicle wave growth in a 2-wave cycle. Dashed lines represent the growing (D3W1), static (D1W2) and regressing (D≥17) phases of the dominant follicle of wave 1. The lines illustrate also the days on which the ultrasound images were taken for this study.

phases. The relationship between the day in the estrous cycle and the sizes of the main structures of the ovary has been studied in detail [21, 22, 23, 29].

There are different imaging techniques that have given significant advances in the understanding of the ovary and the wave-phenomenon of folliculogenesis. The most important imaging technology is ultrasonography, which has been widely used for many years as a research tool in ovarian imaging. Ultrasonography provides an accessible method of sequentially monitoring the ovaries over time in both animals and humans [4, 6, 7, 19, 20, 27, 28, 30, 32, 42, 43].

Figure 2.6 shows an ultrasound image of an ovary of the bovine species with the main ovarian structures identified. The main structures (dominant follicle, subordinate follicles and CL) were identified by a human expert. Follicles appear as black roughly circumscribed areas, like other fluid-filled structures. Although follicles can be confused with other black areas, such as the central cavity of the CL, follicles usually can be distinguished by their spherical appearance. The CL has a different echogenic pattern than that of surrounding tissues [42, 47]. According to a previous study

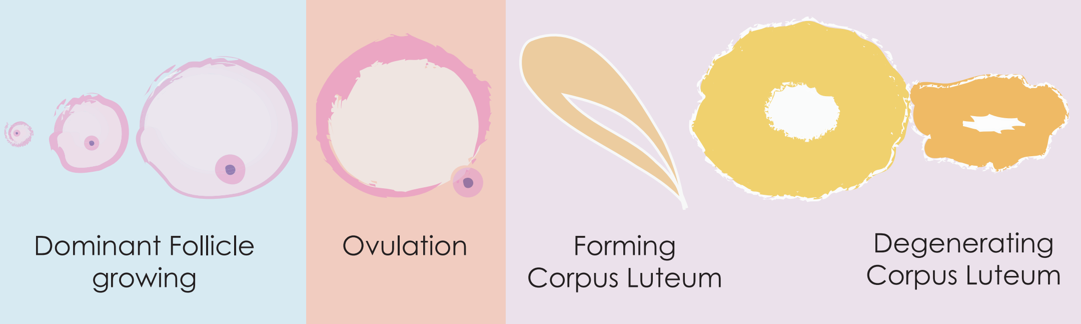


Figure 2.4: Schematic representation of a dominant follicle destined to ovulate. From left to right; the dominant follicle (with an egg inside) grows until it ovulates, releasing an egg. The ruptured dominant follicle collapses forming a corpus luteum (CL). If no fertilization occurs the corpus luteum starts degenerating.

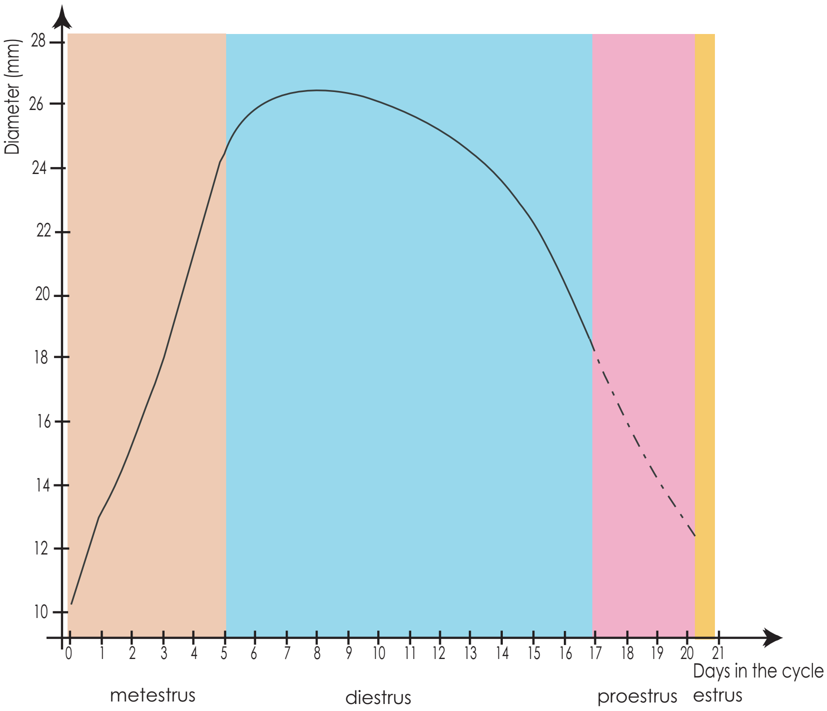


Figure 2.5: Schematic representation of corpus luteum growth during the estrous cycle. This figure expresses the CL diameter in mm. Dashed line represents the period of regression of the CL.

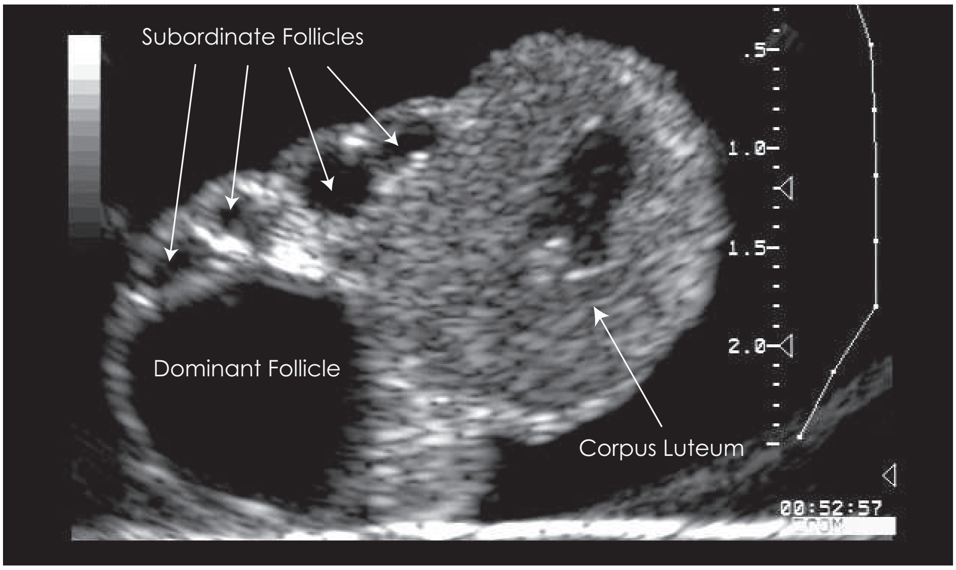


Figure 2.6: Ultrasound image of a bovine ovary with major structures identified [25].

done by Singh et. al. [42] 72.4% of corpora lutea presents a central cavity.

2.2 Classification methods

Classification methods are used to classify a pattern or a set of features into one of a number of categories or class. This section will give a brief survey of some of the most common classification methods. A complete description of the design and implementation for the decision tree classifier and the naïve Bayes classifier will be given in Chapter 4.

Classification methods are developed in two stages. The first stage is the training stage; the classifier learns to distinguish classes from patterns of features taken from a *training set*. In the testing stage the classifier determines the class of new patterns from a *testing set* whose true classes are unknown to the classifier. There are many types of classifiers and they can be applied depending on the information and structure about the patterns to be classified. Generally, any method that tries to classify patterns into certain classes employs some kind of learning. "Learning refers to some form of algorithm for reducing the error on a set of training data" [16].

Pattern classification may use two approaches of learning: supervised classification and unsupervised classification. In supervised classification the classes are known and used in the training stage. In unsupervised classification the classes are unknown or it may be the case that they are known but are not used in the training stage.

Supervised classification is used when there is information about the patterns. Then, given a collection of samples (training set) with classes associated with them, the classifier is able to infer knowledge about the classes. In this way, the supervised classifier is able to classify future samples as belonging to one of the same set of known classes [35].

In unsupervised classification, sometimes called clustering, there is no information about the classes. That is, the training samples used to design the classifier are not labeled by their category membership. The most important reason for using unsupervised classification is when the characteristics of the objects are unknown or there is no real information about them. A second reason is that collecting and labeling a large data set can be very expensive and time consuming. Also, if the characteristics of the objects vary with time, even when they are the exact same object, they may look completely different (e.g. beans that change size and color depending on the season) [13, 16, 49].

2.2.1 Nearest neighbor classification

The nearest neighbor algorithm (NN) is the most intuitive pattern recognition technique. It is a flexible but very computationally expensive method of classification. It is used to classify an unknown feature vector into a class by associating it with the nearest vector (in feature space) of the known class. Each n-element feature vector represents a point in an n-dimensional *feature space*. Similarity between patterns is then defined in terms of distance between points in feature space (more distant points are less similar).

The basic principle of the nearest neighbor algorithm is that samples which fall close together in feature space are likely to belong to the same class. The nearest neighbor classifier stores the training patterns for each class; then the input (test/unknown) pattern is compared against all the stored patterns and assigned to the class of the pattern which is most similar, that is, closest in feature space to the input pattern [14].

This algorithm has some limitations. A problem may occur when the training pattern distributions of two or more classes in feature space overlap. If this occurs, is almost impossible to differentiate between classes hence misclassification can occur.

Another problem is when the training patterns are not strongly representative of their classes and therefore is likely that the classification will fail during the test phase. For this reason, it is important to note that there must be enough patterns in the training set to assure the algorithm will be able to generalize over all possible patterns of each class. Thus, this algorithm usually requires a large number of training patterns in order to have a low error rate resulting in a high storage requirement and a large computation time.

There are approaches to decrease the effects of storage and computation time: the first one is an "editing technique"where patterns that lead to misclassification are removed so that class distributions of training samples do not overlap in feature space. This approach gives a homogeneous set of clusters of samples [49].

Another technique is called condensing and is focused on reducing the number of training samples by removing the patterns in the training set which are deeply embedded in the class (not close to the boundaries of the class regions in feature space) and do not help to reduce the error in the nearest neighbor classification [13, 49].

k–nearest neighbor classifier

A refinement to the nearest neighbor method can be made by examining the nearest k feature vectors; it is called the k-nearest neighbor (k-NN) classifier.

The basic idea of the k-NN is to collect the k nearest neighbors of a pattern x and assign it the class which is most frequently represented among the k nearest neighbors; in other words, a decision is made by taking the majority vote of the k nearest neighbors. A second approach of this classifier is to classify a pattern x in a class that receives a number of votes which is at least equal to a qualifying majority level l, otherwise the pattern is rejected, this approach is known as the (k, l)-NN classifier [14, 15].

Despite the improvements to this classifier, the nearest neighbor and k-NN methods have the big disadvantage that they still require enormous storage to record enough training set pattern vectors, and correspondingly large amounts of computation to search through them to find an optimal match for each test pattern [13].

The general approach for the nearest neighbor learning is by storing all the available training samples with their corresponding class in memory, then for each testing pattern a distance function has to be used to determine which training sample is closest. When the closest training sample is found, the class related to that training sample is assigned as the predicted class to the unknown testing pattern.

A number of distance metrics may be used in nearest neighbor classification and other pattern recognition methods. The most common is the Euclidean distance. A brief description of this, and some alternate measures of distances is given in the following list [16, 49]:

- Euclidean distance: the Euclidean (straight line) distance d_e between two points $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$ in n dimensions is calculated as:

$$d_e(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{2.1}$$

- City block distance: also called Manhattan distance, this metric calculates the distance between two points measured along axes at right angles. This calculation would be suitable for finding distances between points in a city consisting of a grid of intersecting streets. The city block distance d_{cb} between two points $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$ in n dimensions is calculated as follows:

$$d_{cb}(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i|. \tag{2.2}$$

  The city-block metric is a little cheaper to compute than the Euclidean distance so it may be used if the speed of a particular application is important;

- Chebyshev distance: the Chebyshev distance, or maximum value distance, is often used when the execution speed is critical. The Chebyshev distance calculates the absolute magnitude of the difference between two points $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$ and selects the greatest of their differences along any coordinate dimension and is defined as follows:

$$d_{ch}(\vec{x}, \vec{y}) = \max_i |x_i - y_i|. \tag{2.3}$$

- Minkowski distance: the Minkowski distance, also known as L_m distance, is a generalized form of the previous distances. The Minkowski distance d_m between two points $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$ of order m is:

$$d_m(\vec{x}, \vec{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^m \right)^{1/m}. \tag{2.4}$$

  The Minkowski distance of first order $m = 1$ is the city block distance, for the second order $m = 2$ is the Euclidean metric. When the order tends to infinity the Minkowski distance tends to the Chebyshev distance.

2.2.2 Decision tree classifier

A decision tree classifier can be seen as a multistage decision process. A natural way to classify a pattern can be seen as a sequence of questions; therefore instead of using all the set of features together to make a decision, different questions about the subsets of features can be used at different levels of the decision tree.

A classification tree is a conceptually simple approximation to a complex procedure that breaks up the decision into a series of simpler decisions at each node. Figure 2.7 shows an example of a decision tree for a fruit classification adapted from [16].

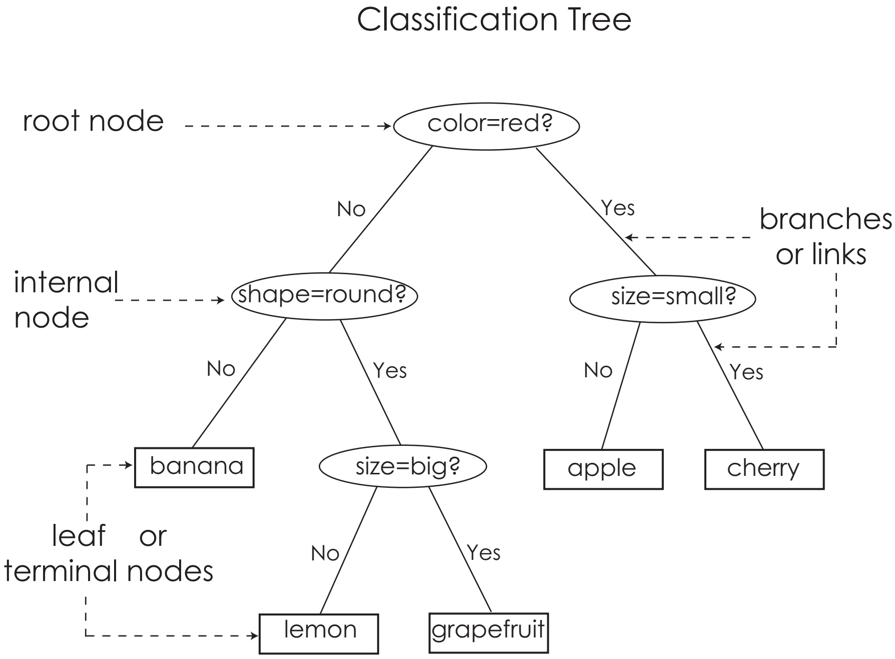


Figure 2.7: Example of a decision tree classifier. The classification in a decision tree proceeds from top to bottom. Starting from the root node. The questions in each node refers to a particular feature of the pattern, which answer leads to the links. Note that this example is a binary tree, but possible values can be numbers, ranges, etc, and not just the binary values such as true/false. The successive internal nodes are visited until a leaf or terminal node is reached. The terminal nodes contain a category/class label, then the classification is made by assigning the class to the pattern.

In Figure 2.7, the root node is displayed at the top of the tree and is connected by directional links or branches to the internal nodes, which are connected also by branches to either internal nodes or leaf nodes (also called terminal nodes). Each internal node is associated with a decision. A class label is associated with each leaf node.

Thus, the classification process is as follows: the pattern to be classified by the decision tree (formed by a set of features) starts at the root node; then the first of a sequence of decisions is made by asking for a particular property of the pattern (features). Depending on the answer, a branch is selected and continues to the internal (descendent) node. The next step is again make a decision in the current node, and so on, until a leaf or terminal node is reached. The leaf has no question or decision to be made, instead it contains a single class label, which is assigned to the pattern being classified. The same class label may appear in different leaves [16, 24, 49].

Decision tree construction is based on the training set and can be expressed recursively. It consists of selecting the most representative or appropriate feature to be place at the root node of the decision tree and makes one branch for each possible value of such a feature. This will split the training set into subsets, one for every value of the feature. Then the process is repeated for each branch recursively, using only the instances that reached such a branch. The decision tree construction stops when the data cannot be split any further or when all the instances at a certain node have the same classification.

According to [24] there are important characteristics about the decision tree classifiers:

- it is feasible to have a tree classifier when all the objects are distinguishable, in other words, the data set which is used to design the classifier has no identical elements with different class labels;
- tree classifiers are very intuitive, the decision process can be traced as a sequence of simple decisions, and can gain a knowledge base in a hierarchical arrangement;
- binary features and features with small number of categories are very useful when designing the tree; the decision can be easily branched out. Also both quantitative and qualitative features are suitable for building the decision tree classifier.

Moreover, decision tree classifiers do not work based on a concept of distance in the feature space which is a great advantage when the objects are described by categorical or mixed-type features since the distance can be very difficult to formulate [24].

Classification trees are used in a wide variety of problems, mainly because they can be compactly stored. Also, it has been demonstrated that they have good generalization performance on a wide range of problems and efficiently classify new samples [49]. Perhaps, the most important advantage of decision tree classifiers is the ability to break down a complex–decision making process into a collection of simpler decisions, thus providing a solution which is easier to interpret and may provide more insight into the structure of the data set [44].

In contrast, an important disadvantage with this classifier is the difficulty of designing an optimal decision tree, possibly leading to a large tree with poor error rates for certain problems. Other difficulties can arise when there are missing features or uncertainty about the features, which bring an increment in complexity with increased size of the tree [49].

2.2.3 Bayesian decision classifier

The Bayesian decision classifier is a statistical pattern recognition technique that classifies an object into the class to which it most likely belongs based on observed features. Using the Bayes decision rule, the classifier assigns a class w_i to a pattern $\vec{x}$ if

$$p(\vec{x}|w_i)p(w_i) > p(\vec{x}|w_j)p(w_j), j \neq i, \tag{2.5}$$

where $p(\vec{x}|w_i)$ is the *class–conditional probability density function* (pdf) of $\vec{x}$ given class w_i, and $p(w_i)$ is the *a priori* probability of class w_i. It is shown in [15, 16, 49] that the decision rule 2.5, also known as Bayes' rule for *minimum error*, minimizes the probability of making an incorrect decision.

Another important rule for this classifier is the Bayes formula or Bayes' theorem from which the *a posteriori* probability $p(w_i|\vec{x})$ of class w_i given the pattern $\vec{x}$, can be expressed in terms of the *a priori* probabilities and the class–conditional pdf:

$$p(w_i|\vec{x}) = \frac{p(\vec{x}|w_i)p(w_i)}{p(\vec{x})} = \frac{p(\vec{x}|w_i)p(w_i)}{\sum_{i=1}^{c} p(\vec{x}|w_i)p(w_i)}, \tag{2.6}$$

where $p(\vec{x})$ is the *unconditional probability distribution*, also called the evidence, and c is the number of classes.

