An investigation into the use of machine learning for determining oestrus in cows

R. Scott Mitchell[1], Robert A. Sherlock[2] and Lloyd A. Smith[1]

[1] Department of Computer Science, University of Waikato, Hamilton, New Zealand.

{rsm1,las}@waikato.ac.nz

[2] Dairying Research Corporation Limited, Hamilton, New Zealand. sherlockr@ruakura.cri.nz

Address for correspondence:

Dr. Lloyd Smith

Department of Computer Science

University of Waikato

Private Bag 3105

Hamilton 2001

NEW ZEALAND

Abstract

A preliminary investigation of the application of two well-known machine learning schemes—C4.5 and FOIL—to detection of oestrus in dairy cows has been made. This is a problem of practical economic significance as each missed opportunity for artificial insemination results in 21 days lost milk production. Classifications were made on normalised deviations of milk volume production and milking order time series data. The best learning scheme was C4.5 which was able to detect 69% of oestrus events, albeit with an unacceptably high rate of "false positives" (74%). Several directions for further work and improvements are identified.

Keywords

Machine learning, oestrus detection, dairy cow.

1. Introduction

This paper describes research aimed at producing a software program to assist dairy farmers in determining when cows are in oestrus, an important practical consideration when artificial insemination is used. Traditionally, oestrus detection relies mainly on visual observation of animal behaviour: a cow in oestrus will stand and allow herself to be mounted by other cows. Such events are readily noted when they take place amongst cows assembled for milking, but may be missed entirely in herds of free-grazing animals which are only brought in for milking twice a day (such as is usual in New Zealand). A widely used technique to increase the detection success rate under these conditions is the use of "tail-paint" which is rubbed off when the cow is mounted. However, regular visual inspection (and re-painting) is labour intensive, particularly in large herds.

With the introduction of milking systems with automatic recording of individual animal production at every milking, there is a possibility of using this data to at least assist in identifying oestrus. Some cows exhibit a characteristic pattern at oestrus of a significantly reduced milk volume followed by a compensatory higher volume at the following milking. Also, in some cases, cows who usually have a well-defined position in the milking order will present themselves for milking well out of that sequence. Such effects are clearly visible in the raw data for only around 5–10% of animals, but this gives some grounds for hope that a machine recognition method may be able to increase this figure to a more useful value. Such an algorithm could be run automatically on a suitable herd database, thus not involving additional farmer time. Even modest success rates for identifying cows in oestrus would be a useful adjunct to traditional methods as long as the number of false positives was not excessive (although in NZ seasonal dairying the dollar cost of failing to get a cow pregnant by missing an oestrus far exceeds that of a "wasted" insemination).

The goal of this research was to investigate the application of two established machine learning methods to this task: C4.5 (Quinlan, 1992) and FOIL (Quinlan, 1990). These methods were chosen because they are well understood among machine learning researchers and practitioners and because they are readily available as part of the WEKA machine learning workbench (Holmes et al., 1994; McQueen et al., 1994a). Milk yield data collected from a Dairying Research Corporation (DRC) herd of 120 cows at the Ruakura Agricultural Research Station for the 1993–94 milking season were used. This body of data was used to answer the following research questions: First, is it possible to determine oestrus events for all cows from milk production and milking order? Second, when an oestrus event is determined by tail paint, is it possible to use these data to more precisely determine the actual time of the oestrus? In the case of the Ruakura research herd, tail paint is observed once weekly, so an oestrus event determined by tail paint may have happened during any of the preceding seven days. If a program can determine the actual day of the oestrus, that will enable the farmer to more closely anticipate the date of the subsequent oestrus.

This study was undertaken as part of the WEKA machine learning project (McQueen et al., 1994a; McQueen et al., 1994b; Mitchell, 1995; De War et al., 1994) at the University of Waikato (WEKA is an acronym for the Waikato Environment for Knowledge Analysis). One of the primary objectives of the WEKA project is to investigate the application of machine learning techniques to problems in agriculture. The project has developed a software system—the WEKA workbench—that integrates a wide range of machine learning algorithms and support tools into a single interactive package (McQueen et al., 1994a; Holmes et al., 1994). The workbench allows the same data to be analysed by many different learning systems and for the results of these analyses to be evaluated in a consistent manner.

2. Machine Learning

Research in the field of machine learning (ML) has been ongoing for a number of years, with the earliest successful algorithms such as ID3 (Quinlan, 1986) developed more than a decade ago. Recently however the advent of more powerful computer systems, and more difficult data analysis problems, has seen a resurgence of interest in ML. This section provides a brief introduction to ML terminology, and to the algorithms and tools used in this study.

Machine learning is the term used to encompass a wide variety of techniques used for the discovery of patterns and relationships in sets of data. The fundamental goal of any machine learning algorithm is to discover meaningful or non-trivial relationships in a set of "training" data and produce a generalisation of these relationships that can be used to interpret new, unseen "test" data. Michie (1991) defines the learning process as follows:

> *"A learning system uses sample data to generate an updated basis for improved classification of subsequent data from the same source, and expresses the new basis in intelligible symbolic form".*

The output of a learning scheme is thus some form of structural description of a dataset, acquired from examples of that data (McQueen et al., 1994a). These descriptions encapsulate the "knowledge" learned by the system and can be represented in different ways. Schemes such as genetic algorithms (Goldberg, 1989), and neural networks generate implicit internal models of the data which are not easily understood by human beings or other machines. In this study we wish to know if the descriptions schemes are consistent with at least some existing human knowledge of the problem domain. This makes the use of schemes such as neural networks impractical. In contrast, the ML algorithms used in this study—C4.5 and FOIL—generate descriptions that can be more readily interpreted by human users.

Figure 1 gives an example of a very simple ML dataset, in the format used by the Waikato machine learning group. The name of the relation, "golf", is given, followed by the names and descriptions of the five attributes in this dataset. For discrete variables such as "outlook" the allowed values are listed, while for continuous—integer or real valued—attributes such as "temperature" a range can be specified if desired. The final attribute, "class", is the classification for this data. It tells us whether or not we should play golf, based on the values of the other four attributes. The attribute section is ended by the "@data" tag and this is followed by the examples. Each examples occupies a single line and contains a value for each attribute, in the order they were listed previously. The goal of a machine learning system is to induce a generalised description or explanation of the class variable—i.e., should we play golf today?—from the examples of the concept given in the dataset.

2.1 C4.5

C4.5 (Quinlan, 1992) is a system for inducing production rules and decision trees from a set of examples. C4.5 uses a “top-down” or “divide-and-conquer” approach to building decision trees. To begin, the set of training examples is partitioned into two or more subsets based on the outcome of a test on the value of a single attribute. The particular test is chosen by an information-theoretic heuristic (the *gain*-ratio criterion) that generally gives close to the optimal partitioning. This procedure is repeated on each of the new subsets and continues until a subset contains only examples of a single class or the partitioning tree has reached a predetermined maximum depth. Figure 2 shows the decision tree induced by C4.5 from the golf data in Figure 1. The tree can be used to classify a test case by starting at the top node of the tree and following the branches corresponding to the attribute values in the test case, until a leaf node is reached. Comparison with the raw golf data will reveal that this tree classifies all of the training cases correctly.

Much of C4.5 is derived from Quinlan’s earlier induction system, ID3 (Quinlan, 1986). The basic ID3 algorithm has been tested and modified by numerous researchers since its invention (Mingers, 1989a; Mingers, 1989b; Utgoff, 1989). C4.5 adds several new and interesting features including: a new criterion for determining the best partitioning of the examples at each decision tree node; “pruned” decision trees, to reduce the chances of overfitting the data; the ability to derive production rules from the unpruned decision tree. These rules are comparable in accuracy to the pruned decision tree, but are more easily interpreted by people.

