CLIFF: Finding Prototypes for Nearest Neighbor Algorithms with Application to Forensic Trace Evidence

Fayola Peters

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Tim Menzies, Ph.D., Chair
Arun Ross, Ph.D.
Bojan Cukic, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2010

Keywords: Forensic Evaluation, Prototype Learning, K-Nearest Neighbor

Abstract

CLIFF: Finding Prototypes for Nearest Neighbor Algorithms with Application to Forensic Trace Evidence

Fayola Peters

Prototype Learning Schemes (PLS) started appearing over 30 years ago (Hart 1968, [22]) in order to alleviate the drawbacks of nearest neighbor classifiers (NNC). These drawbacks include:

1. computation time,
2. storage requirements,
3. the effects of outliers on the classification results,
4. the negative effect of data sets with non-separable and/or overlapping classes,
5. and a low tolerance for noise.

To that end, all PLS have endeavored to create or select a *good* representation of training data which is a mere fraction of the size of the original training data. In most of the literature this fraction is approximately 10%. The aim of this work is to present solutions for these drawbacks of NNC. To accomplish this, the design, implementation and evaluation of CLIFF is described. The basic structure of the CLIFF algorithm involves a ranking measure which ranks the values of each attribute in a training set. The values with the highest ranks are the used as a rule or criteria to select instances/prototypes which obeys the rule/criteria. Intuitively these prototypes best represent the region or neighborhood it comes from and so are expected to eliminate the drawbacks of NNC particularly 3, 4 and 5 above.

With seven(7) standard data sets from the UCI repository [17], the outcome of this work demonstrate that for most cases, CLIFF is statistically the same as or better than those from 1NN rule clssifier as well as three other PLS. Finally in the forensic case study a data set composed of the infrared spectra of the clear coat layer of a range of cars, the performance analysis showed that it is strong with near 100% of the validation set finding the right target. Also, prototype learning is applied successfully with a reduction in *brittleness* while maintaining statistically indistinguishable results with validation sets.

Acknowledgments

The authors would like to thank the WVU Forensics Science Initiative and Royal Canadian Mounted Police (RCMP) particularly Mark Sandercock for the glass database. This work is supported by the Office of Justice Programs, National Institute of Justice, Investigative and Forensic Sciences Division under Grant No. 2003-RC-CX-K001.

Contents

List of Figures

Chapter 1

Introduction

Since the creation of the Nearest Neighbor algorithm in 1967 (Hart [9]), a copious amount of prototype learning schemes (PLS) have appeared to remedy the five (5) major drawbacks associated with the algorithm and it's variations. First, the high computation costs caused by the need for each test sample to find the distance between it and each training sample. Second, the storage requirement is large since the entire dataset needs to be stored in memory. Third, outliers can negatively affect the accuracy of the classifer. Fourth the negative effect of data sets with non-separable and/or overlapping classes and last, the low tolerance to noise. To solve these issues, PLS are used. Their main purpose is to reduce a training set via various selection and/or creation methods to produce *good prototypes* (Figure 1.1). *Good* here meaning that the prototypes are a good representation of the original training data set such that they maintain comparable or improved performance of a nearest neighbour classifier while simultaneously eliminating the effects of the drawbacks mentioned previously.

A review of the literature on prototype learning for this thesis has yielded at least 40 PLS each indicating with experimental proof that their particular design is comparable or better than the standard schemes; published surveys of PLS can be found in [3, 4, 25, 34]. However many of these schemes suffer from at least one of the following disadvantages:

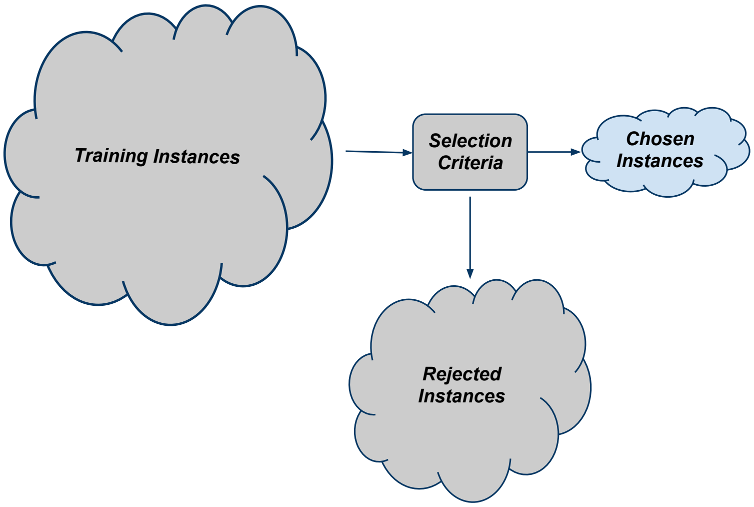


Figure 1.1: The Instance Selection Process.

- computationally expensive
- order effects (where the resulting prototypes are unreasonably affected by the order of the original training set)
- overfitting

The goal of this thesis is not to be unduly critical of the PLS which may succumb to any of the above disadvantages (particularly since many of them have proven to be successful), but rather to present a novel approach to this field of study which overcomes these disadvantages to report little or no loss in recognition of NNC.

Thus, this thesis presents CLIFF, a prototype learning scheme which runs in linear time, is independent of the order of the training set and avoids overfitting. A novel feature of CLIFF is that instead of either removing or adding (un)qualified prototypes from/to a 'prototype list'based on misclassification or clustering, criteria are generated for each target class. These criteria are created using a ranking algoritm called BORE (Best Or Rest) [23] or SBBR (Support Based Baysean Ranking) algorithm, which basically finds the value for each attribute which best represents the specific target class i.e. have the highest ranks. Any of the instances in the training set which adheres to some or all the criteria is or are selected as prototypes. Using this structure the percentage reduction of the training set is directly related to the number of constraints from the criteria used on the prototype selection process, in that the more constraints used the lower the percentage reduction. In this work, to be sure that each class is represented in the final list of prototypes, instances are selected using one contraint at a time until the prototype list is reduced as much a possible without being empty.

After describing the design and operation of CLIFF, its performance is demonstrated by evaluating it using cross-validation experiments with the wide variety of standard data sets from the UCI repository [17].

Next we describe how CLIFF can be used as part of a tool/model for the evaluation of trace forensic evidence. The principal goal of forensic evaluation models is to check that evidence found at a crime scene is (dis)similar to evidence found on a suspect. In our studies of forensic models for evaluation particularly in the sub-field of glass forensics, we conjecture that many of these models succumb to the following flaws:

1. A tiny error(s) in the collection of data;

2. Inappropriate statistical assumptions, such as assuming that the distributions of refractive indices of glass collected at a crime scene or a suspect obeys the properties of a normal distribution;

3. and the use of measured parameters from surveys to calculate the *frequency of occurrence* of trace evidence in a population

In this work we show that CLIFF plays an effective role in the evaluation of forensic trace evidence.

Our research is guided by the following research question:

- Is CLIFF viable as a Prototype Learner for NNC?

  The goal here is to see if the performance of CLIFF is comparable or better than the plain k nearest neighbor (KNN) algorithm and three(3) other PLS. So in our first experiment we compare the performance of predicting the target class using the entire training set to using only the prototypes generated by CLIFF and the other PLS.

1.1 Contributions of this Thesis

The contributions of this thesis are:

- The CLIFF algorithm, a linear time instance selector;
- A measure for the concept of *brittleness* - i.e. how far does a test instance have to move before it changes class;
- A viable method to reduce one effect of noise when performing instance selection.

It is our intent that this work open the eyes of the forensic scientist to the real problem of *brittleness* which exists in current forensic models. We hope in the future that the scientist, when verifying a model, they include a brittleness measure along with their evaluation of forensic evidence as done in this work. This will allow them to be confident that their result comes from a region or neighborhood of similar rather than dissimilar interpretation.

Although we contend that CLIFF can be applied to any type of trace evidence, in future work we hope to acquire more data sets to test CLIFF on. Also, direct comparison with other evaluation models will be investigated.

1.2 Structure of this Thesis

The remaining chapters of this thesis are structured follows:

- Chapter 2 provides a survey of Prototype Learning Schemes over the years;
- Chapter 3 describes the design and operation of CLIFF;
- Chapter 4 presents a detailed description of the experimental procedure followed to analyze standard data-sets using CLIFF;
- Chapter 5 examines a case study in which CLIFF is used as part of a forensic interpretation model to reduce the *brittleness* of published forensic models;
- Chapter 6 conclusions and future work are presented.

Chapter 2

Background and Related Work

In Chapter 1, the optimal goal of PLS as a solution to the drawbacks of NNC is highlighted. To continue this discussion, in this chapter we present a brief survey of PLS starting with one of the earliest - Hart's 1968 Condensed Nearest Neighbor (CNN) [22] to various PLS published in 2008. The reader will see that since 1968, researchers in this field have created PLS which fit into at least one of two(2) categories: 1) instance selection and 2) instance abstraction.

Before moving forward, in the interest of clarity, the following terminology are used throughout the remainder of this thesis: all data-sets refers to supervised data-sets and each data-set consists of rows and columns where each row is referred to as an instance and each column is called an attribute except for the last column which is the target class; a prototype is an instance selected or created to be part of the final reduced training set and finally, consistency is defined as the ability of the final subset of prototypes to correctly classify the original training set.

2.1 Prototype Learning for Nearest Neighbor Classifiers

Research in prototype learning is an active field of study [4,5,7,8,10,11,18,19,26,27,30,38]. A review of the literature in this field has revealed two(2) categories of PLS: 1) instance selection and

2)instance abstraction. Instance selection involves selecting a subset of instances from the original training set as prototypes. Using what Dasarathy terms as *edit rules*, instance selection can take place in four(4) different ways.

1. incremental (CNN [22])
2. decremental (RNN)
3. a combination of 1 and 2
4. border points, non-border points or central points

Instance abstraction involves creating prototypes by merging the instances in a training set according to pre-determined rules. For example, Chang [8] merges two instances if the have the same class, are closer to each other than any other instances and the result of merging does not degrade the performance of NNC. Figure 2.1 shows the PLS surveyed in this work.

Name	Abbreviation	Cite	Type
Condensed Nearest Neighbor	CNN	[22]	Instance Selection
Reduced Nearest Neighbor	RNN	[20]	Instance Selection
Minimal Consistent Set	MCS	[10]	Instance Selection
Prototype Selection by Clustering	PSC	[34]	Instance Selection
The Chang and Modified Chang Algorithms	MCAs	[5, 8]	Instance Abstraction
Learning Vector Quantization	LVQ	[26]	Instance Abstraction

Figure 2.1: PLS surveyed in this work

2.1.1 Instance Selection

Condensed Nearest Neighbor (CNN)

Different PLS use various criteria to determine which instance in a training set is a worthy choice as a prototype. Each also tend to focus on specific goals such as increasing speed or performance or storage reduction. CNN [22] uses a *incremental* strategy where it initializes a random subset of prototypes and adds to the list. Hart's goal with CNN focused on storage reduction with the aim to create a minimal consistent set, i.e. a smallest subset of the complete set that also classifies all the original instances correctly.

Figure 2.2 shows the pseudo-code for CNN which begins by randomly selecting one instance from each target class and stores them as prototypes in a list. These prototypes are then used to classify (using the 1NN rule) the instances in the training set. If any of theses instances are misclassified they are added to the prototype list. This process is repeated until the prototype list can no longer be increased.

Admittedly, although a reduction in the training set with consistency was accomplished, Hart did not achieve his goal of a minimal consistent set with CNN. CNN also suffers with the following disadvantages:

- sensitive to the initial order of input data
- sensitive to noise which can degrade performance

```
INITIAL_PROTOTYPES = [RANDOM(FROM EACH TARGET CLASS)]
PREV = []
CUR = [INITIAL_PROTOTYPES]

REPEAT UNTIL PREV = CUR
  MISCLASSIFIED = classify TRAIN and RETURN MISCLASSIFIED INSTANCES
  PREV = CUR
  CUR = MISCLASSIFIED + CUR
END
```

Figure 2.2: Pseudo-code for CNN

Reduced Nearest Neighbor (RNN)

RNN [20] takes an opposite approach to CNN. Its strategy is *decremental*. So rather than start with a subset of the training set as CNN does, RNN uses the entire training set as initial prototypes and reduces the list. In the end, it is computationally more expensive than CNN, but always produces a subset of a CNN result.

The algorithm begins by setting the initial prototypes as the entire training set. From here, a prototype is removed if and only if its removal does not cause the misclassification of any instance in the training set. This procedure stops when no more prototypes can be removed from the prototype list. Figure 2.3 shows the pseudo-code for RNN.

```
PREV = []
CUR = [TRAIN]

REPEAT UNTIL PREV = CUR
  PREV = CUR
  IF (CUR - (FIRST CUR)) cause misclassification of TRAIN
    CUR = CUR
    CUR = (REST CUR)
  END
END
```

Figure 2.3: Pseudo-code for RNN

Minimal Consistent Set (MCS)

The goal of the MCS is to achieve what Hart [22] failed to achieve: a minimal consistent set. MCS uses a voting strategy which favors instances with the greatest number of like instances (in other words, those with the same class) closer to them than unlike instances.