One way to represent a pattern classifier is in terms of a set of *discriminant functions* $g_i(\vec{x})$, i=1,...,c. Then, the classifier will assign a feature vector $\vec{x}$ to a class w_i if

$$g_i(\vec{x}) > g_j(\vec{x}), j \neq i. \tag{2.7}$$

The classifier is then viewed as a network or machine that computes c discriminant functions, as shown in Figure 2.8 [15, 16], where it assigns to the pattern $\vec{x}$ the class corresponding to the largest discriminant function [14, 15, 16, 24, 49]. "For the minimum–error–rate case, it is possible to simplify things by taking $g_i(\vec{x}) = p(w_i|\vec{x})$, so that the maximum discriminant function corresponds to the maximum a posteriori probability" [16]. There are many options to select a discriminant function. In particular, for minimum–error–rate classification, the discriminant functions can be expressed in a variety of forms which give identical classification results because the decision rules are equivalent [15, 16] as the following equations:

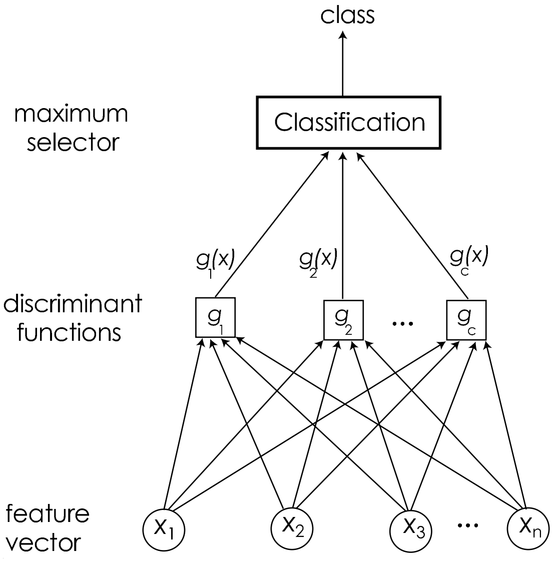


Figure 2.8: Structure of a multi-class pattern classifier. The classifier receives the values of the feature vector $\vec{x}$ as input, and the c discriminant function $g(\vec{x})$ values are calculated. Then, the "maximum selector" determines which of the discriminant values is the maximum, and then categorizes the input pattern into a class w [15, 16].

$$g_i(\vec{x}) = p(w_i|\vec{x}), \tag{2.8}$$

$$g_i(\vec{x}) = p(\vec{x}|w_i)p(w_i), \tag{2.9}$$

$$g_i(\vec{x}) = \log p(\vec{x}|w_i) + \log p(w_i), \tag{2.10}$$

the decision rules are equivalent since removing the evidence (Equation 2.9) or taking the log (Equation 2.10) does not change which discriminant function gives the maximum result.

A special case of Bayes decision rule is the Gaussian classifier, which is one of the most frequently used classifiers in pattern recognition problems. The Gaussian classifier assumes that the class–conditional pdf $p(\vec{x}|w_i)$ is a d-dimensional Gaussian distribution [14, 15, 16, 24, 46, 49], also called normal distribution defined as:

$$p(\vec{x}|w_i) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma_i|}} \exp\left[-\frac{1}{2}(\vec{x}-\mu_i)^t\Sigma_i^{-1}(\vec{x}-\mu_i)\right], \tag{2.11}$$

where Σ_i is a $d \times d$ covariance matrix of the training pattern and μ_i is the mean of class w_i, and $|\Sigma_i|$ is the determinant of Σ_i. Equation 2.10 is particularly useful when $p(\vec{x}|w_i)$ has a Gaussian distribution since substitution of equation 2.11 into equation 2.10 yields the decision function:

$$g_i(\vec{x}) = -\frac{1}{2}(\vec{x}-\mu_i)^t\Sigma_i^{-1}(\vec{x}-\mu_i) - \frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_i| + \log p(w_i). \tag{2.12}$$

The first term in equation 2.12 is also known as the Mahalanobis distance, which is defined as the distance between the feature vector $\vec{x}$ and the mean of the different classes. The second term is a constant which is independent of $\vec{x}$ and i therefore it can be removed without affecting the classification result. During the training process the classifier uses the training set to develop the discriminant function $g_i(x)$ for the class w_i; the covariance matrix Σ_i and the mean μ_i can be estimated for each class w_i from the training set. For example, let's consider a training set formed by n training vectors $(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$. Hence, the mean can be calculated as:

$$\mu_i = \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i, \tag{2.13}$$

and the covariance matrix can be calculated as:

$$\Sigma_i = \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i - \mu_i)(\vec{x}_i - \mu_i)^t. \tag{2.14}$$

Substituting the estimates of the means and the covariance matrices of each class into Equation 2.12 gives the Gaussian classifier; the classification is then achieved by assigning a pattern $\vec{x}$ to a class w_i if $g_i(\vec{x}) > g_j(\vec{x})$ for all $j \neq i$ [49].

Bayesian classification requires knowledge about the classes, since it works essentially with *a priori* probabilities and class–conditioned probabilities. One of the main advantages of this approach is that gives the minimum error rate classification.

A more simple approach to the Bayes classifier is the naïve Bayes model. This works quite well when the dependency relationships among the features used by the classifier are unknown. This technique assumes that the pattern features are conditionally independent. Naïve Bayes classifiers assume that the effect of a feature value on a given class is independent of the values of other features. The naïve Bayes classifier can work surprisingly well in practice, even when the independence assumption is not completely true [16, 37, 50]. This is because the assumption changes the actual posterior probabilities, but rarely, in practice, changes which decision function gives the maximal result.

The Bayesian decision classifier and special cases such as the Gaussian classifier and naïve Bayes model can be studied in more detail in [14, 15, 16, 24, 46, 49].

2.2.4 Support vector machine

The support vector machine (SVM) is a relatively new paradigm for statistical pattern recognition. There are various approaches to this classifier, but we will focus on the basic idea behind the support vector model.

The basic concept relates to linearly separable feature spaces (two sets of points in a two–dimensional space that can be completely separated by a single line). They implement a very

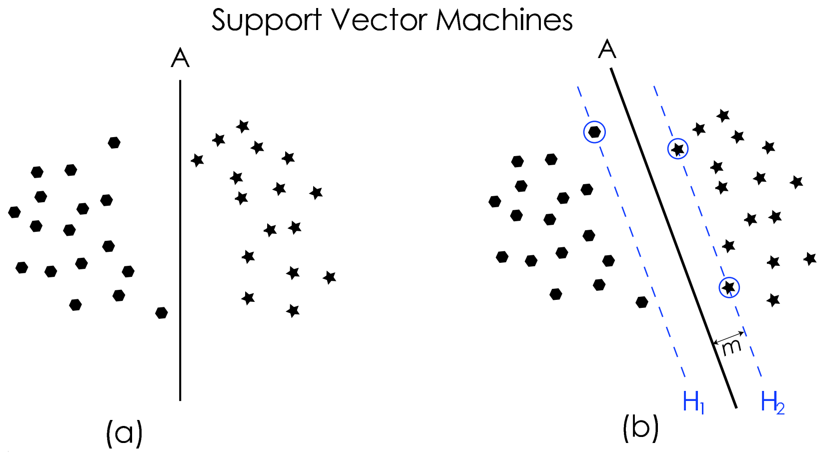


Figure 2.9: Principle of the support vector machine (SVM). (a) shows two linearly separable sets of features with a separating hyperplane A. (b) shows the best separating hyperplane A, that has the maximum possible margin m between the separating hyperplane and the canonical hyperplanes H_1 and H_2. The support vectors are the nearest patterns with a distance m from the separating hyperplane A. The three support vectors are marked by a ring around the data feature points and lie on the canonical hyperplanes H_1 and H_2.

simple idea which is "to find a pair of parallel hyperplanes that leads to the maximum separation between two classes of features so as to provide the greatest protection against errors" [13].

As shown in Figure 2.9(a), there are two linearly separable sets of features, which can be divided or separated by a *separating hyperplane* A. Obviously, there are many possibilities to select a separating hyperplane; in the case of this figure, the hyperplane shows an undesirable option, since the closest feature points are not at the maximum distance and therefore the chosen hyperplane provides low protection against errors.

Therefore, SVM is focused on selecting two parallel hyperplanes that have the maximum possible distance. As shown in 2.9(b) the two parallel hyperplanes, also called *canonical hyperplanes* H_1 and H_2, are characterized by a specific set of feature points that lie on the *canonical hyperplanes* –the so–called *support vectors* (marked by rings in the figure). The separating hyperplane A defines the largest/maximal *margin* which gives the best generalization error of the linear classifier. The term *margin* refers to the perpendicular distance between the separating hyperplane A and each of the two *canonical hyperplanes*. It is expected that the larger the margin, the better the generalization of the classifier [13, 16, 49].

Then, SVM determines the separating hyperplane A for which the margin m is the largest. The support vectors are equally close to the separating hyperplane. Moreover, the selected support vectors are the training samples that define the optimal separating hyperplane, in other words, they are the most informative patterns for the classification task. The canonical hyperplanes (H_1 and H_2

in the figure) are fully defined by *three* support vectors (for 2–D feature spaces). For N-dimensional features spaces, the number of support vectors required is $N+1$ [13].

An important advantage of SVM is the protection against overfitting, since no matter how much data exists in the feature space, the maximum number of vectors used to describe the feature space is $N+1$. Nevertheless the disadvantage of this basic method is that only works when the dataset is linearly separable [13]. This model can be developed further where the separability constraint can be lifted by transforming the data to a feature space of higher dimension where the data will become linearly separable, thus a multiclass classifier can be considered. Moreover this model can be developed in the context of neural networks classifiers, however, this is beyond the scope of this review.

SVM is an extensive topic, we introduced the very basic idea about the support vector model, which is focused to work with linearly separable feature spaces for a binary classification problem, for a more formal and extended review for this model, the reader is refereed to [10, 11, 12, 37, 48, 49].

2.2.5 Clustering

There are problems in pattern recognition where a definition of the classes or even the number of classes is unknown. The problem is not only to classify the given data, but also to define the classes. This method is called unsupervised pattern recognition or clustering (cluster analysis) [13, 14, 16, 24, 40, 46, 49].

The general term "clustering" refers to a number of different methods. Clustering methods aim at discovering the existence of pattern classes in a collection of unlabeled sample patterns. The classical clustering algorithms focus on the general problem of partitioning a given data set into homogeneous groups (clusters) by considering similarities of data points in each group and their relationship to the elements of other groups [14].

There are two main approaches for data clustering: *hierarchical clustering* and *nonhierarchical clustering* or *dynamic clustering* [24].

Hierarchical clustering

Hierarchical clustering methods are among the best known unsupervised methods because of their conceptual simplicity. The procedures can be divided according to two different approaches: *agglomerative* and *divisive* algorithms [13, 14, 16, 49].

The agglomerative algorithm (bottom-up) begins with n clusters, that is, every feature point in the data set is considered as a separate cluster. In the next step, the two most similar points are combined/merged to create a new single cluster. This merging process continues reducing the number of clusters at each step by one. The algorithm stops when all the points are assigned to one cluster. In this algorithm, the natural clusters of feature points in the data set, for a given measure of similarity, are detected by estimating the relative changes in the values of the measures at various stages of the algorithm.

The divisive algorithm (top-down) operates by successively splitting groups, beginning with the whole set of feature points of the data set as a single cluster/group and progressively dividing it into smaller clusters, each one of them with a homogeneous distribution.

NonHierarchical clustering

In dynamic clustering or nonhierarchical clustering, a very popular method is used: the *k–means* algorithm (also known as *c–means* or *basic* ISODATA or iterative relocation). This is one of the various techniques that can be used to simplify the computation and accelerate convergence [14, 16, 24, 49].

The simplest form of the *k–means* algorithm is as follows: the data points are assigned to clusters, where the number of clusters must be specified beforehand. At each iteration of the algorithm, the data points are assigned to clusters on the basis of their similarity with the current cluster representatives (usually they can be assigned to the cluster to whose mean is closest in Euclidian sense). Subsequently the cluster representatives are updated to reflect any changes in the data point assignments. These new cluster models are then used in the next iteration to reclassify the data and the process is continued until a stable partition is obtained or there is no movement from one data points to another cluster. Some of the disadvantages of the *k–means* algorithm are: the sensitivity to the locations of the initial cluster means, and to choose a correct distance measure.

For both hierarchical and nonhierarchical clustering, the clusters are defined as groups of points that are similar according to some “measure”. Usually, similarity is defined as the proximity of the points according to a distance function [46]. Therefore, it is important to know how to measure either the distance among samples or feature points in the data set or the similarity or dissimilarity among them. Many distance metrics exist which are useful for that objective and can be applied for all clustering approaches [14, 16, 49], distance metrics such as Euclidean, city block, Minkowski and Chebyshev distances are defined in Section 2.2.1.

A general survey of the main pattern recognition techniques has been given. All of the different methods present different strengths and many of them could be suitable for the solution of our problem. For this research we selected two classifiers to be implemented: the decision tree classifier and the naïve Bayes classifier.

The decision tree classifier gives a great benefit over many other classifiers, which is interpretability. This classifier leads to a fast classification, employing a sequence of simple queries. Decision trees often produce very simple structures that use only a few features to classify the objects. Furthermore, it has been shown that the decision tree generalize well and can be used to solve a wide range of problems and efficiently classify new samples, providing a natural way to incorporate prior knowledge from human experts, which is very useful when the training set is small [16, 37, 49, 50, 52].

The Bayes decision classifier is a statistical approach to the problem of pattern classification. Its major strength is that not only does it classify the set of features into a number of classes but also it assigns probabilities of being in any of the classes based on the given *prior* probabilities of each class. Specifically the naïve Bayes classifier is a simple model which performs well in practice even when the assumption of conditional independence of individual features is not true.

A detailed explanation about the construction of both classifiers and the features used for training and testing the classifier will be given in the following chapters.

Chapter 3

Materials and Methods

3.1 Image data set

In this section a complete description of the data set will be given. For classification purposes the complete image data set was divided into training and testing sets. The training set was selected for which true classifications are known. A set of five feature parameters were chosen to be powerful discriminators for classification, these feature parameters will be described in the feature selection and extraction section. Ideally the training set should contain as many examples as possible so it includes both common and rare types of feature values. For the testing set the true classifications must also be known in order to determine the classification rate and the accuracy of the classifier.

The ultrasound image data set used for this study was obtained from previous studies by Singh et. al. [42, 43] from the Department of Veterinary Biomedical Science, Western College of Veterinary Medicine, University of Saskatchewan. Ovaries were imaged *in vitro* in parallel planes at 0.5 mm increments using a broad-band (5-9 Mhz) convex-array, ultrasound transducer interfaced with an ATL Ultra Mark 9 HDI ultrasound machine (Advanced Technology Laboratories, Brothell, WA). All images were taken with the same direction and orientation at a resolution 640×480 pixel 8-bit greyscale.

The complete data set was taken from a group of 45 heifers. The animals were ovariectomized (surgical removal of both left and right ovaries) on specific days in the estrous cycle and the ovaries were scanned ultrasonically in water bath. The days chosen for ovariectomy were on day 3 of wave 1 (D3W1), day 1 of wave 2 (D1W2) and after onset of proestrus (day ≥ 17) corresponding to the metestrus, diestrus and proestrus phases respectively. day 0 was defined as the day of the previous ovulation. The days of ovariectomy were selected on the basis of previous studies [3, 21, 43] to represent growing - increasing diameter (D3W1), static - no change in diameter (D1W2) and regressing - decreasing diameter (D$\geq$ 17) phases of the dominant follicle of wave 1 in the estrous cycle (see Figure 2.3).

The 45 pairs of ovaries were divided into two data sets: training data set (denoted as data set A) and testing data set (denoted as data set B). The training data set A consisted of 23 pairs of ovaries that were collected and imaged during metestrus (n = 8, D3W1), diestrus (n = 7, D1W2), and proestrus ($n = 8$, D≥17). Of these animals, 19 of them exhibited a 2-wave pattern while 4 exhibited a 3-wave pattern. The testing data set B consisted of a different group of 22 pairs of ovaries collected and imaged during metestrus ($n = 8$, D3W1), diestrus ($n = 6$, D1W2) and proestrus ($n = 8$, $D \geq 17$). From this group 20 heifers exhibited a 2-wave pattern and 2 heifers a 3-wave pattern.

The ultrasound images used in the present study were acquired *in vitro* to minimize confounding due to intervening tissues, changes in position of the ovary in relation to the transducer and to allow direct digitization of the images from the ultrasound equipment [42, 43].

The complete data set was accompanied by a full set of schematic diagrams containing information about the main structures inside both left and right ovaries based on ultrasound examinations. The diagrams contained information about size and location of the dominant, first subordinate and other subordinate follicles and corpus luteum over both left and right ovaries. The ultrasound examinations commenced at least 2 days before ovulation preceding the estrous cycle under study and continued on a daily basis until the day of ovariectomy to monitor the development of the follicles and corpora lutea. The topographic location and diameter of individual identified follicles and corpora lutea were recorded each day.

An important advantage of the image data set used in this study is that the collection of ultrasound images were accompanied by schematic diagrams with the main structures inside the ovaries identified as either dominant follicle, first subordinate follicle, second subordinate follicle and corpus luteum. The diagrams were annotated by a human expert identifying the main structures based on a daily examination from the day of the last ovulation to the date of ovariectomy and ultrasonically scanned *in vitro*. This information was obtained also from previous studies by Singh et. al. [42, 43].

3.2 Feature selection and extraction

For the feature selection the main objective is to extract the main characteristics of the objects of interest useful for the classification. As we discussed in the reproductive biology shown in Section 2.1, characteristics of the main ovarian structures play an important role when defining the status of the ovary. Figure 2.5 in Chapter 2 showed that the CL has a period of initial growth during the metestrus, followed by a period of maximal size during diestrus and ending with a period of regression during proestrus. Such behavior suggests that the size of the CL is an important feature that reflects the phase in the estrous cycle. Figure 2.3 illustrated the wave-like follicular growth pattern in the estrous cycle, showing the changes in size of the dominant follicle (largest follicle) and subordinate follicles (smaller follicles) along the estrous cycle. Such a behavior also reveals important information about the stage in the estrous cycle.

Therefore, the features chosen to describe the current estrous cycle stage of the animal were :

1. size of the largest follicle (dominant follicle);
2. size of the second largest follicle (first subordinate follicle);
3. size of the third largest follicle (second subordinate follicle);
4. size of the corpus luteum (CL);
5. number of follicles ≥ 2 mm in size.

These features were extracted over both left and right ovaries for a given animal. The size of the dominant, first and second subordinate follicles as well as the CL are defined as the mean of the lengths of their major and minor axis. The number of follicles were counted from left and right ovaries with size ≥ 2 mm.

Dominant follicle (D) features from data set A were extracted by manually measuring the diameter of the largest follicle in the ultrasound images using a Graphical User Interface (GUI) created for this project. The slice (ultrasound image) containing the largest follicle area was located and the largest horizontal and vertical diameters of the follicle within that slice were measured; the size of D was the average of the largest horizontal and vertical diameters. The remaining features, diameter of the first subordinate follicle (S1), diameter of the second subordinate follicle (S2), diameter of the CL (CL) and the number of follicles (NF) were obtained from the schematic diagrams. For the testing data set B, all features were obtained from the schematic diagrams which were recorded by experts.

The complete feature values for training data set A and testing data set B are in tables A.1 and A.3 in Appendix A. Figure 3.1 shows a graph with mean values and standard deviation values for D, S1, S2, CL and NF obtained from the training data set A. The graph is divided in three sections representing the three different stages in the estrous cycle: D3W1 corresponding to *metestrus*, D1W2 corresponding to *diestrus* and finally D$\geq$ 17 corresponding to the *proestrus* stage. All features are measured in millimeters except for the NF which is dimensionless.

Figure 3.2 shows a similar graph with mean and standard deviation values for D, S1, S2, CL and NF features obtained from the testing data set B.

As mentioned before, values for D were extracted by using a graphical user interface (GUI) created in MATLAB (stands for Matrix Laboratory) which is "a high-performance language for technical computing" that is good for many forms of numeric computation and visualization [18].