2.2 FOIL

FOIL (First Order Inductive Learner) is another scheme developed by J. R. Quinlan at the University of Sydney (Quinlan, 1990). It builds on concepts found in ID3 and Michalski’s AQ11 (Michalski and

Chilausky, 1980) to generate descriptions of logical relations using a subset of first-order logic. FOIL analyses a set of positive and negative examples of some relation and produces a structural description of the relation expressed as a set of Horn clauses. In contrast to C4.5, FOIL is a “bottom-up” classifier. Rules are induced for each class, one class at a time. For each class, the training examples are first divided in “positive”—those from the class in question—and “negative”—those from all other classes—cases. FOIL then attempts to find a set of clauses that cover (match) some positive examples but no negative examples. The matching positive examples are removed, and the process repeats until no positive examples remain. A similar procedure is followed for each class. The end result is a set of relational rules for each class in the data. Figure 3 shows the FOIL rules induced from the golf data in Figure 1. Like the C4.5 decision tree in Figure 2, these rules classify all of the training examples correctly, although different attributes and values have been chosen by FOIL.

An important feature of FOIL is its ability to express relationships between the attributes in an example. Zero-order classifiers such as C4.5 can only compare attribute values with constant numbers or symbols, not other attributes. We expected that this feature would be useful when classifying examples representing a time-sequence of data.

C4.5 and FOIL are both examples of “similarity-based” learning schemes, sometimes referred to as learning from examples. These algorithms use only the observed similarities and differences between examples in order to form generalisations (McQueen et al., 1994a). A similarity based learner is “trained” using a set of example data that have been partitioned into two or more *classes* or *concepts*. Each example consists of a set of numerical or symbolic attribute-value pairs. The learner builds a structural description of the concepts encapsulated within the training data. The concept descriptions can then be used to classify a set of unseen test data. If the classification of the test data is known to the user but not to the learning algorithm, the performance of the descriptions with respect to the test examples provides an important measure of the quality of the descriptions.

3. Methodology

Experimental work was carried out using a database of dairy cow milking records created by one of the authors. The database tracks the per-milking performance of 130 identified cows in a DRC research dairy herd over the first few months of the 1993–94 milking season. Experiments were designed to investigate the application of similarity-based learning methods to the recognition and detection of oestrus events.

With most real-world data a significant amount of pre-processing is necessary before the data can be presented to a machine learning scheme. Typical pre-processing steps include the "cleaning" of noisy, anomalous or missing data; selection of useful or interesting attributes—with the possibility of generating new, derived attributes; and generating examples in a form suitable for the particular learning algorithm. Procedures used in this work are described below.

3.1 The WEKA Prolog Evaluator

One of the most important features of the WEKA Workbench is that it allows many different schemes to be run on the same set of data and for the output of each scheme to be evaluated in a consistent fashion (McQueen et al., 1994a). The rules or decision tree produced by a learning scheme are translated into an equivalent Prolog representation and evaluated with respect to the training and test data sets. For each rule or decision tree leaf the evaluator indicates how often the rule or leaf is used, and how many examples it classifies correctly and incorrectly. The evaluator also provides a summary showing how many examples were classified correctly, classified incorrectly, classified into multiple classes or not classified at all. Many schemes—including FOIL—have only minimal internal evaluation methods so the Prolog evaluator is immediately useful for analysing the output from these schemes. Additionally it is a valuable tool for evaluating the performance of different schemes on "neutral ground".

3.2 Generation of Examples

The raw database has a fundamental temporal component, in that the milking data for each cow has been sampled at particular points along a time interval. Thus the data can be considered to be three-dimensional, with a *time* axis in addition to the usual attributes and records. In contrast, input to most machine learning schemes consists of a set of *examples* describing one or more concepts—each example is a set of attribute-value pairs that completely describe a single object in the problem domain.

In the case of the cow milking data, the concepts we are attempting to learn are “in oestrus” or “not in oestrus” for a particular cow on a particular day. In order for the learning system to *predict* oestrus events, it should ideally be trained using time-sequenced series’ of data that lead up to the actual event. To achieve this using a similarity-based learner, a series of milkings for a particular cow must be combined into a single example for presentation to the learning scheme.

Examples were generated from the cow milking database using a “sliding window” approach, illustrated in Figure 4. The series of milking records for each cow was split into a sequence of overlapping “windows”, with the contents of each window becoming a single example. Window sizes of three, five and ten days were used for different experiments. The examples were initially classified as *positive* if an oestrus event had been observed on the final milking in a window, or as *negative* otherwise. This simple classification scheme was modified somewhat for individual experiments as described below.

3.3 Attribute Selection

A fundamental decision that must be made in any experiment of this nature is to determine exactly which attributes or variables should be included in the data given to a machine learning scheme. The attribute selection includes variables present in the raw database and “derived” attributes generated from existing variables. The herd milking data is essentially a relational database, with tables storing various summary and statistical information as well as the individual detailed milking records for each cow. Nearly all machine learning schemes are designed to work with a single flat table of data, thus producing input for a learning system will almost certainly involve some “flattening” of the database (McQueen et al., 1994b). It is important that as little information as possible is lost by this process.

Initially, the only information available to guide the attribute selection process was the domain knowledge of the DRC researchers. Preliminary investigations indicated that a proportion of cows exhibit a characteristic pattern of significantly reduced milk volume at oestrus, with a compensating higher volume at the following milking. It was also observed that cows who usually take a well defined position in the milking order may present themselves out of their normal sequence during oestrus. While these effects were only visually apparent in the raw data for a small percentage of cows, it was hoped that machine learning methods would be able to detect them and possible other useful patterns for most of the herd.

The database records twice-daily milk volumes both as absolute values and as percentage deviations from the herd mean volume for each milking. This latter achieves the important effect of removing the daily variability—typically around 10%—of yield from grazed herds due to changing pasture and environmental conditions (Sherlock and Woolford, 1992). A running mean and standard deviation of the volume deviation is also maintained for each cow. The standard error for individual yield measurements is of the order of 7%. Milking order is stored as a percentage ranking within the herd, with a running mean and standard deviation as for the volume measure. The volume deviation and its running mean and standard deviation are all computed and recorded relative to the overall herd mean for each milking. For the remainder of the paper these variables will be referred to as the “volume deviation”, “volume running mean” and “volume running standard deviation” for a given cow.

Farmer-detected oestrus events are indicated by a simple yes/no field—the FDO (Farmer Detected Oestrus) flag—for each milking. These are data are obtained from direct observations of cow behaviour by the milkers and weekly checks of “tail paint”. The database initially made no distinction between direct observations and tail paint cases. Additional data collected by milking shed technicians was used to construct a new database field that indicated which FDO flags corresponded to tail paint observations and which were direct observations. Since tail paint observations are by definition only accurate to within seven days, these observations were *not* used in the experiments described below unless explicitly stated.

Attributes were chosen so as to minimise the effect of global differences between cows. For example, a particular cow may give consistently above-average milk yields, while another is consistently below-average. However, both cows may exhibit a similar pattern of *changes* in their milk yield during oestrus. Ideally, both of these cows would be treated similarly by the learning scheme regardless of the difference in their average milk yields.

In Experiments 1–3 we considered the difference (delta) between the volume deviation observed for a particular cow, and the volume running mean for the same cow. These deltas were used both as absolute values and normalised using the running standard deviation. Deltas were calculated similarly for the milking order data.

Table 1 shows some example data for an oestrus event. The deltas for the volume deviation have been computed and are shown as absolute values and normalised. Figure 5 and Figure 6 show the raw volume deviation, running mean and deltas in graphical form.

The final attribute used was the number of days since the last FDO flag event for the cow. Since normal oestrus occurs in a regular 18–24 day cycle this was expected to be a useful variable for determining approximately when the next event should occur.

3.4 Detection of oestrus events

The first series of experiments investigated the use of the machine learning schemes to detect the occurrence of oestrus events in the milking data. The set of raw examples generated using the sliding window method was divided into training and test sets for presentation to the learning schemes. The set was randomly split, with 66% of the examples placed into the training set and the remaining 34% in the test set. Each experiment was run several times with different random divisions of the data to minimise the effect of any single anomalous run on the results.