As explained in [10], MCS takes a *top-up* approach as with RNN where at first the entire training set is seen as the initial prototypes. Then for each of these prototypes the distance of its nearest unlike neighbor (NUN), i.e. nearest neighbor with a different class, is found. Next, all nearest like neighbors (NLN) of this prototype whose distances from the prototype are less than that of the NUN are stored. Each of these NLN are counted as a vote toward the prototype for candidacy as a final prototype. The prototype with the most votes is then designated as a candidate and all NLN who contributed to the vote are removed from candidacy consideration as well as from the voting lists of other prototypes.

With the votes now updated, the process is repeated by designating the prototype who now has the most votes as a candidate. This process continues until the list can no longer be reduced. Further, since the goal of MCS is to find the minimal consistent set, the entire strategy is iterative in that the reduced list is now used as input for the next iteration starting with finding the distance of NUN for each prototype. Figure 2.4 shows the pseudo-code for MCS.

```
PREV = []
CUR = [TRAIN]
P* = []   \\ CANDIDATE PROTOTYPE

REPEAT UNTIL PREV = CUR
 PREV = CUR
 CUR = MCS(CUR)
END

MCS(CUR)
REPEAT UNTIL PREV = CUR
  FOR EACH c IN CUR
   DISTANCE_NUN = DISTANCE(c, NUN)
   VOTER_LIST = NLN
   VOTES = COUNT(VOTER_LIST)
  END

  P* = NLN with MAX(VOTES)
  CUR = CUR - P*(VOTER_LIST)
END
```

Figure 2.4: Pseudo-code for MCS

Prototype Selection by Clustering (PSC)

Unlike the previous methods, PSC is uses clusters to aid in prototype selection. Based on the argument that ”...in a training set interior prototypes can be removed with little effect on classification accuracy; and they also argue that border prototypes are critical for classification since they provide relevant information for preserving discrimination between classes” [34], PSC selects prototypes which are border prototypes and also some internal prototypes.

As explained in [34], PSC begins by dividing the training data into regions using a clustering algorithm. Although *Cmeans* is used in their work, it is made clear that any clustering method can be used in its place and so in our experiments for this thesis, *Kmeans* is our algorithm of choice.

After clustering, the internal prototypes are found. This is done by first finding the homogeneous clusters (all instances in the cluster have the same target class). Once identified, the instance which is nearest to the centroid of each cluster is selected as a prototype.

Now to find the border prototypes, those clusters which are non-homogeneous are identified. Next, all instances with the majority target class are found and the border prototypes for this class are those closest to instances of a different target class(es). This is true for instances of each class.

2.1.2 Instance Abstraction

The Chang and Modified Chang Algorithms

Chang’s [8] work on finding prototypes for nearest neighbor classifiers is one of the earliest abstraction methods in the literature. The basic idea of his algorithm is to start with every sample in the training set as a prototype, then merge any two nearest prototype (p1 and p2 to form p*) with the same class as long as the recognition rate is not degraded. The new prototype p* can be formed by simply finding the average of p1 and p2 or the average vector of weighted p1 and p2 (Chang uses the weighted average). p* will have the same class as the individual prototypes p1 and p2. The merging process will continue until the number of incorrect classifications of patterns in the

```
CLUSTER = [KMEANS]

CLUSTERS = CLUSTER ON TRAIN

FOR EACH cluster IN CLUSTERS
  IF cluster = homogeneous
    CENTROID = MEAN(cluster)
    PROTOTYPE = INSTANCE NEAREST CENTROID
  ELSE
    LET CM BE MAJORITY CLASS IN cluster
    FOR EACH CLASS CK IN cluster (CK NOT= CM)
     FOR EACH INSTANCE_K IN CK
       IF INSTANCE_M IN CM IS CLOSEST TO INSTANCE_K
        PROTOTYPE = PROTOTYPE + INSTANCE_M
        IF INSTANCE_K IN CK IS CLOSEST TO INSTANCE_M
         PROTOTYPE = PROTOTYPE + INSTANCE_K
      END
  RETURN PROTOTYPE
END
```

Figure 2.5: Pseudo-code for PSC

data set starts to increase. Figure 2.6, shows Chang's algorithm.

The Modified Chang Algorithm (MCA) [5] was proposed by Bezdek in 1998. In this approach, Chang's algorithm is modified in two(2) ways: 1) the arithmetic mean is used rather than the weighted mean to merge prototypes. 2) the search for prototypes to merge is improved by partitioning the training set by their class labels and only searches within these respective partitions for the nearest prototype to merge. This eliminates the need to check for the criteria of candidate prototypes having the same class.

Learning Vector Quantization (LVQ)

Kohonen [26] in 1990, introduced the learning vector quantization (LVQ) to create prototypes the 1NN classifier using error correction rules. LVQ is a set of learning algorithms for nearest pro-

```
Step 1: Start with an arbitrary point tj
        in B* and assign it to A*
Step 2: For all points tk in B* such that
        class (tk) is not equal to class (tj),
        update bk to be the distance between
        tk ans tj if this distance is smaller
        than the present bk. Otherwise, bk is
        unchanged.
Step 3: Among all the points in B*, find the
        point ts which has the smallest bs
        associated with it.
Step 4: If tj is not the nearest point ts such
        that the classes of tj and ts are
        different, go to Step 6. Otherwise
        continue.
Step 5: Check whether or not d(tj, ts) is less
        than bj. If no, go to Step 6. If yes,
        let bj=d(tj, ts) and continue.
Step 6: Let j=s, move ts from B* to A*, and go
        to Step 2 until B* is empty. When B* is
        empty, the final b1,...,bm are the
        desired ones.
```

Figure 2.6: Chang's algorithm for finding prototypes

totype classification and its basic algorithm, LVQ1, works by first selecting a certain number of prototypes from each class randomly as initial prototypes. This ensures that each class is represented be at least one prototype. These initial prototypes are then updated using the training set with the basic idea that the prototypes will be attracted to training points with the same class label and repelled by those with different class labels. So for an input instance x, find the prototype, m_i (which is denoted by m_c), that is closest to it.Equation 2.1 defines the basic LVQ1 process of how the prototypes are updated [26].

$$m_c(t+1)=m_c(t)+\alpha(t)[x(t)-m_c(t)]ifxandm_chavethesameclass,m_c(t+1)=m_c(t)-\alpha(t)[x(t)-m_c(t)]ifxandm_c \tag{2.1}$$

Here $\alpha(t)$ is a learning rate between 0 and 1 and it decreases with time.

Chapter 3

CLIFF: Tool for Instance Selection

The PLS that we have discussed so far in the previous chapter tend to be based on misclassification, clustering, stochastic search and so on. Rarely does one come across a PLS which considers the idea that some ranges of values for attributes can be critical in selecting prototypes for each class. Thus rows without any of these range values are considered as uncritical. In other words, we consider using techniques practiced in the field of Feature Subset Selection (FSS) for instance selection.

CLIFF was born from this idea, in that the selection of a prototype is dependent on whether each value in an instance satisfies the criteria of being present in a range of values which has been determined to be critical in distinguishing a particular class. Finding these critical range of values is the core of CLIFF and is accomplished by using a Support Based Bayesian Ranking Algorithm (SBBR) [23] which ranks these ranges, so in the end, the critical range is the one with the highest rank. The following section explains the design of CLIFF in detail starting with the core of CLIFF - SBBR - and ending with CLIFF's time complexity.

Before moving forward it is important to note that if a data-set contains attributes whose values are numeric, the values are binned using an equal frequency binning algorithm which sorts attribute values into *N* equal frequency regions making sure that there are no repeats in the numbers which are the boundaries between regions. We refer to the values in these bins as a *range of values* and is represented by a bin number. On the other hand, if an attributes contains discrete values then these are simply refered to as *values*.

3.1 The Support Based Bayesian Ranking Algorithm (SBBR)

The core of CLIFF is to find for a particular attribute a range of values more likely to be present for a particular class. To find these ranges, CLIFF uses a Bayesian ranking measure which includes a support measure (SBBR). First we assume that the target class is divided into one class as *best* and the other classes as *rest* [23]. This makes it easy to find the attribute values which have a high probability of belonging to the current *best* class using Bayes theorem. The theorem uses evidence E and a prior probability $P(H)$ for hypothesis H *best, rest*, to calculate a posteriori probability *P(H|E) = P(E|H)P(H) / P(E)*. When applying the theorem, likelihoods are computed from observed frequencies, then normalized to create probabilities: this normalization cancels out $P(E)$ in Bayes theorem (see Equation 3.1).

```
Let likelihood = like
```

$$P(best|E) = \frac{like(best|E)}{like(best|E) + like(rest|E)} \tag{3.1}$$

Unfortunately, one problem was found using the theorem, according to [23], Bayes theorem is a poor ranking heuristic since it is distracted by low frequency evidence. To alleviate this problem the support measure was introduced. Its purpose was to increase as the frequency of a value increases i.e. like(best|E) is a valid support measure hence Equation 3.2.

```
Let likelihood = like
```

$$P(best|E) * support(best|E) = \frac{like(best|E)^2}{like(best|E) + like(rest|E)} \tag{3.2}$$

So in this work, each attribute value or range of values is ranked according to Equation 3.2. Once ranked, the critical ranges for each attribute are extracted (those with the highest ranks) and used as the criteria for selecting instances from the current *best* class. The whole process is repeated using a different class as *best* and all the others (including the previous *best* class) as *rest*. When a criteria for each class has been found the process ends. Figure 3.1 shows the pseudocode for ranking each value of an attribute for an entire training data set.

```
A = [Attributes]
BEST = [Instances with current best class]
REST = [Instances with other classes]
FREQ = [frequency]
LIKE = [likelihood]
EFB = [EqualFrequencyBinning]
BIN = [Values within Attribute range]
P_BEST = [Probability of BIN in BEST]
P_REST = [Probability of BIN in REST]

BIN_DATA = EFB on data
 FOR EACH attribute in A{
  FOR EACH BIN IN attribute{
   P_BEST = count(BEST) / count(data)
   P_REST = count(REST) / count(data)
   FREQ(BIN|BEST) = count(BIN in BEST) / count(BEST)
   FREQ(BIN|REST) = count(BIN in REST) / count(REST)
   LIKE(BEST|BIN) = FREQ(BIN|BEST) x P_BEST
   LIKE(REST|BIN) = FREQ(BIN|REST) x P_REST
   LIKE_BEST_REST = LIKE(BEST|BIN) + LIKE(REST|BIN)
   RANK = LIKE(BEST|BIN)^2 / LIKE_BEST_REST
   RETURN [BIN, RANK]
  END
END
```

Figure 3.1: Pseudo code for Support Based Bayesian Ranking algorithm for finding the rank of each value in each attribute.

3.2 Instance Selection Using a Criteria

Once the criteria for each class has been found, the question is - how do we use it to select instances? There is the possibility that using the entire criteria could result in zero instances being selected. To avoid this we use one critical range at a time for instance selection until just before there are no more instances to select from or the number of instance selected at between 0 and 15% of the number of instances in the class. Another problem considered is the order in which the critical ranges are used. This problem is solved by sorting them in descending order according to there rank values. Figure 3.2 shows the pseudocode for the process.

```
CRITERIA = [LIST OF CRITICAL RANGES SORTED BY HIGH TO LOW RANK]
N = NUMBER OF INSTANCES IN CLASS
PROTOTYPES = [INSTANCES IN CLASS]

REPEAT UNTIL (PROTOTYPES = PREVIOUS) OR (|PROTOTYPES| = [0 AND 15%] OF N)
  PREVIOUS = PROTOTYPES
  WHILE c IN CRITERIA
    PROTOTYPES = INSTANCES WITH c
    IF PROTOTYPES = EMPTY
      PROTOTYPES = PREVIOUS
  END

  RETURN PROTOTYPES
END
```

Figure 3.2: Instance Selection Using Criteria

3.3 CLIFF: A Simple Example

For this example we use the *weather* data-set shown in Figure 3.3. This data-set contains four(4) attributes that report on the forecast (sunny, overcast and rainy), temperature (hot, mild and cool),

```
forecast  temp  humidty  windy  play
1. sunny     hot   high     FALSE  no
2. sunny     hot   high     TRUE   no
3. overcast  hot   high     FALSE  yes
4. rainy     mild  high     FALSE  yes
5. rainy     cool  normal   FALSE  yes
6. rainy     cool  normal   TRUE   no
7. overcast  cool  normal   TRUE   yes
8. sunny     mild  high     FALSE  no
9. sunny     cool  normal   FALSE  yes
10. rainy     mild  normal   FALSE  yes
11. sunny     mild  normal   TRUE   yes
12. overcast  mild  high     TRUE   yes
13. overcast  hot   normal   FALSE  yes
14. rainy     mild  high     TRUE   no
```

Figure 3.3: A log of some golf-playing behavior

humidity (high and normal) and whether or not the day is windy (true or false). Of the 14 days observed, on nine(9) of them, golf was played and on the remaining five(5) days, no golf was played.