The GUI was used to calculate the diameter of the dominant follicle from the set of image slices that comprise an ovary; the slice with the largest follicle area was selected and the diameter calculated. Figure 3.3 shows the GUI we used for the manual extraction. The image shows an ultrasound image containing the largest (dominant) follicle from a set of images. Follicles appear as

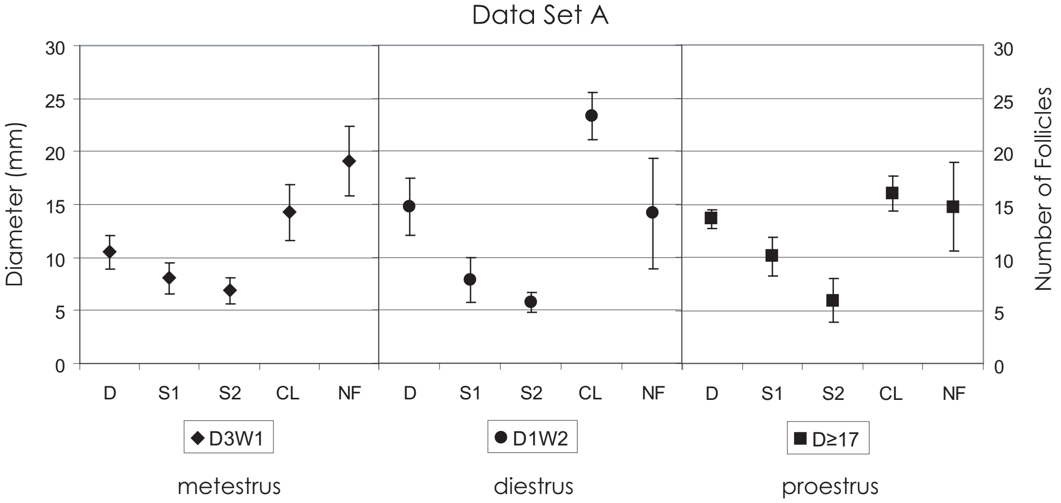


Figure 3.1: This graph shows the mean and standard deviation feature values from the training data set A after the feature extraction. The features (D, S1, S2 and CL) represent size/diameter in millimeters, NF feature is dimensionless, represents the number of follicles.

black circular/elliptical areas in ultrasound images. The right side of the ultrasound image shows the scale on which the ultrasound images were taken, the numbers express dimension in centimeters. That is the distance from, let's say 1.5 to 2.5 in the image is equivalent to 1 centimeter.

To obtain the size of the follicle in millimeters, we have to calculate first the scale of the ultrasound image, which can be done by a click on the "Get Scale" button below the image. The lower section of the figure shows the scale of this particular ultrasound image, for this example 165 pixels are equivalent to 10 mm. This figure also shows two panels that can be used to measure either follicles or corpus luteum areas, there is actually no difference in the way these two diameters are measured. Each panel has two buttons where the horizontal and vertical values can be obtained. After a click on the "Get Horizontal Diameter" a line has to be drawn on the widest horizontal area of the follicle which is determined by eye as shown in the figure. The objective of this application is to be used as a tool to measure the size of the follicles or corpora lutea by an expert.

Figure 3.4 shows an example of the vertical diameter measured. After the "Get Vertical Diameter" button is clicked, the user is able to draw a line on the widest vertical area of the follicle. Thus the GUI displays the mean diameter of the follicle based on the horizontal and vertical diameters. This figure shows an example where the dominant follicle has a mean diameter of 15.87 mm, which was rounded and considered as 16 mm for the feature extraction.

There were a number of unavoidable potential sources of error when extracting the features used in this study from a set of images of an animal's ovaries. The sizes of the follicles were considered regardless of whether they were in their growing or regressing phases since one cannot differentiate

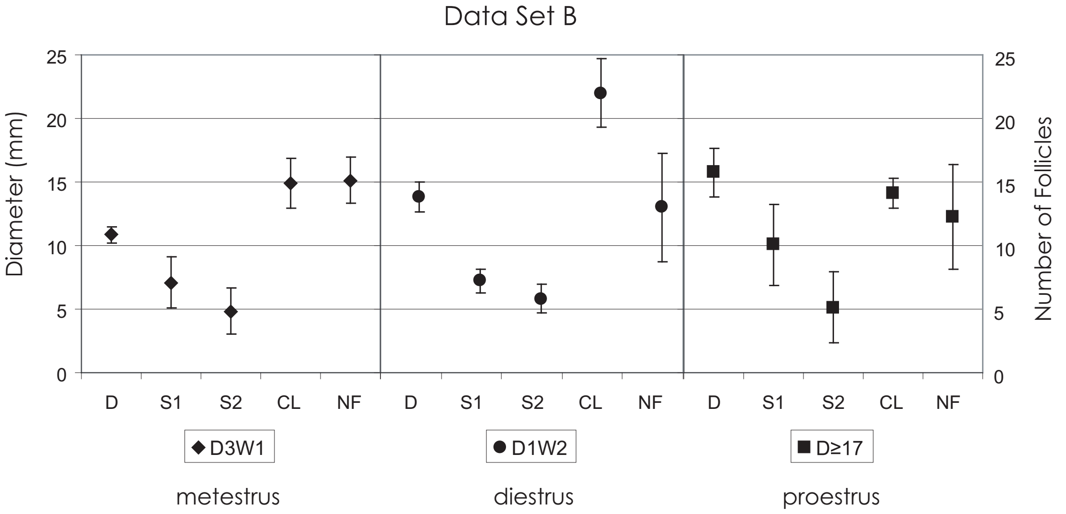


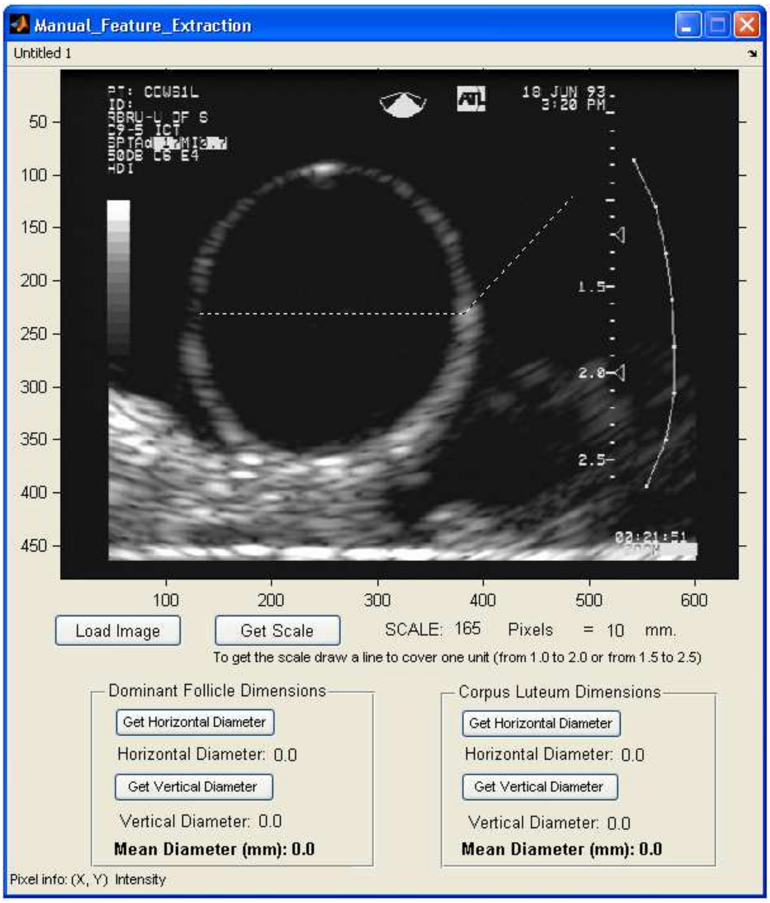
Figure 3.2: This graph shows the mean and standard deviation feature values from the testing data set B after the feature extraction. The features (D, S1, S2 and CL) represent size/diameter in millimeters, NF feature is dimensionless, represents the number of follicles.

a growing dominant follicle from a regressing dominant follicle based on a single day's examination. From Figures 3.1 and 3.2, it can be seen that the mean size of the first subordinate follicle (S1) in class D1W2 has a large value compared to the expected follicle value illustrated in Figure 2.3 from Chapter 2, which suggests that some of the S1 measurements for class D1W2 may in fact have been measurements of the future dominant follicle of wave 2. Similarly some of the S2 values recorded for class D1W2 could have resulted from the future first subordinate follicle of wave 2. For the D$\geq$17 class, the size of the mean for the S1 feature suggests that some values may have resulted from the regressing dominant follicle of wave 1. Similarly, some of the S2 measurements could be the size of the first subordinate follicle of wave 2.

Chapter 4 will show that both naïve Bayes and decision tree classifiers performed surprisingly well, despite the potential sources of error arising from collecting features from only a single pair of images. Moreover, it is necessary to design a classifier that is robust to these errors since they will be unavoidable in practice; a single snapshot of an ovary in time may contain both regressing follicles from one wave and growing follicles from the subsequent wave.

3.3 Classifier design

Two classifiers were implemented, a naïve Bayes classifier and a decision tree classifier. To gain confidence in the correctness of the implementation, the results obtained from both classifiers were compared with an existing machine learning and data mining application called Weka [50]. The



SCALE: 165 Pixels = 10 mm.

To get the scale draw a line to cover one unit (from 1.0 to 2.0 or from 1.5 to 2.5)

Dominant Follicle Dimensions

Horizontal Diameter: 0.0

Vertical Diameter: 0.0

Mean Diameter (mm): 0.0

Corpus Luteum Dimensions

Horizontal Diameter: 0.0

Vertical Diameter: 0.0

Mean Diameter (mm): 0.0

Pixel info: (X, Y) Intensity

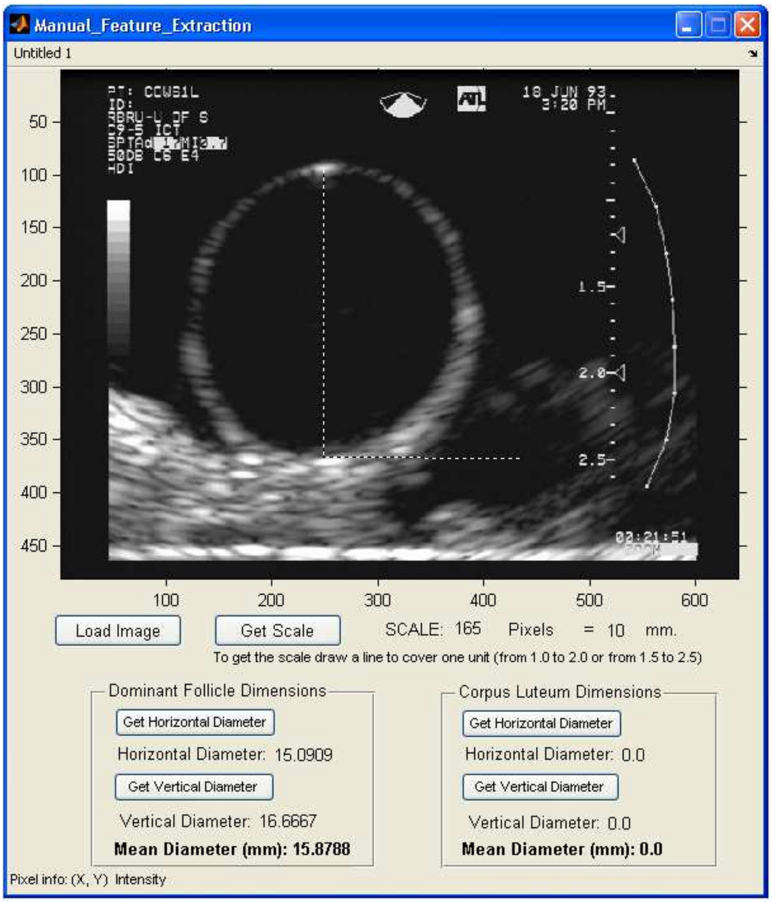
Figure 3.3: Matlab GUI application for the manual feature extraction. The figure displays the main components in the GUI to calculate the diameter of the dominant follicle feature and a line measuring the largest horizontal axe is shown.

results obtained from the classifiers made for this project were in close agreement with the results obtained by the Weka application.

3.3.1 Naïve Bayes classifier

For this study a Bayes decision classifier was built to make the classification of the stages in the estrous cycle. In this section we are going to describe in more detail the methodology we followed to construct this classifier.

This classifier is based on Bayes rule: $p(w_i|\vec{x}) = \frac{p(\vec{x}|w_i)p(w_i)}{p(\vec{x})}$ which was first defined in equation 2.6 in Chapter 2; where $p(w_i|\vec{x})$ is the *a posteriori* probability of class w_i given the pattern vector $\vec{x}$, $p(\vec{x}|w_i)$ is the class–conditional probability density function of a pattern vector $\vec{x}$ given the class w_i, $p(w_i)$ is the *a priori* probability of class w_i and $p(\vec{x})$ is the *unconditional probability distribution*,



SCALE: 165 Pixels = 10 mm.

To get the scale draw a line to cover one unit (from 1.0 to 2.0 or from 1.5 to 2.5)

Dominant Follicle Dimensions

Horizontal Diameter: 15.0909

Vertical Diameter: 16.6667

Mean Diameter (mm): 15.8788

Corpus Luteum Dimensions

Horizontal Diameter: 0.0

Vertical Diameter: 0.0

Mean Diameter (mm): 0.0

Figure 3.4: Matlab GUI application for the manual feature extraction. The figure shows a line measuring the largest vertical axe of the follicle in the ultrasound image. In the lower panel of the GUI, the horizontal and vertical diameters are shown as well as the mean diameter of the follicle, expressed in millimeters.

also called the evidence.

According to Duda et. al. [16] Bayes Rule can also be expressed informally as:

$$posterior = \frac{likelihood \times prior}{evidence}, \tag{3.1}$$

the term *posterior* is the *a posteriori* probability $p(w_i|\vec{x})$, the *likelihood* is the term $p(\vec{x}|w_i)$ which represents the likelihood of a class w_i with respect to the pattern $\vec{x}$, in other words, the class w_i for which $p(\vec{x}|w_i)$ is large is more "likely" to be the true category [16]. The *prior* term is the *a priori* probability of class w_i and the *evidence* is related to the probability $p(\vec{x})$ which can be seen as a scale factor which makes the posterior probabilities to sum to one.

To construct the Bayes classifier, the features were considered to be conditionally independent, also known as the naïve Bayes rule. This is a simplifying assumption which is usually used when the dependency relationships among the features are unknown.

The Bayes classification is defined mainly by the likelihood or class-conditional probability density functions $p(\vec{x}|w_i)$ as well as the prior probabilities $p(w_i)$. The Gaussian density function is the most relevant/used function for this classifier particularly when the feature vectors $\vec{x}$ for a given class w_i are continuous-valued, i.e. the features have numeric values. Then, the numeric values are handled by assuming that they have a "normal" or "Gaussian" probability distribution. The conditional independence assumption allows us to represent the likelihood in terms of:

$$p(\vec{x}|w_i) = p(x_1|w_i) \cdot p(x_2|w_i) \cdot p(x_3|w_i) \cdot \ldots \cdot p(x_n|w_i), \tag{3.2}$$

where $p(x_j|w_i)$ is a 1-D (univariate) normal distribution.

The probability density function for a 1-D normal or Gaussian distribution with mean μ and standard deviation σ is given by the expression:

$$p(\vec{x_j}|w_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(\vec{x}-\mu_i)^2}{2\sigma^2}\right]. \tag{3.3}$$

To construct the Bayes classifier (training stage) we had to first calculate the mean μ and the standard deviation σ for each class and each feature from the training vectors, training data set A. Table 3.1 gives a summary of the feature values for the training vectors that belong to class D3W1 in the training data set A, at the end of the table it shows the mean μ and the standard deviation σ values, for each one of the features. The mean value is calculated as the average of the preceding values expressed as:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{3.4}$$

where μ_j is the mean of feature x_i within class j and n is the number of training samples of class j. The standard deviation is the square root of the sample variance, which can be calculated as follows:

$$\sigma_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_j)^2}, \tag{3.5}$$

where σ_j is the standard deviation of feature x_i within class j and n is the number of training samples of class j.

Table 3.2 and 3.3 give the summary statistics of the feature values for class D1W2 and D≥17 respectively from the training data set A with mean and standard deviation values. In the testing stage of the Bayes classifier the class-conditional density function (likelihood) and prior probabilities have to be calculated in order to get the posterior probability of the class.

Let's consider a test pattern $\vec{x} = (10, 6, 3, 16, 15)$ where the feature values correspond to $D = 10$, $S1 = 6$, $S2 = 3$, $CL = 16$ and $NF = 15$. In order to get a classification for this pattern we first have to calculate the density function or likelihood of such a pattern.

Table 3.1: Training data set A for class D3W1 with summary statistics. This table lists the feature values for class D3W1 in the training data set A, it displays the mean and standard deviation for each feature. The columns represent the different feature values in millimeters for the dominant follicle (D), first subordinate follicle (S1), second subordinate follicle (S2) and corpus luteum (CL). The last column represents a number for the number of subordinate follicles (NF).

Summary Statistics for Class D3W1					
	D (mm.)	*S1* (mm.)	*S2* (mm.)	*CL* (mm.)	*NF* (number)
	11.5	9.0	7.0	17.0	23
	12.0	8.0	6.0	18.0	23
	12.0	8.0	7.0	16.0	18
	10.0	7.0	6.0	11.0	18
	11.0	9.0	7.0	11.0	18
	10.0	9.5	8.0	15.0	19
	7.0	5.0	5.0	13.0	21
	10.5	9.0	9.0	13.0	13
Mean	10.5	8.0	6.8	14.2	19
Std. Dev.	1.6	1.4	1.2	2.6	3

To calculate the density function of the test pattern to be in classes D3W1, D1W2 or D≥17 we used the probability density function for a 1-D normal distribution expressed in Equation 3.3 and the values that were obtained in the training stage (mean and standard deviation values). If we first consider D3W1 as the outcome class, we need to substitute the first testing feature value ($D = 10$) into equation 3.3 as well as $\mu = 10.5$ and $\sigma = 1.62$ from table 3.1. So the value of the probability density function is:

$$p(D = 10|D3W1) = \frac{1}{\sqrt{2\pi} \times 1.62} \exp\left[-\frac{(10 - 10.50)^2}{2 \times (1.62)^2}\right] = 0.234063. \tag{3.6}$$

In the same way, the probability density functions of a D3W1 outcome for S1, S2, CL and NF are calculated in the same way:

$$\begin{aligned} p(S1 = 6|D3W1) &= 0.101717, \\ p(S2 = 3|D3W1) &= 0.00254957, \\ p(CL = 16|D3W1) &= 0.120813, \\ p(NF = 15|D3W1) &= 0.0550628. \end{aligned}$$

Table 3.2: Training data set A for class D1W2 with summary statistics. This table lists the feature values for class D1W2 in the training data set A, it displays the mean and standard deviation for each feature. The columns represent the different feature values in millimeters for the dominant follicle (D), first subordinate follicle (S1), second subordinate follicle (S2) and corpus luteum (CL). The last column represents a number for the number of subordinate follicles (NF).

Summary Statistics for Class D1W2					
	D (mm.)	*S1* (mm.)	*S2* (mm.)	*CL* (mm.)	*NF* (number)
	15.0	8.5	6.0	22.0	15
	20.0	6.0	5.0	24.0	12
	14.0	7.0	6.0	28.0	19
	12.0	8.0	6.0	22.0	12
	16.0	9.0	6.0	23.0	6
	13.0	11.5	7.0	22.0	22
	13.0	5.0	4.0	22.0	13
Mean	14.7	7.8	5.7	23.2	14
Std. Dev.	2.6	2.1	0.9	2.2	5

Now, let's remember that to get a classification based on the Bayes rule we first have to obtain the class-conditional probability density functions (likelihood) and the prior probabilities (prior) as is expressed in Equation 3.1.

Therefore, the probability density functions for the class D3W1 given the testing pattern x is then multiplied by the prior probability of such a class $p(w_i)$. The prior probabilities reflect the prior knowledge or the degree of belief of a resultant class if there is no additional information. For this classifier we consider the prior probabilities to be uniform since we have the same probability of having any of the 3 different classes given the data set. Thus, given a 3-class classification the prior probability assigned to each one of the classes is: $p(w_i) = 0.33$. In reality, the diestrus phase is typically longer than the others - a fact that may warrant the use of non-uniform prior probabilities if this classifier were used to classify animals randomly chosen from a herd.

Using the probability density function, values of each one of the features that form the testing pattern $\vec{x}$ for class $w_i =$ D3W1, and the prior probability of w_i we have:

$$p(\vec{x}|w_i)p(w_i) = 0.2340 \times 0.1017 \times 0.0025 \times 0.1208 \times 0.0550 \times 0.33 = 1.33245 \exp^{-07} . \tag{3.7}$$

A similar calculation for $w_i = D1W2$ would lead to:

Table 3.3: Training data set A for class D≥17 with summary statistics. This table lists the feature values for class D≥17 in the training data set A, it displays the mean and standard deviation for each feature. The columns represent the different feature values in millimeters for the dominant follicle (D), first subordinate follicle (S1), second subordinate follicle (S2) and corpus luteum (CL). The last column represents a number for the number of subordinate follicles (NF).