There were several obvious cases where an oestrus event was missing—that is, oestrus had occurred but was not entered into the database. In order to avoid accidentally treating these missed positive examples as negative, only those examples contained within an "island" around observed oestrus events were used in experimental runs. Typically the examples were restricted to those within eight days on either side of an FDO flag.

The FDO flag in the milking database was set on the basis of behavioural observations made by the farmer at each milking. A cow is in oestrus, however, over a period of time and can be inseminated at any time during that period. An "oestrus event" determined by milk production variation can precede or lag changes in the cow's behaviour—and hence the FDO flag—by as much as 24 hours. Thus it is possible that some oestrus events may be entered into the training set as *negative* examples if the location of the event does not exactly match the database flag. This weakens the definition of a positive example seen by the learner and will lead to a less accurate classification. We experimented with two different approaches to resolve this problem:

1. In Experiment 1, examples falling on the same day as the FDO flag were set as positive examples, but those on the day before and day after the flag were removed from the training set. This meant that any examples trained as negative had to lie more than 24 hours away from the oestrus flag. This reduces the number of positive examples trained as negative, but has the unwanted effect that some positive examples will not be present in the training set.

2. In Experiments 2 and 3, the milkings within 24 hours of each set FDO flag were examined in an attempt to determine exactly where the event had occurred. Milkings where the volume deviation lay more than some number of (running) standard deviations away from the running mean were considered to indicate an oestrus event. These events were entered in the training set as positive examples. Note that the FDO flag events indicated in the raw database were still

treated as positive examples in this experiment. The overall effect was to “spread out” the oestrus observation in cases where its location could not be determined exactly.

Note that all of the examples in the test set remained as originally classified in the database. Figure 7 shows how the examples from the sample event shown earlier would have been classified under the scheme used in Experiments 2 and 3.

3.5 Determining the location of “tail paint” events

“Tail painting” is a technique widely used in New Zealand dairy herds to increase the detection rate of oestrus events that might otherwise go unobserved. However, the regular visual inspection required to maintain accurate records is time consuming and error prone, particularly for large herds. In addition, since the DRC herd tail paintings are only checked once weekly—unlike normal farm practice—these data are only accurate to within the seven days immediately preceding an observation. It is desirable to be able to determine exactly when the oestrus corresponding to a tail paint observation occurred, as this information can be used to help predict more accurately the time of the next oestrus.

The test set for this experiment was constructed entirely from the tail paint oestrus observations in the milking data. Each example consisted of milkings from the six days preceding the tail paint flag, and the day of the flag itself. The actual oestrus event could have occurred at any time during this seven day window.

The training examples were generated from the more accurate direct observations of oestrus. To allow for the possible offset of the actual oestrus from the behavioural observation, the milkings within 24 hours of the observation were examined to determine the most likely location of the event. The milking with the largest difference between the volume deviation and volume running mean was taken to be the actual location of the oestrus. For each observation seven examples were generated using a modification of the sliding window procedure, with a window size of seven days. The examples were arranged so that the most likely oestrus was placed in a different position in each example. This is illustrated in Figure 8. In this way the learning system was trained on examples of oestrus occurring at all times from zero to six days previously. A tail paint observation records a similar situation, where the actual event occurred some time in the past. Thus the concept descriptions learned from the training data should be capable of identifying the true location of the oestrus event in a tail paint observation.

A number of learning runs were also performed using real oestrus events to simulate tail-paint cases. The first oestrus of the season for each cow was used to generate the training data. A series of examples was generated for each event as described earlier. The set of *second* oestrus events was then used to construct the training set. Each test case was generated from a seven day window covering the period of 18–24 days from the first FDO flag for that cow, the period when the second oestrus is most likely to occur. The output of the learning runs again indicated the most likely day for the actual event within this window. The advantage of using simulated tail-paint cases is that the true location of the event is known in most cases and this can be directly compared with the prediction made by the learner, whereas the output of tests using real tail paint cases can only be evaluated by a human expert comparing the classifications with the raw data.

4. Results

This section presents results for each of the experiments described above. Results are included for experiments in both detecting all oestrus events and locating “tail-paint” events. A discussion of these results can be found in the next section of the paper.

The first three experiments described here investigated the detection of oestrus events over the entire herd, as described earlier. The differences between these experiments are in the way the dataset was constructed from the raw database, and in the classification of the examples. All of the results presented are averages over 10 learning runs, each using a different split of the database into training and test sets.

Three columns of values are given for each experiment. These are:

%Correct. The percentage of all examples in the dataset classified correctly by the learner.

%False Positive. The percentage of examples classified as "positive" by the learner that were actually *negative* examples. That is, the proportion of oestrus events predicted by the learner that do not correspond to actual events.

%Correct Positive. The percentage of all positive examples in the dataset that were correctly classified by the learner. That is, the proportion of all the oestrus events in the dataset that were detected by the learning algorithm.

4.1 Experiment 1

In this experiment training examples were generated using a 3-day sliding window, and classified as positive if an oestrus event occurred on the last day of the window. The examples were restricted to within eight days of an oestrus event to avoid overlapping sequences from different events. Training examples within 24 hours of an oestrus event were removed from the dataset. The attributes used were the volume deviation from the running mean, the milking order percentage and the number of days since the previous oestrus.

Table 3 gives the results for this experiment. Results are given for pruned C4.5 trees, C4.5 rules and FOIL rules, for both the training and test datasets, averaged over 10 runs.

4.2 Experiment 2

This experiment was almost identical to the previous, except for the treatment of examples within 24 hours of an oestrus event. In the training set, these examples were considered to be positive if the volume deviation was more than one standard deviation away from the running mean, otherwise they were classified as negative. In testing, any positive classification within 24 hours of an event was treated as a correct classification. Table 4 gives the results for this experiment, averaged over 10 runs.

4.3 Experiment 3

Examples were generated using a 5-day sliding window, restricted to examples within eight days of an oestrus event. Examples within 24 hours of an event were treated as positive (for the training set only) if the volume deviation was more than *two* standard deviations away from the running mean. Attributes used were the volume deviation and milking order deltas from their respective running means; the volume and milking order deltas normalised using their running standard deviations and the number of milkings since the last oestrus event.

This experiment made use of the "tail paint" cases for the first time, to generate additional examples in the *test* set only. A tail paint event indicates that oestrus has occurred sometime during the preceding week. We wanted to determine if the learner was able to identify the exact location (from seven possible days) of the event. The sliding window for the tail paint examples extended backwards in time from the tail paint indication for seven days, so as to cover the entire range when the event could have occurred. The seven examples generated for each tail paint event were placed in the test set as negative examples. The classifications made by the learner on these examples were examined to see if any had been classified as positive—indicating that the learner considered an oestrus event to have occurred at that time. This idea is explored in more detail in the next experiment. Table 5 shows the results for this experiment, averaged over 10 runs.

4.4 Experiment 4

The remaining experiment was concerned with determining the actual location of tail-paint events. Training and test datasets were generated using the procedures described previously. Since the training and test sets were taken from separate parts of the database, there was no need to randomly split the examples as was done in previous experiments.

Attributes identical to those from Experiment 3 were used: volume deviation and order differences from the running mean; the number of standard deviations from the running mean; and the number of days since the last observed oestrus event.

For the learning runs using actual tail paint cases as test data, the output of the learner indicated the most likely day—according to the learning scheme—for the actual oestrus event corresponding to each tail-paint indicator. These predictions were compared to the raw database *by eye* to determine how accurately the learner had performed in each case, i.e. a human judgement was made of the appropriateness of the machine prediction.

In the case of runs using simulated tail paint cases, the classifications made by the learner can be compared directly with the known location of the FDO flag. The results of these runs are shown in Table 6. This table shows the number of classifications made by the learner that were within a given time interval from the location of the oestrus flag. These values are expressed as absolute counts and as a percentage of the test data.

