CLIFF: Tools for 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: Tools for 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 neighbour 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, a collection of new prototype learning schemes (CLIFF1, CLIFF2 and CLIFF3) are described. The basic structure of the CLIFF algorithms 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 represents the region or neighborhood it comes from and so are expected to eliminate the drawbacks of NNC particularly 3, 4 and 5 above.

With 13 standard data sets from the UCI repository [17], the results of this work demonstrate that CLIFF presents results which are statistically the same as those from NNC. 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 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 Neighbour 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*. *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, 26]. However many of these schemes suffer from at least one of the following disadvantages:

- 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', a conjunction of contraints is generated for each target class. These conjuctions are created using a ranking algoritm call BORE (Best Or Rest) [23], which basically finds a range of values 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 all the constraints is or are selected as prototypes. Using this structure the percentage reduction of the training set is related to the number of constraints used on the prototype selection process, in that the more constraints used the lower the percentage reduction. So there is no need to predetermine the number of prototypes desired when using CLIFF.

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]. Our experiments also use FastMap [16] [1] and Feature Subset Selection (FSS) to avoid the curse of dimensionality [31] which negatively affects Nearest Neighbor Classifiers (NNC).

[1] A Fast Algorithm for Indexing, Data Mining, and Visualization of Traditional and Multimedia Datasets

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. 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.

1.1 Statement of Thesis

1.2 Contributions of this Thesis

1.3 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 describes the data sets used along with the preprocessing tools used to avoided the curse of dimensionality [31]
- Chapter 5 presents a detailed description of the experimental procedure followed to analyze the data using CLIFF
- Chapter 6 examines a case study in which CLIFF is used to reduce the *brittleness* of a forensic model
- Chapter 7 conclusions 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. Also included in this chapter, are the evaluation methods generally used to gage the performance of different PLS.

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 [4,5,7,8,10,11,18,19,27,28,32,37]. 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. The following section gives a brief survey of PLS for both categories.

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 *bottom-down* 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 2.1 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

```
LEARNER = 1NN
TRAIN = 80% DATA
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 2.1: Pseudo-code for CNN

Reduced Nearest Neighbor (RNN)

RNN [20] takes an opposite approach to CNN. Its strategy is *top-down*. 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 2.2 shows the pseudo-code for RNN.

```
TRAIN = 80% DATA
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 2.2: 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 2.3 shows the pseudo-code for MCS.

2.1.2 Instance Abstraction

Chang

Changs work deals with finding prototypes for nearest neighbor classifiers. 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. 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 data set starts to increase. Figure 2.4, shows Changs algorithm.

```
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 2.3: Pseudo-code for MCS

2.2 Evaluation of Prototype Learning Schemes

2.2.1 Storage Reduction

2.2.2 Speed Increase

2.2.3 Generalization Accuracy

2.2.4 Noise Tolerance

2.2.5 Probability of Detection and False Alarm

```
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 daistance is smaller
        than the present bk. Otherwise, bk is
        unchanged.
Step 3: Amoung 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. Othewise
        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 2.4: Chang’s algorithm for finding prototypes

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 a 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. The following section explains the design of CLIFF in detail.

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 CLIFF

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. 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
```

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

Once the ranks of the attribute values are found the following is done:

- the highest rank of values of each attribute is collected

- attribute rank values are then sorted from highest to lowest rank

- the top 50% of these attributes are then used in making the criteria for the instance selection of the current *best* class

- now for each of these attributes the range of values or value with the highest rank is used

In the end an instance is only selected if certain attribute values meet the criteria. Figure 3.1 shows the pseudocode for the algorithm. The following section clarifies CLIFF by using a simple example.

```
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

3.2 CLIFF: A Simple Example

```
forecast  temp  humidty  windy  play
sunny     hot   high     FALSE  no
sunny     hot   high     TRUE   no
overcast  hot   high     FALSE  yes
rainy     mild  high     FALSE  yes
rainy     cool  normal   FALSE  yes
rainy     cool  normal   TRUE   no
overcast  cool  normal   TRUE   yes
sunny     mild  high     FALSE  no
sunny     cool  normal   FALSE  yes
rainy     mild  normal   FALSE  yes
sunny     mild  normal   TRUE   yes
overcast  mild  high     TRUE   yes
overcast  hot   normal   FALSE  yes
rainy     mild  high     TRUE   no
```

Figure 3.2: A log of some golf-playing behavior

For this example we use the *weather* data-set shown in Figure 3.2. This data-set contains four(4) attributes that report on the forecast (sunny, overcast and rainy), temperature (hot, mild and cool), 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 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 3.3.

Once the ranks are found for each value for each attribute, the criteria is created using the highest ranked valued in each attribute (note that although the algorithm calls for using the top 50%

```
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 3.3: Finding the rank of *sunny*

of the attributes, for this example we will use all the attributes but *humidity*). So let us assume that the criteria for selecting instances from class *yes* is [rainy, mild, ?, FALSE], the question mark(?) here indicating that the value for *humidity* could be either *high* or *normal*. With this criteria, two(2) out of nine(9) instances would be selected for the class *yes*.

3.3 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 **??**. In this chapter we evaluate CLIFF 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 the CLIFF does a better job of reducing the impact of *brittleness* than any other PLS.

4.1 CLIFF Assessment on Standard Data Sets

4.1.1 Data

Figure 4.1 lists the eight(8) 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

Data Set	Code	Instances	Attributes	Class
Iris	ir	150	4	3
Breast Cancer	bc	286	9	2
Mamography	mm	150	4	3
Heart (Cleveland)	hc	303	13	5
Heart (Hungarian)	hh	297	13	2
Heart (Switzerland)	hs	123	13	5
Heart (Long Beach V.A.)	hv	200	13	5

Figure 4.1: Data Set Characteristics

of these data sets were acquired from the UCI repository [17]. Except for the Iris data-set, all the 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 represented by bin numbers rather than their actual values. In the experiments to follow, the number of bins is set to 10.

4.1.2 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 following sections describes the experiment and discusses the results.

4.1.3 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. So 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 5.6 for the standard data sets. 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 data-set each time. In the end CLIFF is tested and trained 25 times for each data set. The results for this experiment and Experiment 2 are shown in Figure 4.3 to Figure 4.9.

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