Summary Statistics for Class D≥17

	D (mm.)	*S1* (mm.)	*S2* (mm.)	*CL* (mm.)	*NF* (number)
	13.5	10.0	8.0	14.0	18
	13.0	10.5	6.0	17.0	12
	13.0	11.0	5.0	18.0	17
	15.0	7.0	4.0	16.0	14
	13.0	8.0	4.0	15.0	15
	13.5	13.0	7.0	14.0	19
	13.0	11.0	4.0	18.0	17
	15.0	10.0	9.5	16.0	6
Mean	13.6	10.0	5.9	16.0	14
Std. Dev.	0.8	1.8	2.0	1.6	4

$$p(\vec{x}|w_i)p(w_i) = 5.86457 \exp^{-10}, \tag{3.8}$$

and for $w_i = D \geq 17$

$$p(\vec{x}|w_i)p(w_i) = 9.5099 \exp^{-10}. \tag{3.9}$$

Substitution of these values into the Bayes Formula in equation 2.6 yields the following *posteriori* values:

$$\begin{aligned} p(D3W1|\vec{x}) &= 0.9885, \\ p(D1W2|\vec{x}) &= 0.0043, \\ p(D \geq 17|\vec{x}) &= 0.0070. \end{aligned}$$

This indicates that for the unknown testing pattern $\vec{x} = (10, 6, 3, 16, 15)$, the class D3W1 is far more likely to be the true class than classes D1W2 and D≥17. These numbers can turn into percentages so they sum to 100%, resulting in the following:

$$\text{probability of D3W1} = 98.85\%,$$

$$\begin{aligned} \text{probability of D1W2} &= 0.43\%, \\ \text{probability of D}{\geq}17 &= 0.70\%. \end{aligned}$$

The testing pattern $\vec{x} = (10, 6, 3, 16, 15)$ used to evaluate the naïve Bayes classifier was correctly classified as being in class D3W1 or metestrus stage. To see the performance of this classifier we will discuss different experiments and results in Chapter 4.

This simple and intuitive method is the naïve Bayes classifier which classifies a set of feature vectors into a number of classes, it assigns *posterior* probabilities to being in one or another class based on the *a priori* probabilities and the *likelihood* of each one of the classes.

3.3.2 Decision tree classifier

Various decision tree inference algorithms have been developed to solve pattern recognition problems. The most popular are the ID3 and C4.5 algorithms developed by Quinlan [34], and the CART (Classification And Regression Trees) developed by Breiman et al. [9].

Similar to the Bayes classifier, the decision tree classifier takes a feature vector as an input and the classifier returns a decision which is the predicted class for the input pattern. The decision tree reaches the decision by performing a sequence of tests related to the values of the elements of the unknown feature vector. As seen in the decision tree review in Section 2.2.2 in Chapter 2, a decision tree is formed by different nodes, each internal node in the tree corresponds to a decision based on one feature and the branches of such a node are labeled with values of the feature. The leaf nodes in the tree labeled with the class to be returned if that leaf is reached. Although decision trees may be easy to understand and follow, the construction of such trees require the use of heuristics to create a simple yet powerful tree.

Creating a decision tree requires choosing which feature is going to be tested at each node of the tree. One of the great advantages of decision trees is that they estimate the suitability of the different features for separating the patterns representing different classes. The criterion to decide which is the best feature to be placed on a node is based on a metric called information gain (expected amount of information provided by the feature). The information gain gives a numerical value to each one of the features with respect to the training data set. The information gain metric will be explained in detail in this section. The information gain will be used to decide which feature is best for each node, therefore, the best feature is obtained by creating a feature ranking on the basis of the maximum information gain values, calculated for each one of the features for the whole training data set.

The basic idea behind the construction of a decision tree can be seen as a series of steps. The following 3 steps show this process for the creation of a decision tree with discrete or categorical features:

1. select the best feature (feature that scores the highest information gain) to classify a given training data set and assign it to the root node;
2. for each possible value of the chosen feature, a branch will be created from that node. This means the selected feature value will be used to split/partition the data set at each node;
3. determine whether the nodes derived from the branches are terminal or internal nodes. Here for each node a decision is made to continue splitting or to make the node terminal;
    - if the training data set that reached that node contains only patterns from one class (pure classification), then make it a terminal node and assign it the class of the patterns;
    - if the training data set that reached that node is empty, then assign a default class;
    - otherwise, place a new node in the decision tree with the feature that has the highest information gain value related to the current data set (data set that satisfied the previous split value condition of step 2). This new node starts the cycle again (from step 2), this means the algorithm will continue splitting until a terminal node has a pure classification or the data set is empty. In other words, build a decision tree recursively.

The construction of the tree terminates either when all features have been exhausted, or the decision tree perfectly classifies the training data set. A feature is not used on the same path of the decision tree again, that is if a feature is chosen as the best feature it will not be used again by any of its children. The diagram showed in Figure 3.5 illustrates the decision tree construction graphically.

The most important point in the decision tree construction is to choose the feature that will provide the closest to an exact classification. That is to test the most important feature first which will make the most difference to the classification of an example. A more formal algorithm for the decision tree learning from [37] is given in Table 3.4. The algorithm in Table 3.4 is focused on categorical features, that is the split in each internal node would have each one of the possible values of the feature in such internal node. Since the feature values for our classifier are numerical we will have a slightly different approach that can deal with these conditions, because we cannot deal with an infinite number of splits. Thus in each internal node the feature will have a splitting value which will be of the form "feature $\leq x_n$" where x_n is called the split point at node n and can be translated as: "Is the value of the feature lower or equal than the split value?" The use of this split value will allow us to have a binary tree, that means that derived from each node it will only have two possible branches. The first one is the left branch "feature $\leq x_n$" and the second is the right branch "feature $> x_n$".

As seen in the decision tree learning algorithm in Table 3.4, the construction of a decision tree can be expressed recursively (Decision_Tree_Learning function is called inside the Decision_Tree_Learning

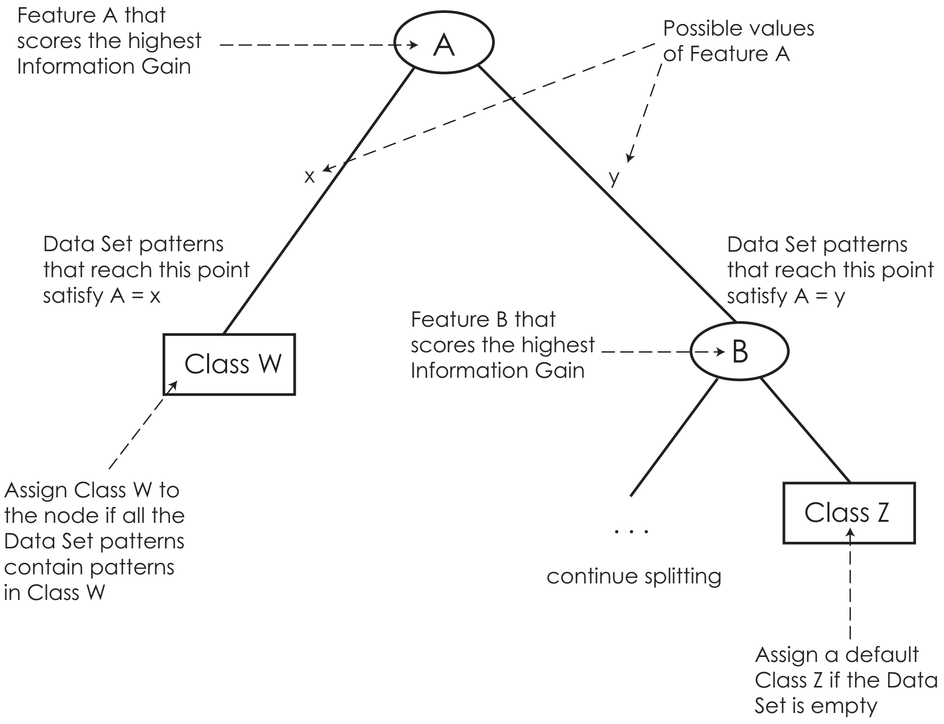

Feature A that scores the highest Information Gain
A
Possible values of Feature A
x
y
Data Set patterns that reach this point satisfy A = x
Data Set patterns that reach this point satisfy A = y
Feature B that scores the highest Information Gain
B
Class W
Assign Class W to the node if all the Data Set patterns contain patterns in Class W
. . .
continue splitting
Class Z
Assign a default Class Z if the Data Set is empty

Figure 3.5: Diagram showing the process in a decision tree creation. The feature with the highest information gain (A) is placed at the root node, this node will have all possible values of such a feature (x and y). The data set patterns that reach the next nodes will be partition so that satisfy $A = x$ (for the left branch) or $A = y$ (for the right branch). This diagram also illustrates the different scenarios while creating the decision tree: the decision tree creates a terminal node when all the patterns belong to one class (class W), it assigns a default class when the data set is empty (class Z) and it will continue splitting while there is not a pure classification and the data set is not empty.

Table 3.4: Decision tree learning algorithm

```
function Decision_Tree_ Learning(DataSet, Features, Default) returns a decision tree
.   inputs:
.           DataSet: set of examples
.           Features: set of features
.           Default: default class
.   if DataSet is empty then return Default
.   else if all DataSet have the same classification then return the classification
.   else if Features is empty then return Majority_Value(DataSet)
.   else
.       best ⟵ Choose_Best_Feature(Features, DataSet)
.       tree ⟵ a new decision tree with root test best
.       m ⟵ Majority_Value(DataSet)
.       for each value $v_i$ of best do
.           DataSet $_i$ ⟵ {elements of DataSet with best = $v_i$}
.           subtree ⟵ Decision_Tree_Learning(DataSet $_i$, Features - best, m)
.           add a branch to tree with label $v_i$ and subtree subtree
.       return tree
```

main function). First, a selection of the best feature is made at the root node, this will split the data set into subsets: one for the data set that satisfies the "$\leq$" (lower equal than the split value) and a second one for the data set that satisfies the "$>$" (greater than the split value). Then the process is repeated recursively for each branch, using only those features that actually reached the branch. When all features have the same classification, that part of the tree stops growing and the class is assigned to that branch (becomes a leaf node).

Hence, the most important part in this construction is to determine which feature will be used to split or to be placed in a node (Choose_Best_Feature function in the algorithm). For that, we need a formal measure of "good" or "bad" feature, so the selected feature would produce the purest possible child nodes. One suitable measure is the expected amount of *information* provided by the feature. The *information* measure is based on the Shannon entropy [39] and represents the amount of information associated with a node of the tree that would be needed to specify whether a new instance should be classified as one of the available classes. Information theory measures information content in *bits* and is defined as:

$$Info(\vec{T}) = -\sum_{i=1}^{n} \frac{freq(w_i, \vec{T})}{|\vec{T}|} \log_2 \frac{freq(w_i, \vec{T})}{|\vec{T}|}, \tag{3.10}$$

where $\vec{T}$ is defined as the training data set, w_i is a class, and $freq(w_i, \vec{T})$ is the frequency or number of times that the class w_i appears in the training data set $\vec{T}$.

Equation 3.10 is a general definition of the information measure. To understand better this concept let's consider an example stated in [37] where it considers a 2-class classification, being the first class named *true* and the second one *false.* Then, consider a training set that contains p positive samples and n negative samples. Thus the estimate of the information contained in a correct answer is:

$$Info\left(\left(\frac{p}{p+n}, \frac{n}{p+n}\right)\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}. \tag{3.11}$$

One bit of information would be enough to know a true-false interrogation like the flip of a fair coin, thus substituting this equation into such a case we get:

$$Info\left(\frac{1}{2}, \frac{1}{2}\right) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1bit. \tag{3.12}$$

Some of the characteristics of the information measure are [50]:

- when the number of either *true's* or *false's* is zero, the information is zero;
- when the number of *true's* and *false's* is equal, the information reaches a maximum value.

Still, with this measure of information we must know how much information we will need after the feature we are currently testing (remainder information). A feature A will divide the data set $\vec{T}$ into subsets $\vec{T_1}, \vec{T_2} \ldots, \vec{T_v}$, depending on the v different values of A. For our case the feature A will divide the data set into 2 subsets $\vec{T_1}$ and $\vec{T_2}$. Each subset $\vec{T_i}$ will have p_i positive samples and n_i negative samples. Hence, the *remainder* information gives the amount of information that we expect would be necessary to assign the class of a new sample, after testing on attribute A. The *remainder* information is calculated as follows:

$$Remainder(A) = \sum_{i=1}^{v}\frac{p_i+n_i}{p+n}Info\left(\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)\right). \tag{3.13}$$

In order to select the best feature to be placed in a node and make a split from the data set, we will require an additional measure; the so called information gain, which is related to the feature. "The information gain from the feature is the difference between the original information and the new requirement (remainder)" [37] or information required given the feature A_k:

$$Gain(\vec{T}, A_k) = Info(\vec{T}) - Remainder_{A_k}(\vec{T}). \tag{3.14}$$

A general formulation for the remainder equation can be expressed as follows [5]:

$$Remainder_{A_k}(\vec{T}) = \sum_{a_k \epsilon D(A_k)}\frac{|\vec{T}_{a_k}^{A_k}|}{|\vec{T}|}Info\left(\vec{T}_{a_k}^{A_k}\right), \tag{3.15}$$

where A_k is the feature which is being tested, $\vec{T}$ is the data set and $\vec{T}_{a_k}^{A_k}$ is the subset for which the feature A_k has the value a_k which belongs to the domain A_k ($D(A_k)$).

Finally, the information gain for the feature with the highest value is the feature which will be placed in the node and taken to split on. To have a better understanding of these concepts, we will illustrate them with some examples. Let's consider the values from the training data set A to create a decision tree and select the best feature to be the root. As mentioned before, for our classification we will have to modify the decision tree algorithm to be able to deal with numerical values.

To get the information gain of a feature it is necessary to first calculate the amount of information, which was defined in equation 3.10. For that we need $|\vec{T}|$ which is the total magnitude or number of elements of the complete training data set $\vec{T}$ ($n = 23$) and $freq(w_i, \vec{T})$ which is the number of elements that correspond to each class in the training data set $\vec{T}$. From the training data set A values displayed in Appendix A.1, we can state that for class D3W1 we have 8 feature vectors, for class D1W2 we have 7 feature vectors and for class D≥17 we have 8 feature vectors. Thus, from the information equation 3.10 we have:

$$Info(\vec{T}) = -\frac{8}{23}\log_2\frac{8}{23} - \frac{7}{23}\log_2\frac{7}{23} - \frac{8}{23}\log_2\frac{8}{23} = 1.58219. \tag{3.16}$$

The next step would be to calculate the remainder value. We can use equation 3.13 and adjust it to our 3-class classification. To do that, first we have to sort the feature values incrementally by the feature, for this example we will consider sorting the feature values of feature D for the training data set A. Once the feature values are sorted, all the repeated values will be collapsed together, see Table 3.5 to illustrate this example. From this table we can see that there are only 12 possible values for the feature D, the table lists such feature values with their related classes, starting with the lowest D value (7) in the training data set A and ending with the highest value (20). At the end of the table, a set of coordinates summarizes the number of samples of each class where the feature D had the indicated value (class). The first coordinate corresponds to D3W1, the second to D1W2 and the third to D≥17 classes. So, for example, for D value equal to 12 there were two instances that had the class D3W1 related, one instance was related to D1W2 class and zero instances were related to class D≥17. In order to calculate the *remainder* value we first have to calculate the $Info\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$ elements modified to our 3-class classification.

Substituting the first feature value D=7 into the information equation we have:

$$Info\left(\left(\frac{1}{1}, \frac{0}{1}, \frac{0}{1}\right)\right) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} - \frac{0}{1}\log_2\frac{0}{1}. \tag{3.17}$$

In practice we can calculate the information without having the individual fractions by applying a log equivalence, from equation 3.17 we can simplify it as:

$$Info\left((1,0,0)\right) = \frac{-1\log_2 1 - 0\log_2 0 - 0\log_2 0 + 1\log 1}{1} = 0, \tag{3.18}$$

we define:

$$0\log_2 0 = 0. \tag{3.19}$$

Table 3.5: Table summarizing the diameter values for the D feature from training data set A. Same values of the feature are collapsed together and their classes listed. The indices at the end of the table summarize the number of times the feature value had a class related, the first coordinate is related to class D3W1, second coordinate to class D1W2 and third to D≥17.

Dominant Feature Values from Training Data Set A											
7.0 (mm)	10.0 (mm)	10.5 (mm)	11.0 (mm)	11.5 (mm)	12.0 (mm)	13.0 (mm)	13.5 (mm)	14.0 (mm)	15.0 (mm)	16.0 (mm)	20.0 (mm)
D3W1	D3W1	D3W1	D3W1	D3W1	D3W1	D1W2	D≥17	D1W2	D1W2	D1W2	D1W2
	D3W1				D3W1	D1W2	D≥17		D≥17		
					D1W2	D≥17			D≥17		
						D≥17					
						D≥17					
						D≥17					
(1,0,0)	(2,0,0)	(1,0,0)	(1,0,0)	(1,0,0)	(2,1,0)	(0,2,4)	(0,0,2)	(0,1,0)	(0,1,2)	(0,1,0)	(0,1,0)

Similarly for the second element with D=10 we have:

$$Info((2,0,0)) = \frac{-2\log_2 2 - 0\log_2 0 - 0\log_2 0 + 2\log 2}{2} = 0. \tag{3.20}$$

We continue doing this for all the values of attribute D. All the results are zero except for the information related to the attribute values D=12, D=13 and D=15 which do not have a pure classification. Then, the information related to D=12, D=13 and D=15 are:

$$\begin{aligned} Info((2,1,0)) &= \frac{-2\log_2 2 - 1\log_2 1 - 0\log_2 0 + 3\log 3}{3} = 0.9182, \\ Info((0,2,4)) &= \frac{-0\log_2 0 - 2\log_2 2 - 4\log_2 4 + 6\log 6}{6} = 0.9182, \\ Info((0,1,2)) &= \frac{-0\log_2 0 - 1\log_2 1 - 2\log_2 2 + 3\log 3}{3} = 0.9182. \end{aligned}$$

Therefore, with these non-zero values, the remainder can be calculated from equation 3.15, with $|\vec{T}^{A_k}_{a_k}|$ as the magnitude of the subset and $|\vec{T}|$=23, we have:

$$Remainder_D(\vec{T}) = \frac{3}{23}Info((2,1,0)) + \frac{6}{23}Info((0,2,4)) + \frac{3}{23}Info((0,1,2)) = 0.47911. \tag{3.21}$$

The *information gain* for feature D (A_k = D) in data set $\vec{T}$ is now calculated from equation 3.14 as:

$$\begin{aligned} Gain(\vec{T}, D) &= Info(\vec{T}) - Remainder_D(\vec{T}), \\ Gain(\vec{T}, D) &= 1.58219 - 0.47911 = 1.1030. \end{aligned}$$

A similar process is done for the rest of the feature groups: S1, S2, CL and NF. Once we calculate the information gain for each one of the features we can choose the one that “gains” the most information to split on.

$$\begin{aligned} Gain(\vec{T}, S1) &= 0.8865, \\ Gain(\vec{T}, S2) &= 0.4297, \\ Gain(\vec{T}, CL) &= 1.1687, \\ Gain(\vec{T}, NF) &= 0.8537. \end{aligned}$$

Based on these results we select the feature CL which has the highest value to be the root of the tree. The decision tree method examines directly the gain weights assigned to the various features. Important features are given a high weight, while unimportant features have low weight and may not be used at all. From this metric we can see that the feature CL is the feature that gives the most information followed by D and S1 features, therefore the best feature for this example is CL.

Once the best feature has been chosen, the feature will be placed in the node (if it is the first one it will be the root node). If the feature is categorical then it would split for each possible value of the feature. For our classifier, that is not the case because we have continuous numeric features. Thus, rather than generate infinitely many branches, we will restrict the possibilities to a two-way, binary split. Therefore a value on which to split on has to be found for such a node. For example, if a testing value for the feature in the root node is $\leq$ *split value* it will be direct to the left child, else if the testing value for the feature is $>$ *split value* it will be send to the right child.