5. Discussion

5.1 Comparison of Learning Schemes

At first glance C4.5 and FOIL both appear to have performed very well in their classification of the herd milking data. The *%Correct* figure in Table 3–Table 5 is consistently in the range of 80–100%. However, these figures do not take into account either the distribution or the relative importance of the classes in the data.

In an oestrus detection application the interesting events occur only very infrequently, approximately once every 21 days for a typical cow. Since the positive events then make up only around 5% of the data, it is possible for a naïve learner to achieve 95% accuracy across the whole dataset merely by classifying all examples as negative. While such a scheme will appear to perform very well, it is clearly useless for detecting the events we are interested in. Therefore, the classification accuracy for positive examples is a far more important measure of performance in the current application.

Ideally the learning scheme should correctly identify all positive examples whilst not misclassifying any negative examples as positive. In practice misclassifications will occur and our objective is then to minimise the number of false positive classifications, so that the majority of positive classifications made by the learner correspond to actual positive instances. This allows for some positive examples to be incorrectly classified as negative, but it is possible that these events could still be detected by other methods.

With respect to the performance of the different learning schemes, in general the rules generated by C4.5 have performed better than either the FOIL rules or C4.5 decision trees. In each of Experiments 1–3 the C4.5 rules have both the lowest false positive and highest correct positive rates on the test datasets. On training data the C4.5 rules show consistently better accuracy than the corresponding decision trees, although the best performance on this data is given by the FOIL rules.

Most of the difference in classification accuracy between schemes can be attributed to differences in pruning strategy. As Tables 1–3 show, FOIL's rules cover the training data extremely well, with less than 5% false positives and over 85% correct positives in most cases. However, many of the rules formed in training will be very specific to small groups of examples, especially considering the low frequency of positive instances. If these rules are then applied directly to test cases, it is unlikely that sufficiently similar instances will be encountered for many of these specific rules to be used. Test accuracy will suffer as a result—this can be clearly seen in the test results for FOIL in Experiments 1–3. The pruning process attempts to reduce this problem by generalising the classification— "pruning out" parts of the rules or tree that are overly specific. This leads to descriptions that generally perform better on unseen test cases, at the expense of some accuracy on the training set.

FOIL performs no pruning on its rules whatsoever; thus while it has shown good results in training the performance of the rules on test data is often quite poor. C4.5 uses a complex pruning algorithm on both rules and decision trees. The effect of this pruning can be clearly seen in the results—the C4.5 trees and rules have outperformed FOIL on the test data in almost all cases.

Table 7 summarises the sizes of the descriptions generated by each algorithm, averaged over 10 runs. Decision tree size is measured as the total number of interior nodes and leaves in the tree. The size of a rule set is the sum of the number of terms in each of the rules.

The rulesets produced by C4.5 were usually much smaller than the corresponding decision trees from the same data. On average, pruned decision trees contained a total of 165 leaves and interior nodes, with the rulesets having an average of 51 terms. A smaller description indicates that "heavier" pruning has been carried out, and in this case the resulting increase in accuracy is quite significant. Obviously pruning can only improve the accuracy of a classification up to a point—if the descriptions are pruned too much they will become overly general and accuracy will begin to decrease again. C4.5 allow the degree of pruned to be set when it is run, although the default setting is generally satisfactory for most applications.

Even with the positive effects of pruning, the results of Experiments 1–3 are rather disappointing. At best, we have nearly 70% of positive examples correctly classified, but this is unfortunately coupled with a large false positive rate close to 75%. It should be remembered that we are judging the performance of the learner based on the accuracy over only 5% of a quite noisy dataset, but still the results are far from adequate for a practical application. There are a number of likely contributing factors to this situation.

Firstly, the dataset may simply not contain enough information to classify more accurately. This situation might occur if there are insufficient examples, if the data were particularly noisy, or if an inappropriate set of attributes had been chosen. For most cows there are only two or three oestrus events and there is a significant amount of natural random variation present in the data which may be enough to mask or obscure important events in some cases.

Secondly, there may be sub-populations of cows within the herd that exhibit different patterns of behaviour at oestrus. For instance, the yield changes for some animals are clearly visible, but for most a visual inspection reveals little. C4.5 is able to describe more subtle or complex patterns than can be detected by eye, but only if a pattern does in fact exist. If there is no detectable pattern for some cows, or there are subgroups with different patterns, many positive examples will be indistinguishable from negative instances, with the result being an excessive false positive rate. The pattern of yield changes around an oestrus event may conform to some continuous distribution of there may be distinct, disjoint subgroups of cows with entirely different characteristics.

Thirdly, there are some features of the C4.5 and FOIL algorithms that may reduce their effectiveness on a dataset of this type. The biggest weakness of FOIL is its lack of any kind of pruning algorithm. FOIL rules generally perform extremely well on training data, and effective pruning could be expected to give high accuracy on test data. FOIL has the advantage of being able to form true first-order rules that compare the values of attributes *within* an instance, so in theory it is well suited to data with a temporal component. Unfortunately this feature is offset by the lack of pruning, as shown in the poor results on the herd milking test data.

Finally, the evaluation heuristic used by C4.5 to determine the best partitioning of cases at each tree node is known to be sub-optimal when presented with highly skewed data such as the milking database (Quinlan, 1992; Mingers 1989b; Mitchell, 1995). The technique used has a tendency to produce highly unequal splits of the examples where the small group—often only two or three instances—is purely of one class. Quinlan argues that this feature is desirable, since it often leads to smaller decision trees. Statistically however, these “small disjuncts” are most likely to be chance occurrences that are unreliable for predicting test cases. In a statistical sense, it is better to have larger groups, possibly containing examples from more than one class.

The evaluation heuristic is a fundamental part of the C4.5 algorithm, so a solution to this problem is not immediately obvious. Ting (1994) has proposed one method that appears to offer some hope. His composite learning method uses an instance-based learner (Aha et al., 1991) to classify the test cases in the small disjuncts. The standard C4.5 decision tree is used for the remaining cases. Both learners are trained in parallel in the same data. The advantage of this system is that the specific instances in the small disjunct are used to classify the test cases, instead of relying on the generalised—and possibly inaccurate—decision tree.

5.2 Practical Oestrus Prediction and Location

Figure 9 summarises the results of Experiment 4 in which the trained machine attempts to localise an oestrus event within a 14 day interval. Over 45% of the test set classifications made by C4.5 lie within ±1 day of the FDO flag location, with nearly 65% within ±2 days of the FDO flag. While obviously these figures could—in theory—be improved, this is nevertheless an encouraging result.

An interesting feature of Figure 9 is the secondary peak centered around an offset of 3–4 days before the FDO flag. This peak accounts for almost a further 20% of the test cases. A sudden sharp decrease in the level of progesterone is known to occur approximately 72 hours before oestrus, and it seems reasonable to conclude that the secondary peak observed in Figure 9 might be associated with this physiological event.

The classifications in Experiment 4 were made partly on the basis of the volume deviation delta from the running mean for each cow. Figure 10 shows the distribution of large changes in yield relative to the location of the FDO flag. For each FDO flag in the database, the largest yield difference from the running mean within one week either side of the flag was found and plotted in Figure 10. The columns were further divided to show the magnitudes of these largest deltas, relative to the running standard deviation at each point. The graph clearly shows that a significant proportion of the maximum deltas—in fact around 27%—fall on the same day as the corresponding FDO flag. Half of these cases had differences of more than three standard deviations from the running mean, with nearly 75% lying more than two standard deviations from the mean. As we move further away from the FDO flag the proportion of very large differences decreases, until at the edges of the graph it is likely that the maximum deviation is as much an indicator of random noise as any useful event.

