WE HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER OUR SUPERVISION BY ALBERT LIONELLE ENTITLED UNSUPERVISED RELIEFF BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF MASTERS OF SCIENCE.

Committee on Graduate Work

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Adviser

Department Head
ABSTRACT OF THESIS

UNSUPERVISED RELIEFF

Many problems such as classification and retrieval problems are inundated with virtually unlimited numbers of features. Images, for example, can be acquired at arbitrarily high resolutions, and web documents can always include other words and parts of speech. Sadly, it is difficult to process all of these features at one time. Additionally, many of these features are irrelevant and may be detrimental to classification algorithms. Feature selection is important in these domains.

ReliefF is a feature selection algorithm that has proved itself across a number of fields [18, 6, 34, 19, 28, 24]. It can select features without making major assumptions about the underlying data distribution. It is a robust algorithm, that works well across domains. However, it is a supervised algorithm, and supervised data is not always available. It can be labor intensive and expensive to label training data, and there are cases when hand labeled data is not ideal. The system we propose eliminates the need for supervised data with ReliefF.

As defined by Kononenko, ReliefF uses supervised data to determine relationships between samples. Using supervised data, Kononenko is also able to show ReliefF is robust to noise and misclassifications of the data samples. As such, we rely on the robustness of ReliefF when implementing our unsupervised solution. Instead of using supervised training categories, we run an unsupervised clustering algorithm, k-means, to classify the training data. We then use the labels created by k-means to select categories for training ReliefF.

Since we are looking for a domain independent feature selection algorithm, we test unsupervised ReliefF and compare it to supervised ReliefF on two different domains. The first domain consists of cat and dog images; the second contains World Wide Web documents from the Open Directory Project. We are able to show the unsupervised trained ReliefF works equivalently in most cases and even better than supervised ReliefF in some cases. This suggests that unsupervised trained ReliefF a viable alternative to supervised ReliefF, whether or not training data is available.
ACKNOWLEDGMENTS

In this world, it is my firm belief that one person cannot do everything alone. We are constantly supported by our friends, our family and our colleagues in most of our endeavors small and large. In return, we support those around us. Rarely, however, do we get a moment to stop and truly thank them for all of the support they provided. First, I would like to thank my graduate committee for all of their hard work and tough questions. Dr. Adele E. Howe, Dr. Ross Beveridge and Dr. Benjamin Clegg for their insight and advice. Dr. Bruce Draper for our constant talks about computer science, both related to my research and other ideas and dreams. My colleagues in the graduate school who constantly offer advice and help when asked. Including a computer after mine had crashed, the night before this thesis was due. To my friends and family who encouraged me over the past few years of work, they are the ones that kept me in touch with reality after long hours of coding and reading. Last but not least, I would like to thank Heather Codanti. She was the one that encouraged me the most to enter graduate school, and encouraged me to stick with it knowing that I was enjoying what I learned. I leave you with another quote, “An investment in knowledge always pays the best interest,” Benjamin Franklin. Graduate school is that investment in knowledge, and the interest of what I have learn has already more than paid for the amount of time I have spent studying.
# TABLE OF CONTENTS

1 Introduction ........................................ 1
   1.1 Problem ........................................... 1
   1.2 Overview of remaining chapters .................... 3

2 Feature Selection .................................... 4
   2.1 Algorithm Review .................................. 4
      2.1.1 Non-parametric Feature Selection Algorithms ... 4
      2.1.2 Relief ....................................... 6
      2.1.3 ReliefF ...................................... 7
   2.2 Classification and Clustering Techniques .......... 7
   2.3 Unsupervised ReliefF .............................. 8

3 Experiments - Vision Data .......................... 10
   3.1 Introduction ...................................... 10
   3.2 Cat-Dog Data ..................................... 10
   3.3 Experimental Procedures .......................... 10
   3.4 ReliefF as a Feature Ranking System .............. 13
   3.5 Results Based on Cluster Differences ............. 16
   3.6 A Closer Look at the ReliefF Values .............. 20
   3.7 Conclusions for the Cat-Dog Data ................. 26

4 Experiments - World Wide Web Documents .......... 29
   4.1 Introduction ...................................... 29
4.2 DMOZ Web Document Data ......................................................... 29
4.3 ReliefF as a Feature Ranking System .......................................... 31
4.4 Results Based on Cluster Differences ....................................... 34
4.5 A Closer Look at ReliefF Values ............................................. 38
4.6 Conclusions for World Wide Web Document Data .................... 41

5 Conclusions .............................................................................. 44

5.1 Discussion ............................................................................. 44
5.2 Future Work ........................................................................ 45

A Full Figures for Cat-Dog Data .................................................. 47

B Full Figures for World Wide Web Data ...................................... 58

REFERENCES ............................................................................ 69
# LIST OF FIGURES

2.1 Experiment Flowchart ................................. 9

3.1 Exemplars from the Cat-Dog data set. .................. 11
3.2 Cat-Dog PvR values using top 100 features ................ 13
3.3 Cat-Dog PvR values using top 1000 features ................. 14
3.4 Cat-Dog PvR values using top 10,000 features ............... 15
3.5 Cat-Dog PvR values with 0.1 threshold .................... 17
3.6 Cat-Dog PvR values with 0.01 threshold ................... 18
3.7 Cat-Dog PvR values with 0.001 threshold .................. 19
3.8 Cat-Dog PvR values across the Supervised Training Set .......... 21
3.9 Cat-Dog PvR values with K-Means set to 2 Clusters ........... 22
3.10 Cat-Dog PvR values with K-Means set to 4 Clusters ........... 23
3.11 Cat-Dog PvR values with K-Means set to 8 Clusters .......... 24
3.12 Cat-Dog Sorted Feature Weights ......................... 25
3.13 Cat-Dog Weighted Features .............................. 27
3.14 Normalized Cat-Dog Weighted Features ..................... 28

4.1 Web Document PvR values using top 100 features .............. 31
4.2 Web Document PvR values using top 1000 features ............ 32
4.3 Web Document PvR values using top 2500 features ............ 33
4.4 Web Document PvR values with 0.1 threshold ................. 35
4.5 Web Document PvR values with 0.01 threshold ................. 36
4.6 Web Document PvR values with 0.001 threshold ............... 37
LIST OF TABLES

3.1 Relevance Table Lookup ......................................................... 12
3.2 Number of Features by Threshold for Cat-Dog Data ...................... 17
4.1 Number of Features by Threshold for World Wide Web Data ............. 34
4.2 Top 10 weighted words across different training types .................... 38
Chapter 1

Introduction

*By failing to prepare, you are preparing to fail.*

- Benjamin Franklin

1.1 Problem

With the evolution of technology, we have become inundated with images and documents. Digitized photos come from many different sources. Some of these sources are controlled such as the passport photos utilized in the U.S. Visit program; others are highly uncontrolled such as a vacationer’s digital camera pictures. Many other photographs are never even examined before they are archived, such as satellite photographs. Simply put, there is too much information and not enough manpower to analyze it. This is also true of documents. With today’s Internet it is possible for anyone to put up a webpage. Some of these pages are highly informative, while others are not. To handle information overload, we seek a way of organizing both image and text into groups thus classifying them. The problem with classification is the noise inherent to many of the features and training sets within some domains. Which pixels of the image are needed to classify it? Which words in a document matter? We need a way to prepare or preprocess samples so that we can gain optimal classification accuracy. The selection of salient features to increase classification accuracy while decreasing processing time is known as feature selection.

Across machine learning, many irrelevant features act as noise in classification processes. To remove these less informative features, we use feature selection. The selection of features is typically designed for a specific domain, so that algorithms can be both more efficient and more accurate. More formally stated, feature selection picks a subset of informative features. This subset represents a minimal set of features needed to represent the data. The remaining features represent redundant or noisy information within the data. In some domains, feature selection is not critical because the number of features is small. However other domains, such as computer vision and document retrieval, produce large feature sets of which many of
these features are noise. Indeed, for this reason, we focus on feature selection mostly within the image and text document retrieval domains.

Image and text documents have several characteristics in common: large feature sets with defining features possibly being a noticeably smaller subset. To put the size in perspective, a 64x64 image that utilizes raw color values for features contains 12,288 features. For text documents, the World Wide Web data document set used in Chapter Four contains 12,579 words after stemming and stop word removal. However, a 64x64 image is barely more than the size of a thumbnail on the desktop; some digital camera images are on the scale of 1024x768 pixels. Even if we crop this image to 768x768, the raw color features would consist of 1,769,472 features. For English language documents, The Oxford English Dictionary contains 290,000 entries and 616,500 word forms [26]. Simply put within the web document and image domains, there are large amounts of data, and within each data sample, the features that define it are even larger. A feature selection algorithm is clearly warranted, but what type of algorithm is needed?

Both the vision and text domains are extremely large. With people collecting images or making pages daily, it is difficult for an algorithm to have access to a significant fraction of the domain for training. To simulate this, we wish to train an algorithm using a small percentage of our training set compared to our test set. We divide our datasets using 30 percent for training and 70 percent for testing. This would show that it could handle large amounts of data while only needing to analyze a small fraction of the data for salient features. Also, much of the vision and text data we receive is filled with noise. Noise being features that do not help with classification of the documents. These noisy features contain very little information in relation to clustering; as such they will cause misclassifications, if an algorithm is not robust against them. However, training it on a fraction of the data could possibly limit an algorithm’s ability to handle the noisy features. Based on this an algorithm should be robust enough to handle noisy classifications within small training sets.

The second property we are interested in for feature selection is that an algorithm doesn’t make assumptions about the distribution of the data. Some feature selection algorithms assume a specific data distribution, for example a Gaussian variant. However, the exact distributions of data in the image domain and web document domains are unknown. Additionally, as the features represent different values, their distributions are most likely different. We seek a single algorithm that can be used to select features across different domains as long as we represent the data in a similar manner.