To find such a *split value* we used a method based on impurity. The split selection method we used for our classifier is the *Gini*-index, this method is an impurity function or measure of inequality of a distribution [17, 51, 52], which is very similar in concept to the information gain measure used previously. The Gini-index, also known as Gini criterion, measures the "goodness" of a split; a split that maximizes the decrease in the node impurity function when moving from one node to the following nodes.

We will give the basic principles of the impurity-based split point selection method known as the Gini-index. For a more formal explanation for this and other efficient methods for finding good split points in the construction of decision trees, refer to [9, 17, 34, 36].

Continuing with our classifier, let's consider the most important feature, CL, which was selected as the root node. For the split selection we know that CL will partition the data set $\vec{T}$ into $\vec{T}_1$ and $\vec{T}_2$. Therefore, $\vec{T}_1$ will have all the feature vectors with $CL \leq v$, and $\vec{T}_2$ will have all the remaining data (feature vectors with $CL > v$), where v is the *split value* and can be any of the different values of CL. Thus, our objective is to select a good split point v for the feature CL.

"Impurity-based split selection methods assess the quality of a feature value v as a potential split point value by calculating the value of an impurity function" [52]. The impurity function we used to get the best split point is the Gini-index which is defined as follows:

$$Gini\left(\left(\vec{T}_1, \vec{T}_2\right)\right) = \frac{|\vec{T}_1|}{|\vec{T}|} Gini\left(\vec{T}_1\right) + \frac{|\vec{T}_2|}{|\vec{T}|} Gini\left(\vec{T}_2\right), \tag{3.22}$$

this function is evaluated for split value v which divides the data set $\vec{T}$ into data set $\vec{T}_1$ and data set $\vec{T}_2$ and where:

$$Gini(\vec{T}) = 1 - \sum_{i=1}^{n} \left(\frac{freq(w_i, \vec{T})}{|\vec{T}|} \right)^2, \tag{3.23}$$

where $freq(w_i, \vec{T})$ is the number of times that class w_i appears in the data set $\vec{T}$ and n is the number of classes.

Let's consider the example to calculate the split value for CL, as for the case to select the CL as the root node. We group the same values together with their corresponding classes as shown in Table 3.6. We first consider the first value in the table for the CL, then for a split value $v = 11$ we get that $|\vec{T}| = 23$ which is the total number of instances in the complete data set, $|\vec{T}_1| = 2$ and $|\vec{T}_2| = 21$, then we get:

Table 3.6: Table summarizing the diameter values for the CL feature from training data set A. Same values of the feature are collapsed together and the corresponding classes are listed. The indices at the end of the table summarize the number of times the feature value had a class related, the first coordinate is related to class D3W1, second coordinate to class D1W2 and third class to D≥17.

Corpus Luteum Feature Values from Training Data Set A										
11.0 (mm)	13.0 (mm)	14.0 (mm)	15.0 (mm)	16.0 (mm)	17.0 (mm)	18.0 (mm)	22.0 (mm)	23.0 (mm)	24.0 (mm)	28.0 (mm)
D3W1	D3W1	D≥17	D3W1	D3W1	D3W1	D3W1	D1W2	D1W2	D1W2	D1W2
D3W1	D3W1	D≥17	D≥17	D≥17	D≥17	D≥17	D1W2			
				D≥17		D≥17	D1W2			
							D1W2			
(2,0,0)	(2,0,0)	(0,0,2)	(1,0,1)	(1,0,2)	(1,0,1)	(1,0,2)	(0,4,0)	(0,1,0)	(0,1,0)	(0,1,0)

$$Gini\left(\left(\vec{T_1},\vec{T_2}\right)\right) = \frac{2}{23}Gini\left(\vec{T_1}\right) + \frac{21}{23}Gini\left(\vec{T_2}\right), \tag{3.24}$$

where:

$$\begin{aligned} Gini(\vec{T_1}) &= 1 - \left(\frac{freq(D3W1,\vec{T_1})}{2}\right)^2 - \left(\frac{freq(D1W2,\vec{T_1})}{2}\right)^2 - \left(\frac{freq(D \geq 17,\vec{T_1})}{2}\right)^2, \\ Gini(\vec{T_2}) &= 1 - \left(\frac{freq(D3W1,\vec{T_2})}{21}\right)^2 - \left(\frac{freq(D1W2,\vec{T_2})}{21}\right)^2 - \left(\frac{freq(D \geq 17,\vec{T_2})}{21}\right)^2. \end{aligned}$$

Substituting the values for the Gini equation we get:

$$\begin{aligned} Gini(\vec{T_1}) &= 0, \\ Gini(\vec{T_2}) &= 0.6621. \end{aligned}$$

Finally the Gini value for a split on $v = 11$ we get:

$$Gini_{11}((\vec{T_1},\vec{T_2})) = 0.6045. \tag{3.25}$$

Similarly we compute the Gini-index for the rest of the values. The split point with the lowest Gini-index is $v = 18$ mm, hence a split value of $CL \leq 18$ mm for the left node and $CL > 18$ mm will generate the purest nodes for that feature. Analyzing table 3.6 we can see that in fact a split on $v = 18$ mm would lead to a completely pure right node, since all the feature vectors with CL > 18 mm values had a class D1W2 (diestrus) associated with them. Since the right child is a pure node that would lead to a pure classification, we will transform this node into a leaf or internal node by associating the class D1W2 (diestrus) to the node. The CL feature will not be used again in a different part of the tree.

The same process (*Decision_Tree_Learning*) algorithm from Table 3.4 is going to be repeated recursively for the left node, when $CL \leq 18$ mm, using only the feature vectors that actually reach this node due to the split ($DataSet_i$) and the features that have not been used (*Features - best*). A new best feature will be selected (*Choose_Best_Feature* function) as well as its split value. If at some point all the feature vectors have the same classification then the *Decision_Tree_Learning* will stop. The design used in the decision tree learning algorithm for choosing the best features is designed to minimize the depth in the final tree.

It is desirable to have the size of the tree as small as possible; smaller trees are more efficient in terms of tree storage and test time requirements and more importantly smaller trees tend to generalize better for the unseen samples because they are less sensitive to the statistical irregularities of the training data set [38]. Figure 3.6 shows the decision tree constructed by using the algorithm explained in this section.

The decision tree construction is closely related to the rule construction. Each path from the root to one of the leafs of the decision tree can be expressed as a rule. For example, one path in

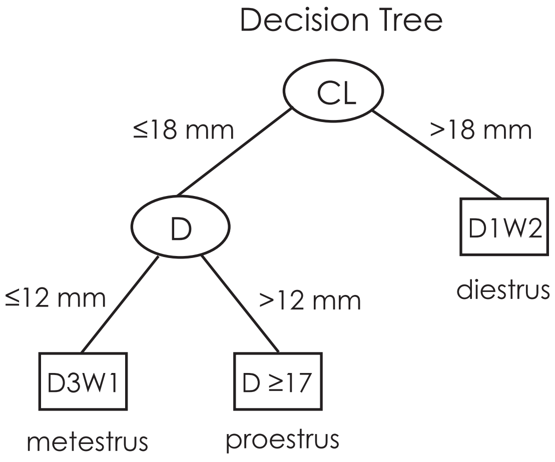


Figure 3.6: Decision tree generated by the training data set A.

the decision tree from Figure 3.6 can be transformed into a rule as: "If CL value is $\leq$ 18 mm, and the D value is $\leq$ 12 mm, then the stage in the estrous cycle is D3W1 (metestrus)".

To decide which class to assign for a test feature vector, we need to answer a series of questions, standing in the nodes of the tree, starting from the root. Let's consider a testing vector $\vec{X}$ from the testing data set B, $\vec{X} = (17, 10, 2, 15, 11)$, where D = 17 mm, S1 = 10 mm, S2 = 2 mm, CL = 15 mm and NF = 11 number of follicles. We then test this unknown feature vector into the decision tree generated by the algorithm starting in the root node, see Figure 3.6. The classification process is as follows: the pattern $\vec{X}$ is placed in the root node and asks for a particular feature, CL. If CL $\leq$ 18 mm then the left node is selected else if CL $>$ 18 mm the right child node is selected. In this case for the testing vector we have CL = 15 mm, therefore the left node is selected. The next step is again to place the pattern in the current node if we did not reach a leaf node. As we did not reach a leaf node we make again a decision in the current node, a question is asked now for the D feature. If D $\leq$ 12 mm then the left node is selected, if D $>$ 12 mm then the right node is selected. For the test node we have a value of D = 17 mm, therefore the right child is selected. Finally as this node is actually a leaf node the process stops, assigning the class D$\geq$17 or proestrus stage. Notice that for this sample just two of the five features were used to correctly classify this testing vector. To see the final performance of this classifier we will discuss different experiments and results in the next chapter.

3.4 Validation methods

Pattern classification algorithms can be separated in two stages: the *training* and *testing* stages. The use of an appropriate training and testing methodology is crucial to the validity of the classifier.

The basic idea of training and testing methodologies is that the data used in the *training* stage (training data set) should be different from the data used in the *testing* stage (testing data set). One of the most common problems when designing the classification algorithms is to *re-use* the training data set. That is, the data set used in the training stage is also used in the testing stage, this is also known as "testing on the training data" or "re-substitution method". This is a *flawed* methodology because it gives an overoptimistic misclassification error rate, also known as "apparent error rate" [14, 44].

Another important aspect when choosing a methodology for training and testing is to use a correct partition of the data set into training data set and testing data set. This is mainly important when the amount of data is limited. In this case we would like to use as much of the data set as possible to build the classifier (training stage), and enough unseen data to test the performance of the classifier (testing stage) and obtain an accurate performance estimate.

The performance of the classifier can be determined by testing the classifier with an independent *testing* data set which has not previously been seen by the classifier. To make a performance evaluation of a classifier we usually want to know the expected error rate. "The error rate is the probability of making erroneous classification for a future random chosen sample" [14]. The predicted classification made by the classifier is compared with the actual class of the test pattern. If the two match, there is no error. If they do not match an error has occurred. Then the overall performance is measured by the number of errors divided by the number of samples [52]:

$$Error\ rate = \frac{number\ of\ errors}{number\ of\ samples} \tag{3.26}$$

The error rate estimation is a single measure of performance that treats equally all correct and incorrect classifications. Hence this measure of performance can be complemented with a confusion matrix, which is useful to identify how the error rate is decomposed.

A confusion matrix is an m-by-m table, where m is the number of classes. The row labels in the confusion matrix are the actual class labels and the column labels are the predicted class labels. The value x of an entry (i, j) in the confusion matrix indicates that the pattern with true class i was classified as class j a total of x times.

The desired result is a diagonal matrix, meaning that all patterns were classified correctly. If there were errors in the classification, they will be observable in the matrix. Optionally an additional column may be added to the matrix to show the total number of patterns being classified. Table 3.7 illustrates a 3-by-3 confusion matrix example in which the class a was classified correctly for the 3 testing patterns, in the same way class b was classified correctly for the 4 testing patterns. Finally for class c it was classified correctly 4 of the 5 total of testing patterns but it was classified incorrectly for one pattern, classifying it as class a.

Table 3.7: Example of a 3-by-3 confusion matrix.

Confusion Matrix				
Classified as:	a	b	c	Total
a	**3**	0	0	3
b	0	**4**	0	4
c	1	0	**4**	5

Hold–out estimate

A general and obvious method for training and testing and therefore evaluating a classifier is known as the *hold–out estimate*. This method suggests to split the data set into two (mutually exclusive) subsets; one subset is used for designing the classifier (training stage) and the other subset to test the performance of the classifier (testing stage), which is used to estimate the error rate. This method is a single train-and-test experiment which can use a half-and-half partition, so 50% of the data set will be selected as the training data set and the remaining 50% will be the testing data set[14, 49].

Cross–validation

Cross–validation (also know as the rotation method or k-fold cross–validation) is another method of evaluating a classifier by dividing the training set into several parts, and, in turn, one part is excluded to test the classifier. The data is randomly divided into k subsets. Then, we use one subset (i.e. testing set) to test the classifier which was trained on the remaining $(k - 1)$ subsets. Thus, the training and testing stages will be done k times, each of them using a different testing set. The cross validation methodology is frequently used for small data sets in which data is scarce for both training and testing.

A common way to split the data set in cross–validation technique is by using k as 5 or 10, meaning each subset will contain 20% or 10% respectively of the total data set. An example of a $k = 5$ split is shown in Figure 3.7. An important advantage of this approach is that eventually the whole data set will be used in both training and testing stages; every pattern will be part of the testing set once and $k - 1$ times as part of the training set. A disadvantage of this method is that the training has to be rerun k times, which means it takes k times as much computation to make an evaluation of the classifier [14, 24, 35, 44, 49].

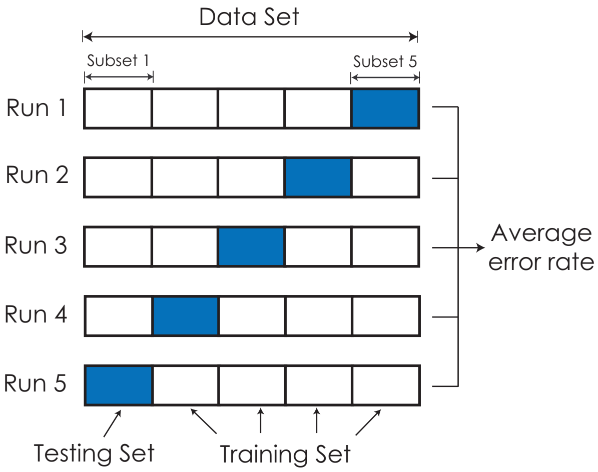


Figure 3.7: Cross validation methodology. The figure shows the split for cross validation method of $k = 5$, the data set is then divided in k (five) subsets. The training and testing stages is done five times, each of them using a different test set and the other $k - 1$ (four) subsets are put together to form the training set. The error rate is estimated as the average error of all five runs. Every data appears in a testing set exactly once and in a training set $k - 1$ times.

The error rate for this methodology e_c is calculated as the average across the k tests as follows:

$$e_c = \frac{1}{k} \sum_{i=1}^{k} e_i, \tag{3.27}$$

where e_i is the error rate for the ith data partition.

CHAPTER 4

ESTROUS PHASE CLASSIFICATION (EXPERIMENTS AND RESULTS)

A complete explanation of the experiments and results will be given in this chapter along with the performance evaluation of the decision tree and naïve Bayes classifiers. Assessment of the classifier performance is an important part of any pattern recognition classifier. Performance evaluation is important not only to measure the accuracy of the classifier but also the need for improvements.

Four different experiments were done for this project. For experiment 1, both the decision tree and naïve Bayes classifiers used a hold-out methodology for training and testing, dividing the data set into two halves (half-and-half), the first half (data set A) to train the classifier and the second half (data set B) to test the classifier. For experiment 2, both classifiers used a k-fold cross validation methodology with $k = 5$ using the complete data set (data set A and B). For experiment 3, the patterns for animals that exhibited a 3-wave follicular growth pattern were eliminated from data set A and B to form data sets A$'$ and B$'$ respectively; both classifiers were trained with data set A$'$ and tested with data set B$'$. For experiment 4, both classifiers were extended to a 4-class classification, a new class (D6W1) was incorporated to the data set A and data set B. This experiment used the hold-out methodology (half-and-half) for training and testing.

4.1 Experiment 1: 3-class classification using hold–out estimate methodology

The objective of this experiment was to test the hypothesis that by using ultrasound detected features of the bovine ovaries (size of the dominant follicle, size of the two largest subordinate follicles, size of the corpus luteum and number of follicles with size ≥ 2 mm) we can determine automatically the stage in the estrous cycle as either class 1: D3W1 (metestrus), class 2: D1W2 (diestrus) or class 3: D≥ 17 (proestrus) based on a single day's examination. This experiment was constructed using the hold–out methodology for training and testing the decision tree and naïve Bayes classifiers implemented to test this hypothesis.

Table 4.1: Results from experiment 1. Confusion matrix resulting from the classification of data set B by the decision tree classifier. The complete 22 patterns from the testing data set B were classified correctly, the classification rate was 100%.

Decision Tree Confusion Matrix				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**8**	0	0	8
D1W2	0	**6**	0	6
D≥17	0	0	**8**	8

For experiment 1, we divided the complete data set ($n = 45$ heifers with $n = 45$ pairs of ovaries) into data set A ($n = 23$ pairs of ovaries) and data set B ($n = 22$ pairs of ovaries). Hence, both the decision tree and the naïve Bayes classifier were trained using data set A and tested using data set B. The mean feature values from data sets A and B are shown in Figures 3.1 and 3.2 respectively. The graphs express the mean diameter in millimeters of D, S1, S2, and CL and their standard deviations. In the case of the NF (number of follicles) feature, the value is dimensionless. The complete feature values for training data set A and testing data set B can be seen in tables A.1 and A.3 in Appendix A.

4.1.1 Decision tree classifier

The tree derived from the decision tree algorithm described in Chapter 3 Subsection 3.3.2 is illustrated in Figure 3.6, this tree was trained using data set A. The classification proceeds from top to bottom, starting at the root node (CL). The left branch connects to the internal node (D) if the CL feature value is ≤ 18 and the right branch connects to the leaf node (class D1W2) if CL > 18, consecutively the left branch of the internal node (D) connects to the leaf node (class D3W1) if D ≤ 12 and the right branch connects to the leaf node (class D≥17) if D > 12. When the leaf node is encountered, the pattern is assigned to the class corresponding to that leaf node.

The confusion matrix for the decision tree classification of data set B is shown in table 4.1. The classification rate of the decision tree classifier for experiment 1 was 100% (22 of 22 patterns were classified correctly). This is an excellent result as it suggests that extremely high classification rates can be achieved through a decision tree that makes only two comparisons in the worst case, and requires only two features to be extracted from the input images: D and CL features.

4.1.2 Naïve Bayes classifier

The performance of the naïve Bayes classifier was trained with data set A and evaluated using the data set B. The resulting confusion matrix is shown in Table 4.2. The matrix shows that 19 patterns were classified correctly and 3 classified incorrectly for a classification rate of 86.36%. All patterns of the D3W1 class were classified correctly, however, the classifier misclassified two D1W2 patterns (one classified as D3W1 and the second as D≥17), and one D≥17 pattern (classified as D1W2).

Table 4.2: Results from experiment 1. Confusion matrix resulting from the classification of data set B by the naïve Bayes classifier. A total of 19 patterns were classified correctly and 3 patterns were classified incorrectly. The classification rate was 86.36%.

Naïve Bayes Confusion Matrix				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**8**	0	0	8
D1W2	1	**4**	1	6
D≥17	0	1	**7**	8

4.2 Experiment 2: 3-class classification using cross-validation methodology

The objective of this experiment was to test the hypothesis that by using ultrasound detected features of the bovine ovaries (size of the dominant follicle, size of the two largest subordinate follicles, size of the corpus luteum and number of follicles with size $\geq$ 2 mm) we can determine automatically the stage in the estrous cycle as either class 1: D3W1 (metestrus), class 2: D1W2 (diestrus) or class 3: D≥ 17 (proestrus) based on a single day's examination. This experiment was constructed using the k-fold cross validation methodology for training and testing the decision tree and naïve Bayes classifiers implemented to test this hypothesis. This experiment was designed to have the best use of the available data since eventually all the patterns in the data set will be used in both training and testing stages, different from experiment 1 that used half of the data set for training and half of the data set for testing.