Figure 10 shows a secondary peak at 72 hours before the FDO flag, similarly to Figure 9. This may indicate that the large changes in yield may also be associated with the pre-oestrus decrease in progesterone level. If this is true, many negative examples in our training sets would have included patterns similar to those found in the true positive examples, thus reducing the level of discrimination between positive and negative examples. If this effect could be taken advantage of, we might expect to see a reduction in the false positive rate from earlier experiments as well as improved accuracy in the identification of tail paint cases.

A preliminary analysis of the distribution of individual cows with respect to the data in Figure 10 did not produce any startling results, that is, the FDO flags for a given cow were not in general confined to any particular part of the graph. This lends weight to the idea that the changes in yield around oestrus time follow some continuous distribution—sometimes a particular cow will show the effect strongly and sometimes nothing will be detectable at all.

The graphs in Figure 9 and Figure 10 have sufficiently similar shapes that it is tempting to conclude that C4.5 is somehow learning the distribution of maximum yield deltas. It is likely that this is happening, since much of the training data comes directly from this distribution. Changes in yield do not always identify the oestrus event exactly, and C4.5 appears to have done as well as could be expected given this noisy data. It is possible that other attributes could be used, either in combination with, or instead of, the volume deltas to better identify the exact time of oestrus and improve classification accuracy.

6. Conclusions

While the performance of the learning schemes is not yet at a level suitable for a practical application, they have shown better accuracy than human experts on the same data. Only 5–10% of events are visible to the eye in the raw data, but C4.5 is able to identify nearly 70% of events, albeit with a high false positive rate. With “time localisation” of the event into a seven-day window, C4.5 can successfully place 65% of events to within 48 hours of their actual location. Although this is clearly far from the “perfect” learner, it is well on the way to becoming a useful practical aid. It should be noted that an eventual fielded system would require minimal effort from farmers—all of the necessary data is collected and analysed automatically.

It may be possible to improve learning accuracy by better training. For example, the current classifications based on the FDO flag could be replaced with an absolute physical measure such as milk progesterone level. Reducing the amount of noise in the data—by improving the measurement accuracy of the milking machinery—would certainly help, although it is hard to quantify this effect in terms of classification accuracy. Some of the other automatically measured variables may also be relevant to the classification, in particular the electrical conductivity of the milk. Further study is required to assess the relative contributions of the different variables to the learning process. A more thorough analysis of the rules and decision trees produced by the learning schemes would be useful in this respect.

Further work is also needed to investigate the nature of the distribution of cow performance variations around oestrus events. If this distribution proves to be too narrow, or is multi-modal, then we may have to abandon hope of a single learning strategy that can be applied to the entire herd. There is almost certainly not enough data to learn individual cow behaviours, but there may be a possibility of grouping together animals with similar behaviours. The current work should be extended to cover more farms over successive seasons, looking at the consistency of behaviours from different groups of cows of the same age on successive lactations.

Acknowledgements

This work is supported by the New Zealand Foundation for Research, Science and Technology. We gratefully acknowledge the assistance of Steve Garner and the other members of the Waikato machine learning group who have worked on the development of the WEKA machine workbench, and Peter Copeman for his assistance with the DRC herd records.

References

Aha, D.W., Kibler, D and Albert, M.K. (1991) Instance-based learning algorithms. Machine Learning, 6: 37–66.

Garner, S.R. (1994) ARFF—the WEKA dataset format. World Wide Web hypertext document at http://www.cs.waikato.ac.nz/ml/workbench/arff.html.

Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley Publishing Company Inc., Reading, MA, 412 pp.

Holmes, G., Donkin, A. and Witten, I.H. (1994) WEKA: A machine learning workbench. Working Paper 94/9, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

McQueen, R.J., Garner, S.R., Nevill-Manning, C.G. and Witten, I.H. (1994) Applying machine learning to agricultural data. Computers and Electronics in Agriculture, 12 :275–293.

McQueen, R.J., Neal, D., De War, R. and Garner, S.R. (1994) Preparing and processing relational data through the WEKA machine learning workbench. Working Paper, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

Michalski, R.S. and Chilausky, R.L. (1980) Learning by being told and from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. Int. J. Policy Anal. Info. Systems, 4: 125–161.

Michie, D. (1991) Methodologies from machine learning in data analysis and software. The Computer Journal, 34(6): 559–565.

Mingers, J. (1989) An empirical comparison of pruning methods for decision-tree induction. Machine Learning, 4: 227–243.

Mingers, J. (1989) An empirical comparison of selection measures for decision tree induction. Machine Learning, 3: 319–342.

Mitchell, R.S. (1995) The application of machine learning techniques to time-series data. Working Paper 95/15, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

Quinlan, J.R. (1986) Induction of decision trees. Machine Learning, 1: 81–106.

Quinlan, J.R. (1990) Learning logical relations from definitions. Machine Learning, 5: 239–266.

Quinlan, J.R. (1992) C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 302 pp.

Schlünsen, D., Roth, H., Schön, H., Paul, W. and Speckmann, H. (1987) Automatic health and oestrus control in dairy husbandry through computer aided systems. Journal of Agricultural Engineering Research, 38: 262–279.

Sherlock, R.A. and Woolford, M.W. (1992) Automated cow and machine performance monitoring in the Ruakura milk harvester. In: A.H. Ipema, A.C. Lippus, J.H.M. Metz and W. Rossing (Editors), Proceedings of the International Symposium on Prospects for Automatic Milking, November 1992, Wageningen, The Netherlands. Purdoc Scientific Publishers.

Ting, K.M. (1994) The problem of small disjuncts: its remedy in decision trees. In: R. Elio (Editor), Proceedings of the Tenth Canadian Conference on Artificial Intelligence, May 1994, Banff, Alberta. Canadian Society for Computational Studies of Intelligence, pp. 91–97.

Utgoff, P.E. (1989) Incremental induction of decision trees. Machine Learning, 4: 161–186.

De War, R. and Neal, D.L. (1994) WEKA machine learning project: Cow culling. Working Paper, 94/12, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

Table 1. Sample oestrus event showing volume deltas and standard deviations used to determine class.

Day, Time	1, AM	1, PM	2, AM	2, PM	3, AM	3, PM	4, AM	4, PM	5, AM	5, PM
Volume deviation (% of herd mean)	18.2	38.9	21.9	36.8	-48.1	161.7	36.6	55.4	22	28.1
Running mean	25.6	25.6	26.1	26.7	26.3	26.1	29.4	34.3	39.2	42.6
Running standard deviation	17.3	17.3	17.3	17.2	17.3	18	19.5	21.5	23.7	25.9
Delta from running mean, absolute	-7.4	13.3	-4.2	10.1	-74.4	135.6	7.2	21.1	-17.2	-14.5
Delta from running mean, normalised	-0.43	0.77	-0.24	0.59	-4.30	7.53	0.37	0.98	-0.73	-0.56

Table 2. Summary of Experiments 1–3.

Experiment	Window Size	Attributes	Comments
1	3 days	Volume deviation deltas from cow running mean. Milking order deltas from cow running mean. Time since last FDO flag.	Examples restricted to within 8 days either side of an FDO flag. Examples from days adjacent to FDO flag removed from training set.
2	3 days	As above	Examples restricted to within 8 days either side of an FDO flag. Examples from day either side of FDO flag reclassified as positive if volume deviation more than one standard deviation from running mean.
3	5 days	As above, but with normalised volume deltas (number of standard deviations from running mean	As above, except examples re-classified if volume deviation more than two standard deviations from running mean.

Table 3. Classification results for Experiment 1.

Dataset	% Correct	% False positive	% Correct positive
Training			
C4.5 pruned trees	96.7	43.4	9.7
C4.5 rules	96.6	27.5	42.0
FOIL rules	98.9	3.4	37.7
Testing			
C4.5 pruned trees	96.5	62.6	7.8
C4.5 rules	96.3	34.7	41.0
FOIL rules	96.8	75.0	3.3

Table 4. Classification results for Experiment 2.