To create a criteria for each class in this data-set, we first divide it into *best* and *rest*. For this example, let us say that all the instances with the *yes* class are *best* while the others are *rest*. Now we find the ranks of each value in each attribute, so let K = 14 (total number of instances), *best* = 9 and *rest* = 5. To find the rank of *sunny* in forecast the following calculations are completed in Figure 3.4.

Once the ranks are found for each value for each attribute, the criteria is created using the highest ranked values in each attribute to create [attribute, value] pairs. For this example we will use the following as the criteria for the *yes* class:

```
[[forecast, rainy] [temp, mild] [humidity, high] [windy, FALSE]]
```

```
E = sunny
P(best) = 9/14
P(rest) = 5/14
freq(E|best) = 2/9
freq(E|rest) = 3/5
like(best|E) = 2/9 * 9/14 = 2/14
like(rest|E) = 3/5 * 5/14 = 3/14
P(best|E) = (2/14) / (2/14 + 3/14) = 0.40

P(best|E) * support(best|E) = (2/14)^2 / (2/14 + 3/14)
= 0.06
```

Figure 3.4: Finding the rank of *sunny*

With this criteria, after applying [forecast, rainy], we are left with three instances (4, 5 and 10):

```
rainy     mild  high     FALSE  yes

rainy     cool  normal   FALSE  yes

rainy     mild  normal   FALSE  yes
```

Since this is about 33% of the data in the *yes* class, the second pair [temp, mild] in the criteria is used on the three instances. This yield two(2) instances (4 and 10) reducing the data in the *yes* class to about 22%:

```
rainy     mild  high     FALSE  yes

rainy     mild  normal   FALSE  yes
```

With the algorithm preferring a reduction of between 0 and 15%, the third pair in the criteria is applied to the two(2) instances. The result of this is zero instances left and so the result of the previous pair is kept.

3.4 CLIFF: Time Complexity

Intuitively, time complexity for CLIFF can be considered in terms of 1) ranking each value in each attribute, a $O(m)$ operation where m represents attributes, 2) finding the criteria for each class, a *O(m)* + *O(k)* operation where k represents the class, and 3) selecting instances from each class using the criteria a $O(n)$ operation where n represents the number of instances.

Assuming that $n > m > k$ (which is the case for all data-sets used in this thesis), this process yields a complexity of $O(m)$ + *O(m)* + *O(k)* + $O(n)$ which reduces to $O(n)$.

Chapter 4

CLIFF Assessment

In Chapter 3, we discussed the design and operation of CLIFF with the help of a simple example. We also showed its time complexity as linear - $O(n)$ in Section 3.4. In this chapter, we look at how CLIFF performs against other PLS with three(3) evaluation methods. CLIFF is evaluated by first comparing its performance with three(3) PLS spanning the decades from 1968 (Hart's CNN [22]) to MCS in 1994 [10] and finally 2010 PSC [34]. We then move on to examine the noise tolerance of each PLS studied here by introducing artificial noise to the training sets. Finally we take a look at what we call the *brittleness* measure. Brittleness, discussed in Chapter 5, is defined as *a tiny change in the input data can lead to a major change in the output*. As the reader will see, we view instance selection as a viable method to decrease *brittleness* and the following sections will show that CLIFF does a better job of reducing the impact of *brittleness* than any other PLS.

4.1 Data Sets

Figure 4.1 lists the seven(7) data sets used to assess CLIFF. The number of instances and attributes per instance are shown for each data set, along with the number of distinct classes of instances. All of these data sets were acquired from the UCI repository [17]. Except for the Iris data-set, all the

Data Set	Code	Instances	Attributes	Class
Breast Cancer	bc	286	9	2
Dermatology	dm	366	34	6
Heart (Cleveland)	hc	303	13	5
Heart (Hungarian)	hh	297	13	2
Iris	ir	150	4	3
Liver (Bupa)	lv	345	6	2
Mamography	mm	150	4	3

Figure 4.1: Data Set Characteristics

attribute values of the data-sets are discrete and so do not require any pre-processing. However, since the attribute values for Iris are numeric, we discretize them using an equal frequency binning algorithm so that ranges of values are ranked rather than each individual value. In the experiments to follow, the number of bins is set to 10.

4.2 Experimental Method

We evaluate CLIFF as a prototype learning scheme on standard data sets in cross validation experiments. Its performance compared with CNN, MCS and PSC is measured using probability of detection (pd) and probability of false alarm (pf) completed as follows: By allowing A, B, C and D to represent true negatives, false negatives, false positives and true positives respectfully, it then follows that *pd* also known as recall, is the result of true positives divided by the sum of false negative and true positives *D / (B + D)*. While pf is the result of: *C / (A + C)*. The *pd* and pf values range from 0 to 1. When there are no false alarms $pf = 0$ and at 100% detection, *pd* = 1.

The results were visualized using *quartile charts* as in [37]. To generate these charts the performance measures for the *pd*s and pfs are sorted to get the median, lower and upper quartile of numbers. For our quartile charts, the upper and lower quartiles are marked with black lines; the median is marked with a black dot; and the vertical bars are added to mark the 50% percentile value. Figure 4.2 shows an example where the upper and lower quartiles are 39% and 59% respectively, while the median is 49%.

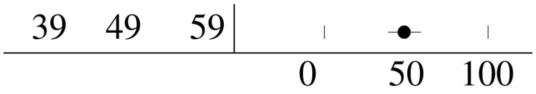


Figure 4.2

Finally, the Mann-Whitney U test was used to test for statistical difference of the different PLS. These results are shown as rank values starting at one(1). The lower the rank value, the better the performance of the prototype learner. Please note that prototype learners with the same rank value, are not statistically different.

The following sections describes the experiments and discusses the results.

4.3 Experiment 1: Is CLIFF viable as a Prototype Learning Scheme for NNC?

The goal here is to see if the performance of CLIFF is comparable or better than the plain k nearest neighbor (KNN) algorithm, and the CNN, MCS and PSC prototype learners. In this experiment we compare the performance of predicting the target class using the entire training set to using only the prototypes generated by the prototype learners including CLIFF. To accomplish this, our experiment design follows the pseudo code given in Figure 5.8 for the standard data sets. For each data set, tests were built from 20% of the data, selected at random. The models/prototypes were then learned from the remaining 80% of the data.

This procedure was repeated 5 times, randomizing the order of data in each data-set each time. In the end CLIFF is tested and trained 25 times for each data set.

```
DATA = [bc dm hc hh ir lv mm]
LEARNER = [KNN]
PLS = [KNN CLIFF CNN MCS PSC]
STAT_TEST = [Mann Whitney]

FOR EACH PLS
  REPEAT 5 TIMES
    FOR EACH data IN DATA
     TRAIN = random 80% of data
     TEST = data - TRAIN

     \\Construct model from TRAIN data
     r_TRAIN = Reduce TRAIN with PLS
     MODEL = Train LEARNER with r_TRAIN

     \\Evaluate model on test data
     [pd, pf] = MODEL on TEST
   END
  END
END
```

Figure 4.3: Pseudo code for Experiment 1

4.3.1 Results from Experiment 1

The results for this experiment and Experiment 2 (discussed in the following section) are shown in Figure 4.7 to Figure 4.13. For each data-set, each figure shows the results for a *clean* data-set (without noise) and a *noisy* data-set (with noise). For both the clean and noisy data-sets, the [pd, pf] quartile charts, the percentage of the data-set used for training after using the PLS and the rank values showing the significant difference between the PLS and KNN are presented.

Let us focus on the *clean* results for each data-set from Figure 4.1. First, we compare the CLIFF results with those of the baseline KNN. As shown, despite using only 9% to 15% of the training set, the pds and pfs results of CLIFF compares favorably with those of KNN showing similar or better rank values in most cases. For example, the Mammography(mm) data set (Figure 4.13) ranks as number one(1) for both the pd and pf results while KNN ranks at number three(3). In Figure 4.12, the Liver(lv) data set exhibits statistically similar results for pd while for pf CLIFF has a much better statistical performance than KNN. Figure 4.8 to Figure 4.11 also show encouraging results for CLIFF as compared with KNN, however Figure 4.7, the Breast Cancer(bc) data set shows an exception, with the pd and pf statistical results for KNN better than CLIFF.

Next, let us consider PLS. As compared with the other PLS, CLIFF does as well as or better than the others in all but one(1) case. However, although for the *bc* data set, CNN present statistically better pd and pf results, it does so with 62% of the training data for median pd and pf values of 55% and 39% respectively, while CLIFF only needed 11% of the training data for median pd and pf values of 67% and 20% respectively.

4.4 Experiment 2: How well does CLIFF handle the presence of noise?

The goal here is to see if the CLIFF works well in the presence of noise. This is important because according to [40]:

> In the presence of class noise, ... there are two main problems that can occur. The first is that very few instances will be removed from the training set because many instances are needed to maintain the noisy (and thus overly complex) decision boundaries. The second problem is that generalization accuracy can suffer, especially if noisy instances are retained while good instances are removed.

With that said, in this experiment we repeat Experiment 1, only this time noise is introduced to training data by randomly changing the target class of 10% of the instances in the training data to any other target class value. The the results here will indicate how well the 1NN classifier along with the different PLS are able to predict the correct target class even if some of its training data is faulty [40].

4.4.1 Results from Experiment 2

The *noisy* tables in Figure 4.7 to Figure 4.13 displays the results of Experiment 2. The first thing to recognize is that compared to the *clean* tables, there is a general degradation of the pd and pf results for each data set. However in some cases such as the pd results for *bc* the pds can degrade by as little as 1%. The second thing to recognize is that while the *sizes* of the training data basically remains the same for CLIFF (differences of no more that 2%) for all data sets, the other PLS display a general increase in the training set sizes except for PSC whose training size decrease from 42% to 40%.

4.5 Experiment 3: Can CLIFF reduce brittleness?

In this work, *brittleness* refers to the following:

> *Brittleness* is a measure of whether a solution (predicted target class) comes from a region of similar solutions or from a region of dissimilar solutions. Or, looking at this another way, how far would a test instance have to move before a different target class is predicted.

Take for example the *Before CLIFF* chart in Figure 5.1, the classes *versicolor* and *virginica* obviously show severe overlap. Also, the versicolor test instance represented by the purple square does not have to move very far before it can change its predicted target class to *virginica*. Looking now at the *After CLIFF* chart in Figure 5.1, after applying CLIFF, a subset of instances are selected as prototypes. This increases the distance from the *versicolor* test instance to a prototype with the *virginica* class thereby reducing *brittleness*.

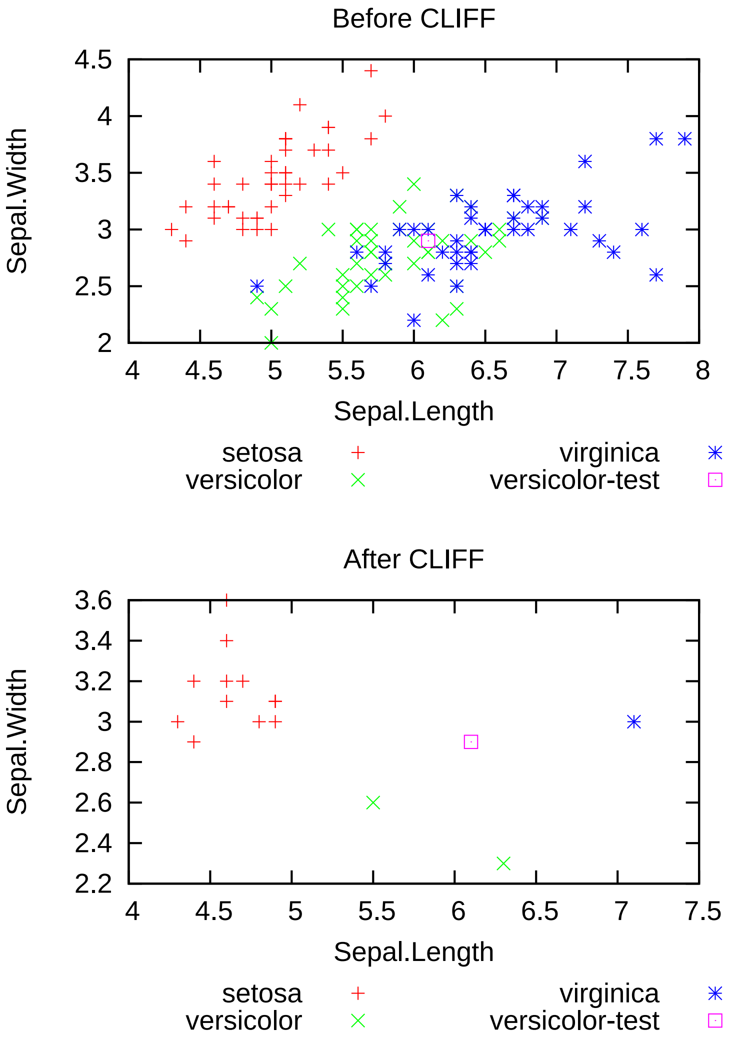


Figure 4.4

With this measure in mind, the goal here is to see how each prototype learning scheme studied here reduces the brittleness of the KNN model where k=1. Brittleness will be measured by distance each test instance moves before changing its original predicted target class. Intuitively, the further away the test instance has to move the less *brittle* the model. The experiment design for *brittleness* can be done in conjunction with Experiment 1 by collecting the distances of each test instance with a predicted target class to the nearest training instance with a different target class. This is done for each prototype learning scheme and 1NN. The distances generated are joined, sorted and labelled according to their position in the list. for example, let us say that A and B are PLS with the distance values of [2, 2, 2, 3, 4, 3, 77] and [6, 7, 3, 9, 1, 1, 1, 100] respectively. After being joined, sorted and labelled the result is as follows:

PLS	B	B	B	A	A	A	A	A	B	A	B	B	B	A	B
Sort	1	1	1	2	2	2	3	3	3	4	6	7	9	77	100
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Label	1	1	1	5	5	5	8	8	8	10	11	12	13	14	15

As shown, labels are assigned according to the position of a value in the list, however, if values are the same, the mean of their position values are used as a label.