```
DATA = [ir bc mm hc hh hs hv]
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 4.2: Pseudo code for Experiment 1

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	
							0 50 100

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	
							0 50 100

Figure 4.3: Probability of Detection (PD) and Probability of False Alarm (PF)results -iris

bc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	knn	1	100	26	50	93	
	cnn+knn	1	65	20	49	91	
	mcs+knn	1	22	37	46	57	
	psc+knn	1	16	24	42	77	
	cliff+knn	1	13	5	31	98	
pf	cliff+knn	1	13	2	12	88	
	knn	1	100	7	32	73	
	cnn+knn	1	65	7	33	77	
	psc+knn	1	16	17	38	73	
	mcs+knn	1	22	38	50	60	
							0 50 100

bc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cnn+knn	1	65	37	60	83	
	psc+knn	1	18	15	57	81	
	knn	1	100	29	51	77	
	mcs+knn	1	26	37	50	62	
	cliff+knn	1	12	0	37	98	
pf	cliff+knn	1	12	0	22	88	
	knn	1	100	19	35	70	
	cnn+knn	1	65	12	40	63	
	psc+knn	1	18	14	44	78	
	mcs+knn	1	26	35	44	57	
							0 50 100

Figure 4.4: Probability of Detection (PD) and Probability of False Alarm (PF)results -breastcancer

mm	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	14	57	69	76	
	knn	1	100	61	68	73	
	cnn+knn	1	55	52	65	77	
	mcs+knn	2	10	47	56	68	
	psc+knn	2	19	34	56	70	
pf	cliff+knn	1	14	21	30	40	
	knn	1	100	28	33	40	
	cnn+knn	1	55	23	35	48	
	mcs+knn	2	10	33	44	53	
	psc+knn	2	19	28	44	64	
							0 50 100

mm	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	cliff+knn	1	13	62	69	77	
	cnn+knn	2	60	50	60	71	
	knn	2	100	50	59	67	
	psc+knn	3	10	40	60	74	
	mcs+knn	4	20	44	56	63	
pf	cliff+knn	1	13	24	33	42	
	cnn+knn	2	60	27	40	49	
	knn	2	100	32	40	48	
	psc+knn	3	10	27	40	61	
	mcs+knn	4	20	37	43	52	
							0 50 100

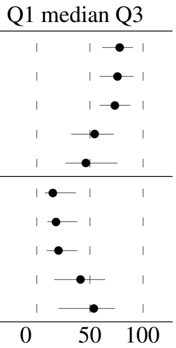
Figure 4.5: Probability of Detection (PD) and Probability of False Alarm (PF)results -mam

hc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	psc+knn	1	35	8	25	39	
	cliff+knn	1	10	0	22	50	
	knn	1	100	9	20	38	
	mcs+knn	1	41	9	20	35	
	cnn+knn	1	65	0	20	40	
pf	cliff+knn	1	10	2	9	22	
	mcs+knn	2	41	5	11	21	
	cnn+knn	2	65	6	11	24	
	psc+knn	2	35	6	12	23	
	knn	2	100	6	12	24	
							0 50 100

hc	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	psc+knn	1	40	11	25	36	
	cliff+knn	1	9	0	21	50	
	cnn+knn	1	77	0	20	50	
	mcs+knn	1	47	0	20	39	
	knn	1	100	0	18	38	
pf	cliff+knn	1	9	2	9	22	
	cnn+knn	2	77	7	13	22	
	mcs+knn	3	47	8	13	23	
	knn	3	100	8	15	24	
	psc+knn	4	40	9	16	25	
							0 50 100

Figure 4.6: Probability of Detection (PD) and Probability of False Alarm (PF)results -hc

hh	PLS	rank	size%	25%	50%	75%
pd	cnn+knn	1	59	62	78	90
	cliff+knn	1	49	59	76	92
	knn	1	100	59	74	88
	mcs+knn	2	18	32	55	72
	psc+knn	2	15	27	46	75
pf	cliff+knn	1	49	8	15	36
	knn	1	100	10	18	39
	cnn+knn	1	59	9	21	39
	mcs+knn	2	18	16	41	65
	psc+knn	2	15	21	54	73



hh	PLS	rank	size%	25%	50%	75%
pd	cliff+knn	1	45	58	76	92
	knn	1	100	57	75	85
	cnn+knn	1	79	60	75	88
	mcs+knn	2	26	36	50	61
	psc+knn	3	18	17	45	68
pf	cliff+knn	1	45	8	21	39
	knn	1	100	14	24	38
	cnn+knn	1	79	11	26	40
	mcs+knn	2	26	35	48	58
	psc+knn	3	18	32	50	82

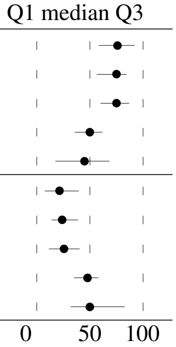
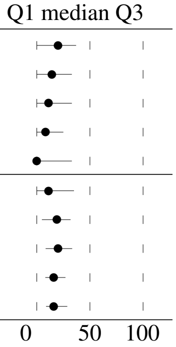