The third major problem we come to within our domain is supervision. Supervision is a process in which trained personal classify the data. However, supervised algorithms tend to be very expensive to implement. This expense usually isn’t the execution time, but the cost of developing training sets. It requires skilled manpower to classify documents. Not only is this an additional cost on industry, it also is rare that a quality training set exists. This can be a big problem for feature selection algorithms in domains that do not have
ample amounts of training data such as World Wide Web images, satellite photographs or personal documents. Based on this, a feature selection algorithm within the vision and web document domains should be unsupervised.

One algorithm that shows strong potential in these domains is ReliefF. Relief, originally defined by Kira and Rendell, is an algorithm that is efficient in selecting attributes based on binary classifications [16]. As some classification problems are multi-class, Kononenko modified Relief naming it ReliefF to include multiple classes [18]. Relief has been used across a number of domains, e.g., cancer genes [34], knowledge databases [3], decision trees [19] and computer vision [6]. As such, it has been shown to work on domains in which there is little information about the data distribution. Additionally, Kononenko shows that ReliefF is robust to noisy classifications of data and features.

While ReliefF covers our first two requirements, it needs supervised training data. As mentioned above, supervised training is expensive and requires trained manpower to pick and classify samples into categories. This process is expensive and time consuming, and there exist very few large training sets within the domains of computer vision and web document retrieval. This report focuses on making ReliefF unsupervised. We assume that through robustness across different domains, ReliefF is able to define valid features using unsupervised learned labels, instead of supervised labels.

1.2 Overview of remaining chapters

Chapter Two reviews the feature selection literature in relation to machine learning. We then discuss Relief and its extension ReliefF, and its requirement of supervised training data. Finally Chapter Two finishes with our proposal to use k-means to make ReliefF unsupervised. Chapter Three describes our testing procedures within the Vision domain, and the results of our experiments. It shows that unsupervised ReliefF performs equivalently to supervised ReliefF. Chapter Four contains results from our second domain, World Wide Web documents. We are able to show that over all unsupervised ReliefF performs equivalently to the supervised version. Finally, Chapter Five contains the conclusions generated from both test domains.
Chapter 2

Feature Selection

People will accept your ideas much more readily if you tell them Benjamin Franklin said it first.

David H. Comins

2.1 Algorithm Review

The feature selection problem spans many different domains, with a corresponding variety of approaches. If a problem has more features than are needed for evaluation, researchers usually include a form of feature selection. Feature selection received its initial foundation from Bayesian Statistics. Some statisticians approach feature selection by modeling the distribution of the data. Using a priori probabilities, feature selection can be solved given class and feature distributions. Because distribution models estimate model parameters, they are termed parametric models. However, because they are making assumptions about the data, the internal functions of the model change as the data changes. This makes it difficult to use statistical models for our purpose.

The other major area of study for feature selection is artificial intelligence. In artificial intelligence, most feature selection models try not to assume distributional models. Since they do not assume models for the data, they are usually termed non-parametric. For our purposes, non-parametric models are ideal. Within the field of AI, there are numerous feature selection algorithms. As it would be infeasible to list them all, we will only cover some of the most common ones. It should be noted, that many feature selection algorithms are domain specific. Trevor et al. mention this point in passing as do many other machine learning books [12]. For our research, we seek a non-domain specific feature selector.

2.1.1 Non-parametric Feature Selection Algorithms

Sequential Forward Selection (SFS) [22] and Sequential Backward Selection (SBS) [36] both utilize depth first search of the features in an exhaustive attempt to add (SFS) or remove (SBS) features from the feature space. These algorithms suffer from two major problems for our purposes. First is the nesting problem;
once a feature is added or removed, it is never examined again. The second is the exhaustive nature of the search. When we are looking for algorithms that deal with millions of features, exhaustive searches become infeasible within our context. There are variations of SFS and SBS, but they tend to be even more complex while suffering from the same problems making them inapplicable [15]. Focus and Focus2 are also inapplicable as they are exhaustive searches very sensitive to noise [1, 2].

On the other hand, Jain and Zongker show that the Max-Min algorithm is very fast [15]. However, Max-Min performs very poorly when compared to other algorithms. In terms of more efficient algorithms, the Branch and Bound algorithm performs quickly, but it relies on monotonic evaluation criteria that are difficult to determine for our domain [23].

Artificial neural networks (ANN) have been used for their adaptability in unknown domains. ANNs are adaptive learning models comprised of densely interconnected nodes. Each series of connections have input nodes, hidden nodes and output nodes. Through large learning sets weights are derived for each connection in the network. One way they have been used for feature selection is that when a weight or output node drops below a certain threshold that node and its associated connections are pruned. Some cases of this are [21, 30, 5]. Unfortunately, it is impractical to use ANNs within the domains of machine vision and web based text documents as the feature sets are extremely large, thus making the number of input nodes large. This is furthered by a scaling problem that happens when using neural networks. In general, it usually is better to have more hidden nodes than less. For domains where the distribution is unknown, it also is standard to have fully connected neural networks [12]. This causes the problem of 12,000+ features to explode in the number of calculations that must be performed to prune only a few features. In addition to this, after a group of features is removed, it must be retrained, making this a very iterative and expensive pruning process. Finally, neural networks generally need more samples than features for training. While this data does exist, the domains we are interested in are usually trained with a smaller sample size than features.

Genetic algorithms (GA) have also been used to select features. This step is fairly obvious, as many feature selections utilize search. Genetic algorithms are specialized search heuristics that perform well in many domains without exhaustively searching the data. For a genetic algorithm, each feature is either “on” or “off” within the chromosome bit string. The GA then searches the space of possible features by selecting them and trying them in the problem. The fitness function usually is defined based on the desired use of the samples; which for our work is classification and cluster strength. Some of the first uses of genetic algorithms for feature selection were done by Siedlecki and Sklansky [31]. As genetic algorithms are related strongly with search, they were compared to the SBS method, and the genetic algorithm outperformed SBS noticeably [33]. Overall, genetic algorithms have been used with some promising results, but for our work the fact they need to reclassify the dataset after each search attempt make them infeasible.
Tree based classification techniques commonly referred to as decision trees have also been used for feature selection. Developed by Quinlan [27] and Breiman [7] independently, they recursively classify groups into sets based on the information gain or entropy of each feature. Since they split the features based on the information in each feature, they also are a way of selecting features [12]. However, the conical algorithms of ID3 and C4.5 degrade when dealing with large amounts of irrelevant features [2, 17]. For our work, it is well known there are many irrelevant features, so these algorithms are not ideal.

Relief is a simple feature selection algorithm that has received notable attention. It has been used in regression trees [28], as attribute weighting methods [35], computer vision [6], decision trees instead of ID3 or C4.5 [19], cancer genes [34] and inductive logic programming [24] to just name a few. The relief algorithm can be computed efficiently and performs well. It is applied in a preprocessing step before a model is learned [16] and is one of the more successful preprocessing algorithms [32]. The main advantage of the relief algorithm is that it doesn’t make any assumptions about the data distribution or even about the domain. It simply assumes that like samples should have similar features. It also is robust to noise and large feature sets with small sample sets [6]. Originally created by Kira and Randell [16] and later modified by Kononenko [18], Relief is a promising algorithm for our research.

2.1.2 Relief

The main idea of the original Relief algorithm [16] is to estimate the quality of an attribute according to how well it distinguishes between similar instances. Given a randomly selected sample in a binary-class problem, it considers the two nearest neighbors: one from its own class and one from the other class. The one from its own class is termed the nearest hit while the other is the nearest miss. Assigning a weight of 0 for a hit and a weight of 1 for a miss, it then weights each feature based on equation 2.1. The $\text{diff}$ function takes the difference between the features in each sample. This difference can be city-block distance or any other distance that fits your domain. When finished taking samples, it averages the weight for each feature. The features that contain less information ideally have a weight of 0 or less, and the more informative features have higher weights.

$$W_f = W_f - \text{diff}(\text{Near}_f - \text{hit}_f)^2 + \text{diff}(\text{Near}_f - \text{miss}_f)^2$$  \hspace{1cm} (2.1)

The Relief algorithm has been used as a wrapper for learning and clustering algorithms. Unfortunately, Relief was only designed for two class problems, and it has had issues with noise and incomplete datasets. To address this latter problem, Kononenko developed a series of relief algorithms (A-F), with his final robust multi-class version being ReliefF.
2.1.3 ReliefF

Relief’s original version uses 1-Nearest-Neighbor as its base training classifier. ReliefF takes the K nearest misses minus the K nearest hits to weigh each feature individually as seen in formula 2.2 instead of only taking one hit from each type [18]. A sample is classified as a hit if it is one of the k-nearest samples from the same category as the training sample, and a sample is classified as a miss if it is one of the k-nearest samples from any other category. The weight it generates for each feature can then be used to rank each feature by importance. Kononenko’s formula gives a wide range of weights for each feature.

\[
W_f^+ = \frac{\sum_{m \in \text{misses}} |s_f - m_f| - \sum_{h \in \text{hits}} |s_f - h_f|}{|\text{Hits}| \ast |\text{Misses}| \ast \text{Range}(f)}
\]  

(2.2)

Based on the ranking, any feature under a threshold is ignored when running the classification algorithm. The threshold is usually determined by the domain. In most cases, when ideal thresholds are unknown, the weights are graphed, and the threshold is selected arbitrarily when the graph shows a steep decline.

ReliefF proves to be a robust algorithm that handles noise and incomplete data sets fairly well. This compensates for most of the original algorithm’s weaknesses. It is a very strong algorithm, but as stated above, supervision is an expensive requirement. In order to talk about how we modify ReliefF to remove its dependence on supervised data to determine a hit and miss, we must first briefly discuss clustering.