Dataset	% Correct	% False positive	% Correct positive
Training			
C4.5 pruned trees	93.4	13.1	30.0
C4.5 rules	93.5	8.2	50.6
FOIL rules	99.3	0.1	86.3
Testing			
C4.5 pruned trees	90.6	59.5	14.0
C4.5 rules	91.9	20.1	44.1
FOIL rules	92.0	65.9	5.8

Table 5. Classification results for Experiment 3.

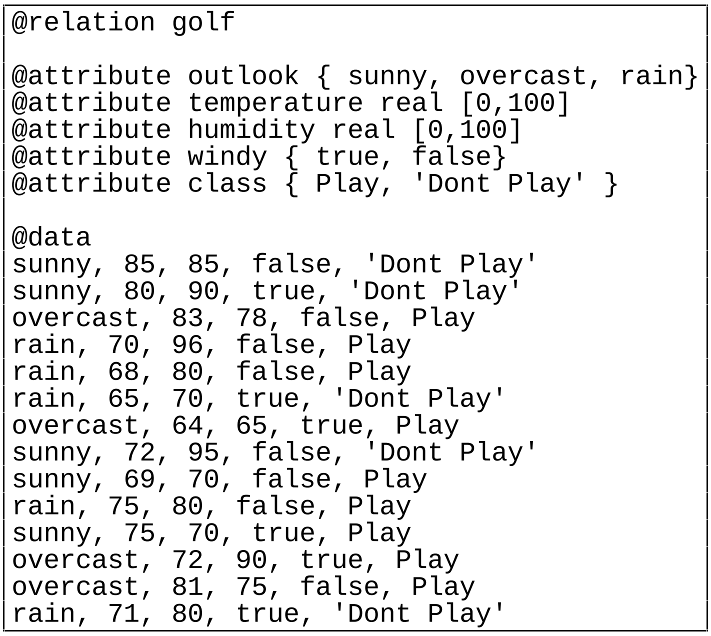
Dataset	% Correct	% False positive	% Correct positive
Training			
C4.5 pruned trees	91.6	35.7	64.2
C4.5 rules	85.3	48.8	66.3
FOIL rules	99.6	0	94.9
Testing			
C4.5 pruned trees	87.8	76.9	48.8
C4.5 rules	82.2	73.7	68.7
FOIL rules	96.3	68.0	20.0

Table 6. Classification results for Experiment 4—simulated tail paint examples.

Offset (days)	Count	Percentage
-6	2	2.4
-5	5	6.0
-4	8	9.5
-3	8	9.5
-2	5	6.0
-1	11	13.1
0	14	16.7
1	13	15.5
2	10	12.0
3	4	4.8
4	2	2.4
5	2	2.4

Table 7. Average size of classification descriptions for Experiments 1–3.

Experiment	C4.5 Unpruned Decision Tree	C4.5 Pruned Decision Tree	C4.5 Rules	FOIL Rules
1	635	17.2	24.2	719.9
2	919.2	153	62.9	1276.5
3	877.4	323.6	64.6	1085.9



```
@relation golf

@attribute outlook { sunny, overcast, rain}
@attribute temperature real [0,100]
@attribute humidity real [0,100]
@attribute windy { true, false}
@attribute class { Play, 'Dont Play' }

@data
sunny, 85, 85, false, 'Dont Play'
sunny, 80, 90, true, 'Dont Play'
overcast, 83, 78, false, Play
rain, 70, 96, false, Play
rain, 68, 80, false, Play
rain, 65, 70, true, 'Dont Play'
overcast, 64, 65, true, Play
sunny, 72, 95, false, 'Dont Play'
sunny, 69, 70, false, Play
rain, 75, 80, false, Play
sunny, 75, 70, true, Play
overcast, 72, 90, true, Play
overcast, 81, 75, false, Play
rain, 71, 80, true, 'Dont Play'
```

Figure 1. Mitchell, Sherlock and Smith

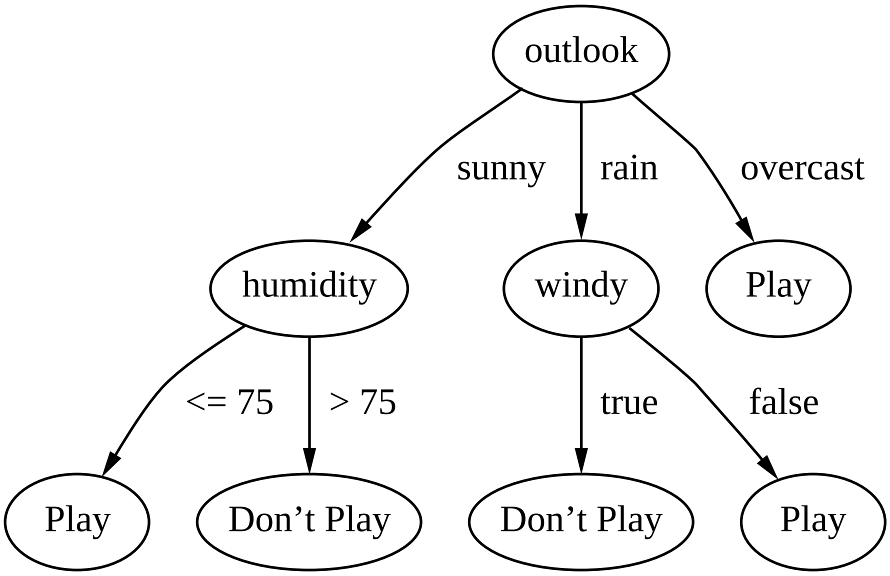


Figure 2. Mitchell, Sherlock and Smith

```
'Play' :- outlook=“overcast”.
'Play' :- temperature<=70,windy=“false”.
'Play' :- temperature>72,temperature<=75.
'Dont Play' :- outlook=“sunny”,humidity>70.
'Dont Play' :- outlook=“rain”,windy=“true”.
```