4.5.1 Results from Experiment 3

Figure 4.5 and Figure 4.6 present the results for this experiment. The pattern is very clear: CLIFF does a much better job of reducing *brittleness* in all cases than any of the other PLS. Each chart in Figure 4.5 represents results for the different data sets used in this work. They all show that the CLIFF test instances have to move further away (most of the time) before there is a change in their target classes. These results are confirmed by a Mann Whitney statistical test (Figure 4.6), which show that the CLIFF results are statistically different and better than the other PLS.

4.6 Summary

Collectively, the results of the above experiments indicate that CLIFF may be effective in the field of forensic interpretation where a very low false alarm rate pf is desired. Here, CLIFF can be used to help lower the pfs of a forensic interpretation model. The results of Experiment 1 indicate this possibility in the median pf results where CLIFF's values ranges from 0 to 47 and are lower or the same as the baseline (KNN) pf results.

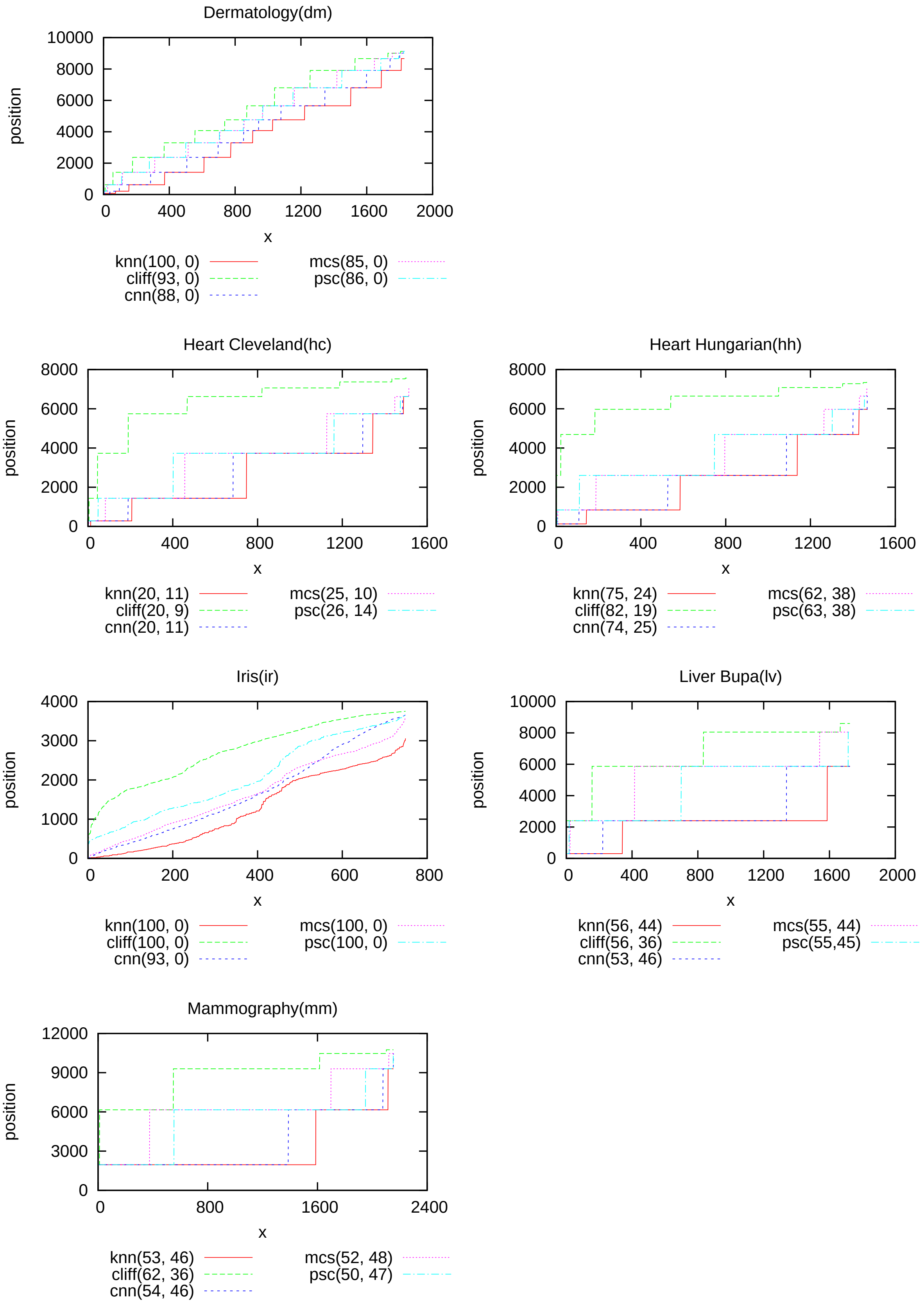


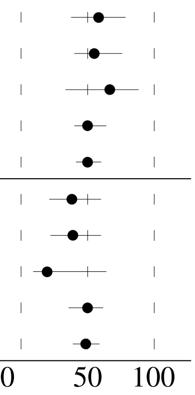
Figure 4.5: Position of distance values for PLS

Clusters	PLS	Significance
Breast Cancer (bc)	cliff	1
	mcs	2
	psc	2
	cnn	3
	knn	3
Dermatology (dm)	cliff	1
	mcs	2
	psc	2
	cnn	3
	knn	3
Heart Cleveland (hc)	cliff	1
	mcs	2
	psc	2
	cnn	2
	knn	2
Heart Hungarian (hh)	cliff	1
	mcs	2
	psc	2
	cnn	2
	knn	2
Iris (ir)	375	1
	mcs	2
	psc	3
	cnn	3
	knn	4
Liver Bupa (lv)	cliff	1
	mcs	2
	psc	2
	cnn	3
	knn	3
Mammography (mm)	cliff	1
	mcs	2
	psc	2
	cnn	3
	knn	3

Figure 4.6: Summary of Mann Whitney U-test results for Experiment 3 (95% confidence): In the Significance column, indicates that CLIFF is better than other PLS with the greatest brittleness reduction reported.

Clean Breast Cancer Results

bc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	knn	1	100	38	58	79	
	cnn+knn	1	62	40	55	76	
	cliff+knn	2	11	33	67	89	
	psc+knn	2	15	40	50	64	
	mcs+knn	3	22	41	50	60	
pf	knn	1	100	21	38	60	
	cnn+knn	1	62	22	39	60	
	cliff+knn	2	11	9	20	64	
	psc+knn	2	15	36	50	61	
	mcs+knn	3	22	39	49	59	



Noisy Breast Cancer Results

bc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	knn	1	100	41	57	67	
	cliff+knn	1	11	39	57	82	
	mcs+knn	2	25	43	57	63	
	cnn+knn	2	71	44	54	65	
	psc+knn	3	19	37	49	64	
pf	cliff+knn	1	11	17	35	58	
	knn	1	100	32	40	53	
	cnn+knn	2	71	33	42	54	
	psc+knn	3	19	33	50	62	
	mcs+knn	2	25	37	44	54	

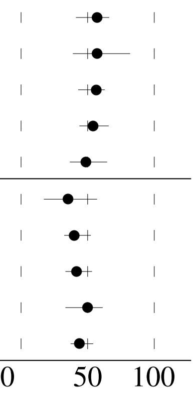
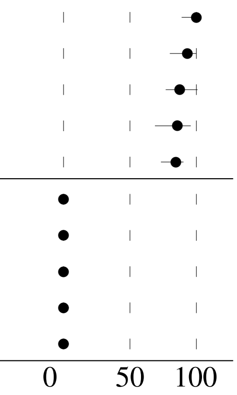


Figure 4.7: Clean and noisy results for breast cancer.

Clean Dermatology Results

dm	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	knn	1	100	89	100	100	
	cliff+knn	2	13	80	93	100	
	cnn+knn	3	27	77	88	100	
	psc+knn	4	10	69	86	96	
	mcs+knn	4	11	73	85	91	
pf	knn	1	100	0	0	2	
	cliff+knn	1	13	0	0	3	
	cnn+knn	1	27	0	0	3	
	psc+knn	1	10	0	0	5	
	mcs+knn	1	11	0	0	5	

Noisy Dermatology Results

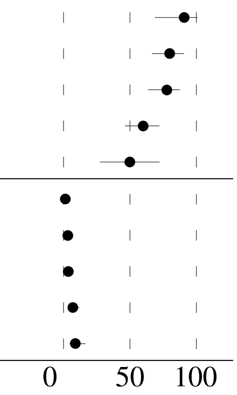
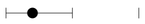
dm	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	13	69	91	100	
	cnn+knn	2	94	67	80	90	
	knn	2	100	64	78	88	
	mcs+knn	3	27	46	60	73	
	psc+knn	4	22	27	50	73	
pf	cliff+knn	1	13	0	1	6	
	cnn+knn	2	94	2	3	6	
	knn	2	100	2	4	8	
	mcs+knn	3	27	4	7	12	
	psc+knn	4	22	5	9	16	

Figure 4.8: Clean and noisy results for dermatology

Clean Heart (Cleveland) Results

hc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	psc+knn	1	42	9	26	42	
	mcs+knn	1	36	9	25	42	
	cnn+knn	1	67	0	20	50	
	cliff+knn	1	11	0	20	42	
	knn	1	100	0	20	40	
pf	cliff+knn	1	11	3	9	22	
	mcs+knn	2	36	5	10	25	
	cnn+knn	2	67	6	11	23	
	knn	2	100	6	11	23	
	psc+knn	3	42	7	14	23	
							0 50 100

Noisy Heart (Cleveland) Results

hc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	12	0	17	47	
	psc+knn	2	40	10	21	33	
	cnn+knn	2	86	0	17	38	
	mcs+knn	2	48	0	17	33	
	knn	3	100	0	18	33	
pf	cliff+knn	1	12	3	10	22	
	cnn+knn	2	86	8	13	23	
	knn	2	100	8	16	23	
	mcs+knn	3	48	9	14	26	
	psc+knn	4	40	10	18	25	
							0 50 100

Figure 4.9: Clean and noisy results for heart (Cleveland).

Clean Heart (Hungarian) Results

hh	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	9	68	82	90	●
	knn	1	100	65	75	83	●
	cnn+knn	1	65	57	74	85	●
	psc+knn	2	14	50	63	75	●
	mcs+knn	2	19	53	62	71	●
pf	cliff+knn	1	9	10	19	31	●
	knn	1	100	16	24	33	●
	cnn+knn	1	65	13	25	37	●
	mcs+knn	2	19	28	38	46	●
	psc+knn	2	14	28	38	48	●
							0 50 100

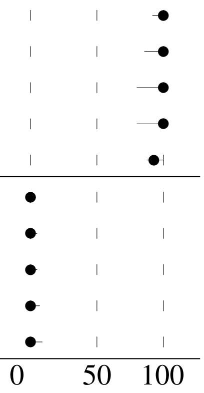
Noisy Heart (Hungarian) Results

hh	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	7	68	79	89	●
	cnn+knn	2	76	60	65	72	●
	knn	2	100	58	64	69	●
	mcs+knn	2	25	51	59	68	●
	psc+knn	3	19	38	53	68	●
pf	cliff+knn	1	7	11	21	32	●
	knn	2	100	29	35	41	●
	cnn+knn	2	76	28	36	41	●
	mcs+knn	2	25	28	37	47	●
	psc+knn	3	19	28	44	61	●
							0 50 100

Figure 4.10: Clean and noisy results for heart (Hungarian)

Clean Iris Results

ir	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	knn	1	100	92	100	100	
	psc+knn	1	9	86	100	100	
	cliff+knn	1	15	80	100	100	
	mcs+knn	1	4	80	100	100	
	cnn+knn	1	14	88	93	100	
pf	knn	1	100	0	0	4	
	psc+knn	1	9	0	0	6	
	cnn+knn	1	14	0	0	6	
	cliff+knn	1	15	0	0	7	
	mcs+knn	1	4	0	0	9	



Noisy Iris Results

ir	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	13	56	78	100	
	knn	2	100	69	83	92	
	cnn+knn	3	33	38	67	80	
	mcs+knn	3	10	42	63	78	
	psc+knn	4	18	30	50	67	
pf	cliff+knn	1	13	0	5	19	
	knn	2	100	0	9	14	
	mcs+knn	3	10	6	16	25	
	cnn+knn	3	33	9	17	29	
	psc+knn	3	18	10	25	38	

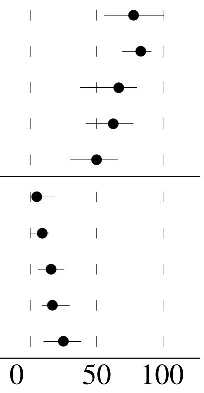
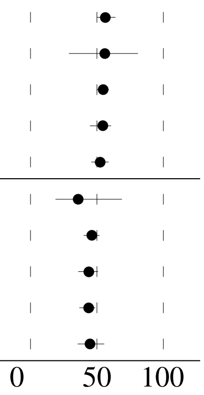


Figure 4.11: Clean and noisy results for iris.