Figure 4.7: Probability of Detection (PD) and Probability of False Alarm (PF)results -hh

hs	PLS	rank	size%	25%	50%	75%
pd	knn	1	100	0	20	38
	cnn+knn	1	82	0	14	33
	mcs+knn	1	58	0	11	33
	psc+knn	1	52	0	8	25
	cliff+knn	1	19	0	0	33
pf	cliff+knn	1	19	0	11	35
	cnn+knn	2	82	5	19	32
	knn	2	100	8	20	33
	psc+knn	3	52	8	16	28
	mcs+knn	3	58	9	16	29



hs	PLS	rank	size%	25%	50%	75%
pd	knn	1	100	0	17	38
	cnn+knn	1	84	0	13	33
	mcs+knn	1	61	0	11	33
	cliff+knn	1	16	0	0	33
	psc+knn	1	56	0	0	30
pf	cliff+knn	1	16	0	16	33
	mcs+knn	1	61	8	16	29
	cnn+knn	1	84	8	16	30
	knn	1	100	9	18	29
	psc+knn	1	56	8	19	27

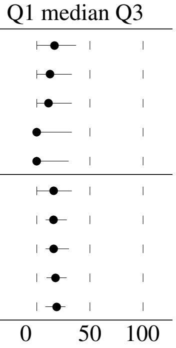


Figure 4.8: Probability of Detection (PD) and Probability of False Alarm (PF)results -hs

hv	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	knn	1	100	13	27	38	
	cnn+knn	1	84	14	25	38	
	mcs+knn	1	57	13	25	36	
	psc+knn	1	49	0	25	38	
	cliff+knn	2	10	0	13	33	
pf	cliff+knn	1	10	0	16	34	
	psc+knn	1	49	10	18	25	
	mcs+knn	1	57	10	19	26	
	knn	1	100	10	19	27	
	cnn+knn	1	84	10	19	27	
							0 50 100

hv	PLS	rank	size%	25%	50%	75%	Q1 median Q3
pd	mcs+knn	1	58	11	27	33	
	cnn+knn	2	88	11	25	33	
	psc+knn	2	51	8	22	33	
	knn	2	100	11	20	36	
	cliff+knn	3	9	0	20	36	
pf	cliff+knn	1	9	4	17	32	
	mcs+knn	1	58	11	18	27	
	cnn+knn	1	88	11	18	28	
	knn	1	100	9	19	25	
	psc+knn	1	51	12	19	24	
							0 50 100

Figure 4.9: Probability of Detection (PD) and Probability of False Alarm (PF)results -hv