Figure 3. Mitchell, Sherlock and Smith

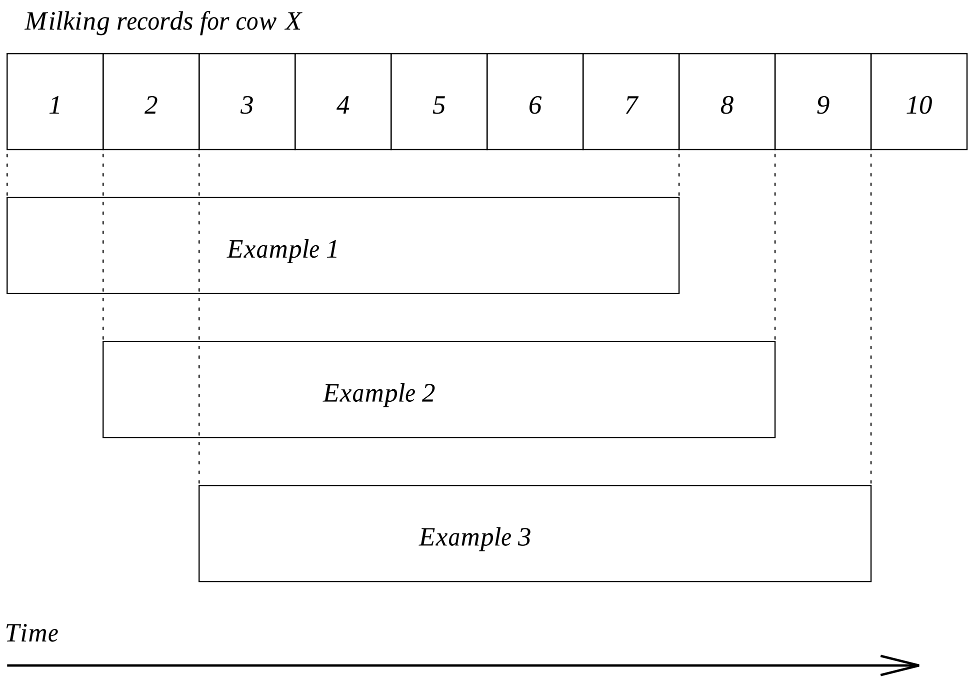


Figure 4. Mitchell, Sherlock and Smith

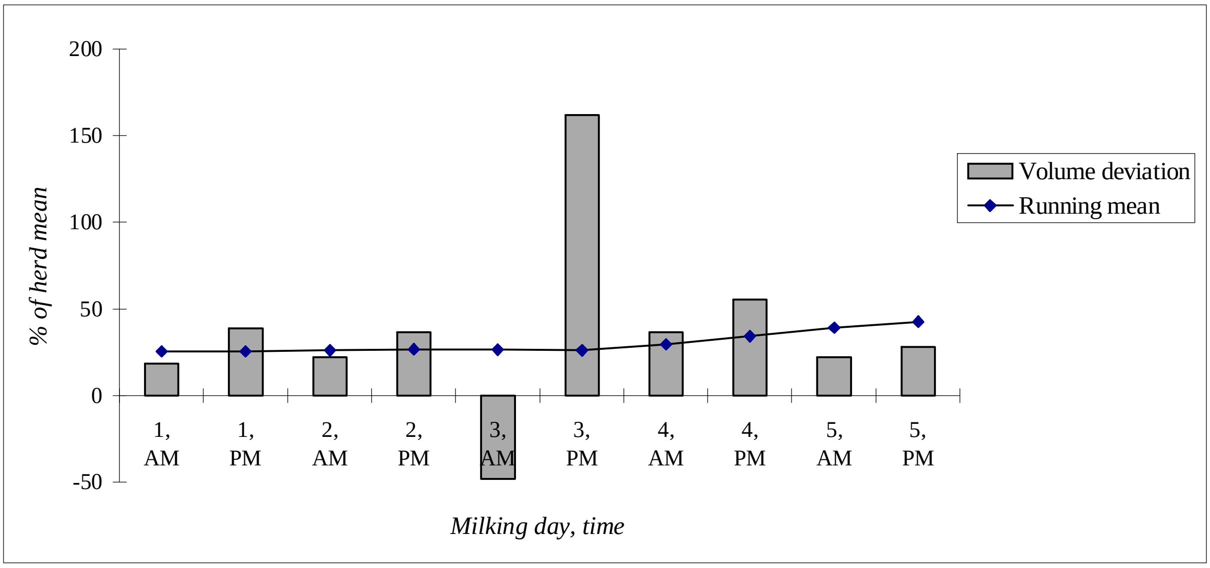


Figure 5. Mitchell, Sherlock and Smith

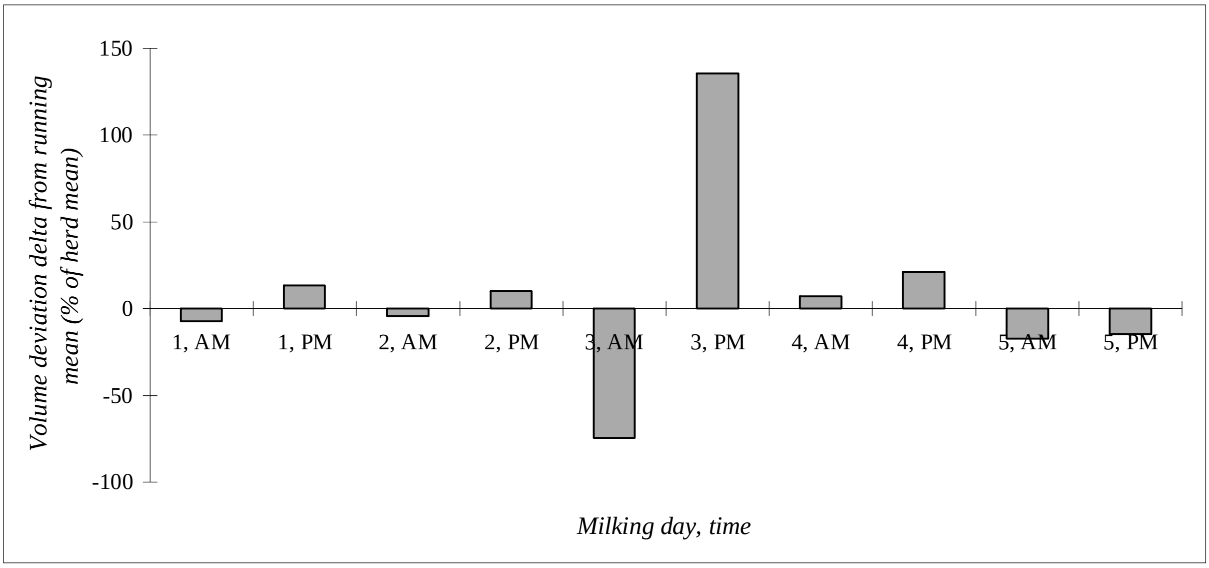
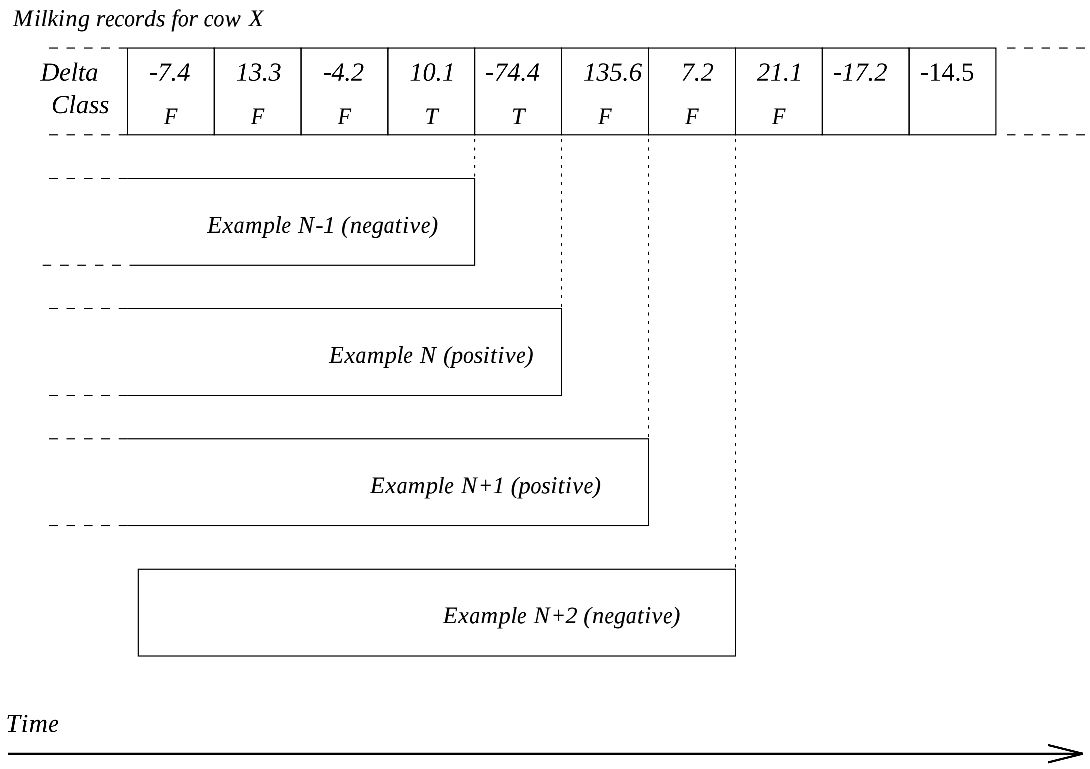


Figure 6. Mitchell, Sherlock and Smith



Milking records for cow X

Delta	*-7.4*	*13.3*	*-4.2*	*10.1*	*-74.4*	*135.6*	*7.2*	*21.1*	*-17.2*	-14.5
Class	*F*	*F*	*F*	*T*	*T*	*F*	*F*	*F*		