Clean Liver (Bupa) Results

lv	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	knn	1	100	50	56	64	
	cliff+knn	1	9	29	56	80	
	mcs+knn	1	25	50	55	60	
	psc+knn	1	13	45	55	60	
	cnn+knn	1	59	46	53	58	
pf	cliff+knn	1	9	19	36	68	
	cnn+knn	2	59	40	46	53	
	knn	2	100	36	44	50	
	mcs+knn	3	25	37	44	48	
	psc+knn	3	13	36	45	55	



Noisy Liver (Bupa) Results

lv	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	mcs+knn	1	26	44	55	61	
	knn	1	100	44	54	60	
	cnn+knn	1	57	47	54	59	
	psc+knn	1	15	42	53	58	
	cliff+knn	1	9	30	48	74	
pf	cliff+knn	1	9	21	42	68	
	knn	1	100	38	44	54	
	mcs+knn	1	26	39	45	54	
	psc+knn	1	15	41	47	57	
	cnn+knn	1	57	41	48	54	

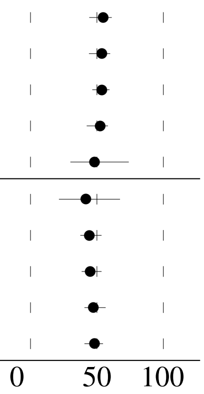


Figure 4.12: Clean and noisy results for liver (Bupa).

Clean Mammography Results

mm	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	8	49	62	77	
	cnn+knn	2	57	47	54	61	
	knn	3	100	45	53	57	
	mcs+knn	3	17	48	52	57	
	psc+knn	4	10	44	50	56	
pf	cliff+knn	1	8	21	36	45	
	cnn+knn	2	57	39	46	52	
	knn	3	100	41	46	51	
	mcs+knn	3	17	41	48	52	
	psc+knn	4	10	43	47	55	
							0 50 100

Noisy Mammography Results

mm	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	9	51	63	76	
	mcs+knn	2	19	44	51	57	
	psc+knn	2	11	43	50	60	
	cnn+knn	3	61	35	48	57	
	knn	4	100	34	46	53	
pf	cliff+knn	1	9	22	37	52	
	mcs+knn	2	19	39	48	54	
	psc+knn	2	11	41	50	57	
	cnn+knn	3	61	42	51	65	
	knn	4	100	44	54	63	
							0 50 100

Figure 4.13: Clean and noisy results for mammography

Chapter 5

Case Study: Solving the Problem of Brittleness in Forensic Models Using CLIFF

5.1 Introduction

The results in Chapter 4 are a good indication that CLIFF can be beneficial in the field of forensic interpretation. To explore the use of CLIFF as part of a proposed forensic model for the interpretation of trace evidence, a case study is conducted involving spectra data collected from the clear coat paint of cars. Specifically, CLIFF is used as a means to reduce brittleness in our proposed forensic model.

The principal goal of forensic interpretation models is to check that evidence found at a crime scene is (dis)similar to evidence found on a suspect. In creating these models, attention is given to the significance level of the solution however the *brittleness* level is never considered. The *brittleness* level is a measure of whether a solution comes from a region of similar solutions or from a region of dissimilar solutions. We contend that a solution coming from a region with a low level of brittleness i.e. a region of similar solutions, is much better that one from a high level of brittleness - a region of dissimilar solutions. This is because, intuitively, a solution from low brittleness is less likely to signal a false alarm.

The concept of *brittleness* is not a stranger to the world of forensic science, in fact it is recognized as the “fall-off-the-cliff-effect”, a term coined by Ken Smalldon. In other words, Smalldon recognized that tiny changes in input data could lead to a massive change in the output. Although Walsh [39] worked on reducing the brittleness in his model, to the best of our knowledge, no work been done to quantify brittleness in current forensic models or to recognize and eliminate the causes of brittleness in these models.

In our studies of forensic models for evaluation particularly in the sub-field of glass forensics, we conjecture that brittleness is caused by the following:

1. A tiny error(s) in the collection of data;

2. Inappropriate statistical assumptions, such as assuming that the distributions of the refractive index of glass collected at a crime scene or a suspect obeys the properties of a normal distribution;

3. and the use of measured parameters from surveys to calculate the *frequency of occurrence* of trace evidence in a population

In this study we quickly eliminate the two(2) latter causes of brittleness by using a simple classification method, k-nearest neighbor (KNN) which are neither concerned with the distribution of data nor the frequency of occurrence of the data in a population. To reduce the effects of errors in data collection, CLIFF is used to augment KNN (we refer to this combination as *the CLIFF Avoidance Model - CAM*). As explained in Chapter 3, CLIFF selects samples from the data which best represents the region or neighborhood it comes from. In other words, we expect that samples which contain errors would be poor representatives and would therefore be eliminated from further analysis. This leads to neighborhoods with different outcomes being further apart from each other. An example of this is shown in the $Before$ and $After$ CLIFF charts repeated here in Figure 5.1.

In the end our goal for this case study is threefold. First we want to develop a new generation of forensic models which avoids inappropriate statistical assumptions. Second, the new models must not be *brittle*, so that they do not change their interpretation without sufficient evidence and third, provide not only an interpretation of the evidence but also a measure of how reliable the interpretation is, in other words, what is the brittleness level of the model.

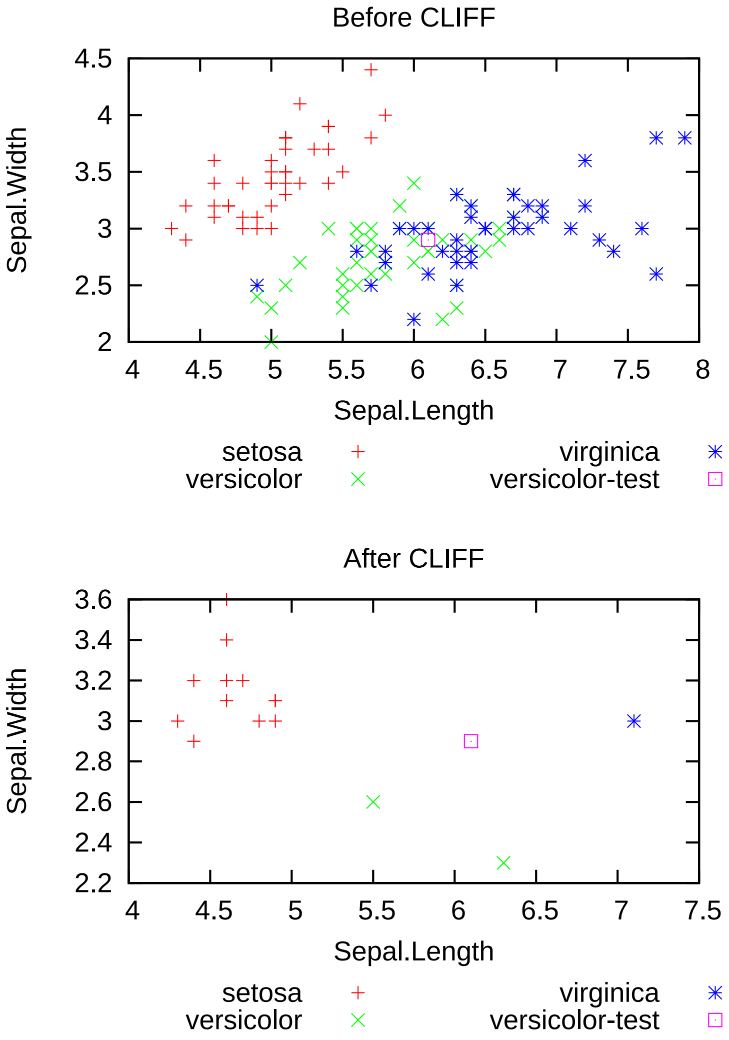


Figure 5.1

Our research is guided by the following research questions:

- Is CAM a strong forensic model?

- Does CAM reduce brittleness?

5.2 Motivation

This work is in part motivated by a recent National Academy of Sciences report titled “Strengthening Forensic Science” [35]. This report took special notice of forensic interpretation models stating:

> With the exception of nuclear DNA analysis, ...no forensic method has been rigorously shown to have the capacity to consistently, and with a high degree of certainty, demonstrate a connection between evidence and a specific individual or source. [35]

The concern voiced in that statement is exactly what CLIFF is meant to alleviate: Solving the problem of lack of *consistency* by reducing brittleness. Before exploring our proposed model, CAM, we will look at four(4) of the early standard glass forensic model which are prone to high levels of brittleness.

5.2.1 Glass Forensic Models

This section provides an overview of the following glass forensic models used in this work to show brittleness.

1. The 1978 Seheult model [36]

2. The 1980 Grove model [21]

3. The 1995 Evett model [15]

4. The 1996 Walsh model [39]

Seheult 1978

Seheult [36], examines and simplifies Lindley's [31] 6th equation for real-world application of Refractive Index (RI) Analysis. According to Seheult:

> A measurement x, with normal error having known standard deviation σ, is made on the unknown refractive index Θ_1 of the glass at the scene of the crime. Another measurement y, made on the glass found on the suspect, is also assumed to be normal but with mean Θ_2 and the same standard deviation as x. The refractive indices Θ are assumed to be normally distributed with known mean μ and known standard deviation τ. If I is the event that the two pieces of glass come from the same source($\Theta_1 = \Theta_2$) and $\bar{I}$ the contrary event, Lindley suggests that the odds on identity should be multiplied by the factor
>
> $$\frac{p(x,y|I)}{p(x,y|\bar{I})} \tag{5.1}$$
>
> In this special case, it follows from Lindley's 6th equation that the factor is
>
> $$\frac{1+\lambda^2}{\lambda(2+\lambda^2)^{1/2}}{}^{-\frac{1}{2(1+\lambda^2)}\cdot(u^2-v^2)} \tag{5.2}$$
>
> Where
>
> $$\lambda = \frac{\sigma}{\tau}, u = \frac{x-y}{\sigma\sqrt{2}}, v = \frac{z-\mu}{\tau(1+\frac{1}{2}\lambda^2)^{\frac{1}{2}}}, z = \frac{1}{2}(x+y)$$

Grove 1980

By adopting a model used by Lindley and Seheult, Grove proposed a non-Bayesian approach based on likelihood ratios to solve the forensic problem. The problem of deciding whether the fragments have come from common source is distinguished from the problem of deciding the guilt or innocence of the suspect. To explain his method, Grove first reviewed Lindley's method. He argued that we should, where possible, avoid parametric assumptions about the underlying distributions.

Hence, in discussing the respective roles of θ_1 and θ_2 Grove did not attribute any probability distribution to an unknown parameter without special justification. So when considering (θ_1 != θ_2), $\bar{I}$ can be interpreted as saying that the fragments are present by chance entailing a random choice of value for θ_2. The simplified likelihood ratio obtained from the Grove's derivation is:

$$\frac{\tau}{\sigma} \cdot e^{\left\{ \frac{-(X-Y)^2}{4\sigma^2} + \frac{(Y-\mu^2)}{2\tau^2} \right\}} \tag{5.3}$$

We are of course only concerned with the evidence about I and $\bar{I}$ so far as it has the bearing on the guilt or innocence of the suspect. Grove also considered the Event of Guilty factor $\underline{G}$ in the calculation of likelihood ratio (LR). Therefore the LR now becomes

$$p(X,Y|G)/p(X,Y|\bar{G}) \tag{5.4}$$

Here in the expansion event $\underline{T}$, that fragments were transferred from the broken window to the suspect and persisted until discovery and event $\underline{A}$,that the suspect came into contact with glass from other source. Here p(A/G)=p(A/$\bar{G}$)=Pa and p(T/G)= Pt. The resulting expression is

$$\frac{P(X,Y,S|G)}{P(X,Y,S|\bar{G})} = 1 + Pt\left\{ \left(\frac{1}{Pa} - 1\right) \frac{p(X,Y|I)}{p(X,Y|\bar{I})} - 1 \right\} \tag{5.5}$$

Evett 1995

Evett et al used data from forensic surveys to create a Bayesian approach in determining the statistical significance of finding glass fragments and groups of glass fragments on individuals associated with a crime [15].

Evett proposes that likelihood ratios are well suited for explaining the existence of glass fragments on a person suspected of a crime. A likelihood ratio is defined in the context of this paper as the ratio of the probability that the suspected person is guilty given the existing evidence to the probability that the suspected person is innocent given the existing evidence. The given evidence, as it applies to Evett's approach, includes the number of different sets of glass and the number of fragments in each unique group of glass.