2.2 Classification and Clustering Techniques

Classification problems have been studied at least as much as feature selection. Humans like to classify data into groups. As such, in many different domains, classification algorithms have been applied to categorize data samples [12]. Classification algorithms can be placed into two defined groups: supervised and unsupervised. Within the supervised domains, we have Bayesian Networks, Decision Trees, Support Vector Machines and many other machine learning techniques. However, for our purposes we are interested in unsupervised classification techniques. Most unsupervised classification techniques are clustering algorithms.

Overall, clustering is an extensive subject that is beyond the scope of this work. For further information on clustering, see [14]. Since we realize clustering is fairly domain specific, we wanted to limit our option when defining our solution. As such, we chose to use K-means as it has proved over and over again that it is a strong generalized algorithm that works across many different domains. It also is a fairly conical algorithm for clustering, which made it ideal for our purposes.

Clustering algorithms take data samples and group them together by similar or defining features. There are many different clustering algorithms, each with their own advantages and disadvantages. The problem with clustering techniques is that their advantages and disadvantages are data dependent. Jain et al. state
that it is the user’s dilemma when determining which clustering algorithm to use [14]. Many users just try algorithms on test datasets to see if they are suited for a domain.

Hierarchical clustering algorithms have the nice property that one can pick and choose the number of clusters from a hierarchy of links between them. At the first stage, there are \( N \) clusters were \( N \) is the number of samples. Each stage then reduces the number of clusters until only a single cluster is left. For hierarchical clustering algorithm exemplars, we have single link and complete link algorithms. These algorithms are fairly straightforward, but their uses tend to be fairly dependent upon the domain [14].

Partitional algorithms, on the other hand, tend to be limited by a set number of clusters or by some other clustering criteria (e.g. entropy). K-means is an algorithm that takes this approach. Using random starting points for centroids, it clusters based on the distance between samples grouping to the closest centroid of that cluster. It then takes the mean of the cluster and starts over using the mean as the centroid. In practice random restarts are sometimes needed because of its non-deterministic nature. It is fairly simple and is commonly used as the conical clustering algorithm for many experiments [12].

Lastly, we should not forget about other machine learning techniques that are also used for clustering. For example, ever-adaptive neural networks are also used for clustering. They seek to learn a way to classify groups. Self-organizing maps (SOM) have also been used for clustering. These maps provide a framework of weights that organize a data sample into its ideal cluster location [12].

### 2.3 Unsupervised ReliefF

Relief and its many versions are supervised based on using supervised labeled training data. Our goal is to see if ReliefF can be used without having supervised training data. This is accomplished in the following manner:

1. Cluster the unsupervised data using K-Means. While we could try to substitute any unsupervised clustering algorithm, we use K-Means for its simplicity and baseline it provides. K-Means has one major variable parameter to set, the number of clusters. As such, we will use different cluster counts testing each respectively as a different label system. We start with a cluster count equal to the number of supervised clusters then double it for each test.

2. Assign labels to each sample based on the cluster id determined by K-Means. (e.g. “cluster 0”). While these labels may not have semantic meaning to humans, they do have meaning for the actual workings of the ReliefF algorithm.

3. Train ReliefF using the labels defined in Step 2. A hit would be classified as a hit if it has the same label as the exemplar. A miss if it has a different label. This is the same as the ReliefF algorithm, but
instead of using the supervised labels for training the algorithm we used those determined by K-Means.

4. Select features based on the weights generated by ReliefF.

Given this, we are able to select features for each data sample. The features generated by ReliefF trained with supervised labels may be different than those generated by ReliefF trained with unsupervised learned labels. Are the features equivalent for ranking or classifying the data samples? To test this, we take only the features for each sample and sort the weights by distances. Using the original supervised labels we are able to test if the images are classified correctly, even if they are using different features. This is also shown in Figure 2.1. The figure only shows one run of K-Means for room constraints, however, we actually have three different runs of K-Means all with a different cluster counts, different labels and thus different weights.

---

**Figure 2.1: Experiment Flowchart**

This figure represents the flow of our experiments. Using the 30% test data, we train ReliefF using the original supervised labels and using labels generated by K-Means clustering. We then select features based on weights. Using only the selected features, we classify the remaining 70% of the data and compare against supervised labels for accuracy.
Chapter 3

Experiments - Vision Data

All cats are gray in the dark.
Benjamin Franklin (1706-1790)

3.1 Introduction

Computer vision is one of two primary domains of interest for our research. The vision domain has two important characteristics for our work. The first is the large number of features that exist within a single image. A 64 pixel wide by 64 pixel high image with three color channels has 12,288 features. However in most applications, a 64x64 pixel image is small. The second characteristic is while there are many images, there are very few supervised training sets. Those that exist are small or expensive. As a result, the number of training samples is usually much smaller than the number of features. With these challenges, vision is an interesting domain to test. Additionally, ReliefF has been used in the past with varying amounts of success in high dimensional data spaces [6].

3.2 Cat-Dog Data

The data set we use is the cat-dog data set used previously in [6, 9]. The data set consists of 100 unique cat images and 100 unique dog images. Each image is 64x64 pixels. Each cat and dog image was hand selected and categorized. The images were registered by hand-aligning the eyes, although there is considerable error in the registration. Figure 3.1 contains some examples from the dataset.

3.3 Experimental Procedures

For testing we choose to stick with standard procedures for the domain whenever possible. Each image was prepossessed by subtracting the mean pixel value from every pixel, and then dividing every pixel by the square root of the sum of squared pixel values.
The Cat-Dog data set contains 100 cat images and 100 dog images. Each image is roughly registered by eye coordinates and scaled to be 64x64 pixels.

This creates images with a zero mean value and a unit length, when images are interpreted on feature vectors. It also has the property that Euclidean distance between two vectors in feature space is inversely proportional to their correlation [12]. Each feature is represented by its raw normalized pixel value. For a distance measure between images, we used the L2-Norm, also known as the Euclidean distance [13].

For clustering, we use basic distances between images. K-Means is used with random restarts, and convergence was achieved when no sample changed clusters between iterations. For these tests we chose 2, 4, and 8 clusters respectively when generating ReliefF values.

We generated ReliefF weights for both supervised and unsupervised data. For the supervised data, we used the predefined categories. For the unsupervised data, we used the clusters generated by k-means. A sample is considered a ReliefF hit if it is nearby and a member of the same cluster label, and a ReliefF miss if it is nearby but a member of a different cluster label. To train ReliefF, we randomly sampled 30 percent of the data and used that as our training set. We ran each test seven times and generated an average weight for each run. This helped eliminate outliers.

We test by taking the top N features by weight. Using this procedure we are able to compare supervised trained ReliefF versus unsupervised trained ReliefF while holding either the salience weight threshold or the number of features constant. For the Cat-Dog data, we tested the top 100, 1000 and 10,000 features from each training type. Additionally, we test by taking features based on weights higher than the thresholds 0.001, 0.01 and 0.1. This latter test is done to compare the differences in the actual weights generated by the different types of training for ReliefF.

To analyze the data across domains, we use precision versus recall curves. This test is calculated with the other 70 percent of the data (no preference was given to cat or dog). As we are both interested in recall and precision, one value needs to be fixed while the other remains continuous. We fix recall values to every 0.1 step between 0 and 1 giving us 1 percent to 100 percent of the samples returned for that category. We then
<table>
<thead>
<tr>
<th>Relevance Table</th>
<th>Relevant (same category)</th>
<th>Not Relevant (different category)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>True positive (TP)</td>
<td>False positive (FP)</td>
</tr>
<tr>
<td>Not Retrieved</td>
<td>False negative (FN)</td>
<td>True negative (TN)</td>
</tr>
</tbody>
</table>

Table 3.1: Relevance Table Lookup

The relevance determines precision and recall curves. Using precision and recall, we are more interested in the samples that are retrieved and classified than samples that are never classified. In essence the PvR curve is an ROC curve that asks two questions. Precision asks if all the retrieved images are relevant, while recall asks if all the relevant images were retrieved.

calculate the continuous precision based on that recall value.

\[
\text{recall} = \frac{TP}{TP + FN} \tag{3.1}
\]

\[
\text{precision} = \frac{TP}{TP + FP} \tag{3.2}
\]

Precision versus recall curves for a single sample is generated by ordering the samples by distance (K-Nearest-Neighbor) from that sample. We then take N samples until a set recall score is reached given by equation 3.1. We then calculate the precision from those N samples using equation 3.2. To calculate the averaged precision for our test, we average the precision across all samples, and then average it again across each test [4].

The question we ask is one of retrieval. If we take an exemplar and we want a percentage our dataset from our database, how many of the returned samples are actually labeled from that category or are from different categories? A sample was determined relevant if it was from the same category as the other sample as shown in table 3.1

For example, we take our calico cat, “cali”, and ask for at least five other cats to be returned from our database. That would give us a recall of 0.05 or 5 percent. Assuming three dogs were considered closer to our calico than the fifth cat returned, that would give us a precision of 0.625 or 63.5 percent. We never examine the dogs not returned as we do not care about the dogs not returned only the cats that should have been returned. We do however care about the dogs that were returned when we only wanted cats.

We ran each test seven times to eliminate outliers. For an additional baseline we ran KNN utilizing the full feature set and generated the precision versus recall curve. Our criteria for a successful test was based on the supervised precision versus recall scores. If the k-means generated results were better or equivalent to the score, then it shows that unsupervised ReliefF is possible. The full feature set was included in all of our tests, as we like both supervised and unsupervised ReliefF selected features to work equivalently to the full set of features.
3.4 ReliefF as a Feature Ranking System

The major variable for k-means is the number of clusters one selects. For our initial test, we selected two as that was the number of categories the original dataset contained. We also tested four and eight clusters to see if the number of clusters made any difference. By changing the number of clusters, we also changed how ReliefF could categorize the data. As such, we fix the number of features used and test based on those features. For our tests we took the top 100, 1000 and 10,000 features determined by ReliefF across each training type. Figures 3.2, 3.3 and 3.4 show the differences in precision versus recall scores across the number of features used.