Figure 7. Mitchell, Sherlock and Smith

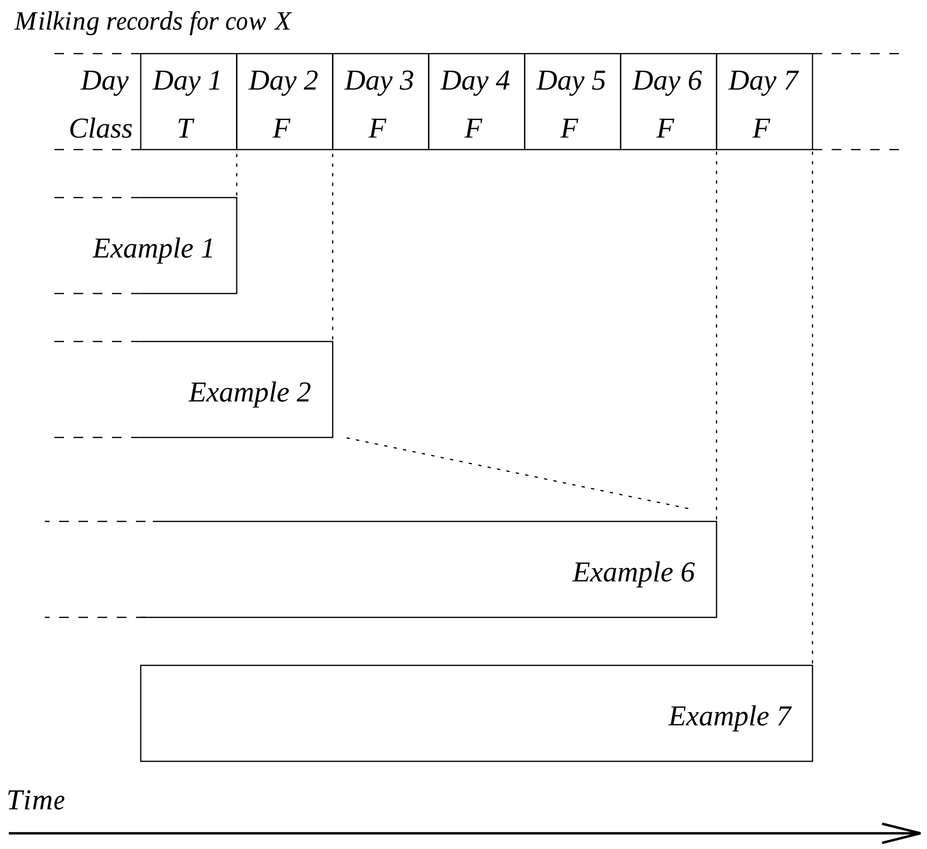


Figure 8. Mitchell, Sherlock and Smith

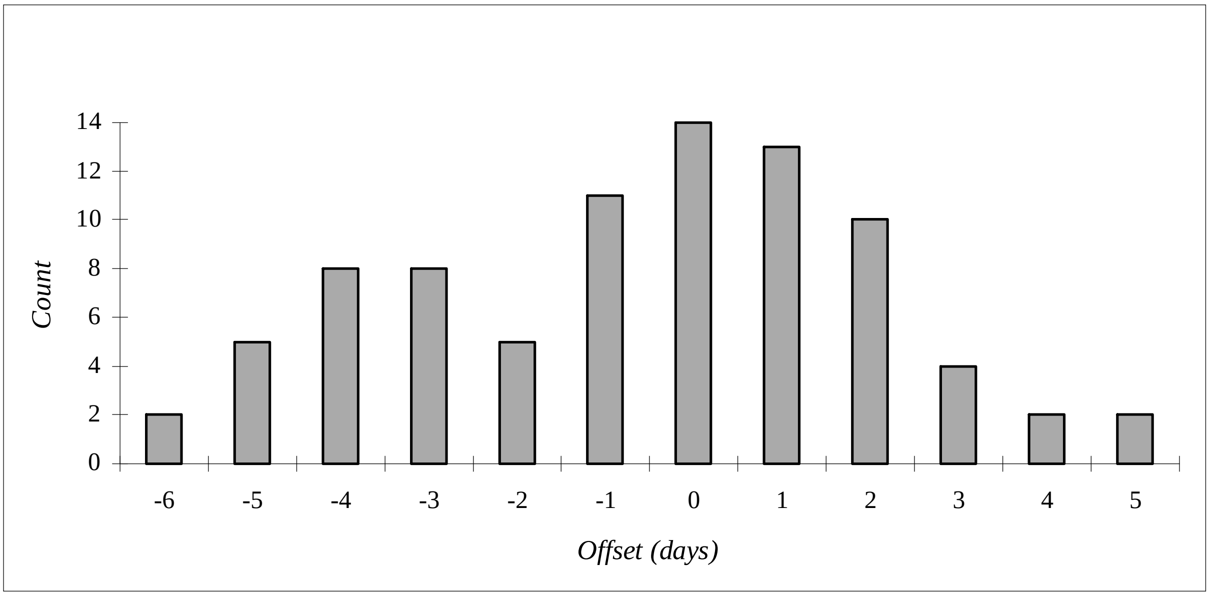


Figure 9. Mitchell, Sherlock and Smith

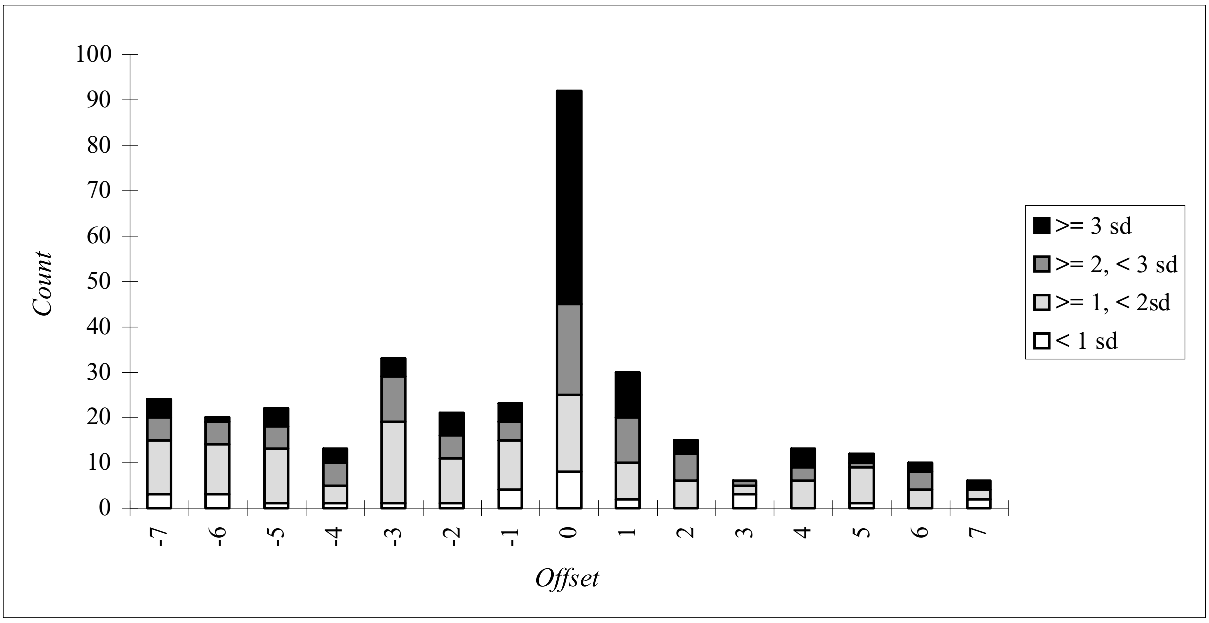


Figure 10. Mitchell, Sherlock and Smith

List of Figures