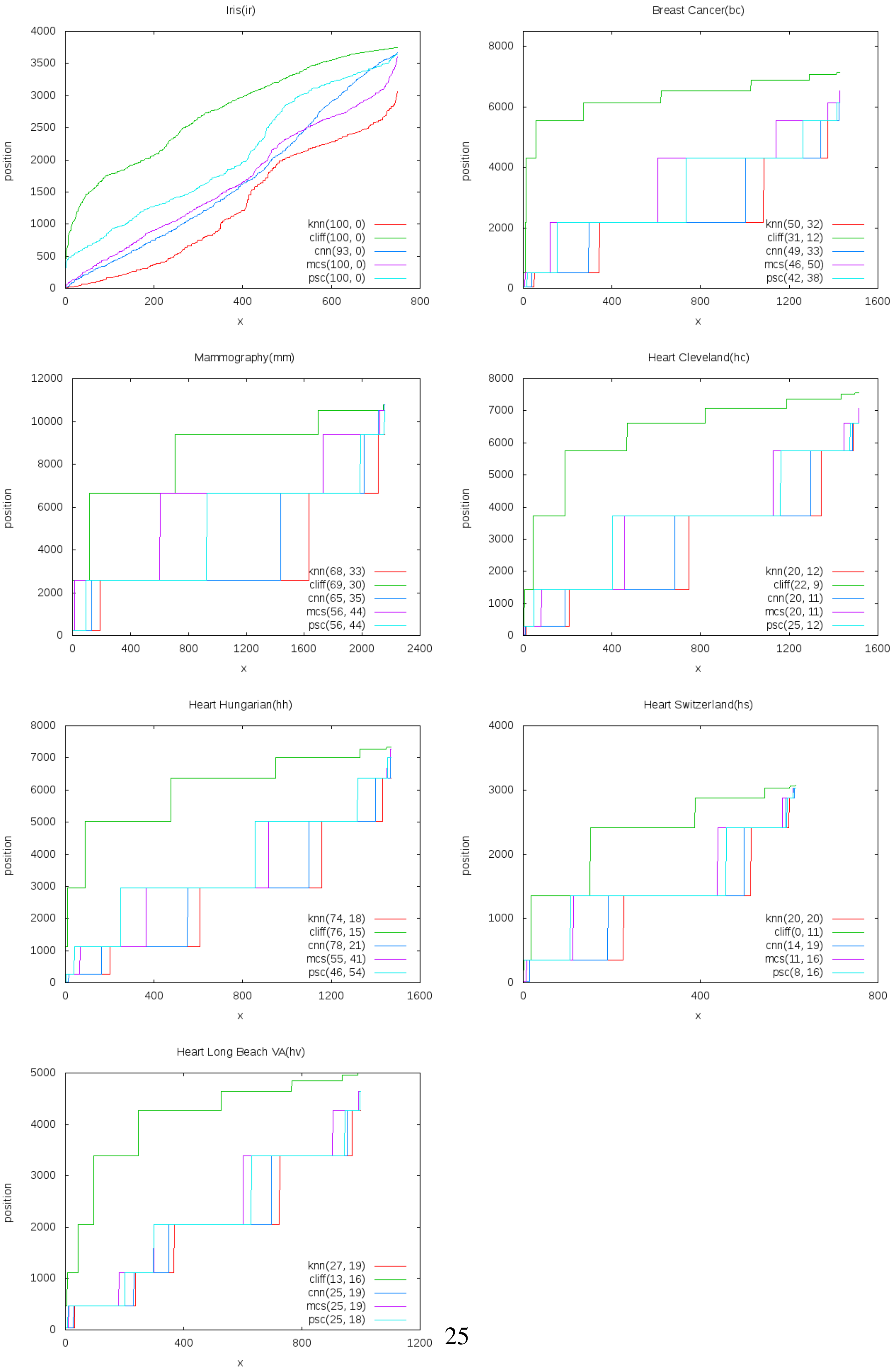

Iris(ir)
position
x
knn(100, 0)
cliff(100, 0)
cnn(93, 0)
mcs(100, 0)
psc(100, 0)
Breast Cancer(bc)
knn(50, 32)
cliff(31, 12)
cnn(49, 33)
mcs(46, 50)
psc(42, 38)
Mammography(mm)
knn(68, 33)
cliff(69, 30)
cnn(65, 35)
mcs(56, 44)
psc(56, 44)
Heart Cleveland(hc)
knn(20, 12)
cliff(22, 9)
cnn(20, 11)
mcs(20, 11)
psc(25, 12)
Heart Hungarian(hh)
knn(74, 18)
cliff(76, 15)
cnn(78, 21)
mcs(55, 41)
psc(46, 54)
Heart Switzerland(hs)
knn(20, 20)
cliff(0, 11)
cnn(14, 19)
mcs(11, 16)
psc(8, 16)
Heart Long Beach VA(hv)
knn(27, 19)
cliff(13, 16)
cnn(25, 19)
mcs(25, 19)
psc(25, 18)

Figure 4.10

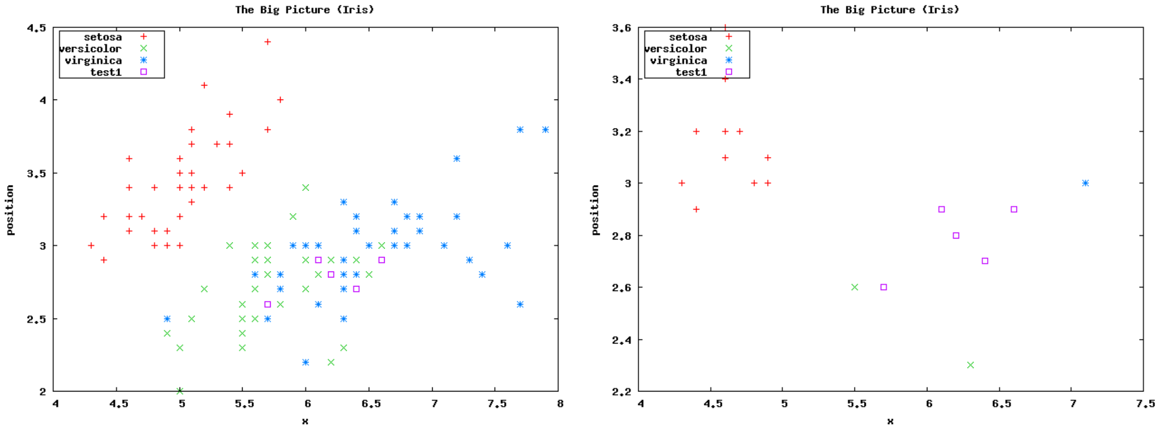

The Big Picture (Iris)
setosa
versicolor
virginica
test1
position
x
4
4.5
5
5.5
6
6.5
7
7.5
8
2
2.5
3
3.5
4
4.5
The Big Picture (Iris)
setosa
versicolor
virginica
test1
position
x
4
4.5
5
5.5
6
6.5
7
7.5
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6

Figure 4.11

Chapter 5

Case Study: Solving the Problem of Brittleness in Forensic Models

5.1 Introduction

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 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.

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 [38] 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 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 quickly eliminate the two(2) latter causes of brittleness by using simple classification methods such as 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, a novel prototype learning algorithm (PLA) is used to augment KNN. Basically this PLA 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 neighbourhoods with different outcomes being futher apart from each other.

In the end our goal for this work 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.

Our research is guided by the following research questions:

- Using KNN as a model, what is the best K for each data set?

- Are the results of using KNN better or comparable to current models which use statistical assumptions and surveys
- Does prototype learning reduce brittleness?
- Do the results of applying a PLA differ significantly from results of not applying a PLA?

5.2 Visualization of Brittleness

This work is 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, however, 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], p6

In this study we visualize the inconsistencies of four(4) of these forensic methods in one way. By simply plotting the measurements derived from evidence from a crime scene denoted as x and suspect (y), against the results of interpretation.

The rest of this section gives details of the four(4) forensic models evaluated in this work, followed by the visualization of these models to highlight their brittleness which results in the inconsistencies in model results.

5.3 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 [38]

5.3.1 Seheult 1978

Seheult [36], examines and simplifies Lindley's [33] 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
>
> $$\lambda = \frac{\sigma}{\tau}, u = \frac{x-y}{\sigma\sqrt{2}}, v = \frac{z-\mu}{\tau(1+\frac{1}{2}\lambda^2)^{\frac{1}{2}}}, z = \frac{1}{2}(x+y)$$

5.3.2 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

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

5.3.3 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.

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

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

5.3.4 Walsh 1996

The equation presented by Walsh [38] 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

5.4 Visualization of Brittleness in Models

The result of applying the visualization technique i.e. plotting the measurements derived from evidence from a crime scene denoted as x and suspect (y), against the results of interpretation on the glass forensic models are shown in Figure 5.1.

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 z axis of all the models represent the likelihood ratio (LR) generated from these models, in other words, the significance of the match/non-match of evidence to an individual or source.