The Lambert, Satterthwaite and Harrison (LSH) survey used empirical evidence to supply probabilities relevant to Evett's proposal. The LSH survey inspected individuals and collected glass fragments from each of them. These fragments were placed into groups based on their refractive index (RI) and other distinguishing physical properties. The number of fragments and the number of sets of fragments were recorded, and the discrete probabilities were published. In particular, there are two unique probabilities that are of great interest in calculating Evett's proposed likelihood ratio.

- S, the probability of finding N glass *fragments* per group
- P, the probability of finding M *groups* on an individual.

The following symbols are used by Evett to express his equations:

- P_n is the probability of finding n groups of glass on the surface of a person's clothes
- T_n is the probability that n fragments of glass would be transferred, retained and found on the suspect's clothing if he had smashed the scene window
- S_n is the probability that a group of glass fragments on a person's clothing consists of n fragments
- f is the probability that a group of fragments on person's clothing would match the control sample
- λ is the expected number of glass fragments remaining at a time, t

Evett utilizes the following equations to determine the likelihood ratio for the first case described in his 1994 paper. In this case, a single window is broken, and a single group of glass fragments is expected to be recovered.

$$LR = \frac{P_0 T_n}{P_1 S_n f} + T_0 \tag{5.6}$$

$$T_n = \frac{e^{-\lambda}\lambda^n}{n!} \tag{5.7}$$

Walsh 1996

The equation presented by Walsh [39] is similar to one of Evett's. The difference is that Walsh argues that instead of incorporating grouping and matching, only grouping should be included. Walsh says this is because match/non-match is really just an arbitrary line. He examines the use of a technique in interpreting glass evidence of a specific case. This technique is as follows:

$$\frac{T_L P_0 p(\bar{X}, \bar{Y} | S_y, S_x)}{P_1 S_L f_1} \tag{5.8}$$

Where

- T_L = the probability of 3 or more glass fragments being transferred from the crime scene to the person.
- P_0 = the probability of a person having no glass on their clothing
- P_1 = the probability of a person having one group of glass on their clothing
- S_L = the probability that a group of glass on clothing is 3 or more fragments
- $\bar{X}$ and $\bar{Y}$ are the mean of the control and recovered groups respectively
- S_x and S_y are the sample standard deviations of the control and recovered groups respectively
- f_1 is the value of the probability density for glass at the mean of the recovered sample

- $p(\bar{X},\bar{Y}|S_y,S_x)$ is the value of the probability density for the difference between the sample means

5.2.2 Visualization of Brittleness in These Models

The visualization of the model described above are shown in Figure 5.2. For the first two(2) models the x and y axes represent the mean refractive index (RI) values of evidence from a crime scene and suspect respectively. While the x axis of the Walsh model represents $f1$ is the value of the probability density for glass at the mean of the recovered sample and the y axis represents the value of the probability density for the difference between the sample means. The x and y axes of the Evett model represents λ and $f-values$ respectively. The green, red and blue points in the charts represent the likelihood ratios (LR) generated from these models, in other words, the significance of the match/non-match of evidence to an individual or source. Here the *green* and *blue* points have a LR of 10 (match) while the *red* points have a LR of 0 (non-match).

Using data donated by the Royal Canadian Mounted Police (RCMP), values such as the RI ranges and their mean, were extracted to generate random samples for the forensic glass models. In all four(4) models 1000 samples are randomly generated for the variables in each model. For instance, in the Seheult model, each sample looks like this: [x, y, σ, μ, τ]. The symbols are explained in 5.2.1.

In Figure 5.2 - the Seheult and Grove models, brittleness or Smalldon's "fall-off-the-cliff-effect" is clearly demonstrated. These models proposed by Seheult (Section 5.2.1) and Grove (5.2.1) respectively, show how the likelihood ratio changes as we try different values from the refractive index of from glass from two sources (x and y). This model could lead to incorrect interpretations if minor errors are made when measuring the refractive index of glass samples taken from a suspect's clothes. Note how, for both Seheult and Grove, how tiny changes in the blue points can lead to a dramatic change in the likelihood ratio from ten(10) to zero(0).

The Evett model shows similar brittleness as in the Seheult and Grove models. However, Walsh [39] is the clear exception. As mentioned earlier, Walsh's model is the only one of these models which attempted to reduce *brittleness*. As shown in Figure 5.2 the result of this is the creation of a clear boundary between the LR. Hence *brittleness* only exists at the boundary.

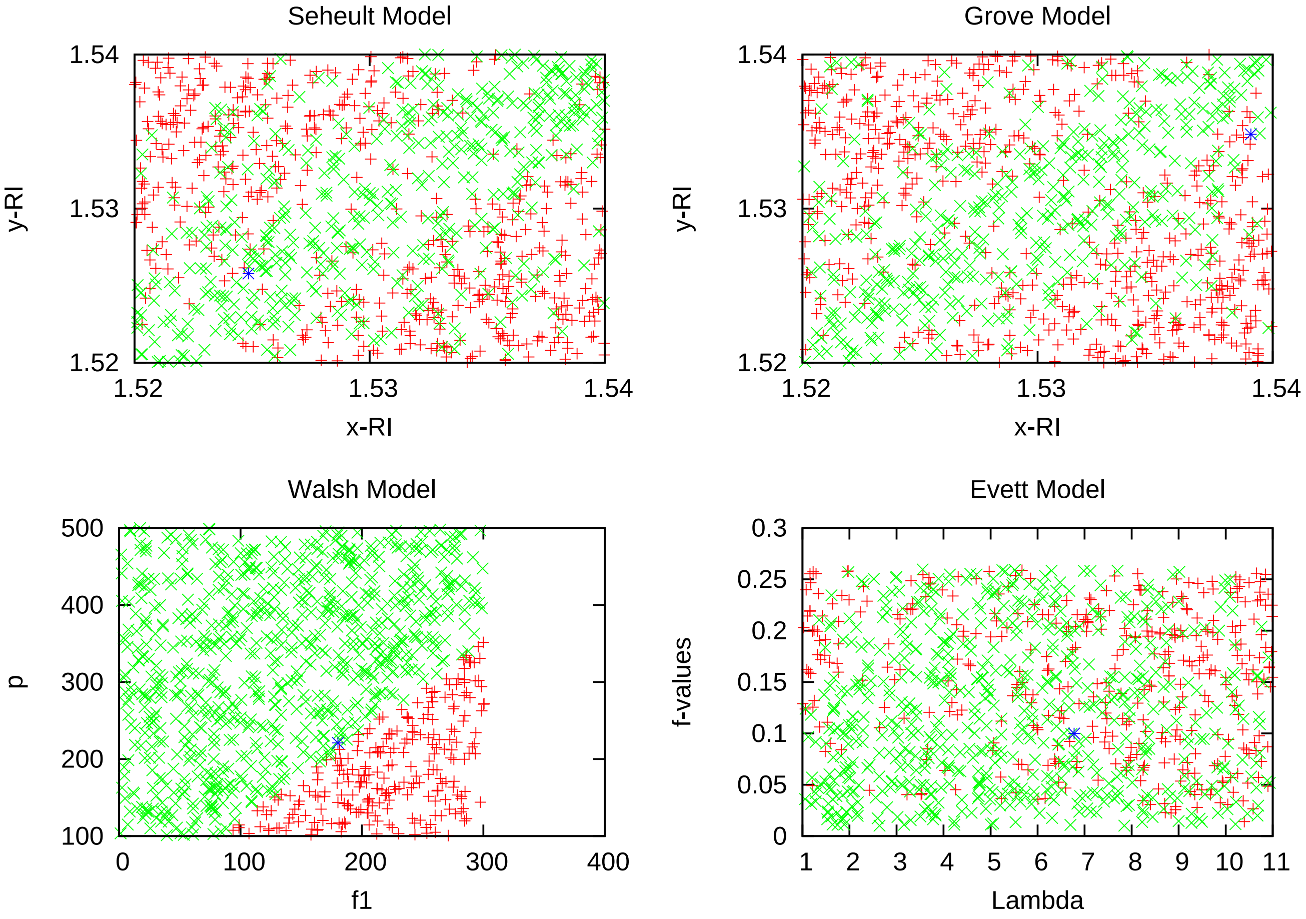


Figure 5.2: Visualization of four(4) glass forensic models.

From these visualizations it is obvious that the concern of the National Academy of Sciences report [35] mentioned earlier in this section is a valid one. So how can this concern be alleviated? We propose not only including a *brittleness* measure to a forensic method as a solution, but also moving away from forensic models which use surveys [13–15, 36, 39], and statistical assumptions [21, 36, 39].

The following sections gives details of CAM (the CLIFF Avoidance Model) as well as the data set used to evaluate the models.

5.3 The CLIFF Avoidance Model (CAM)

If standard forensic interpretation models are brittle what can we do?

We the found the answer to this question in the work of [32], and also in our exploration of the intuition that to reduce *brittleness*, data with dissimilar outcomes should not be close neighbors. To that end we introduce CAM, a forensic interpretation model designed to reduce *brittleness* by avoiding inappropriate statistical assumptions, and present a measure of how strong the model is.

The Design of CAM is deeply rooted in the work of [32]. In their study, analysis is done using chemometrics, an application of mathematical, statistical and/or computer science techniques to chemistry. The chemometric analysis done by [32], uses computer science techniques to analyze the absorbance spectra of the clear coat layer of a range of cars. The analysis proceeded as follows:

- Agglomerative hierarchical clustering (AHC) for grouping the data into classes
- Principal component analysis (PCA) for reducing dimensions of the data
- Discriminant analysis for classification i.e. associating an unknown sample to a group or region

This technique produced a strong model which achieved an overall classification accuracy of 91.61% . This encouraging result and other insights gained from this study led to the design of CAM. The goal of CAM is not only to create a strong forensic model but also to show how strong the model is. To achieve these goals, CAM includes a brittleness measure as well as a method to reduce brittleness - CLIFF. Also, in an effort to keep CAM simple, we substituted different tools to preform the analysis done in [32]. For instance $K - means$ is used instead of AHC for grouping the data into classes. $FastMap$ [16] is used for dimensionality reduction and K-nearest neighbor is used for classification. The basic operation of CAM is shown in Figure 5.3. The data is collected and the dimensions is reduced if necessary. Clusters are then created from the data and classification is done along with a brittleness measure (further discussed in Section 5.3.4). Finally, we test if brittleness can be reduced by CLIFF (Instance Selection).

The details of CAM are discussed below.

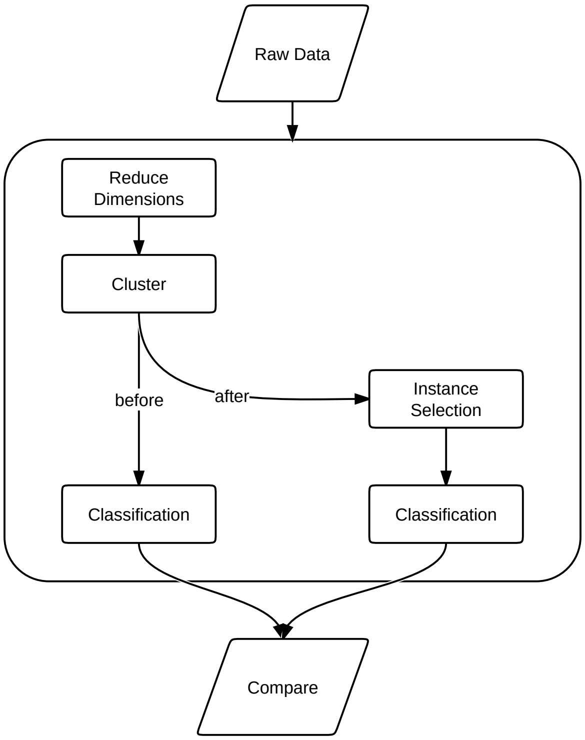


Figure 5.3: Proposed procedure for the forensic evaluation of data

5.3.1 Dimensionality Reduction

The data used in our experiments contains 1151 attributes and 185 instances. Using the data set as is would cause us to create a model that is computationally expensive and likely to produce unacceptable results such as a high false positive values caused by redundant and noisy data. To avoid this foreseen problem, we turn to dimensionality reduction.

Dimensionality Reduction refers to reducing high-dimensional data to low dimensional data. This is accomplished by attempting to summarize the data by using less terms than needed. While this reduces the overall information available and thus a level of precision, it allows for easy visualization of data otherwise impossible to visualize. Some algorithms that can be used for Dimensionality Reduction include Principle Component Analysis (PCA), and FastMap. These are discussed below.

Principal Component Analysis

PCA can be defined as "the orthogonal projection of the data onto a lower dimensional linear space". In other words, looking at our data set, our goal is to project the data onto a space having dimensionality that is less than 1,151 ($M < 1{,}151$) while maximizing the variance of the projected data [6]. The result of this serves two main purposes:

1. To simplify analysis and

2. To aid in the visualization of the data

To achieve this goal, the data set is transformed to a new set of variables which are not correlated and which are ordered so that the first few principal components (PCs) retain most of the variation present in all of the original variables [24]. Let us look at an example. Figure 5.4 shows a visualization of Fisher's five-dimensional iris data set on a scatter plot. First, PCs are extracted from the four continuous variables (sepal-width, sepal-length, petal-width, and petal-length). Second, these variables are projected onto the subspace formed by the first two components extracted. Finally this two-dimensional data is shown on a scatter-plot in Figure 5.4. The fifth dimension (species) is represented by the color of the points on the scatter-plot.