For experiment 2 the complete data set $n = 45$ pairs of ovaries were used to evaluate both classifiers using the k-fold cross validation methodology. This experiment used both data set A and data set B ($A \cup B$) for training and testing the classifiers. The mean feature values and standard

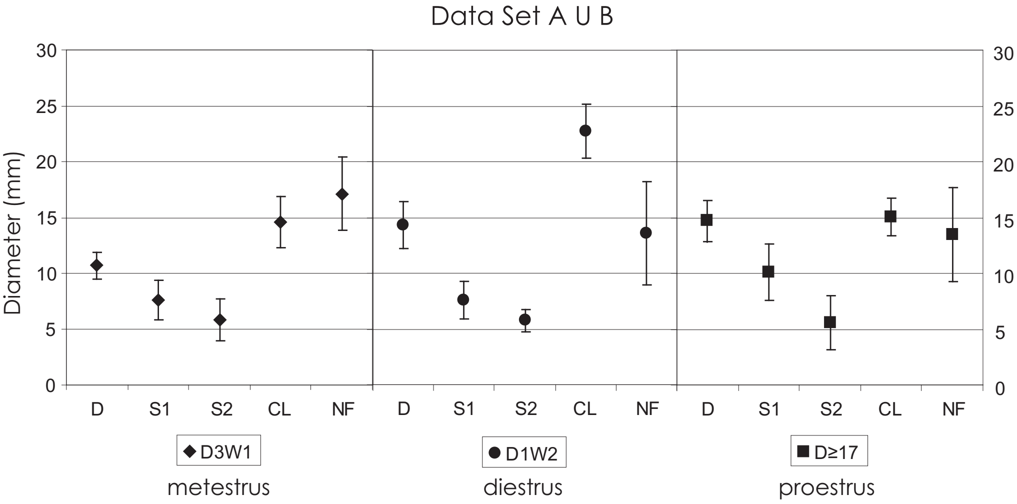


Figure 4.1: This graph shows the mean and standard deviation feature values from the complete data set formed by data set A and data set B ($A \cup B$), which is composed of 45 pairs of ovaries. The features (D, S1, S2 and CL) represent size/diameter in millimeters, NF feature is dimensionless, represents the number of follicles.

deviation values from both data sets is shown in Figure 4.1. For a complete summary of mean and standard deviation feature values of data set $A \cup B$ see Table B.4 in Appendix B.

This experiment used the cross validation methodology with k=5. Therefore the classifier will be trained with 80% of the data set $A \cup B$ and tested with the remaining 20%. Each one of the subsets will contain a total of 9 patterns. See Table B.1 in Appendix B for a complete list of the data set $A \cup B$ randomly divided into 5 subsets with both feature values and their true classification labels.

4.2.1 Decision tree classifier

Cross-validation: run 1

For the first run of the cross validation method the decision tree classifier was trained with subsets 1, 2, 3 and 4 from the data set $A \cup B$ and tested with subset 5 (this can be illustrated in Figure 3.7). The decision tree inferred from these subsets is illustrated in Figure 4.2. Notice this decision tree is identical to the tree inferred from experiment 1.

The confusion matrix for the decision tree classification for run 1 using subsets 1, 2, 3 and 4 as the training set and subset 5 as the testing set is shown in Table 4.3. The classification rate of the decision tree classifier for run 1 was 100%, the matrix shows that the testing data set $n = 9$ (subset 5) was classified correctly. Subset 5 was composed of 4 patterns of class D3W1, 2 patterns of class D1W2 and 3 patterns of class D≥17.

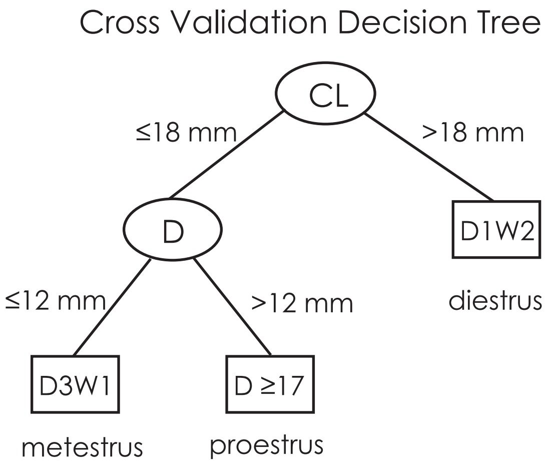


Figure 4.2: Experiment 2: decision tree obtained in the classification by using cross validation methodology. The same decision tree was generated for run 1, run 2, run 3 and run 5.

Table 4.3: Results from experiment 2. Confusion matrix resulting from run 1 for the cross validation by the decision tree classifier. All the testing patterns (subset 5) were classified correctly, the classification rate was 100%.

Decision Tree Confusion Matrix for Run 1				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**4**	0	0	4
D1W2	0	**2**	0	2
D≥17	0	0	**3**	3

Cross-validation: run 2

For run 2 the decision tree classifier was trained using a training set that was composed of subsets 1, 2, 3 and 5 and was evaluated using the subset 4 as the testing set. The decision tree inferred from this training set was identical to the decision tree of run 1 (see Figure 4.2).

The resulting confusion matrix for the decision tree classification for run 2 is shown in Table 4.4. The matrix shows that 9 of 9 patterns were classified correctly for a classification rate of 100%. The testing set (subset 4) was composed of 3 patterns of class D3W1, 5 patterns of class D1W2 and 1 pattern of class D≥17.

Table 4.4: Results from experiment 2. Confusion matrix resulting from run 2 for the cross validation by the decision tree classifier. All the testing patterns (subset 4) were classified correctly, the classification rate was 100%.

Decision Tree Confusion Matrix for Run 2				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**3**	0	0	3
D1W2	0	**5**	0	5
D≥17	0	0	**1**	1

Table 4.5: Results from experiment 2. Confusion matrix resulting from run 3 for the cross validation by the decision tree classifier. All the testing patterns (subset 3) were classified correctly, the classification rate was 100%.

Decision Tree Confusion Matrix for Run 3				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**3**	0	0	3
D1W2	0	**1**	0	1
D≥17	0	0	**5**	5

Cross-validation: run 3

For run 3 the decision tree classifier was trained using a training set that was composed of subsets 1, 2, 4 and 5 and was evaluated using the subset 3 as the testing set. The decision tree inferred from this training set was identical to the decision tree of run 1 and run 2 (see Figure 4.2).

The resulting confusion matrix for the decision tree classification for run 3 is shown in Table 4.5. The matrix shows that 9 of 9 patterns were classified correctly for a classification rate of 100%. The testing set (subset 3) was composed of 3 patterns of class D3W1, 1 pattern of class D1W2 and 5 patterns of class D≥17.

Cross-validation: run 4

For run 4 the decision tree classifier was trained using a training set that was composed of subsets 1, 3, 4 and 5 and was evaluated using subset 2 as the testing set. The decision tree inferred from this training set was slightly different from the other decision trees derived from run 1, 2 and 3. Figure 4.3 shows the decision tree inferred from this run. It is important to note that the decision tree generated by this run (training set: subsets 1, 3, 4 and 5) was very similar to the previous

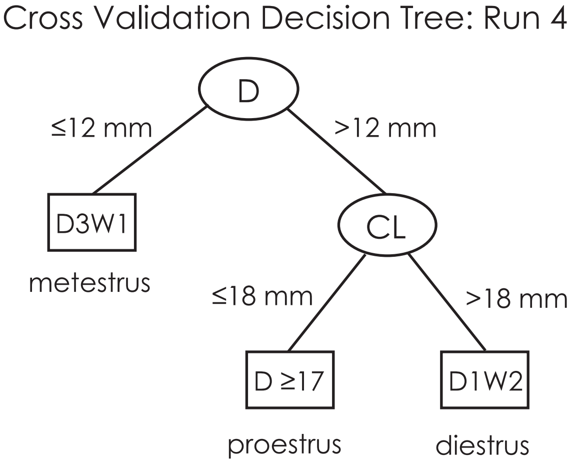


Figure 4.3: Experiment 2: decision tree obtained in the classification by using cross validation methodology. This decision tree was generated for run 4.

Table 4.6: Results from experiment 2. Confusion matrix resulting from run 4 for the cross validation by the decision tree classifier. All the testing patterns (subset 2) were classified correctly, the classification rate was 100%.

Decision Tree Confusion Matrix for Run 4				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**3**	0	0	3
D1W2	0	**2**	0	2
D≥17	0	0	**4**	4

runs; using only D and CL features to make the classification. The split values for D and CL were the same, 12 and 18 mm respectively. In fact, for this run both features D and CL contained the same amount of information gain (measure to select the best feature to split on), that means that any of the two different features D or CL could have been chosen as the root node (best feature), in this case the inference algorithm selected the first feature in the data set (D) to be the root node.

The resulting confusion matrix for the decision tree classification for run 4 is shown in Table 4.6. The matrix shows that 9 of 9 patterns were classified correctly for a classification rate of 100%. The testing set (subset 2) was composed of 3 patterns of class D3W1, 2 patterns of class D1W2 and 4 patterns of class D≥17.

Table 4.7: Results from experiment 2. Confusion matrix resulting from run 5 for the cross validation by the decision tree classifier. All the testing patterns (subset 1) were classified correctly, the classification rate was 100%.

Decision Tree Confusion Matrix for Run 5				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**3**	0	0	3
D1W2	0	**3**	0	3
D≥17	0	0	**3**	3

Cross-validation: run 5

For run 5 the decision tree classifier was trained using a training set that was composed of subsets 2, 3, 4 and 5 and was evaluated using the subset 1 as the testing set. The decision tree inferred from this training set was identical to the decision tree from run 1, run 2 and run 3 (see Figure 4.2).

The resulting confusion matrix for the decision tree classification for run 5 is shown in Table 4.7. The matrix shows that 9 of 9 patterns were classified correctly for a classification rate of 100%. The testing set (subset 1) was composed of 3 patterns of class D3W1, 3 patterns of class D1W2 and 3 patterns of class D≥17.

The final classification rate for this experiment can be calculated using equation 3.27 which is the average of the error rates from all runs. Thus the classification rate for experiment 2 using decision tree classifier is 100%.

4.2.2 Naïve Bayes classifier

The performance of the naïve Bayes classifier for experiment 2 was evaluated using the same cross validation methodology. Similarly to the decision tree classification, for run 1 the Bayes classifier was trained using subsets 1, 2, 3 and 4 from the data set $A \cup B$ and tested with subset 5. The resulting confusion matrix for run 1 was identical to the decision tree confusion matrix shown in Table 4.3 for run 1. The classification rate was 100% with 9 of 9 patterns classified correctly.

For run 2 the naïve Bayes classifier was trained using a training set that was composed of subsets 1, 2, 3 and 5 and was evaluated using the subset 4 as the testing set. The resulting confusion matrix for this classification is shown in Table 4.8. The matrix shows that 8 of 9 patterns were classified correctly and 1 classified incorrectly for a classification rate of 88.88%. All patterns of the D1W2 and D≥17 classes were classified correctly, 2 patterns of D3W1 class were classified correctly and 1 pattern was classified incorrectly as D≥17.

Table 4.8: Results from experiment 2. Confusion matrix resulting from run 2 for the cross validation by the naïve Bayes classifier. 8 of the 9 testing patterns were classified correctly, the classification rate was 88.88%.

Naïve Bayes Confusion Matrix for Run 2				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**2**	0	1	3
D1W2	0	**5**	0	5
D≥17	0	0	**1**	1

Table 4.9: Results from experiment 2. Confusion matrix resulting from run 4 for the cross validation by the naïve Bayes classifier. 8 of the 9 testing patterns were classified correctly, the classification rate was 88.88%.

Naïve Bayes Confusion Matrix for Run 4				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**3**	0	0	3
D1W2	0	**2**	0	2
D≥17	1	0	**3**	4

For run 3 the resulting confusion matrix was identical to run 3 for the decision tree shown in Table 4.5. The classification rate was 100% with 9 of 9 patters classified correctly.

For run 4 the naïve Bayes classifier was trained using a training set that was composed of subsets 1, 3, 4 and 5 and was evaluated using the subset 2 as the testing set. The resulting confusion matrix for this classification is shown in Table 4.9. The matrix shows that 8 of 9 patterns were classified correctly and 1 classified incorrectly for a classification rate of 88.88%. All patterns of the D3W1 and D1W2 classes were classified correctly; 3 patterns of D≥17 class were classified correctly and 1 pattern was classified incorrectly as D3W1.

For run 5 the classification rate was 100% with 9 of 9 patters classified correctly.

The final classification rate for this experiment can be calculated using equation 3.27 which is the average of the error rates from all runs. Thus the classification rate for experiment 2 using the naïve Bayes classifier was 95.55%. This is a better result compared to experiment 1, this could be due to the fact that cross-validation methodology has a more efficient use of the data set, meaning that more feature vectors (80% of the data set) were used for the training stage (different from the 50% of the data set used in experiment 1), providing more information to get a better classification.

4.3 Experiment 3: 3-class classification for animals with 2-wave follicular patterns

The objective of this experiment was to test the hypothesis that by using ultrasound detected features of the bovine ovaries (size of the dominant follicle, size of the two largest subordinate follicles, size of the corpus luteum and number of follicles with size ≥ 2 mm) of animals presenting only 2-wave follicular growth patterns we can determine automatically the stage in the estrous cycle as either class 1: D3W1 (metestrus), class 2: D1W2 (diestrus) or class 3: D$\geq$ 17 (proestrus) based on a single day's examination. This experiment was proposed to verify if animals with 2 and 3 wave follicular patterns were confused during the classification. This experiment was constructed using the hold–out methodology for training and testing the decision tree and naïve Bayes classifiers implemented to test this hypothesis.

For experiment 3, the animals that exhibited a 3-wave follicular growth pattern were removed from data sets A and B to form data sets A$'$ and B$'$ respectively. For data set A, 4 samples in the D$\geq$17 class were removed leaving 8 patterns from D3W1, 7 patterns from D1W2 and 4 patterns from D$\geq$17; a total of 19 patterns. For data set B, 2 samples in class D$\geq$17 exhibiting a 3-wave follicular pattern were removed, leaving 8 patterns from D3W1, 6 patterns from D1W2 and 6 patterns from D$\geq$17; a total of 20 patterns.

The decision tree and naïve Bayes classifiers were then trained using the data set A$'$, and tested using data set B$'$. The mean feature values of this experiment are expressed in Figure 4.4 for the data set A$'$ and Figure 4.5 for data set B$'$. The graphs express the mean diameter in millimeters of D, S1, S2, and CL. In the case of the NF feature, the value is dimensionless. From the figures related to the original data sets A and B (Figures 3.1 and 3.2), there is not a notable difference between data set A and data set A$'$ and data set B and data set B$'$ respectively due to the small number of patterns that presented a 3-wave follicular pattern and the fact that the 3-wave animals belonged only to the D$\geq$17 classes.

4.3.1 Decision tree classifier

For this experiment animals with a 3-wave follicular growth pattern were eliminated, thus the classifier was trained using data set A$'$ and tested using data set B$'$. The decision tree inferred from data set A$'$ was identical to the decision tree of experiment 1 (see Figure 3.6). The confusion matrix for the decision tree classification using the testing data set B' is shown in table 4.10. The classification rate was 100% with all patterns classified correctly.

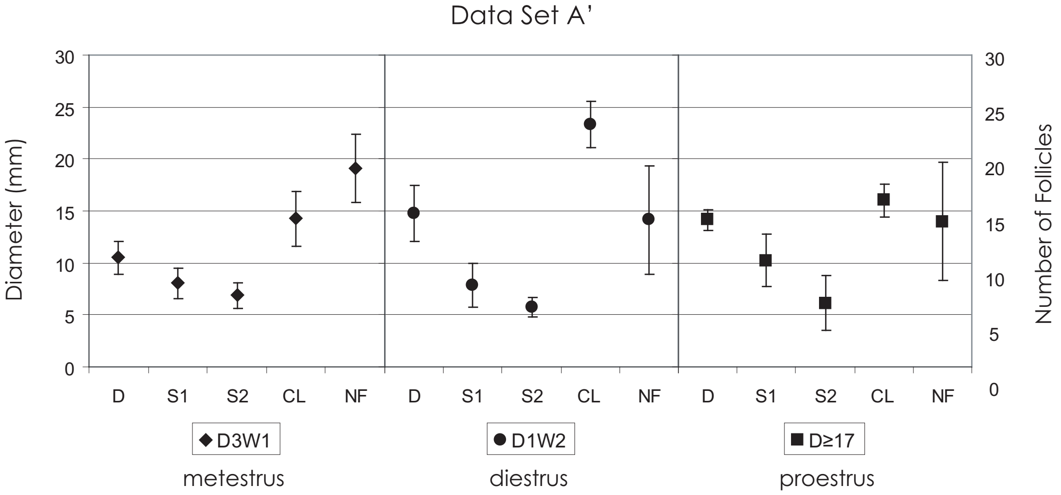


Figure 4.4: This graph shows the mean and standard deviation feature values from the training data set A′ . The features (D, S1, S2 and CL) represent size/diameter in millimeters, NF feature is dimensionless, represents the number of follicles.

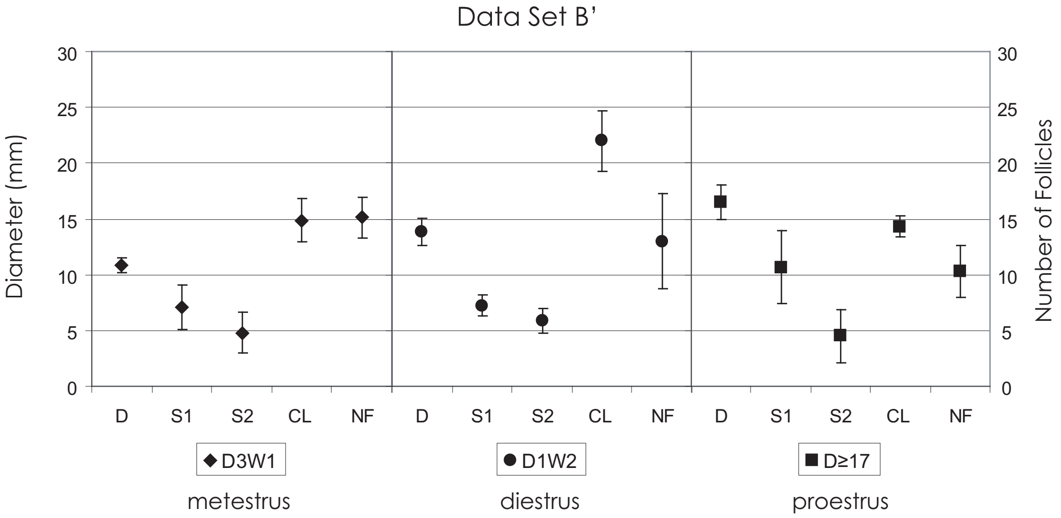


Figure 4.5: This graph shows the mean and standard deviation feature values from the testing data set B′ . The features (D, S1, S2 and CL) represent size/diameter in millimeters, NF feature is dimensionless, represents the number of follicles.

Table 4.10: Results from experiment 3. Confusion matrix resulting from training with data set A' and testing with data set B' for the decision tree classifier. All the testing patterns were classified correctly, the classification rate was 100%.

Decision Tree Confusion Matrix				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**8**	0	0	8
D1W2	0	**6**	0	6
D≥17	0	0	**6**	6

4.3.2 Naïve Bayes classifier

The performance of the naïve Bayes classifier was evaluated using data set B′. The resulting confusion matrix is shown in Table 4.11. The matrix shows that 90% (18 of 20) patterns were classified correctly.

All instances for D≥17 class were classified correctly while D3W1 and D1W2 had one misclassification each and were classified as D≥17 and D3W1 respectively. Interestingly, the 3-wave patterns that were eliminated for this experiment were in fact classified correctly in experiment 1, which suggests the 2-wave and 3-wave patterns were not confused during the classification and presented similar characteristics.

Table 4.11: Results from experiment 3. Confusion matrix resulting from training with data set A' and testing with data set B' for the naïve Bayes classifier. 18 of 20 testing patterns were classified correctly, the classification rate was 90%.