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

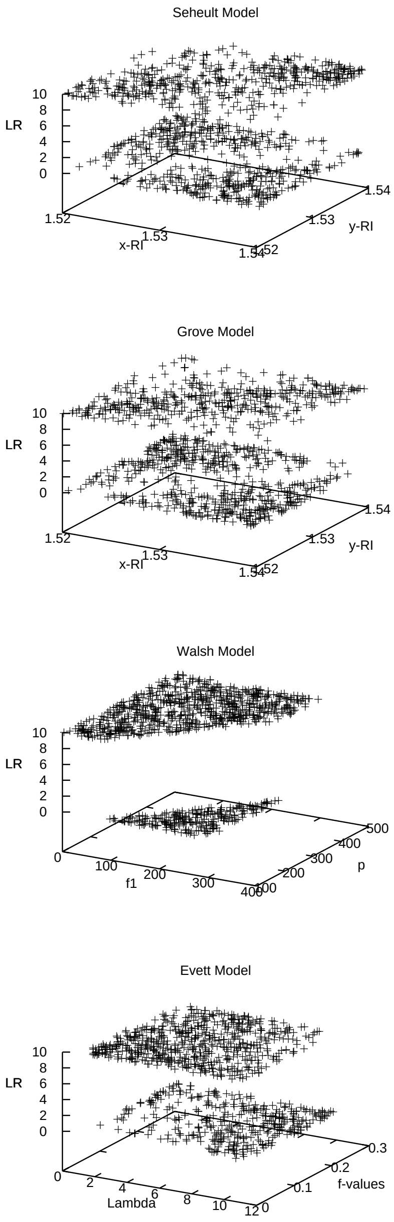


Figure 5.1: Visualization of four(4) glass forensic models

instance, in the Seheult model, each sample looks like this: [x, y, σ, μ, τ]. The symbols are explained in 5.3.1.

In Figure 5.1 - the sehult and grove models, brittleness or Smalldon's "fall-off-the-cliff-effect" is clearly demonstrated. These models proposed by Seheult (Section 5.3.1) and Grove (5.3.2) respectively, show how the likelihood ratio changes (on the vertical axis) 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, near the region where x=y, how tiny changes in the x or y refractive indexes can lead to dramatic changes in the likelihood ratio (from zero to one).

The visualization of the Evett (5.3.3) and Walsh (5.3.4) models show similar brittleness when the likelihood ratios are 0 and 1. For Walsh, values located at the edge of a cliff a LR=1 can easily become LR=0 at the smallest change in the $f1$ or p values. While Evett will cause problems because a small change occurs with any sample it is possible for the LR to change.

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 a Bayesian approach [13–15, 36, 38], and statitical assumptions [21, 36, 38].

The following sections gives details of the models used in this work as well as the data set used to evaluate the models.

5.5 Introduction

If standard methods are brittle what can we do? We seek our answer to this question in the work of [25], and we explore an intuition that to reduce brittleness, data with dissimilar outcomes should not be close neighbors. In this section the details of CLIFF's core procedure and tools are discussed. Included in this discussion is a sub-section which further explores our intuition for brittleness reduction and our tool borne from this intuition - the CLIFF selector.

The Design of CLIFF is deeply rooted in the work of [25]. In their work, analysis is done using

Chemometrics, an application of mathematical, statistical and/or computer science techniques to chemistry. In the work done by [25], Chemometrics using computer science techniques is applied to analyze the infrared 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 100% accuracy i.e. when validated by removing random samples from the model, all the samples were correctly assigned. The goal of CLIFF is not only to create a strong forensic model but also to show how strong the model is. To achieve this CLIFF includes a brittleness measure as well as a method to reduce brittleness. Also, in an effort to keep CLIFF simple, we substituted different tools to preform the analysis done in [25]. For instance Kmeans is used instead of AHC for grouping the data into classes. FastMap is used for dimensionality reduction and K-nearest neighbor is used for classification. The basic operation of CLIFF is shown in Figure 5.2. 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 5.9). Finally, we test if brittleness can be reduced using a novel prototype learning technique (Instance Selection).

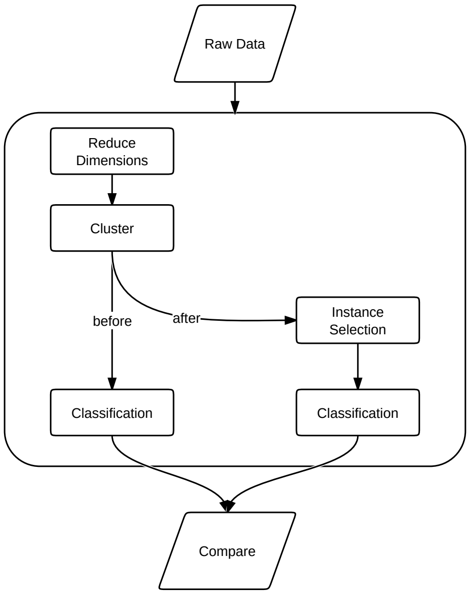


Figure 5.2: Proposed procedure for the forensic evaluation of data

5.6 Dimensionality Reduction

5.6.1 Principal Component Analysis

The goal of Principal component analysis (PCA) is to reduce the number of variables or dimensions of a data set which has a large number of correlated variables while maintaining as much of the data variation as is possible. 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 5.3 shows a visualization of Fisher's five-dimensional iris data on a two-dimensional 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 5.3. The fifth dimension (species) is represented by the color of the points on the scatter-plot.