In the interest of speed, in CAM we use *FastMap* to reduce the dimensions of the data set to M=4.

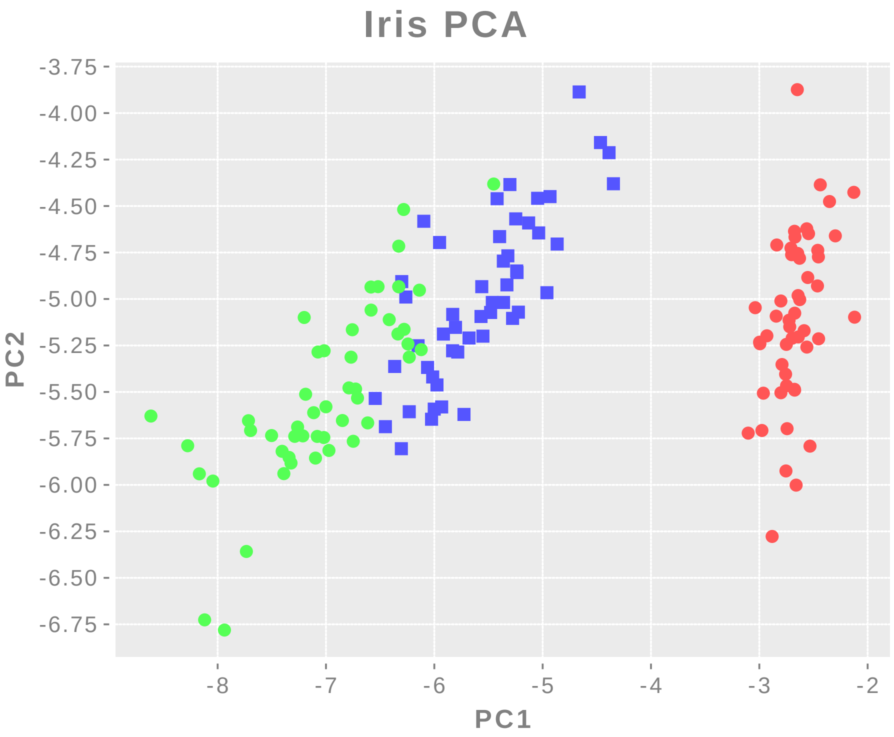


Figure 5.4: PCA for iris data set

FastMap

In FastMap [16], the basis of each reduction is using the cosine law on the triangle formed by an object in the feature space and the two objects that are furthest apart in the current (pre-reduction) space (see Figure 5.5). These two objects are referred to as the pivot objects of that step in the reduction phase (M total pivot object sets). Finding the optimal solution of the problem of finding the two furthest apart points is an N squared problem (where N is the total number of objects), but this is where the heuristic nature of FastMap comes into play. Instead of finding the absolute furthest apart points, FastMap takes a shortcut by first randomly selecting an object from the set, and then finding the object that is furthest from it and setting this object as the first pivot point. After the first pivot point is selected, FastMap finds the points farthest from this and uses it as the second pivot point. The line formed by these two points becomes the line that all of the other

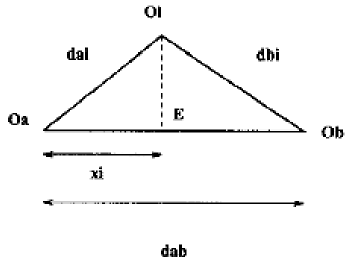


Figure 5.5: Example of using the cosine law to find the position of *Oi* in the dimension *k*

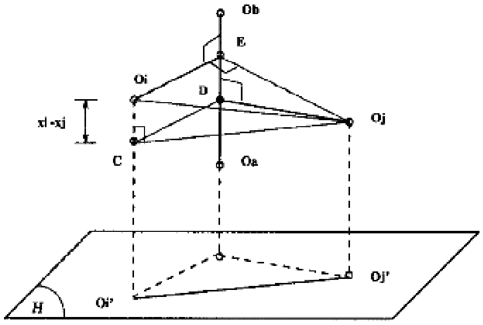


Figure 5.6: Projects of points O_i and O_j onto the hyper-plane perpendicular to the line O_aO_b

points will be mapped to in the new M dimension space. (Further details of this algorithm can be found elsewhere [16]).

FASTMAP uses the following equation to calculate x_i, or the position of object O_i in the reduced space:

$$x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{2d_{a,b}} \tag{5.9}$$

This technique can be visualized by imagining the hyper-plane perpendicular to the line formed by pivot points, O_a and O_b, and projecting the new point onto this plane (see Figure 5.6). FASTMAP requires only $2D$ passes over D documents.

5.3.2 Clustering

Clustering is the second step in CAM and can be defined as the grouping of the samples into groups whose members are similar in some way. The samples that belong to two different clusters are dissimilar. The major goal of clustering is to determine the intrinsic grouping in the set of unlabelled data. In most of the clustering techniques, distance is the major criteria. Two objects are similar if they are close according to the given distance.

CAM clusters using K-means. The Figure 5.7 represents the pseudo code for the K-means algorithm. The idea behind K-means clustering is done by assuming some arbitrary number of centroids, then the objects are associated to nearest centroids. The centroids are then moved to center of the clusters. These steps are repeated until a suitable level of convergence is attained.

```
k = [1, ..., Number of clusters]
STOP = [Stopping criteria]

FOR EACH instance IN DATA
 WHILE STOP IS FALSE
  // Calculate membership in clusters
  FOR EACH instance X IN DATA
   FIND NEAREST CENTROID_k
   ADD TO CLUSTER_k
  END

  // Recompute the centroids
  FOR EACH CLUSTER
   FIND NEW CENTROIDS
  END

  // Check stopping criteria
  [TRUE or FALSE] = STOP
 END
END
```

Figure 5.7: Pseudo code for K-means

5.3.3 Classification with KNN

K-nearest neighbor (KNN) classification is a simple classification method usually used when there is little or no prior knowledge about the distribution of data. KNN is described in [12] as follows: Stores the complete training data. New samples are classified by choosing the majority class among the k closest examples in the training data. For our particular problem, we used the Euclidean, i.e. sum of squares, distance to measure the distance between samples and k = 1.

5.3.4 The Brittleness Measure

Calculating the brittleness measure is a novel operation of CAM. We use the brittleness measure in this work to determine if the results of CAM comes from a region where all the possible results are (dis)similar. For the purpose of this work the optimal result will come from a region of similar results. To make this determination, using the 1NN classifier as a baseline for CAM, each instance from a test set is assigned to a target class. Next, the distances from their nearest unlike neighbor (NUN) i.e. the distance from an instance with a different class are recorded. The result is two(2) lists of distances for 1NN and CAM. Recall that brittleness is a small change can result in a different outcome, so here the closer the distances of CAM to 1NN the more brittle the model. So an ideal result will have the greatest distance between CAM and 1NN.

The brittleness measure will give an output of either *high* or *low*: high indicating that there is no significant difference between the CAM and 1NN distances, while *low* indicates the opposite. The significance of these values was calculated using the Mann-Whitney U test. This is a non-parametric test which replaces the distance values with their rank or position inside the population of all sorted values.

Equation 5.10 embodies our definition of brittleness: if the significance of CAM values are less than or equal to the 1NN values, then an unacceptable level of brittleness is present in the model.

$$[CAM <= 1NN] ==> BRITTLENESS \tag{5.10}$$

In this chapter, we evaluate CAM as a forensic model on a data set donated by [32] in cross validation experiments. First, we describe the data set and experimental procedures. Next we present results which show the probability of detection (pd), probability of false alarm (pf) and brittleness level of CAM.

5.4 Data Set and Experimental Method

The data set used in this work contains 37 samples each with five(5) replicates (37 x 5 = 185 instances). Each instance has 1151 infrared measurements ranging from 1800-650cm-1. For our experiments we took the original data set and created four (4) data sets each with a different number of clusters (3, 5, 10 and 20) or groups. These clusters were created using the K-means algorithm (Figure 5.7). The effectiveness of CAM is measured using pd, pf and brittleness level. The brittleness level measure is conducted as described in Section 5.3.4.

5.4.1 Experiment 1: CAM as a forensic model?

Our goal is to determine if CAM is an adequate model for forensic evaluation. In other words, can it be used in preference to current statistical models? To answer this question, our experiment design follows the pseudo code given in Figure 5.8 for the four (4) data sets created from the original data set. For each data set, tests were built from 20% of the data, selected at random. The models were learned from the remaining 80% of the data.

This procedure was repeated 5 times, randomizing the order of data in each project each time. In the end CAM is tested and trained 25 times for each data set.

```
DATA = [3, 5, 10, 20]
LEARNER = [1NN]

REPEAT 5 TIMES
 FOR EACH data IN DATA
  TRAIN = random 80% of data
  TEST = data - TRAIN

  \\Construct model from TRAIN data
  MODEL = Train LEARNER with TRAIN
  \\Evaluate model on test data
  [pd, pf] = MODEL on TEST
 END
END
```

Figure 5.8: Pseudo code for Experiment 1

Results from Experiment 1

Figure 5.9 shows the 25%, 50% and 100% percentile values of the *pd*, pf for 1NN and CAM as well as the *size* and *rank* results which indicate the percentage of the training set used and the statistical difference between 1NN and CAM respectively. The *pd* and pf results for both 1NN and CAM are promising with CAM showing that 50% of the pd values are at or above 91% for the data set with 3 clusters and at 100% for the other data sets, while 1NN shows all at 100%. For pf, 50% of the values for CAM are at 4% for 3 clusters and 0% for the others, while 1NN shows all at 0%. These results show that our model is highly discriminating and can be used successfully in the evaluation of trace evidence.

5.4.2 Experiment 2: Does CAM reduce brittleness?

The first experiment shows that 1NN and CAM creates strong models for forensic interpretation, with high pd's and low pf's. However there is no indication of whether or not CAM reduces

3	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	1nn	1	100	90	100	100	
	cam	1	25	63	91	100	
pf	1nn	1	100	0	0	4	
	cam	2	25	0	4	16	
							0 50 100

5	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	1nn	1	100	91	100	100	
	cam	1	30	88	100	100	
pf	1nn	1	100	0	0	3	
	cam	1	30	0	0	3	
							0 50 100

10	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	1nn	1	100	92	100	100	
	cam	1	36	67	100	100	
pf	1nn	1	100	0	0	0	
	cam	1	36	0	0	3	
							0 50 100

20	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	1nn	1	100	33	100	100	
	cam	1	53	75	100	100	
pf	1nn	1	100	0	0	0	
	cam	1	53	0	0	3	
							0 50 100

Figure 5.9: Results for Experiment 1 for the 4 data sets distinguished by the number of clusters 3, 5, 10 and 20.

brittleness. We accomplish this with Experiment 2. The design for this experiment can be seen in Figure 5.10. It is similar to that in Figure 5.8, however, CLIFF, the instance selector in CAM is included and is described in Chapter 3.

Results from Experiment 2

In Figure 5.11 and Figure 5.12 the pattern is clear. The distances of the test set for CAM have to move further away in order to change their predicted target classes than those of 1NN (see Figure 5.11. This result is further established by Figure 5.12 which shows that for all the data sets, the brittleness level is low, i.e. CAM's list of distances is significantly better than 1NN's list of

```
DATA = [3, 5, 10, 20]
LEARNER = [1NN]
STAT_TEST = [Mann Whitney]
SELECTOR = [CLIFF]

REPEAT 5 TIMES
 FOR EACH data IN DATA
  TRAIN = random 90% of data
  TEST = data - TRAIN

  \\CLIFF selector: select best from clusters
  N_TRAIN = SELECTOR with TRAIN

  \\Construct model from TRAIN data
  MODEL = Train LEARNER with N_TRAIN
  \\Evaluate model on test data
  [brittleness] = STAT_TEST on the CAM and 1NN lists of distances
  [pd, pf, brittleness] = MODEL on TEST
 END
END
```

Figure 5.10: Pseudo code for Experiment 2

distances.

5.4.3 Summary

In summary, by using CAM, inappropriate statistical assumptions about the data are avoided. We found a successful way to reduce any brittleness to create strong forensic interpretation models. One important point to note here also is this: In order to evaluate data sets with multiple attributes, a host of new statistical models has been built [1,2,28,29,41,42]. This has been the case with forensic scientists building these models for glass interpretation when using the elemental composition of glass rather than just the refractive index. On the other hand, with CAM an increase in the number of attributes used, does not signal the need to create a new model, it works with any data set.