Naïve Bayes Confusion Matrix				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**7**	0	1	8
D1W2	1	**5**	0	6
D≥17	0	0	**6**	6

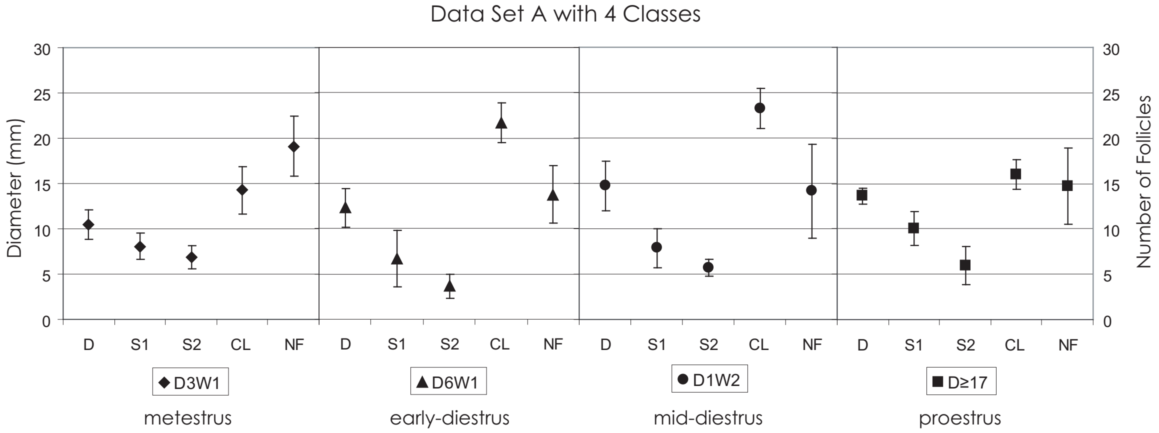


Figure 4.6: This graph shows the mean and standard deviation feature values from data set A with 4 classes: D3W1, D6W1, D1W2 and D$\geq$17. The features (D, S1, S2 and CL) represent size/diameter in millimeters, NF feature is dimensionless, represents the number of follicles.

4.4 Experiment 4: 4-class classification using hold–out estimate methodology

The objective of this experiment was to test the hypothesis that by using ultrasound detected features of the bovine ovaries (size of the dominant follicle, size of the two largest subordinate follicles, size of the corpus luteum and number of follicles with size $\geq$ 2 mm) we can determine automatically the stage in the estrous cycle as either class 1: D3W1 (metestrus), class 2: D6W1 (early-diestrus), class 3: D1W2 (mid-diestrus) or class 4: D$\geq$ 17 (proestrus) based on a single day's examination. This experiment was constructed using the hold–out methodology for training and testing the decision tree and naïve Bayes classifiers implemented to test this hypothesis.

Experiment 4 is an extension of the 3-class (D3W1, D1W2 and D$\geq$17) classification of this study. Additional patterns were incorporated to this experiment to have an additional class: D6W1 (day 6 of wave 1). The four classes: D3W1, D6W1, D1W2 and D$\geq$17 correspond roughly to metestrus, early-diestrus, mid-diestrus and proestrus phases of the estrous cycle respectively.

The patterns corresponding to D6W1 were incorporated to data set A and data set B. Figure 4.6 shows the graph for data set A with 4 classes and Figure 4.7 shows the graph for data set B with 4 classes.

The feature values for D6W1 were extracted in the same way as the feature values from the rest of the classes as D, S1, S2, CL and NF. For each animal, both left and right ovaries were ovariectomized and imaged *in vitro* during D6W1 (early-diestrus) $n = 9$ pairs of ovaries for data set A and $n = 6$ pairs of ovaries for data set B.

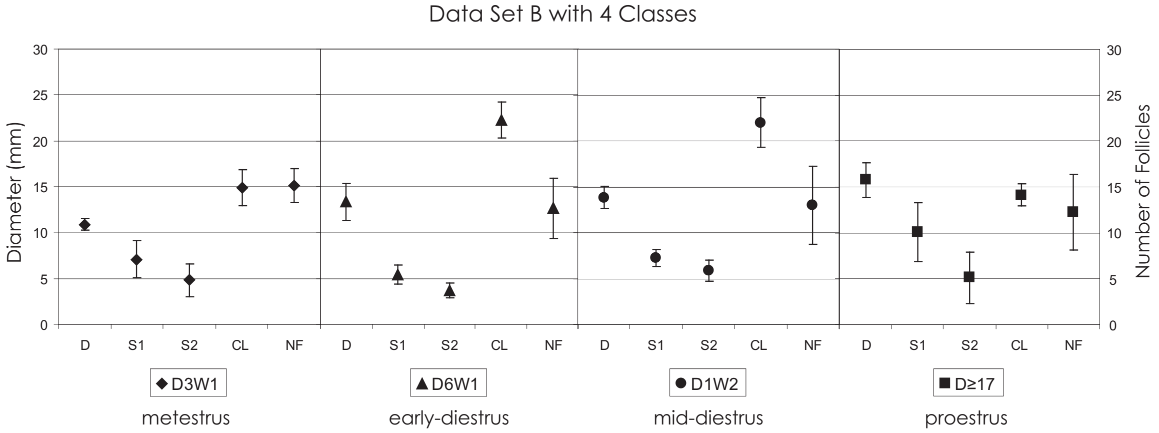


Figure 4.7: This graph shows the mean and standard deviation feature values from data set B with 4 classes: D3W1, D6W1, D1W2 and D≥17. The features (D, S1, S2 and CL) represent size/diameter in millimeters, NF feature is dimensionless, represents the number of follicles.

Therefore, the training data set A with 4 classes consisted of 32 animals; $n_1 = 8$ related to D3W1 (metestrus), $n_2 = 9$ related to D6W1 (early-diestrus), $n_3 = 7$ related to D1W2 (mid-diestrus) and $n_4 = 8$ related to D≥17 (proestrus). Data set B with 4 classes was used as the testing set with a total of $n = 28$ animals; $n_1 = 8$ related to D3W1 (metestrus), $n_2 = 6$ related to D6W1 (early-diestrus), $n_3 = 6$ related to D1W2 (mid-diestrus) and $n_4 = 8$ related to D≥17 (proestrus).

4.4.1 Decision tree classifier

The decision tree classifier for this experiment was trained using data set A with 4 classes and tested with data set B with 4 classes. For experiment 4 the tree obtained from the decision tree classifier is illustrated in Figure 4.8. In the same way the classification proceeds from top to bottom leading to a leaf node related to a class. In this tree all features were used except for the NF feature. It is important to note that this tree added S1 and S2 features in addition to the D and CL to the decision tree, in contrast to the trees obtained from previous experiments were the only features appearing in the decision trees were D and CL. The inference algorithm used the next best features from the data set S1 and S2 to distinguish between classes D6W1 and D1W2.

The evaluation of the classifier was made by using data set B with 4 classes. The resulting confusion matrix is shown in Table 4.12. The matrix shows that 20 patterns were classified correctly and 8 patterns classified incorrectly for a classification rate of 71.43%. From this result it is possible to see that all patterns of D3W1 and D≥17 classes were classified correctly, however, the classifier misclassified D6W1 and D1W2 classes, 5 of 6 patterns of D6W1 were incorrectly classified as D1W2 and 3 of 6 patterns of D1W2 were incorrectly classified as D6W1.

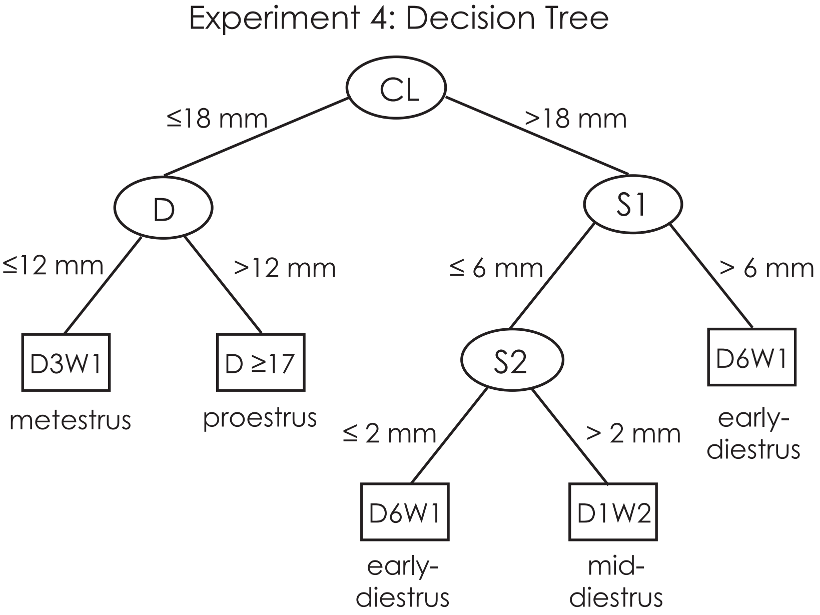


Figure 4.8: Experiment 4: decision tree obtained in the classification by using data set A as the training set including four classes: D3W1, D6W1, D1W2 and D≥17 corresponding to metestrus, early-diestrus, mid-diestrus and proestrus respectively.

4.4.2 Naïve Bayes classifier

The performance of the naïve Bayes classifier was evaluated using data set B with 4 classes. The resulting confusion matrix is shown in Table 4.13. The matrix shows that 18 patterns were classified correctly and 10 patterns classified incorrectly for a classification rate of 64.29%.

From this result we can see that this classifier has difficulty distinguishing classes D6W1 and D1W2. Classes D3W1 and D≥17 also had some misclassification in contrast to the decision tree classification for this experiment that classified all patterns correctly for D3W1 and D≥17 classes.

Table 4.12: Results from experiment 4. Confusion matrix resulting from the decision tree classifier by using data set A as the training set and data set B as the testing set including 4 classes: D3W1, D6W1, D1W2 and D≥17. A total of 20 patterns were classified correctly and 8 patterns incorrectly, the classification rate was 71.43%.

Decision Tree Confusion Matrix					
Classified as:	D3W1	D6W1	D1W2	D≥17	Total
D3W1	**8**	0	0	0	8
D6W1	0	**1**	5	0	6
D1W2	0	3	**3**	0	6
D≥17	0	0	0	**8**	8

Table 4.13: Results from experiment 4. Confusion matrix resulting from the naïve Bayes classifier by using data set A as the training set and data set B as the testing set including 4 classes: D3W1, D6W1, D1W2 and D≥17. A total of 18 patterns were classified correctly and 10 patterns incorrectly, the classification rate was 64.29%.

Naïve Bayes Confusion Matrix					
Classified as:	D3W1	D6W1	D1W2	D≥17	Total
D3W1	**6**	0	0	2	8
D6W1	0	**2**	4	0	6
D1W2	0	2	**3**	1	6
D≥17	0	0	1	**7**	8

CHAPTER 5

DISCUSSION AND CONCLUSIONS

This chapter provides a discussion of the thesis with a summary of the different experiments and conclusions derived from the results of the experiments. Finally, future work resulting from this thesis is discussed.

Due to the wave-like follicular growth, explained in Section 2.1, it was conjectured that the size of the dominant follicle, the size of the two largest subordinate follicles, and the total number of subordinate follicles would be useful features for distinguishing between the different stages in the estrous cycle: metestrus, diestrus and proestrus phases. The estrus stage was not considered due to its short duration and the lack of available data. The size of the corpus luteum is also a useful feature for reproductive phase discrimination.

The results of the present study from experiment 1, 2 and 3 supported the hypothesis that by using ultrasound detected features of the bovine ovaries (size of the dominant follicle, size of the two largest subordinate follicles, size of the corpus luteum and number of follicles with size ≥ 2 mm) we can automatically and robustly determine the stage in the estrous cycle as either class 1: D3W1 (metestrus), class 2: D1W2 (diestrus) or class 3: D$\geq$ 17 (proestrus) based on a single day's examination. The results described in Chapter 4 showed that both the decision tree and naïve Bayes classifiers performed considerably well and confirmed that the features chosen to describe the stage in the estrous cycle were sufficient information to produce a correct 3-class classification, despite the potential sources of error arising from collecting features from only a single day's examination of both ovaries.

5.1 Experiment 1: 3-class classification using hold–out estimate methodology

Experiment 1 was trained and tested using the hold-out estimate methodology. The classifiers were trained with data set A and tested with data set B with a total of 22 and 23 heifers respectively; using animals that presented 2 and 3 wave patterns of follicular activity. The decision tree classifier performed perfectly, classifying all the testing instances correctly for a classification rate of 100%. The decision tree used only the CL and D features for the classification. The size of the CL was generally larger in the D1W2 (diestrus) class ($\simeq$23.2mm for data set A and $\simeq$22mm for data set B) and smaller in class D3W1 (metestrus, $\simeq$14.25mm for data set A and $\simeq$14.88mm for data set B) and D$\geq$ 17 (proestrus, $\simeq$16mm for data set A and $\simeq$14.12mm for data set B). The size of D was larger in both the D1W2 (diestrus) and D$\geq$17 (proestrus) classes, which had similar values as: $\simeq$14.7mm and $\simeq$13.62mm for data set A respectively, and $\simeq$13.8mm and $\simeq$15.7mm for data set B respectively. The size of D for D3W1 (metestrus) class was smaller with mean values of $\simeq$10.5mm for data set A and $\simeq$10.8mm for data set B. For a complete summary statistics about the features of data set A and B see Appendices A.2 and A.4.

The naïve Bayes classifier classified 86.36% of the instances correctly ($n = 19$) and 13.64% incorrectly ($n = 3$). Patterns for class D3W1 (metestrus) were classified correctly, nevertheless, the classifier misclassified two patterns for D1W2 and one for D$\geq$17 classes.

5.2 Experiment 2: 3-class classification using cross-validation methodology

Experiment 2 was trained and tested using the k-fold cross-validation methodology, using the complete data set $A \cup B$ for training and testing and $k = 5$. Both classifiers were trained with 80% of the data set $A \cup B$ and tested with the remaining 20% for total of $k = 5$ runs (each time the test set was different). The decision tree performed perfectly, classifying all the test instances correctly for the 5 runs (average classification rate 100%). The decision trees derived from this experiment were identical to experiment 1, using only CL and D features for the classification. The naïve Bayes had an average classification rate of 95.5% over the 5 runs, which was better than the classification rate for experiment 1 (86.3%). This could be due to the fact that cross-validation methodology is characterized to have a more efficient use of the data considering two factors: one factor is that more data set instances were provided in the training stages (80% of the data set for each run) compared to the 50% of the data set used for experiment 1 giving more information to have a better classification. The second factor is that the testing data set was smaller, only 20% of the data set instances were used for testing in each run (different from the 50% of the data set used in experiment 1).

5.3 Experiment 3: 3-class classification for animals with 2-wave patterns

Experiment 3 was trained and tested using the hold-out estimate methodology. The classifiers were trained with data set A$'$ and tested with data set B$'$ which included only animals that exhibited 2-wave follicular growth patterns (animals with 3-wave follicular patterns were eliminated from the data sets). The decision tree classifier classified 100% of the instances correctly with a decision tree identical to experiment 1 and 2 using only D and CL features for the classification. The naïve Bayes classifier exhibited a small improvement classifying 90% of the instances correctly compared to experiment 1 (classification rate of 86.39%) that used the same hold-out methodology for training and testing. This experiment suggested that the extraction of animals that exhibited 3-wave follicular growth patterns did not eliminate any error or improve noticeably the performance of the classifiers: in experiment 1 the instances that corresponded to animals with 3-wave follicular patterns were classified correctly by both classifiers. This suggested that the instances with both follicular growth patterns (2-wave and 3-wave follicular patterns) presented similar characteristics on the days of the ultrasound examination of the ovaries. Evaluation of the classifiers using a larger data set with animals containing both follicular patterns is required to fully demonstrate their insensitivity to 2-wave and 3-wave patterns of follicular growth. This would achieve an important level of robustness since it is not currently possible to determine whether an animal exhibits a 2 or 3 wave follicular patterns without daily examination.

5.4 Experiment 4: 4-class classification using hold–out estimate methodology

Experiment 4 was an extension to the previous 3-class classification experiments with the inclusion of an additional class: D6W1 (early-diestrus) giving a 4-class classification. The 4 classes corresponded roughly to D3W1 (metestrus), D6W1 (early-diestrus), D1W2 (mid-diestrus) and D≥17 (diestrus). This experiment was trained and tested using the hold-out estimate methodology. The classifiers were trained with data set A with four classes and tested with data set B with four classes (the original data sets A and B plus the additional D6W1 data). The decision tree classifier had a classification rate of 71.43%, with 20 of 28 instances classified correctly. The decision tree derived from this experiment used four of the five available features: D, S1, S2 and CL to perform the classification. Patterns for class D3W1 (metestrus) and D≥17 (diestrus) were classified correctly, nevertheless, patterns for classes D6W1 (early-diestrus) and D1W2 (mid-diestrus) were confused between them; with 5 of 6 instances of D6W1 class classified as D1W2 and 3 of 6 instances of D1W2 classified as D6W1. An important note from this result is that even though some instances (from D6W1 and D1W2 classes) were classified incorrectly by the decision tree, the instances were classified as either D6W1 or D1W2 and were not confused with the rest of the classes. The naïve Bayes classifier had more difficulty distinguishing among classes, it classified 64.29% of the instances correctly (n=18) and 35.71% incorrectly (n=10). For this classifier patterns for class D3W1 (metestrus) were classified correctly, however, patterns from classes D6W1, D1W2 and D≥17 were confused among the different classes, with a special emphasis in D6W1 and D1W2 classes. Both classifiers had a low classification rate compared to experiments 1, 2 and 3 suggesting that the selected features used to build the classifiers were insufficient for distinguishing between the early- and mid-diestrus classes. The results of this experiment suggested that a selection of additional features could be required to accurately determine the stage in the estrous cycle for a 4-class classification based on a single day's ultrasound examination of the ovaries.

In conclusion, the experiments revealed that the performance of the decision tree classifier for experiments 1, 2 and 3 achieved the best results, giving a classification rate of 100% to detect the stage in the estrous cycle. For experiment 4 the decision tree classifier gave a classification rate of 71.43% (which was the first approach to a four-class classification). The decision tree derived from experiments 1, 2 and 3 was identical giving a very simple solution to the classification problem with a decision tree that was small and easy to understand and interpret. Another important advantage of this result is that, although all the features were used in the training stage, the decision tree inference algorithm determined that only two features from the training patterns were needed (CL and D). This implied that for a three-class classification the most discriminating of the features chosen were the size of the corpus luteum and size of the dominant follicle. This is a good result as it suggests that extremely high classification rates can be achieved through a decision tree that makes only two comparisons in the worst case, and requires only two features to be extracted from the input images. Thus, these two features may be sufficient to construct a robust three-class classifier, although a larger-scale experiment would be needed to verify this hypothesis.

The success of the decision tree classifier based on only two features is somewhat surprising, given the errors that can arise in feature extraction due to the potential presence of follicles belonging to different waves in a single image as was discussed in Section 2.1. That such a simple decision tree solves such an apparently complicated classification problem so well is rather astonishing and offers the potential for extremely fast, reliable, and consistent automatic decision making.

The performance of the naïve Bayes classifier achieved reasonably good results, improving from 86.36% (experiment 1) to 90% (experiment 3) when the patterns from 3-wave animals were removed. An increase in the classification rate (95.55%) was achieved when the cross-validation methodology was used for training and testing in experiment 2. For experiment 4, the classification rate was of 64.29% when attempting to make a 4-class classification including a D6W1 (early-diestrus) class. The decision tree inference algorithm for this experiment determined that four of the five available features from the training patterns were needed (D, S1, S2 and CL) to perform a classification.

The work herein constitutes the third stage of what could become a fully automated system for determining the current reproductive phase of mammals on the basis of a single ultrasound examination. The first stage of such a system would be the segmentation of the relevant ovarian structures. If the size of the dominant follicle and size of the CL are a sufficiently rich feature set, then the follicle segmentation problem can be solved fairly easily. Potočnik reported that his algorithm correctly segments nearly 100% of large follicles greater than 10mm [33]. Segmentation of the CL is the subject of current research. For the second step, one need only recognize the largest follicle, and measure its diameter. Thus, if future work can achieve a robust segmentation algorithm for the CL, the entire process could be fully automated.

The work presented in this thesis provides extensive opportunities for future work such as:

- selection of different features that could give additional information to improve the performance of the 4-class classification. Different features could better describe the stages in the estrous cycle so that patterns from the early-diestrus (D6W1) and mid-diestrus stages (D1W2) for a for a 4-class classification could be better distinguished. Some features that warrant consideration are: echotexture characteristics of follicular images (walls and antrum), difference in size between the dominant follicle (D) and the first subordinate follicle (S1), difference in size between the first subordinate (S1) and second subordinate follicle (S2);
- more extensive experiments with additional data to complement both the decision tree and naïve Bayes classifiers. The results obtained from this project were generated based on a limited amount of ultrasound images of bovine ovaries, it would be desirable to get more ultrasound images during different days in the estrous cycle, so we could get more information to extend the current classifiers to a higher number of classes such as the different days in the estrous cycle;
- development of a classifier that could distinguish between two or three wave patterns based on a sequence of images from a single animal;
- a fully automated system for determining the current stage in the estrous cycle could be achieved by successfully automating the segmentation and measurement of the relevant ovarian structures.