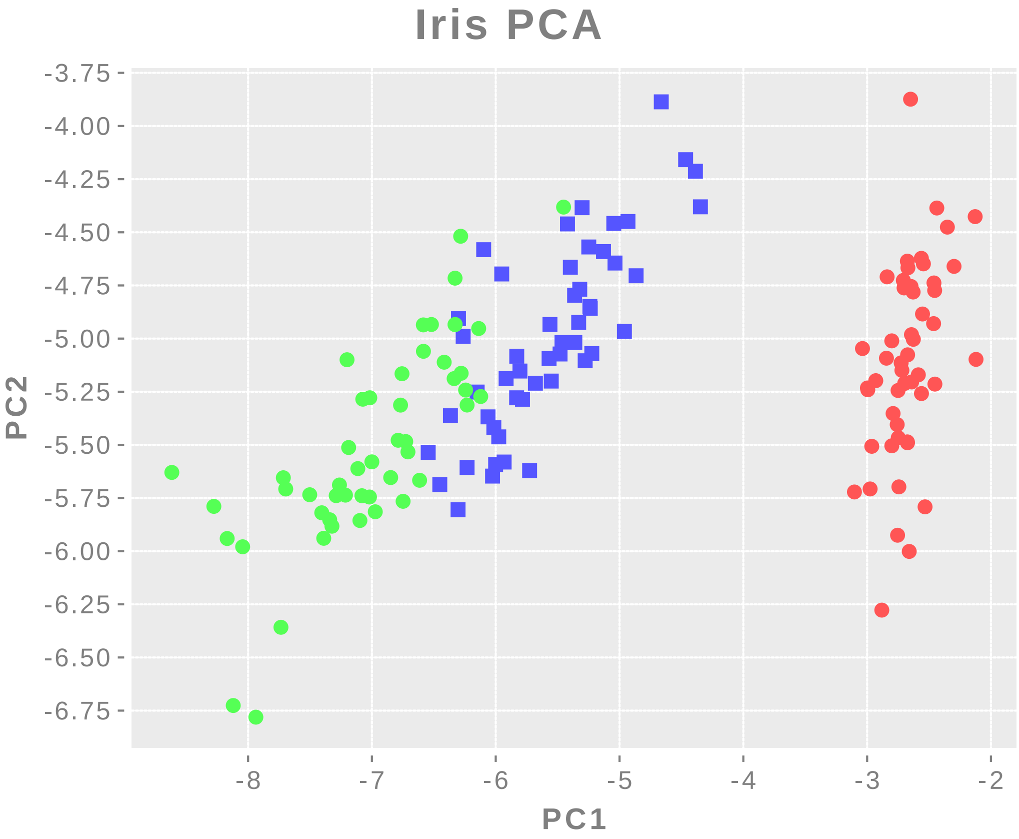


Figure 5.3: PCA for iris data set

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 are Principle Component Analysis (PCA), and FastMap.

The data used in this work contains 1,151 variables and 185 samples. To perform an analysis on this data set we must first reduce the number of variables used. In [25], PCA is used to perform dimensionality reduction. 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]. In [25], two techniques - Pearson correlation and covariance for comparison of the two, were used to determine an appropriate value for M (M = 4).

To speed things up a little, in our model we use *FastMap* to reduce the dimensions of the data set. In FastMap 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. 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 points will be mapped to in the new M dimension space. (Further details of this algorithm can be found elsewhere [16]).

To determine the appropriate value for M using FastMap, we experimented experimented with different values for M. Figure 5.4 shows results for various K-nearest neighbor classifiers (discussed further in Sections 5.8 and 4), with M fixed at 2, 4, 8 and 16. When M is 2 or 4 100% of the validation samples are predicted correctly (pd) and 0% are predicted incorrectly (pd). For this reason, our model model is analysed using M = 4.

5.7 Clustering

Clustering is the second step in the CLIFF tool 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.

CLIFF clusters using K-means. The Figure 5.5 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.

5.8 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. Finally, to determine a value for k, we investigated the performance of six (6) KNN classifiers where k is fixed at 2, 4, 8 and 16. Figure 5.4 shows the results which indicate that using KNN classifiers where k is equal to 4, 8 or 16, the validation of samples is 100%. For CLIFF k = 4 is used.