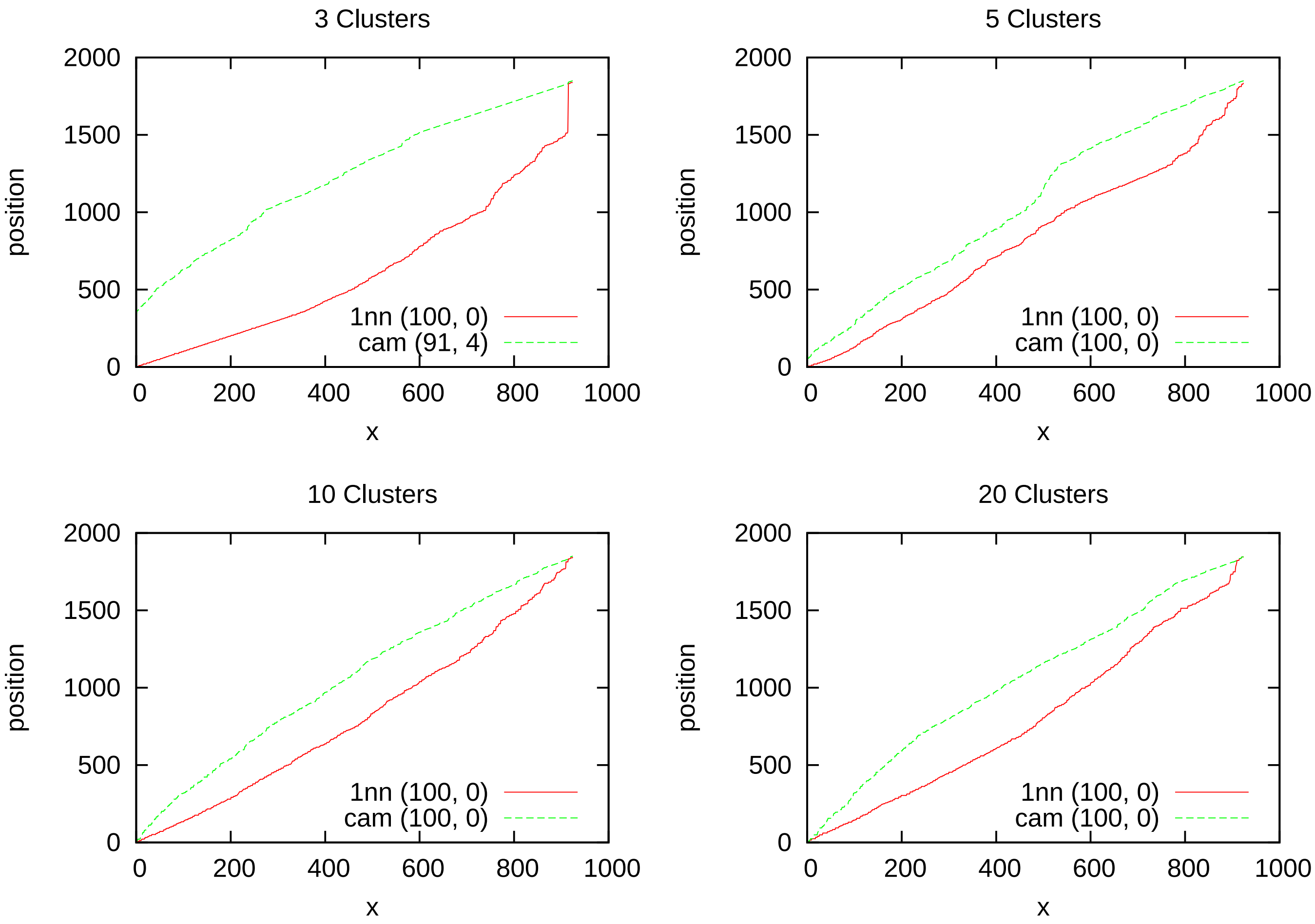


Figure 5.11: Position of values in 1NN and CAM population with data set at 3, 5, 10 and 20 clusters.

Clusters	Forensic Models	Significance
3	1nn cam	low
5	1nn cam	low
10	1nn cam	low
20	1nn cam	low

Figure 5.12: Summary of Mann Whitney U-test results for Experiment 2 (95% confidence): In the Significance column, *low* indicates that CAM is better than just 1NN

Chapter 6

Conclusions and Future Work

This chapter summarizes the thesis and suggest directions for future research.

6.1 Summary of the Thesis

Our goal in this thesis has been to evaluate the hypothesis that *there are critical ranges of values for each attribute of a data set which can be used to select instances from the data set to be used as prototypes to improve the performance of nearest neighbor classifiers*. To this end we have presented a prototype learning scheme called CLIFF, that uses a ranking algorithm to create a criteria for each class in a data set. These criterion are then used to select instances as prototypes. We showed that CLIFF:

1. has a time complexity of $O(n)$;
2. reduces training sets to a range of 9 to 15%;
3. has *pd* and pf results which compares favorably with 1NN and other PLS in several standard data sets;

4. does not significantly increase the number of instance selected in the presence of noise as compared with other PLS;

5. reduces *brittleness* substantial in most data sets used.

In response to the concern of the National Academy of Sciences (NAS) which stated in a report [35] that, "...no forensic method has been rigorously shown to have the capacity to consistently, and with a high degree of certainty, demonstrate a connection between evidence and a specific individual or source", we also designed and implemented a forensic interpretation model called *the CLIFF Avoidance Model*, CAM, which successfully uses 1NN as the baseline model, and CLIFF to reduce its *brittleness* in order to achieve the desired consistency of forensic interpretation. In other words, with CAM, for an interpretation to change, there must be sufficient evidence.

6.2 Future Work

6.2.1 Using CLIFF with Other Classifiers

Although CLIFF was created to eliminate the drawbacks of nearest neighbor classifiers, the next step would be to try CLIFF with other standard classifiers particularly decision tree learners which have a tendency to produce complex theories [33]. We conjecture that reducing the size to the training set with CLIFF beforehand will produce shorter trees and therefore simpler theories.

6.2.2 Using CLIFF to Optimized Feature Subset Selection

In Chapter 5 we reduced the dimensionality of the data set used with FastMap [16]. With the new data set CLIFF was still able to produce favorable results. It would be interesting to see if the same can happen using feature subsets.

Bibliography

[1] CGG Aitken and D Lucy. Evaluation of trace evidence in the form of multivariate data. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY SERIES C-APPLIED STATISTICS*, 53(Part 1):109–122, 2004.

[2] CGG Aitken, D Lucy, G Zadora, and JM Curran. Evaluation of transfer evidence for three-level multivariate data with the use of graphical models. *COMPUTATIONAL STATISTICS & DATA ANALYSIS*, 50(10):2571–2588, JUN 20 2006.

[3] JC Bezdek and LI Kuncheva. Some notes on twenty one (21) nearest prototype classifiers. In Ferri, FJ and Inesta, JM and Amin, A and Pudil, P, editor, *ADVANCES IN PATTERN RECOGNITION*, volume 1876 of *LECTURE NOTES IN COMPUTER SCIENCE*, pages 1–16. 2000.

[4] JC Bezdek and LI Kuncheva. Nearest prototype classifier designs: An experimental study. *INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS*, 16(12):1445–1473, DEC 2001.

[5] J.C. Bezdek, T.R. Reichherzer, G.S. Lim, and Y. Attikiouzel. Multiple-prototype classifier design. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(1):67–79, Feb 1998.

[6] C.M Bishop. *Pattern Recognition and Machine Learning*. New York, NY, Springer, 2006.

[7] Jos Ramn Cano, Francisco Herrera, and Manuel Lozano. Stratification for scaling up evolutionary prototype selection. *Pattern Recognition Letters*, 26(7):953 – 963, 2005.

[8] Chin-Liang Chang. Finding prototypes for nearest neighbor classifiers. *Computers, IEEE Transactions on*, C-23(11):1179–1184, Nov. 1974.

[9] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, Jan 1967.

[10] B.V. Dasarathy. Minimal consistent set (mcs) identification for optimal nearest neighbor decision systems design. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(3):511–517, Mar 1994.

[11] V. Susheela Devi and M. Narasimha Murty. An incremental prototype set building technique. *Pattern Recognition*, 35(2):505 – 513, 2002.

[12] Richard O. Duda and Peter E.Hart. *Pattern classification and scene analysis*. A Wiley-Interscience Publication, New York: Wiley, 1973.

[13] Ian Evett. A quantitative theory for interpreting transfer evidence in criminal cases. *Applied Statistics*, 33(1):25–32, 1984.

[14] Ian Evett and John Buckleton. The interpretation of glass evidence. a practical approach. *Journal of the Forensic Science Society*, 30(4):215–223, 1990.

[15] Ian Evett and J. Lambert. Further observations on glass evidence interpretation. *Science and Justice*, 35(4):283–289, 1995.

[16] Christos Faloutsos and King-Ip Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 163–174, New York, NY, USA, 1995. ACM.

[17] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[18] Utpal Garain. Prototype reduction using an artificial immune model. *Pattern Anal. Appl.*, 11(3-4):353–363, 2008.

[19] Salvador Garca, Jos Ramn Cano, and Francisco Herrera. A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition*, 41(8):2693 – 2709, 2008.

[20] G. Gates. The reduced nearest neighbor rule (corresp.). *Information Theory, IEEE Transactions on*, 18(3):431 – 433, 1972.

[21] D.M. Grove. Interpretation of forensic evidence using a likelihood ratio. *Biometrika*, 67(1):243–246, April 1980.

[22] P. Hart. The condensed nearest neighbor rule (corresp.). *Information Theory, IEEE Transactions on*, 14(3):515 – 516, may 1968.

[23] O. Jalali, T. Menzies, and M. Feather. Optimizing requirements decisions with keys. In *Proceedings of the PROMISE 2008 Workshop (ICSE)*, 2008. Available from `http://menzies.us/pdf/08keys.pdf`.

[24] I. Jolliffe. *Principal component analysis*. Springer-Verlag, 175 Fifth Avenue, NY, USA, 2002.

[25] SW Kim and BJ Oommen. A brief taxonomy and ranking of creative prototype reduction schemes. *PATTERN ANALYSIS AND APPLICATIONS*, 6(3):232–244, DEC 2003.

[26] T. Kohonen. Improved versions of learning vector quantization. pages 545 –550 vol.1, jun 1990.

[27] Teuvo Kohonen and Panu Somervuo. Self-organizing maps of symbol strings. *Neurocomputing*, 21(1-3):19 – 30, 1998.

[28] RD Koons and J Buscaglia. Interpretation of glass composition measurements: the effects of match criteria on discrimination capability. *JOURNAL OF FORENSIC SCIENCES*, 47(3):505–512, MAY 2002.

[29] RD Koons and JA Buscaglia. The forensic significance of glass composition and refractive index measurements. *JOURNAL OF FORENSIC SCIENCES*, 44(3):496–503, MAY 1999.

[30] Y Li, M Xie, and T Goh. A study of project selection and feature weighting for analogy based software cost estimation. *Journal of Systems and Software*, 82:241–252, 2009.

[31] DV LINDLEY. PROBLEM IN FORENSIC-SCIENCE. *BIOMETRIKA*, 64(2):207–213, 1977.

[32] E. Liszewski, S. Lewis, J. Siegel, and J. Goodpaster. Characterisation of automotive paint clear coats by ultraviolet absorption microspectrophotometry with subsequent chemometric analysis. 2010.

[33] T. Menzies and Y. Hu. Just enough learning (of association rules): The TAR2 treatment learner. In *Artificial Intelligence Review*, 2007. Available from `http://menzies.us/pdf/07tar2.pdf`.

[34] J. Olvera-Lpez, J. Carrasco-Ochoa, and J. Martnez-Trinidad. A new fast prototype selection method based on clustering. *Pattern Analysis amp; Applications*, 13:131–141, 2010. 10.1007/s10044-008-0142-x.

[35] Committee on Identifying the Needs of the Forensic Sciences Community;Committee on Applied and National Research Council. Theoretical Statistics. *Strengthening Forensic Science*

in the United States: A Path Forward. National Academies Press, 500 Fifth Street, N.W., Lockbox 285, Washington, DC 20055, 2009.

[36] Allan Seheult. On a problem in forensic science. *Biometrika*, 65(3):646–648, December 1978.

[37] Burak Turhan, Tim Menzies, Ayse B. Bener, and Justin Di Stefano. On the relative value of cross-company and within-company data for defect prediction. *EMPIRICAL SOFTWARE ENGINEERING*, 14(5):540–578, OCT 2009.

[38] CJ Veenman and MJT Reinders. The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 27(9):1417–1429, SEP 2005.

[39] K. Walsh, J. Buckleton, and C. Triggs. A practical example of the interpretation of glass evidence. *Science and Justice*, 36(4):213–218, 1996.

[40] D. Randall Wilson and Tony R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000. 10.1023/A:1007626913721.

[41] G. Zadora. Evaluation of evidence value of glass fragments by likelihood ratio and Bayesian Network approaches. *ANALYTICA CHIMICA ACTA*, 642(1-2, Sp. Iss. SI):279–290, MAY 29 2009.

[42] G. Zadora and T. Neocleous. Likelihood ratio model for classification of forensic evidence. *ANALYTICA CHIMICA ACTA*, 642(1-2, Sp. Iss. SI):266–278, MAY 29 2009.