REFERENCES

[1] G.P. Adams, K. Kot, and O.J. Ginther. Selection of a dominant follicle and suppression of follicular growth in heifers. *Animal Reproduction Science*, 30:259–271, 1993.

[2] G.P. Adams, K. Kot, C.A. Smith, and O.J. Ginther. Effect of the dominant follicle on regression of its subordinates heifers. *Canadian Journal of Animal Science*, 73:267–275, 1993.

[3] G.P. Adams, R.L. Matteri, J.P. Kastelic, J.C.H Ko, and O.J. Ginther. Association between surges of follicle-stimulating hormone and the emergence of follicular waves in heifers. *Journal of Reproduction and Fertility*, 94:177–188, 1992.

[4] G.P. Adams and R.A. Pierson. Bovine model for study of ovarian follicular dynamics in humans. *Theriogenology*, 43:113–120, 1995.

[5] N.B. Amor, S. Benferhat, and Z. Elouedi. Naive Bayes vs Decision Trees in intrusion detection systems. In *ACM Symposium on Applied Computing*, pages 420–424, 2004.

[6] A.R. Baerwald, G.P. Adams, and R.A. Pierson. Characterization of ovarian follicular wave dynamics in women. *Biology of Reproduction*, 69:1023–1031, 2003.

[7] A.R. Baerwald, G.P. Adams, and R.A. Pierson. A new model for ovarian follicular development during the human menstrual cycle. *Fertility and Sterility*, 80(1):116–122, July 2003.

[8] M.H. Beers. *The Merck Manual of Medical Information.* Merck Research Laboratories, second home edition, 2003.

[9] L. Breiman, J. H. Friedman, R.A Olshen, and C. J. Stone. *Classification and Regression Trees.* Monterey, CA. Wadsworth and Brooks, 1984.

[10] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[11] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[12] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods.* Cambridge University Press, 2000.

[13] E.R. Davies. *Machine Vision. Theory, Algorithms, and Practicalities.* Morgan Kaufmann Publishers, third edition, 2005.

[14] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall International, 1982.

[15] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[16] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* John Wiley & Sons, second edition, 2001.

[17] J.L. Gastwirth. The estimation of the Lorenz curve and Gini index. *The Review of Economics and Statistics*, 54(3):306–16, August 1972.

[18] A. Gilat. *MATLAB an Introduction with applications.* John Wiley, 2005. Documentation available at http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html.

[19] O.J. Ginther. *Ultrasonic Imaging and Animal Reproduction: Fundamentals (Book 1).* Equiservices Publishing, 1995.

[20] O.J. Ginther. *Ultrasonic Imaging and Animal Reproduction: Horses (Book 2).* Equiservices Publishing, 1995.

[21] O.J. Ginther, J.P. Kastelic, and L. Knopf. Composition and characteristics of follicular waves during the bovine estrous cycle. *Animal Reproduction Science*, 20:187–200, 1989.

[22] O.J. Ginther, J.P. Kastelic, and L. Knopf. Intraovarian relationships among dominant and subordinate follicles and the corpus luteum in heifers. *Theriogenology*, 32(5):787–795, 1989.

[23] O.J. Ginther, L. Knopf, and J.P. Kastelic. Temporal associations among ovarian events in cattle during oestrous cycle with two and three follicular waves. *Reproduction Fertility*, 87:223–230, 1989.

[24] L.I. Kuncheva. *Combining Pattern Classifiers Methods and Algorithms.* John Wiley & Sons, 2004.

[25] I. Maldonado-Castillo, M.G. Eramian, R.A. Pierson, J. Singh, and G.P. Adams. Classification of bovine reproductive cycle phase using ultrasound-detected features. In *Fourth Canadian Conference on Computer and Robot Vision*, pages 258–265. IEEE Computer Society, 2007.

[26] A.R. Peters and P.J.H. Ball. *Reproduction in Cattle.* Blackwell Science, second edition, 1995.

[27] R.A. Pierson and G.P. Adams. Computer-assisted image analysis, diagnostic ultrasonography and ovulation induction: Strange bedfellows. *Theriogenology*, 43:105–112, 1995.

[28] R.A. Pierson and O.J. Ginther. Ultrasonography of the bovine ovary. *Theriogenology*, 21:495–504, 1984.

[29] R.A. Pierson and O.J. Ginther. Follicular populations during the estrous cycle in heifers: Part I. Influence of day. *Animal Reproduction Science*, 124:165–176, 1987.

[30] R.A. Pierson and O.J. Ginther. Reliability of diagnostic ultrasonography for identification and measurement of follicles and detecting corpus luteum in heifers. *Theriogenology*, 29:21–37, 1987.

[31] R.A. Pierson and O.J. Ginther. Follicular populations during the estrous cycle in heifers: Part III. Time of selection of ovulatory follicle. *Animal Reproduction Science*, 16:81–95, 1988.

[32] R.A. Pierson and O.J. Ginther. Ultrasonic imaging of the ovaries and uterus in cattle. *Theriogenology*, 29:21–37, 1988.

[33] B. Potočnik and D. Zazula. Automated analysis of a sequence of ovarian ultrasound images. Part I: segmentation of single 2D images. *Image and Vision Computing*, 20:217–225, 2002.

[34] J.R. Quinlan. Induction of Decision Trees. In Jude W. Shavlik and Thomas G. Dietterich, editors, *Readings in Machine Learning.* Morgan Kaufmann, 1990. Originally published in *Machine Learning* 1:81–106, 1986.

[35] B.D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[36] L. Rokach and O. Maimon. Top-Down induction of Decision Trees classifiers- a Survey. *IEEE Transcations on Systems, Man, and Cybernetics- PartC: Applications and Reviews*, 35(4):476–487, November 2005.

[37] S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach.* Prentice Hall, second edition, 2003.

[38] S.R. Safavian and D. Landgrebe. A survey of Decision Tree classifier methodology. In *IEEE Transactions on Systems, Man, And Cybernetics*, volume 21, pages 660–674, May/June 1991.

[39] C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, Illinois., 1949.

[40] L.G. Shapiro and G.C. Stockman. *Computer Vision.* Prentice Hall, 2001.

[41] J. Singh. *Bovine Ovary: Morphologic and Biochemical Kinetics.* PhD thesis, University of Saskatchewan, 1997.

[42] J. Singh, R.A. Pierson, and G.P. Adams. Ultrasound image attributes of bovine corpus luteum: Structural and functional correlates. *Journal of Reproduction and Fertility*, 109:35–44, 1997.

[43] J. Singh, R.A. Pierson, and G.P. Adams. Ultrasound image attributes of bovine ovarian follicles and endocrine and functional correlates. *Journal of Reproduction and Fertility*, 112:19–29, 1998.

[44] M. Sonka and J.M. Fitzpatrick. *Handbook of Medical Imaging*, volume 2: Medical Image Processing and Analysis. SPIE Press, 2000.

[45] T. Sutton. *Introduction to Animal Reproduction.* E.I. Sutton Consulting, second edition, 2000.

[46] C.W. Therrien. *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics.* John Wiley & Sons, 1989.

[47] J.W. Tom, R.A. Pierson, and G.P. Adams. Quantitative echotexture analysis of bovine corpora lutea. *Theriogenology*, 49:1345–1352, 1998.

[48] V.N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[49] A.R. Webb. *Statistical Pattern Recognition.* John Wiley & Sons, second edition, 2002.

[50] I.H. Witten and E. Frank. *Data Mining, Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers, second edition, 2005.

[51] K. Xu. How has the literature on Gini's index evolved in the past 80 years? Department of Economics at Dalhousie University working papers archive, Dalhousie, Department of Economics, June 2003.

[52] N. Ye. *The Handbook of Data Mining.* Lawrence Erlbaum Associates Publishers, 2003.

Appendix A

Bovine Ultrasound Image Data Set

A.1 Data set A

The feature values for data set A are described in Table A.1.

Table A.1: This table summarizes the complete feature values for data set A. A total of 23 heifers form this group with $n = 8$ heifers correspond to D3W1 class, $n = 7$ heifers correspond to D1W2 class and $n = 8$ heifers correspond to D≥17 class. The columns represent the different features values in millimeters used in this study: D represents the diameter of the dominant follicle, S1 is the first subordinate follicle, S2 is the second subordinate follicle, CL is the diameter of the corpus luteum and NF represents the number of follicles.

Feature Values for Data Set A						
Heifer	D (mm.)	S1 (mm.)	S2 (mm.)	CL (mm.)	NF (number)	Stage
Heifer 1	11.5	9.0	7.0	17.0	23	D3W1
Heifer 2	12.0	8.0	6.0	18.0	23	D3W1
Heifer 3	12.0	8.0	7.0	16.0	18	D3W1
Heifer 4	10.0	7.0	6.0	11.0	18	D3W1
Heifer 5	11.0	9.0	7.0	11.0	18	D3W1
Heifer 6	10.0	9.5	8.0	15.0	19	D3W1
Heifer 7	7.0	5.0	5.0	13.0	21	D3W1
Heifer 8	10.5	9.0	9.0	13.0	13	D3W1
Heifer 9	15.0	8.5	6.0	22.0	15	D1W2
Heifer 10	20.0	6.0	5.0	24.0	12	D1W2
Heifer 11	14.0	7.0	6.0	28.0	19	D1W2
Heifer 12	12.0	8.0	6.0	22.0	12	D1W2
Heifer 13	16.0	9.0	6.0	23.0	6	D1W2
Heifer 14	13.0	11.5	7.0	22.0	22	D1W2
Heifer 15	13.0	5.0	4.0	22.0	13	D1W2
Heifer 16	13.5	10.0	8.0	14.0	18	D≥17
Heifer 17	13.0	10.5	6.0	17.0	12	D≥17
Heifer 18	13.0	11.0	5.0	18.0	17	D≥17
Heifer 19	15.0	7.0	4.0	16.0	14	D≥17
Heifer 20	13.0	8.0	4.0	15.0	15	D≥17
Heifer 21	13.5	13.0	7.0	14.0	19	D≥17
Heifer 22	13.0	11.0	4.0	18.0	17	D≥17
Heifer 23	15.0	10.0	9.5	16.0	6	D≥17

The summary statistics of data set A are listed in Table A.2, containing the mean feature values and the standard deviation values.

Table A.2: Summary Statistics for data set A. This table lists the mean feature and standard deviation values for the 3 classes D3W1, D1W2 and D≥17 for the complete data set A formed by a total of 23 heifers.

Summary Statistics for Data Set A						
	D (mm.)	*S1* (mm.)	*S2* (mm.)	*CL* (mm.)	*NF* (number)	*Class*
Mean	10.50	8.06	6.87	14.25	19.12	D3W1
Std. Dev.	1.62	1.47	1.24	2.65	3.27	D3W1
Mean	14.71	7.85	5.71	23.28	14.14	D1W2
Std. Dev.	2.69	2.13	0.95	2.21	5.20	D1W2
Mean	13.62	10.06	5.93	16.00	14.75	D≥17
Std. Dev.	0.87	1.86	2.07	1.60	4.20	D≥17

A.2 Data set B

The feature values for data set B are described in Table A.3.

Table A.3: This table summarizes the complete feature values for data set B. A total of 22 heifers form this group with $n = 8$ heifers correspond to D3W1 class, $n = 6$ heifers correspond to D1W2 class and $n = 8$ heifers correspond to D≥17 class. The columns represent the different features values in millimeters used in this study: D represents the diameter of the dominant follicle, S1 is the first subordinate follicle, S2 is the second subordinate follicle, CL is the diameter of the corpus luteum and NF represents the number of follicles.

Feature Values for Data Set B						
Heifer	D (mm.)	S1 (mm.)	S2 (mm.)	CL (mm.)	NF (number)	Stage
Heifer 1	10.0	6.0	3.0	16.0	15	D3W1
Heifer 2	11.0	4.0	3.0	17.0	14	D3W1
Heifer 3	11.0	9.0	4.0	14.0	12	D3W1
Heifer 4	12.0	7.5	6.5	14.0	14	D3W1
Heifer 5	11.0	5.0	4.0	11.0	16	D3W1
Heifer 6	11.0	8.0	6.0	17.0	16	D3W1
Heifer 7	10.0	7.0	4.0	15.0	18	D3W1
Heifer 8	11.0	10.0	8.0	15.0	16	D3W1
Heifer 9	13.0	7.5	6.5	19.0	20	D1W2
Heifer 10	16.0	5.5	4.0	21.0	13	D1W2
Heifer 11	14.0	8.0	6.5	25.0	9	D1W2
Heifer 12	13.0	8.0	7.0	24.0	14	D1W2
Heifer 13	14.0	7.0	6.0	24.0	8	D1W2
Heifer 14	13.0	7.5	5.0	19.0	14	D1W2
Heifer 15	17.0	10.0	2.0	15.0	11	D≥17
Heifer 16	17.0	15.0	4.0	16.0	11	D≥17
Heifer 17	13.0	10.5	10.0	15.0	19	D≥17
Heifer 18	15.0	12.0	9.0	13.5	10	D≥17
Heifer 19	16.0	5.0	4.0	14.0	13	D≥17
Heifer 20	19.0	11.0	4.0	14.0	6	D≥17
Heifer 21	14.0	6.0	4.0	12.0	17	D≥17
Heifer 22	15.0	11.0	4.0	13.5	11	D≥17

The summary statistics of data set B are listed in Table A.4, containing the mean feature values and the standard deviation values.

Table A.4: Summary statistics for data set B. This table lists the mean feature and standard deviation values for the 3 classes D3W1, D1W2 and D≥17 for the complete data set B formed by a total of 22 heifers.

Summary Statistics for Data Set B						
	D (mm.)	*S1* (mm.)	*S2* (mm.)	*CL* (mm.)	*NF* (number)	*Class*
Mean	10.87	7.06	4.81	14.87	15.12	D3W1
Std. Dev.	0.64	2.00	1.81	1.95	1.80	D3W1
Mean	13.83	7.25	5.83	22.00	13.00	D1W2
Std. Dev.	1.16	0.93	1.12	2.68	4.28	D1W2
Mean	15.75	10.06	5.12	14.12	12.25	D≥17
Std. Dev.	1.90	3.21	2.79	1.21	4.09	D≥17

Appendix B

Experiment Results

B.1 Cross validation results

Data values from data set A and data set B for the cross validation evaluation are listed in Table B.1.

Table B.1: This table summarizes the complete feature values for data set A and data set B. Both data sets were sorted randomly over all classes for the cross-validation evaluation. A total of 45 heifers form this group with $n = 16$ heifers correspond to D3W1 class, $n = 13$ heifers correspond to D1W2 class and $n = 16$ heifers correspond to D≥17 class.

Feature Values for $A \cup B$ for Cross Validation						
Subset	**D (mm.)**	**S1 (mm.)**	**S2 (mm.)**	**CL (mm.)**	**NF (number)**	**Stage**
Subset 1	15.0	11.0	4.0	13.5	11	D≥17
Subset 1	19.0	11.0	4.0	14.0	6	D≥17
Subset 1	10.5	9.0	9.0	13.0	13	D3W1
Subset 1	12.0	8.0	6.0	18.0	23	D3W1
Subset 1	16.0	9.0	6.0	23.0	6	D1W2
Subset 1	12.0	8.0	6.0	22.0	12	D1W2
Subset 1	13.0	8.0	4.0	15.0	15	D≥17
Subset 1	10.0	6.0	3.0	16.0	15	D3W1
Subset 1	13.0	5.0	4.0	22.0	13	D1W2
Subset 2	13.0	11.5	7.0	22.0	22	D1W2
Subset 2	14.0	6.0	4.0	12.0	17	D≥17
Subset 2	10.0	7.0	4.0	15.0	18	D3W1
Subset 2	13.5	10.0	8.0	14.0	18	D≥17
Subset 2	13.0	10.5	6.0	17.0	12	D≥17
Subset 2	10.0	9.5	8.0	15.0	19	D3W1
Subset 2	16.0	5.5	4.0	21.0	13	D1W2
Subset 2	16.0	5.0	4.0	14.0	13	D≥17
Subset 2	11.0	9.0	7.0	11.0	18	D3W1
Subset 3	14.0	7.0	6.0	24.0	8	D1W2
Subset 3	17.0	15.0	4.0	16.0	11	D≥17
Subset 3	15.0	12.0	9.0	13.5	10	D≥17
Subset 3	12.0	8.0	7.0	16.0	18	D3W1
Subset 3	13.0	11.0	4.0	18.0	17	D≥17
Subset 3	15.0	7.0	4.0	16.0	14	D≥17
Subset 3	11.0	4.0	3.0	17.0	14	D3W1
Subset 3	11.0	8.0	6.0	17.0	16	D3W1
Subset 3	13.5	13.0	7.0	14.0	19	D≥17
Subset 4	14.0	7.0	6.0	28.0	19	D1W2
Subset 4	13.0	7.5	5.0	19.0	14	D1W2
Subset 4	14.0	8.0	6.5	25.0	9	D1W2

Continue on next page

Table B.1 – continue from previous page

Subset	**D (mm.)**	**S1 (mm.)**	**S2 (mm.)**	**CL (mm.)**	**NF (number)**	**Stage**
Subset 4	11.0	9.0	4.0	14.0	12	D3W1
Subset 4	15.0	8.5	6.0	22.0	15	D1W2
Subset 4	12.0	7.5	6.5	14.0	14	D3W1
Subset 4	15.0	10.0	9.5	16.0	6	D≥17
Subset 4	7.0	5.0	5.0	13.0	21	D3W1
Subset 4	20.0	6.0	5.0	24.0	12	D1W2
Subset 5	13.0	11.0	5.0	18.0	17	D≥17
Subset 5	13.0	8.0	7.0	24.0	14	D1W2
Subset 5	10.0	7.0	6.0	11.0	18	D3W1
Subset 5	11.0	10.0	8.0	15.0	16	D3W1
Subset 5	17.0	10.0	2.0	15.0	11	D≥17
Subset 5	11.0	5.0	4.0	11.0	16	D3W1
Subset 5	13.0	7.5	6.5	19.0	20	D1W2
Subset 5	13.0	10.5	10.0	15.0	19	D≥17
Subset 5	11.5	9.0	7.0	17.0	23	D3W1

Table B.2: Results from experiment 2. Confusion matrix resulting from the classification using the cross validation technique by the decision tree classifier. The complete 45 patterns were eventually used for training and testing, the classification rate was 100%.

Decision Tree Confusion Matrix Combined				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**16**	0	0	16
D1W2	0	**13**	0	13
D≥17	0	0	**16**	16

Table B.3: Results from experiment 2. Confusion matrix resulting from the classification using the cross validation technique by the naïve Bayes classifier. The complete 45 patterns were eventually used for training and testing. The classification rate was 95.55%.

Naïve Bayes Confusion Matrix Combined				
Classified as:	D3W1	D1W2	D≥17	Total
D3W1	**15**	0	1	16
D1W2	0	**13**	0	13
D≥17	1	0	**15**	16

Table B.4: Summary statistics for data set A and data set B used for cross validation. This table lists the mean feature and standard deviation values for the 3 classes D3W1, D1W2 and D≥17 for the complete data set used in this project $A \cup B$.

Summary Statistics for $A \cup B$						
	D (mm.)	*S1* (mm.)	*S2* (mm.)	*CL* (mm.)	*NF* (number)	*Class*
Mean	10.68	7.56	5.84	14.56	17.12	D3W1
Std. Dev.	1.2	1.77	1.84	2.27	3.28	D3W1
Mean	14.30	7.57	5.76	22.69	13.61	D1W2
Std. Dev.	2.09	1.65	0.99	2.42	4.64	D1W2
Mean	14.68	10.06	5.53	15.06	13.5	D≥17
Std. Dev.	1.8	2.53	2.41	1.68	4.21	D≥17