![Precision vs Recall Using Top 100 Features](image)

Figure 3.2: Cat-Dog PVR values using top 100 features

Precision versus Recall curves across training types using a fixed feature count of 100. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the top 100 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. The features generated by both supervised trained ReliefF values (yellow) and unsupervised trained with two k-means clusters (green) perform poorly between 1 percent and 19 percent recall, but they improve as they reach later recall values. K-means with eight and four clusters both perform fairly well given that they are only using 100 of the 12,288 features for sampling. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set. For the full figure, see Appendix A.

Figure 3.2 clearly shows that only taking the top 100 features, unsupervised trained ReliefF performs
qualitatively better than taking the top 100 features of the supervised trained ReliefF features. Comparing the first 19 percent recall, unsupervised ReliefF does better than supervised. After the first 19 percent we once again reach a point where we are seeing marginal gains as the majority of the opposing category has already been retrieved. Unsupervised trained with two k-means clusters also has this problem with the top 100 features ranked. However, k-means trained with four and eight show remarkable improvements. This shows the ranking provided by the unsupervised trained ReliefF is strong. If we are only interested in precision and not efficiency for the first 5 percent recall, the full set performs better. After the first 5 percent, however, all ReliefF versions perform noticeably better than the full set. While not fully studied, we found a major difference in the time it takes to run distance measures using only 100 features compared to 12,288 features.

**Figure 3.3: Cat-Dog PvR values using top 1000 features**

Precision versus Recall curves across training types using a fixed feature count of 1000. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the top 1000 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. The features generated by the supervised trained ReliefF values (yellow) still perform poorly between 1 percent and 19 percent recall, but they improve as they reach later recall values. The k-means with two clusters is now performing competitively against the full feature set and the other unsupervised trained ReliefF selected features. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set. For the full figure, see Appendix A

14
Figure 3.3 shows a noticeable improvement in the unsupervised trained results. Given that we only return one possible image, the k-means trained with eight clusters actually generates better results than the full feature set. However, it steeply drops until 5 percent recall which it crosses the full feature set along with the rest of the ReliefF learned results. K-Means trained with two clusters shows a noticeable improvement over the results shown in Figure 3.2. All three unsupervised trained ReliefF versions perform better than the supervised trained ReliefF. This shows that the rankings they provide are valid using only the top 1000 features, while supervised actually needs more than a 1000 features to provide equivalent information. It should also be noted that 1000 features is one-third the total number of features used by the smallest of the unsupervised threshold tests, leading us to believe there are many redundant features not including those from using gray-scale images. As ReliefF does not eliminate redundant features, further study should be applied using Bins’s proposed system which eliminates redundant features from ReliefF generated features [6].

Figure 3.4: Cat-Dog PvR values using top 10,000 features
Precision versus Recall curves across training types using a fixed feature count of 10,000. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the top 10,000 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set. For the full figure, see Appendix A.
Using the extreme of 10,000 features supervised trained ReliefF finally shows improvement over the previous two feature cutoffs (100 and 1000). Seen in Figure 3.4, supervised trained ReliefF performs fairly equivalent to unsupervised trained ReliefF. While there are peaks in which supervised performs slightly better, they are marginal compared to the other feature ranking systems.

### 3.5 Results Based on Cluster Differences

In addition to fixing the number of features, we can also fix the threshold in which we collect features. Using thresholds of 0.1, 0.01 and 0.001, figures 3.5, 3.6 and 3.7 contain the results across the different cluster sizes but with the threshold fixed in each one.

Figure 3.5 shows the different ReliefF learned feature sets and the full feature set (show in red on all graphs). There are really two different tests being examined in this figure. The first test is against the full set of features, all 12,288. If we were only interested in the first 5 percentile of recall, than the full set performs slightly better than all of the ReliefF generated feature sets. However, the performance loss is noticeable when using the full feature set, so that may not be a valid option. Given the first 13 percentile of recall, the unsupervised sets (green, blue, orange) consistently perform better than the supervised training set (yellow). If we are only interested in the first 13 percent, unsupervised trained ReliefF is the better choice for feature reduction even if supervised training data is available. Finally, around 25 percent recall all of the ReliefF sets are equivalent, so based on the quality of the results it does not matter which one you use. Training data is not always available, so having the ability to substitute unsupervised training instead of supervised is ideal. These results show that this not only is possible, but better to do in cases. Not shown in this graph (see Appendix A for the full graph), the supervised trained features actually performed marginally better after 30 percent recall. This is a side effect of the precision versus recall curves. Let us assume we are looking for the cats. Assuming all the dogs have already been returned before we have 30 percent of the cats returned, then the only direction for precision to go after that point is up. Basically everything not returned is a cat, so as we increase recall, precision will rise.

Something that should not be ignored, however, is the number of features required for the 0.1 threshold. With a threshold at 0.1, the supervised training set used only 355 features. K-Means with two clusters used 3005 features to represent the data. K-Means with four clusters used 4446, and k-means with 8 clusters used 8310 features. While this is barely over half the total features, it still is a noticeable feature difference. While we did not do a full study on time differences, we found the time difference between 355 features and 3005 features was negligible. With the difference in the quality of the results returned, both versions are comparable. To easily compare feature counts versus threshold and training types see Table 3.2.

Using ReliefF with a 0.01 threshold yielded similar results across all clustering types (see Figure 3.6).
Figure 3.5: Cat-Dog PvR values with 0.1 threshold

Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the ReliefF weighted features higher than 0.1. The supervised trained ReliefF features (yellow) performs worse than the unsupervised trained features (green, purple, orange) until 15 percent recall. Before 5 percent recall, the full feature set performs marginally better than the reduced features sets. However, after the first 5 percent, the unsupervised trained feature sets perform noticeably better than the full feature set. From 15 percent until 100 percent recall all ReliefF selected feature sets perform equivalently and show an improvement over the full feature set. For the full figure, see Appendix A

<table>
<thead>
<tr>
<th>ReliefF Threshold</th>
<th>Supervised</th>
<th>K-Means 2 Clusters</th>
<th>K-Means 4 Clusters</th>
<th>K-Means 8 Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>355</td>
<td>3005</td>
<td>4446</td>
<td>8310</td>
</tr>
<tr>
<td>0.01</td>
<td>3947</td>
<td>7827</td>
<td>10,475</td>
<td>11,448</td>
</tr>
<tr>
<td>0.001</td>
<td>5027</td>
<td>8643</td>
<td>10,982</td>
<td>11,680</td>
</tr>
</tbody>
</table>

Table 3.2: Number of Features by Threshold for Cat-Dog Data

The different training types generated by varying number of features across the ReliefF thresholds. Overall, the unsupervised training data generated more features than the supervised for each threshold, but the classification results generated were equivalent.
They are all better or equivalent to the baseline, and throughout most recall values the precision values are equivalent to each other. Overall k-means with four clusters shows a bit better early results, but this evens out in the end when it is equivalent to the others. K-Means with eight clusters doesn’t show any improvements. For this threshold, supervised used 3947 features. K-Means with two clusters used 7827 features. K-Means with four clusters used 10,475 features, and k-means with eight clusters used 11,448 features. As you can see this threshold is starting to get extreme for unsupervised ReliefF. However, even the elimination of roughly 1000 features caused a noticeable difference in early to middle precision versus recall values.

Figure 3.6: Cat-Dog PvR values with 0.01 threshold
Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the ReliefF weighted features higher than 0.01. Compared to the features thresholded at 0.1, the supervised trained ReliefF features (yellow) performs competitively against the unsupervised trained (green, purple, orange) trained features. The green line is trained using two clusters. The purple uses four clusters, and the orange uses eight clusters. This is consistent throughout figures where the cluster counts are variable. For the first 11 percent recall there is a noticeable drop in the precision scored generated by the supervised trained feature set. This drop is mirrored by the unsupervised trained sets, but not nearly as extreme from 5 percent to 14 percent. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set. From 20 percent until 100 percent recall, all ReliefF selected feature sets perform equivalently and show an improvement over the full feature set. For the full figure, see Appendix A

The last ReliefF threshold that we used was 0.001. This threshold was a bit extreme in all cases for the number of features. It however showed that even picking a small threshold, all four versions of training
ReliefF were qualitatively equivalent to each other. Figure 3.7 maps out the differences between each cluster type. As we can see, there is essentially no difference between the types. Unsupervised still shows a bit more variation as recall increases, but over all it is equivalent to the rest. However, they are all better than the full feature set. For the number of features used, supervised used 5027 features which is still under half the total number of features. K-means with two clusters used 8643. K-means with four clusters used 10,982, and k-means with eight clusters used 11,680 features.

![Precision vs Recall w/0.001 Threshold](image)

Figure 3.7: Cat-Dog PvR values with 0.001 threshold

Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the ReliefF weighted features higher than 0.001. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets but the difference between sets is lessened (as is the number of features). However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set. From 20 percent until 100 percent recall, all ReliefF selected feature sets perform equivalently and show an improvement over the full feature set. For the full figure, see Appendix A

From these results we can see KNN with ReliefF works equivalently for both unsupervised ReliefF training sets and supervised ReliefF training sets, but at the cost of higher over all feature weights. When only using thresholds to determine the number of features to select, these weights provide us with more features. We propose two possible explanations. The first has to do with noise. As we are using unsupervised training data, there are a number of samples misclassified. If a sample doesn’t actually promote ideal feature ranking, it would keep the ReliefF weights from adjusting themselves properly leaving the weights high. The second
idea has to do with the number of clusters. Theoretically in order to represent more clusters, one needs more 
features. Since ReliefF did not know the true number of clusters when training unsupervised, it generated 
the weights needed to rank based on more than two clusters. This would cause a general overall higher 
weight for certain features, as it would distinguish between clusters that actually shouldn’t have been there 
(i.e. distinguishes between black and white pictures).