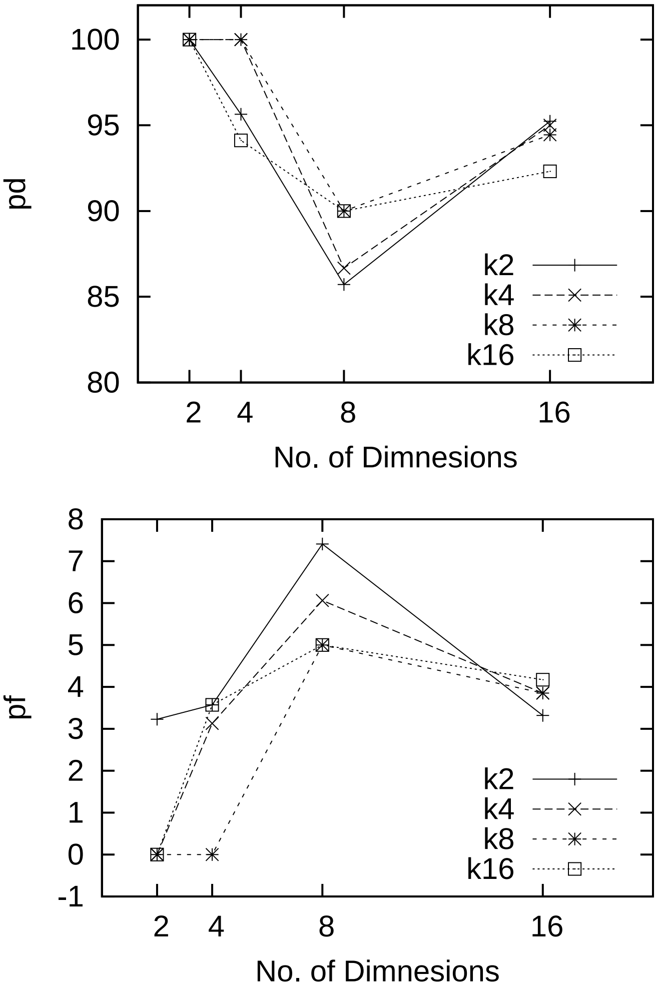


Figure 5.4: Probability of detection (pd) and Probability of False alarms (pf) using fixed values for dimensions and fixed k values for k-nearest neighbor

```
DATA = [3, 5, 10, 20]
k = [1, ..., Number of clusters]
STOP = [Stopping criteria]

FOR EACH data IN DATA
 N = count(data)
 WHILE STOP IS FALSE
  // Calculate membership in clusters
  FOR EACH data point 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 5.5: Pseudo code for K-means

5.9 The Brittleness Measure

Calculating the brittleness measure is a novel operation of CLIFF. We use the brittleness measure in this work to determine if the results of CLIFF 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 each sample from a validation set, once each sample from this set has been classified, the distance from the nearest unlike neighbor (NUN) i.e. the distance from a sample with a different class and the distance from the nearest like neighbor (NLN) i.e. the distance from a sample with the same class is recorded. Recall that brittleness is a small change can result in a different outcome, so here the closer the distances of NUN to NLN the more brittle the model. So an ideal result will have the greatest distance between NUNs and NLNs.

The brittleness measure will give an output of either *high* or *low*: high indicating that there is no significant difference between the NUN and NLN values, 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 5.9 embodies our definition of brittleness: if the significance of NUN values are less than or equal to the NLN values, then an unacceptable level of brittleness is present in the model.

$$[NUN <= NLN] ==> BRITTLENESS \tag{5.9}$$

In this chapter, we evaluate CLIFF as a forensic model on a data set donated by [25] 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 CLIFF before and after the use of the selector.

5.10 Data Set and Experimental Method

The data set used in this work is donated by [25]. It contains 37 samples each with five(5) replicates (37 x 5 = 185 instances). Each instance has 1151 infrared measurements ranging from 1800-650cm-1. (Further details of this algorithm can be found elsewhere [25]). 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 5.5).

The effectiveness of CLIFF is measured using pd, pf and brittleness level (high, low) completed as folows: 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 brittleness level measure is conducted as follows: First we calculate Euclidean distances between the validation or testing set which has already been validated and the training set. For each instance in the validation set the distance from its nearest like neighbor (NLN) and its nearest unlike neighbor (NUN) is found. Using these NLN and NUN distances from the entire validation set a Mann-Whitney U test was used to test for statistical difference between the NLN and NUN distances. The following sections describes two experiments and discusses their results.

5.11 Experiment 1: KNN as a forensic model?

Our goal is to determine if KNN 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 5.6 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 CLIFF is tested and trained 25 times for each data set.

5.11.1 Results from Experiment 1

Figure 5.7 shows the 25%, 50% and 100% percentile values of the *pd*, *pf* and position values in each data set when r=1 (upper table) and r=2 (lower table. Next to these is the brittleness signal where *high* signals an unacceptable level of brittleness and *low* signals an acceptable level of brittleness. The results show that the brittleness level for each data set is *low*. The *pd* and *pf* results are promising showing that 50% of the pd values are at or above 95% for the data set with

```
DATA = [3, 5, 10, 20]
LEARNER = [KNN]
STAT_TEST = [Mann Whitney]

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

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

Figure 5.6: Pseudo code for Experiment 1

3 clusters and at 100% for the other data sets. While 50% of the pf values are at 3% for 3 clusters and 0% for the others. These results show that our model is highly discriminating and can be used successfully in the evaluation of trace evidence.

5.12 Experiment 2: Can brittleness be reduced?

The first experiment shows that KNN creates strong models for forensic evaluation, with high pd's, low pf's and low brittleness levels. With experiment 2 we want to find out if these results can be improved by reducing brittleness further. Since we believe that it is the nearness of unlike neighbors which causes the brittleness (See Equation 5.9), in this section we evaluate the CLIFF selector which selects a subset of data from each cluster which best represents the cluster in hopes that this increases the distance between like neighbors and therefore decrease brittleness while maintaining comparable pd and pf results from experiment 1. Also we expect that the position

		Before percentiles			
Clusters	Types	25%	50%	75%	Brittleness Level
3	pd	90	95	100	low
	pf	0	3	4	
	position	264	614	1068	
5	pd	94	100	100	low
	pf	0	0	0	
	position	374	855	1225	
10	pd	75	100	100	low
	pf	0	0	0	
	position	361	783	1254	
20	pd	0	100	100	low
	pf	0	0	3	
	position	377	762	1256	

		Before percentiles			
Clusters	Types	25%	50%	75%	Brittleness Level
3	pd	89	94	100	low
	pf	0	0	4	
	position	419	905	1351	
5	pd	94	100	100	low
	pf	0	0	0	
	position	437	903	1297	
10	pd	50	100	100	low
	pf	0	0	3	
	position	442	908	1354	
20	pd	0	67	100	low
	pf	0	0	0	
	position	437	896	1345	

Figure 5.7: Results for Experiment 1 for the 4 data sets distinguished by the number of clusters. Here for the upper and lower tables n=4 is used while r=1 is used for the upper table and r=2 for the lower table.

values well be greater than those in the experiment 1.

The design for this experiment can be seen in Figure 5.8. It is similar to that in Figure 5.6, however, the CLIFF selector is included and is described in 3.1.