Overall, however, it seems that ReliefF does not need to know the exact number of clusters as long as a 
user adjusts the threshold correctly. This is supported by evidence that even with the larger thresholds the 
unsupervised versions perform equivalently or better than the supervised trained version of ReliefF.

3.6 A Closer Look at the ReliefF Values

Our next step is to analyze the ReliefF values themselves, and how the number of features selected through 
a set threshold or by picking the top N perform compared to other feature sets trained in the same manner. 
Figures 3.8, 3.9, 3.10 and 3.11 show the precision and recall curves across the ReliefF feature counts while 
we hold the training types fixed. Did more features always perform better? In actuality, we found this wasn’t 
the case. Usually, there is a “sweet” spot for each one that performs optimally for the training data given. All 
of the figures include the full set as a comparable baseline.

The supervised data is the only one that produced noticeably different results. In the starting percentile, 
it actually performs noticeably worse than the baseline. It also is more sporadic. This is probably because 
the number of features selected just didn’t have the informative power to represent the ideal cluster results. 
Looking at the three k-means figures, we see the same story for all of them, more features didn’t mean better.

The ReliefF weights are shown in figure 3.12. It is obvious the unsupervised training set caused ReliefF 
to learn much higher weights than the supervised training set. This stems from the fact that the unsupervised 
versions both contained noise and had to define features for more than one category. ReliefF trained with 
K-Means set to eight clusters (green) shows this result the most. The weight curve shows a number of 
obvious drops as the weights progress. One drop happens to be around 1000 features, which explains why 
1000 features classified fairly well for this feature set. Another drop happens around 5000 features. The 
other unsupervised trained feature sets do have noticeable drops, but not nearly as strong as the eight cluster 
solution. This would explain why they tend to have similar feature counts and results most of the time. The 
supervised curve is fairly smooth showing the ReliefF actually had difficulty finding exact features to help 
with classification. These variations are further explored by looking at the actual features selected.

One area we explored was unsupervised ReliefF’s ability to remove redundant features. The images we 
used were gray scale images, so each feature was actually duplicated three times (RGB) when training ReliefF 
and selecting features. As supervised ReliefF does not handle redundant features, we assumed unsupervised
Figure 3.8: Cat-Dog PvR values across the Supervised Training Set
The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the supervised trained ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). Overall, we can see the supervised data varied fair amounts based on the number of features selected. Still for the first 5 percent of recall, all of them performed worse than the full feature set. However, after the first 5 percent they all performed better than the full feature set. See Appendix A for the full figure.
Figure 3.9: Cat-Dog PvR values with K-Means set to 2 Clusters

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the unsupervised trained with two clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). For the unsupervised data, which on average provided more features than the supervised when using a set value for the threshold (i.e. 0.1) more did not mean better. After the five percent recall, all of them perform equivalently to each other. The 1000 features version, which is one-third of the 0.1 threshold feature count, does performs worse only for only the first two percentile of recall. Overall, all of them perform better than the full feature set, after 5 percent recall. See Appendix A for the full figure.
Figure 3.10: Cat-Dog PvR values with K-Means set to 4 Clusters

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). For the unsupervised data, which on average provided more features than the supervised when using a set value for the threshold (i.e. 0.1) more did not mean better. All versions perform fairly equivalent to each other. This is a major drop between 10,000 features to 100 features, yet the curves have mostly the same results. All of them perform better than the full feature set after 5 percent recall, but vary before 5 percent. See Appendix A for the full figure.
Figure 3.11: Cat-Dog PvR values with K-Means set to 8 Clusters

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the unsupervised trained with eight clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). For the unsupervised data, which on average provided more features than the supervised when using a set value for the threshold (i.e. 0.1) more did not mean better. All versions perform fairly equivalent to each other. This is a major drop between 10,000 features to 100 features, yet the curves have mostly the same results. All of them perform better than the full feature set after 5 percent recall, but vary before 5 percent. See Appendix A for the full figure.
Figure 3.12: Cat-Dog Sorted Feature Weights
This figure shows the ReliefF weights sorted by weight and separated by training types. The blue line shows the supervised weights, which is a fairly smooth line. This explains the low feature count when using thresholds. The green line is the other extreme of ReliefF trained by K-Means with eight clusters. The overall feature counts are high, and there are noticeable drop offs in the graph. One drop off is near 1000 features, which explains the increase in performance when the top 1000 features were selected. The overlap of the red and yellow lines (two clusters and four clusters) supports the results that both training types returned a similar number of features.
ReliefF would also have similar issues. This was confirmed as most all features came in triplets of redundant weights. For example, the top 1000 features corresponding to the red, green and blue values is only 334 unique pixels. This clearly shows a need to experiment with Bins’s proposed system that removes redundant features while using ReliefF.

Looking at figure 3.13 we can see the weights applied to each feature across the different training sets. The whiter points show highly important features and the black ones show the features that were below the threshold. Gray points show lower weight features. As there are very few actual high weight features most of the figure shows black and dark grays. To analyze these features better, figure 3.14 is a normalized version of the features.

The supervised feature weight image is almost black, which illustrates that most of the features scored as irrelevant. The unsupervised training sets contain more gray points, which shows that ReliefF defined more features as salient features. While many of the features overlap (i.e. the left jaw bone), some feature weights vary greatly between the images. However, most of the features that contain more information for the supervised data are represented in the unsupervised training sets. This proves promising as unsupervised does find the same features, but it needs the additional features to compensate for the noise and more categories.

### 3.7 Conclusions for the Cat-Dog Data

Overall for the Cat-Dog data set, we find that unsupervised ReliefF works just as well as unsupervised, but it requires greater examination of the thresholds. For unsupervised ReliefF the overall values were higher, so either taking the top N features or using a higher threshold may be needed. This is supported by the fact that precision versus recall values are equivalent if not better for the unsupervised features compared to the supervised features. It further is supported by the ranking tests, in which unsupervised performed better than supervised using less features. Both versions of ReliefF also outperform the full feature set after the first 5 percent recall. As this is just one domain, our next area of interest is testing the same algorithm on World Wide Web documents.
Figure 3.13: Cat-Dog Weighted Features
The four images show an image representation of the ReliefF feature weights. The top left consists of the weights from the supervised learned ReliefF values. The top right contains the weights for the unsupervised trained ReliefF with K-Means set to two clusters. The bottom left uses K-Means set to four clusters and the bottom right uses K-Means set to eight clusters. These features are unnormalized showing the vast majority of features containing low feature weights. As the number of clusters increase, we can see the increase in number of feature weights. The increase in the overall weights comes from both the amount of noise in classification and the number of categories ReliefF had to represent.
Figure 3.14: Normalized Cat-Dog Weighted Features
The four images show an image representation of the ReliefF feature weights. The top left consists of the weights from the supervised learned ReliefF values. The top right contains the weights for the unsupervised trained ReliefF with K-Means set to two clusters. The bottom left uses K-Means set to four clusters and the bottom right uses K-Means set to eight clusters. The Normalized features help show the overlap between features across training types. As we can see, some features such as the left jaw bone overlap, but other features such as cat ears do not show up in the supervised trained weights.
Chapter 4

Experiments - World Wide Web Documents

*On the Internet, nobody knows you’re a dog.*

Peter Steiner, cartoon in The New Yorker, July 5, 1993

4.1 Introduction

It is estimated that the size of the Web now is billions of documents, with Google reporting four billion searchable pages in May of 2004 [11]. The number of documents makes it difficult to search the Web for viable information, and some companies are turning towards directories to narrow the search space for document retrieval. Currently, the larger directory sites such as Yahoo and the Open Directory Project are using humans to classify web pages into categories. This is both an expensive and time consuming process, and demand for research is growing in the automated classification of text into categories. However, the English language can easily generate over 260,000 words [26] with 50,000 commonly used words [20]. The number of Web pages makes categorization a daunting task. Feature selection is clearly warranted for eliminating the unneeded words in each document. Previously, we have used supervised trained ReliefF for webpage classification on internal projects. This led us to believe unsupervised trained ReliefF would work within the domain.

4.2 DMOZ Web Document Data

We used categories and Web pages from The Open Directory Project (dmoz.org) as our dataset. Open Directory is a project where human volunteers manually classify documents into directory categories. The volunteers are not trained in classification, and rarely are they experts in their area. More traditionally, they are enthusiasts in the category they are classifying. However, with that said, a number of well known search engines including Google use the Open Directory Project for their Web page directory listings. We chose this dataset for scalability and for the challenges it presents.
To limit the size of our initial tests, we trained and tested on the specific category of ‘Games:Video Games’. The six categories used were: action, board games, cards, racing, role-playing, and sports. This was a more difficult task for classification than merely classifying at a higher level such as ‘sports’ vs. ‘health’ vs. ‘politics’, etc., because pages were related to each as they all fit under the umbrella of ‘video games’. The corpus consisted of 3,173 web documents. The corpus was made up of 732 documents from action, 146 from board games, 50 from cards, 1057 from racing, 1075 from rollplaying and 113 from sports. This was divided into seven randomly sampled variants without preference to categories. ReliefF training and classification data was made up of about 30/70 respectively as with image data. Overall, the dataset is fairly challenging.

We chose a bag-of-words (BOW) feature set for analysis. These were all English language pages, which amounts to thousands of words in our feature set. Given that the number of commonly used English words is around 50,000 words [20], we needed to minimize the number of words to analyze. Web documents were stripped of all HTML and JavaScript code using HTMLParser [10] and punctuation removed. All words were cross-checked for validity and removed if it did not exist in a dictionary file [29], and stop words (common words such as ‘a’, ‘or’, and ‘the’) were removed. Each word was then stemmed using the Porter Stemmer [25]. Term frequency counts were then counted on the remaining words with the help of the CMU Statistical Language Modeling Toolkit [8]. The basic feature set consisted of 12,579 terms. Each feature was then normalized using TFIDF, which is fairly standard along with stemming, stop word removal and dictionary checking within the domain of Web document retrieval [4].

For our experiments we use the same procedure presented in Chapter 3. As 2, 4 and 8 clusters would have very little meaning for the dataset, we instead used 6, 12 and 24. We use the Cosine Distance between documents, which has been used in web document retrieval [4].

Throughout all of our tests we have calculated the standard deviation. However, differences between the Cat-Dog dataset runs are minimal (0.001) compared to the differences between the DMOZ dataset runs. The inherent difference in the size of the set and the category distributions make the DMOZ dataset a much more difficult dataset to classify. As such, the standard deviation ranges from 0.01-0.03 for the tests, with an average of 0.02. This causes the following tests to look significantly different at first, but once we calculate the overlap of the standard deviations, we find that many of the curves are equivalent. Additionally, since the categories are unbalanced, the precision of a random classifier is not intuitive to calculate. Given the distribution of the categories, we calculate the average precision of a random classifier to be 0.09 or 9 percent. This is approximately constant overall all fixed recall values. This baseline, not drawn in every figure, helps us determine if ReliefF performs better than random even though the results are fairly low. In all cases, it does.
4.3 ReliefF as a Feature Ranking System

Once again, the major variable for k-means is determined by the number of clusters. We tested on six clusters first as that was the number of supervised categories, then twelve and twenty-four clusters to represent the data. We fix the number of features used. For our tests we took the top 100, 1000 and 2500 features determined by ReliefF across each training type. 2500 features were selected as anything over 2500 would mean we would be taking random 0 weight features for all four test sets. Figures 4.1, 4.2 and 4.3 show the differences in precision versus recall scores across the number of features used.

![Precision vs Recall Using Top 100 Features](image)

Figure 4.1: Web Document PvR values using top 100 features

Precision versus Recall curves across training types using a fixed feature count of 100. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the top 100 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. Selecting the same number of features for both unsupervised trained ReliefF and supervised trained ReliefF proved to have similar results when mapping out the precision and recall curves. Both supervised (yellow) and the unsupervised trained ReliefF feature sets (green, purple, orange) started with low precision values until 10 percent or higher recall. In the first 10 percent recall, the full feature set has the highest precision without much competition. However, we are only comparing 100 features against 12,579 features. For the full figure, see Appendix B.

Figure 4.1 shows the difference between ReliefF reduced feature sets across unsupervised and supervised training when we fix the number of features to 100. This number is a fairly low feature count given the
results that will be presented in Section 4.4. When comparing the curves, there is actually a fair amount of crossover. The supervised (yellow) and k-means trained with 12 clusters (purple) Relieff generated values run are indistinguishable throughout the graph. K-Means with six (green) and twelve (orange) clusters also are indistinguishable with their average standard deviations overlapping. Before 10 percent recall the full feature set performs significantly better than the Relieff generated features. This shows that taking only 100 of the 12,579 features may not be enough to fully represent the categories. After 10 percent recall, K-Means with 6 and 24 clusters trains features that are equivalent and eventually better than the full feature set. By 50 percent recall all of the curves are higher than the full feature set and equivalent to each other.

**Figure 4.2: Web Document PvR values using top 1000 features**

Precision versus Recall curves across training types using a fixed feature count of 1000. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the top 1000 Relieff weighted features. By sampling this way, we are using Relieff as a ranking system and comparing each training type against others with the exact same number of features. As the number of features increase so does the precision for all of the reduced feature sets. In the first percentile, the full feature set still performs better. After the first percentile until the 7 percent to 11 percent recall, the reduced features do better. Not shown on this figure, at 40 percent recall all of the reduced features cross the full feature set line again. Both the supervised and unsupervised trained Relieff features produce qualitatively equivalent precision versus recall curves with an average standard deviation of 0.02. For the full figure, see Appendix B.

In Figure 4.2 the differences between the different feature training types becomes small. Both supervised and unsupervised feature sets start lower than the full feature set, quickly out perform it only to drop below it.
between 7 percent and 11 percent recall. They all then slowly increase precision until they are equivalent with it around 50 percent recall (see Appendix B for the full figure). The standard deviation between the precision values was 0.02 making all of the curves overlap. This shows that the ReliefF trained with unsupervised labeled data performs equivalently to ReliefF trained with supervised data. However, unlike the vision data, unsupervised doesn’t do better in the earlier percentages of recall, but it still performs equivalently. Similar to the curves in Figure 4.1, k-means with six clusters (green) and k-means with twenty-four clusters (orange) tend to follow very similar paths.

Figure 4.3: Web Document PvR values using top 2500 features

Precision versus Recall curves across training types using a fixed feature count of 2500. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the top 2500 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. 2500 features was selected as it was still within the range provided by the thresholds, but very high compared to the number of features the unsupervised sets were using. However, the unsupervised sets performed very well with 2500 features, and k-means with 6 clusters (green) actually is noticeably different between recall 5 percent and recall 15 percent. Beyond that time there really is no difference between any of the sets. Not shown, but around 40 percent recall all of the lines cross and are equivalent to the full feature set. For the full figure, see Appendix B.

In Figure 4.3 we see the same story as Figure 4.2. We however do witness a switch in the results for this figure. Instead of supervised (yellow) and k-means with 12 clusters (purple) pairing together, k-means with 6
clusters is closer to the supervised Relieff features. However, the standard deviations for the supervised set and k-means with 6 clusters overlap each other making the two curves equivalent. We do notice though that even with the overlap, k-means with 6 clusters is still slightly better than the supervised set at recall of 12 percent. Overall all, the reduced feature sets are roughly equivalent to each other from recall 1 to 5 percent and then again from 15 to 100 percent. The full set performs better in spots, but the reduced sets perform better around 5 percent to 15 percent recall and 50 percent to 100 percent recall.

Based on these three figures, it should be sufficient to say the supervised trained Relieff and the unsupervised trained Relieff rank features in a similar manner. However, they both had difficulties outperforming the full feature set. This leads us to believe more features are needed, yet this is limited by the weights of the features. As we shall see in the next section, the weights change drastically as they decrease.

### 4.4 Results Based on Cluster Differences

As with the Cat-Dog data we set our thresholds at 0.1, 0.01 and 0.001 across the different Relieff generated weights. This also helps us examine the actual differences between the weights and their effect on classification values. Figures 4.4, 4.5 and 4.6 contain the results across the different cluster sizes but with the threshold fixed in each one.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>29</td>
<td>36</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>0.01</td>
<td>1657</td>
<td>116</td>
<td>75</td>
<td>24</td>
</tr>
<tr>
<td>0.001</td>
<td>4222</td>
<td>237</td>
<td>203</td>
<td>106</td>
</tr>
</tbody>
</table>

Table 4.1: Number of Features by Threshold for World Wide Web Data

The different training types generated by varying numbers of features across the Relieff thresholds. For the WWW data, the majority features of are presumably noise, making it difficult for the unsupervised trained Relieff to differentiate. This caused the majority of the weights to drop to zero or below, explaining the low feature counts.

In Figure 4.4 the part that stands out the most is the difference in the first 10 percent recall versus precision values. The precision for all of the feature selection schemes is extremely low compared to the full set shown in red. Looking further into this we find that the difference stems from the number of features. Using the supervised trained weights and setting the threshold at 0.1, we only included 29 of the 12,579 features. This explains the significant drop in processing time, but also the poor precision for the early percentiles. For the K-means with six clusters, the 0.1 threshold returned 36 features. For K-means with twelve clusters the threshold returned 12 features, and finally for K-means with twenty-four clusters the threshold returned only 9 features. Table 4.1 shows the number of features compared to the weight threshold. The number of features combined with the early precision values leads us to believe that the 0.1 threshold was too extreme for all Relieff trained features. However, after the 10th percentile, the features learned by Relieff actually started
Figure 4.4: Web Document PvR values with 0.1 threshold

Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the ReliefF weighted features higher than 0.1. In this figure, there is a major difference in between the full feature set and the reduced feature sets. However, after the first 10 percent, the reduced feature sets perform noticeably better than the full feature set. For the full figure, see Appendix B.
Figure 4.5: Web Document PvR values with 0.01 threshold
Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the ReliefF weighted features higher than 0.01. In this figure, the sets using the unsupervised features perform noticeably worse than both the supervised reduced feature sets and the full set from 1 percent to 10 percent recall. However, after 20 percent recall, the unsupervised sets perform better than both the full feature set and the supervised. For the full figure, see Appendix B.

Figure 4.5 shows a noticeable difference between the supervised trained ReliefF features and the unsupervised trained ReliefF. However, after 20 percent recall the unsupervised actually performs noticeably better than the supervised feature set. This also stems from the number of features. The supervised set threshold at 0.01 contains 1657 features, but the six, twelve and twenty-four unsupervised sets contain 116, 75 and 24 features respectively. Since the feature count is still fairly low, it makes sense there is very little change in the early percentiles as before. However, it is also obvious that more features are needed to represent precision in the early percentiles. For the supervised and full set of features, the peak around 7 percent comes from the fact many of the documents contained frames. These frames all looked similar but really had little information content in them making them difficult to classify correctly.