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

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 NLN and NUN
  [pd, pf, brittleness] = MODEL on TEST
 END
END
```

Figure 5.8: Pseudo code for Experiment 2

5.12.1 Results from Experiment 2

Figure 5.9 shows results for 5 and 10 clusters remain the same for 50% of the pd and pf values while for 3 and 20 clusters the pd's have decreased to 82% and 67% respectively. Also the brittleness level remains low for each data set. The results shown in Figure 5.9 does not provide any information about the difference between the low level of brittleness between Figure 5.7 and Figure 5.9, however the model remains strong. Figure 5.10 illustrates the reduction of brittleness after the CLIFF selector is applied. Mann Whitney U test was also applied to these results to see if there was a statistical difference between the before and after results. The test indicated that the $after$ results are better than $before$ (see Figure 5.11). So brittleness can be reduced while maintaining comparable results.

In summary, by using CLIFF, inappropriate statistical assumptions about the data are avoided. We found a successful way to reduce any brittleness found, to create strong forensic evaluation models. One important point to note here also is this: In order to evaluate data sets with multiple variables, a host of new statistical models has been built [1,2,29,30,39,40]. 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 indices. On the other hand, with CLIFF an increase in the number of variables used does not signal the need to create a new model, it works with any data set.

		After percentiles			
Clusters	Types	25%	50%	75%	Brittleness Level
3	pd	49	82	100	low
	pf	0	9	20	
	position	787	1228	1609	
5	pd	94	100	100	low
	pf	0	0	0	
	position	563	988	1532	
10	pd	60	100	100	low
	pf	0	0	3	
	position	578	1048	1463	
20	pd	0	67	100	low
	pf	0	0	3	
	position	601	1081	1481	

		After percentiles			
Clusters	Types	25%	50%	75%	Brittleness Level
3	pd	89	100	100	low
	pf	0	0	5	
	position	633	1047	1432	
5	pd	90	100	100	low
	pf	0	0	0	
	position	507	982	1465	
10	pd	100	100	100	low
	pf	0	0	0	
	position	506	968	1426	
20	pd	0	80	100	low
	pf	0	0	0	
	position	495	957	1424	

Figure 5.9: Results for Experiment 2 for the 4 data sets distinguished by the number of clusters. Here for the upper and lower tables n=4 is used while r=1 is used for the upper table and r=2 for the lower table.

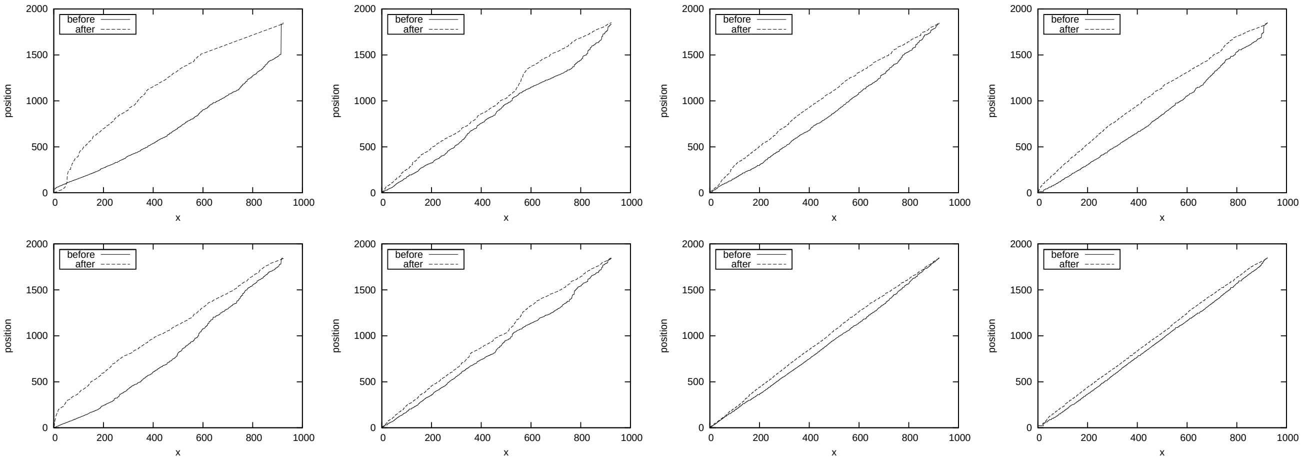


Figure 5.10: Position of values in the 'before' and 'after' population with data set at 3, 5, 10 and 20 clusters. The first row shows the results for r=1 while the second row shows the results for r=2

Clusters	Treatments	Significance
3	before after	-1
5	before after	-1
10	before after	-1
20	before after	-1

Figure 5.11: Results for Experiment 2 of before and after results. -1 indicates that the after is better than before

Chapter 6

Conclusion

The principal purpose of this work was to address the concern of the National Academy of Sciences which stated in a report [35] that:

> With the exception of nuclear DNA analysis, however, 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], p6

Our answer to this is, a novel approach for the evaluation of trace forensic evidence. With a data set donated by [25], made up of the infrared spectra of the clear coat layer of a range of cars, we showed that:

- CLIFF creates strong models with *low* brittleness levels
- The CLIFF selector, based on a PLA, can further reduce the brittleness of a model
- The levels of brittleness differ significantly before and after the use of the CLIFF selector (Mann Whitney U test)

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.

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] N. Karslake, S. Lewis, and W. Bronswijk. Characterisation of automotive paint clear coats by atr-fr-ir with subsequent chemometric analysis. 2009.

[26] 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.

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

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

[29] 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.

[30] 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.

[31] F Korn, BU Pagel, and C Faloutsos. On the "dimensionality curse" and the "self-similarity blessing". *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 13(1):96–111, JAN-FEB 2001.

[32] 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.

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

[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] 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.

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

[39] 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.

[40] 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.