Figure 4.6 tells a similar story using the 0.001 threshold. However, there is a noticeable increase in
Figure 4.6: Web Document PvR values with 0.001 threshold

Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the ReliefF weighted features higher than 0.001. In this figure, we see a notable change in the unsupervised results compared to the other figures. This change, while still not better than either the supervised set or the full set in the early 30 percentiles, leads us to believe that more features are needed for the unsupervised set. Not shown in the figure, is after 30 percent recall the unsupervised performs equivalently to the supervised set. For the full figure, see Appendix B.
performance for the unsupervised values, showing they are approaching the ideal feature count for their representation. The supervised dataset contains 4,222 features, while the unsupervised datasets contain 237, 203 and 106 features respectively with six, twelve and twenty-four clusters.

Overall, the unsupervised ReliefF weights do not produce high enough weights for large number of features. Because of these low overall weights, just thresholding the values proved a difficult test for the World Wide Web data, and may not have been the suitable. However, the ranking test shown in the previous section, along with the results here support the conclusion that ReliefF trained with unsupervised labels performs equivalently to ReliefF trained with supervised labels.

4.5 A Closer Look at ReliefF Values

Once again we want to analyze the ReliefF values themselves, and how the number of features selected through a set threshold or by picking the top N perform compared to other feature sets trained in the same manner. Figures 4.7, 4.8, 4.9 and 4.10 show the differences.

<table>
<thead>
<tr>
<th>Supervised Training Set</th>
<th>K-Means w/6 Clusters</th>
<th>K-Means w/12 Clusters</th>
<th>K-Means w/24 Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>reconstruct</td>
<td>addison</td>
<td>reconstruct</td>
<td>reloc</td>
</tr>
<tr>
<td>den</td>
<td>lucia</td>
<td>tanaka</td>
<td>reconstruct</td>
</tr>
<tr>
<td>bin</td>
<td>tanaka</td>
<td>addison</td>
<td>addison</td>
</tr>
<tr>
<td>militia</td>
<td>sept</td>
<td>intrepid</td>
<td>hindi</td>
</tr>
<tr>
<td>magazin</td>
<td>keith</td>
<td>lucia</td>
<td>hindu</td>
</tr>
<tr>
<td>directori</td>
<td>ten</td>
<td>sept</td>
<td>lucia</td>
</tr>
<tr>
<td>domain</td>
<td>historian</td>
<td>militia</td>
<td>tanaka</td>
</tr>
<tr>
<td>cursori</td>
<td>spiller</td>
<td>keith</td>
<td>keith</td>
</tr>
<tr>
<td>oldi</td>
<td>danish</td>
<td>creator</td>
<td>sept</td>
</tr>
<tr>
<td>wilkinson</td>
<td>summarile</td>
<td>credit</td>
<td>volum</td>
</tr>
</tbody>
</table>

Table 4.2: Top 10 weighted words across different training types

This table shows top 10 words based on the ReliefF generated weight value for each word. As we can see there are some words that overlap across different categories such as ‘reconstruct’ and ‘militia’, but other words are specific to categories such as ‘oldi’ for the supervised training set. These words do not correspond with the category names, but they do have enough meaning to differentiate categories. The words seem to have odd endings as these are the words after stemming occurs. For example, ‘volum’ really is ‘volume’.

All of the Figures seem to show considerable differences between them. However, a closer look at the values reveals they actually end up equivalent when they pass 50 percent recall (see Appendix B). Before that point, the higher feature lines all follow the same pattern with 0.02 standard deviation of each-other and the lower feature lines also follow similar patterns. One area which needs to be examined in the future is how exactly are the ReliefF values being affected by the text data? Some of the curves are fairly variable, and what pages exactly would cause that? One thing to note, which leads us into looking at the actual ReliefF weight curve, is for k-means trained on 24 clusters, only 288 features had a weight above zero. That means taking 1000 and 2500 features effectively took features in alphabetically order after the first 288 features.

38
Figure 4.7: Web Document PvR values across the Supervised Training Set
The figure illustrates Precision versus Recall values only using the full feature set (red, 12,579 features) and the supervised trained ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). Overall, we can see the supervised data varied fair amounts based on the number of features selected. Eventually, however, they all evened out to roughly be equivalent to the full feature set and each other. For the full figure, see Appendix B.
This could explain some of the large variations we receive for that data. For a closer look at the top 10 words used, see 4.2.

Figure 4.8: Web Document PvR values with K-Means set to 6 Clusters

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,279 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). For the unsupervised data, which on average actually produced lower feature counts than the supervised, lower did not mean better. On average, as one increased the number of features there early levels of precision also increased. Versions with over 100 features performed fairly equivalent to each other. See Appendix B for the full figure.

Looking at figure 4.11 we find that there really is little difference between the weights of the features themselves. Actually, these results are opposite from the vision results shown in chapter 3. Given these weights, ReliefF is assuming that most of the features are noise, since so few of the features actually have meaning. The noise generated only compounds this as the algorithm is unable to differentiate between feature noise and cluster noise. However, even with these results ReliefF is able to pick out a few key features that actually end up providing more useful information in the later stages of recall. There also is a fundamental difference about the representation power of the data itself. For an image, a group of pixels is needed to truly represent something with meaning. It may be a small group or large group depending on the meaning, but a single pixel really doesn’t have the representational power by itself. With words, this does not hold true. A single word could describe the entire document, while the rest of the words are just ‘filler’ words. Of course,
with a word the opposite can be true, it may be the combination of key-words that give detail. The English language is variable, and as such, the features generated also can be very variable.

**Figure 4.9: Web Document PvR values with K-Means set to 12 Clusters**

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,279 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). For the unsupervised data, which on average actually produced lower feature counts than the supervised, lower did not mean better. On average, as one increased the number of features there early levels of precision also increased. Versions with over 100 features performed fairly equivalent to each other. See Appendix B for the full figure.

### 4.6 Conclusions for World Wide Web Document Data

Overall, our World Wide Web document dataset is a very difficult set to classify. Because it is a difficult set to classify, it also is a difficult set to build a minimal unsupervised training set. The large amounts of noise generated both by non-informative features and from misclassifications caused some difficulty for unsupervised trained ReliefF. However, unsupervised trained ReliefF still had the representational power to perform better in the later recall percentiles. Additionally, as one increased the number of features the precision also increased considerably in the early levels. This leads us to believe there still is an ideal “sweet spot” for the number of features to select. However, exploring this spot in detail is beyond the scope of this thesis and falls under future work.
Figure 4.10: Web Document PvR values with K-Means set to 24 Clusters

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,279 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). For the unsupervised data, which on average actually produced lower feature counts than the supervised, lower did not mean better. On average, as one increased the number of features there early levels of precision also increased. Versions with over 100 features performed fairly equivalent to each other. See Appendix B for the full figure.
The sorted weight curve for the World Wide Web document data is extremely steep. Supervised is only 0.02 points higher than unsupervised in many cases. This shows that some words have extremely high informative power when classifying documents, but other words are fillers causing noise.
Chapter 5

Conclusions

*God gave us our memories so that we might have roses in December.*

James Matthew Barrie

5.1 Discussion

Many problems such as classification and retrieval problems are inundated with virtually unlimited numbers of features. Images, for example, can be acquired at arbitrarily high resolutions, and web documents can always include other words and parts of speech. Sadly, it is difficult to process all of these features at one time. Additionally, many of these features are irrelevant and may be detrimental to classification algorithms. Feature selection is important in these domains.

ReliefF is a feature selection algorithm that has proved itself across a number of fields \[18, 6, 34, 19, 28, 24\]. It can select features without making major assumptions about the underlying data distribution. It is a robust algorithm, that works well across domains. However, it is a supervised algorithm, and supervised data is not always available. It can be labor intensive and expensive to label training data, and there are cases when hand labeled data is not ideal. The system we propose eliminates the need for supervised data with ReliefF.

As defined by Kononenko, ReliefF uses supervised data to determine relationships between samples. Using supervised data, Kononenko is also able to show ReliefF is robust to noise and misclassifications of the data samples. As such, we rely on the robustness of ReliefF when implementing our unsupervised solution. Instead of using supervised training categories, we run an unsupervised clustering algorithm, k-means, to classify the training data. We then use the labels created by k-means to select categories for training ReliefF. Since we are looking for a domain independent feature selection algorithm, we test unsupervised ReliefF and compare it to supervised ReliefF on two different domains. The first domain consists of cat and dog images.

For the Cat-Dog data we show unsupervised trained ReliefF works just as well if not better in certain cases than supervised trained ReliefF. At first, it seems like it is working but with the caveat of needing more
features, but this isn’t the case. The rank based feature tests show that even taking only 1000 of the 12,288 features the unsupervised trained ReliefF works better than supervised for recall values 0.01 to 0.15. After 15 percent, however, they work equivalently.

The second domain consists of World Wide Web text documents. We show that also in that domain, the ReliefF trained with supervised labels and ReliefF trained with unsupervised labels select equivalent features. All of which are also better than random, but in early percentages of recall worse than the full feature set. After 0.3 recall, all versions are equivalent. Additionally, the weights generated are overall very low weights, with only one-third of the weights higher than zero. The World Wide Web document is a difficult domain both with variations in the number of samples per category, and how the features interact with the data. One word can have more meaning than one pixel. Further study is needed in this area with both versions of ReliefF, as both versions had equal amounts of difficultly in the domain.

Overall, the proposed system is a feasible alternative to the supervised version. It may not always be the best system to use, as some of our results show the full feature set actually works better for certain percentages of recall, but it does work when supervised data is not an option.

\section*{5.2 Future Work}

Throughout this study, there are a number of key areas which we would have liked to address, but time and scope did not permit. The most notable of these topics are as follows:

1. Dataset Differences: The two domains of computer vision and World Wide Web documents are vast domains, yet we have only tested one dataset from each. Additionally, the datasets we have are fairly different in the number of categories, size of categories and balance between the categories. A larger study should include variations of the dataset to examine if variances between ReliefF for the domains were determined by the differences in the datasets or the differences of the semantics between the domains. For example, we could include an image dataset that had six categories and 100 images each. We also could include a World Wide Web dataset that had two categories, or test vision categories that had unbalanced category counts. While we purposely examined two very different datasets, this was not an exhaustive study.

2. Additional Domains: A domain independent feature selection algorithm is our second defining goal for feature selection. Our studies show that unsupervised ReliefF works well independent of two domains. However, as with supervised ReliefF, there are a number of additional domains that need to be tested. Bioinformatics may be a good domain for unsupervised ReliefF. It also traditionally has large feature sets with small training sets. A larger study should include other domains which supervised ReliefF
has shown to be a good feature selection algorithm.

3. Redundant Features: Supervised ReliefF does not handle redundant features. As we expected, unsupervised ReliefF does not eliminate redundant features. However, many of our interest domains have redundant features that may or may not have meaning. Bins proposed a system of using correlation to remove redundant features. This system should also work for unsupervised ReliefF, and we feel would be a good addition to the system. As such, in the future we would like to use Bins's proposed system with unsupervised ReliefF for removal of redundant features.

4. Clustering: The most notable area for examination, in our opinion, is the actual affect K-Means has on unsupervised ReliefF. K-Nearest-Neighbor is at the heart of both ReliefF and K-Means. Is there an intrinsic relationship between the two making K-Means ideal no matter what the domain? Additionally, as the K-Means cluster labels are labeling based on the distance, how does that affect the misses? We have noticed that unsupervised ReliefF actually may be able to represent documents better as it ignores human-defined semantics. For example, dogs may be separated into groups of pointy-eared dogs and floppy eared dogs. A closer analysis of the clusters themselves, and ReliefF's relationship to K-Means is needed. What are the clusters that are being formed? Is the medium similar to the feature counts? Do other clustering algorithms work better or worse than K-Means for unsupervised ReliefF?

Overall, we there are a number of variations to unsupervised ReliefF. Most of them are in the training/naming of the actual clusters. However, the true question is in what situations will unsupervised ReliefF select better or equivalent features compared to the features selected by supervised ReliefF. We are able to show that it does for our two test datasets, but larger studies may be needed.
Appendix A

Full Figures for Cat-Dog Data
Figure A.1: Cat-Dog PvR values with 0.1 threshold: Full

Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the ReliefF weighted features higher than 0.1. The supervised trained ReliefF features (yellow) performs worse than the unsupervised trained features (green, purple, orange) until 15 percent recall. Before 5 percent recall, the full feature set performs marginally better than the reduced features sets. However, after the first 5 percent, the unsupervised trained feature sets perform noticeably better than the full feature set. From 15 percent until 100 percent recall all ReliefF selected feature sets perform equivalently and show an improvement over the full feature set.
Figure A.2: Cat-Dog PvR values with 0.01 threshold: Full

Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the ReliefF weighted features higher than 0.01. Compared to the features thresholded at 0.1, the supervised trained ReliefF features (yellow) performs competitively against the unsupervised trained (green, purple, orange) trained features. However, even for the first 11 percent recall there is a noticeable drop in the precision scored generated by the supervised trained feature set. This drop is mirrored by the unsupervised trained sets, but not nearly as extreme from 5 percent to 14 percent. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set. From 20 percent until 100 percent recall, all ReliefF selected feature sets perform equivalently and show an improvement over the full feature set.
Figure A.3: Cat-Dog PvR values with 0.001 threshold: Full Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the ReliefF weighted features higher than 0.001. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets but the difference between sets is lessened (as is the number of features). However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set. From 20 percent until 100 percent recall, all ReliefF selected feature sets perform equivalently and show an improvement over the full feature set.
Figure A.4: Cat-Dog PvR values using top 100 features: Full Precision versus Recall curves across training types using a fixed feature count of 100. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the top 100 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. The features generated by both supervised trained ReliefF values (yellow) and unsupervised trained with two k-means clusters (green) perform poorly between 1 percent and 19 percent recall, but they improve as they reach later recall values. K-means with eight and four clusters both perform fairly well given that they are only using 100 of the 12,288 features for sampling. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set.
Figure A.5: Cat-Dog PvR values using top 1000 features: Full Precision versus Recall curves across training types using a fixed feature count of 1000. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the top 1000 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. The features generated by the supervised trained ReliefF values (yellow) still perform poorly between 1 percent and 19 percent recall, but they improve as they reach later recall values. The k-means with two clusters is now performing competitively against the full feature set and the other unsupervised trained ReliefF selected features. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set.
Figure A.6: Cat-Dog PvR values using top 10,000 features: Full Precision versus Recall curves across training types using a fixed feature count of 10,000. The red line represents the full feature set. The full feature set includes all 12,288 features. All other lines are samples using only the top 10,000 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. Before 5 percent recall, the full feature set still performs marginally better than the reduced features sets. However, after the first 5 percent, the ReliefF trained feature sets perform noticeably better than the full feature set.
Figure A.7: Cat-Dog PvR values across the Supervised Training Set: Full

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the supervised trained ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). Overall, we can see the supervised data varied fair amounts based on the number of features selected. Still for the first 5 percent of recall, all of them performed worse than the full feature set. However, after the first 5 percent they all performed better than the full feature set.
Figure A.8: Cat-Dog PvR values with K-Means set to 2 Clusters: Full

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the unsupervised trained with two clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). For the unsupervised data, which on average provided more features than the supervised when using a set value for the threshold (i.e. 0.1) more did not mean better. After the five percent recall, all of them perform equivalently to each other. The 1000 features version, which is one-third of the 0.1 threshold feature count, does performs worse only for only the first two percentile of recall. Overall, all of them perform better than the full feature set, after 5 percent recall.
The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). For the unsupervised data, which on average provided more features than the supervised when using a set value for the threshold (i.e. 0.1) more did not mean better. All versions perform fairly equivalent to each other. This is a major drop between 10,000 features to 100 features, yet the curves have mostly the same results. All of them perform better than the full feature set after 5 percent recall, but vary before 5 percent.
The figure illustrates Precision versus Recall values only using the full feature set (red, 12,288 features) and the unsupervised trained with eight clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 10,000). For the unsupervised data, which on average provided more features than the supervised when using a set value for the threshold (i.e. 0.1) more did not mean better. All versions perform fairly equivalent to each other. This is a major drop between 10,000 features to 100 features, yet the curves have mostly the same results. All of them perform better than the full feature set after 5 percent recall, but vary before 5 percent.
Appendix B

Full Figures for World Wide Web Data
Figure B.1: Web Document PvR values with 0.1 threshold: Full
Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the ReliefF weighted features higher than 0.1. In this figure, there is a major difference in between the full feature set and the reduced feature sets. However, after the first 10 percent, the reduced feature sets perform noticeably better than the full feature set.
Figure B.2: Web Document PvR values with 0.01 threshold: Full Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the ReliefF weighted features higher than 0.01. In this figure, the sets using the unsupervised features perform noticeably worse than both the supervised reduced feature sets and the full set from 1 percent to 10 percent recall. However, after 20 percent recall, the unsupervised sets perform better than both the full feature set and the supervised.
Figure B.3: Web Document PvR values with 0.001 threshold: Full Precision versus Recall curves across training types. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the ReliefF weighted features higher than 0.001. In this figure, we see a notable change in the unsupervised results compared to the other figures. This change, while still not better than either the supervised set or the full set in the early 30 percentiles, leads us to believe that more features are needed for the unsupervised set. After 30 percent recall the unsupervised performs equivalently to the supervised set.
Figure B.4: Web Document PvR values using top 100 features: Full

Precision versus Recall curves across training types using a fixed feature count of 100. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the top 100 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. Selecting the same number of features for both unsupervised trained ReliefF and supervised trained ReliefF proved to have similar results when mapping out the precision and recall curves. Both supervised (yellow) and the unsupervised trained ReliefF feature sets (green, purple, orange) started with low precision values until 10 percent or higher recall. In the first 10 percent recall, the full feature set has the highest precision without much competition. However, we are only comparing 100 features against 12,579 features.
Precision vs Recall Using Top 1000 Features

Figure B.5: Web Document PvR values using top 1000 features: Full Precision versus Recall curves across training types using a fixed feature count of 1000. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the top 1000 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. As the number of features increase so does the precision for all of the reduced feature sets. In the first percentile, the full feature set still performs better. After the first percentile until the 7 percent to 11 percent recall, the reduced features do better. Not shown on this figure, at 40 percent recall all of the reduced features cross the full feature set line again. Both the supervised and unsupervised trained ReliefF features produce qualitatively equivalent precision versus recall curves with an average standard deviation of 0.02.
Figure B.6: Web Document PvR values using top 2500 features: Full
Precision versus Recall curves across training types using a fixed feature count of 2500. The red line represents the full feature set. The full feature set includes all 12,579 features. All other lines are samples using only the top 2500 ReliefF weighted features. By sampling this way, we are using ReliefF as a ranking system and comparing each training type against others with the exact same number of features. 2500 features was selected as it was still within the range provided by the thresholds, but very high compared to the number of features the unsupervised sets were using. However, the unsupervised sets performed very well with 2500 features, and k-means with 6 clusters (green) actually is noticeably different between recall 5 percent and recall 15 percent. Beyond that time there really is no difference between any of the sets. Not shown, but around 40 percent recall all of the lines cross and are equivalent to the full feature set.
The figure illustrates Precision versus Recall values only using the full feature set (red, 12,579 features) and the supervised trained ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). Overall, we can see the supervised data varied fair amounts based on the number of features selected. Eventually, however, they all evened out to roughly be equivalent to the full feature set and each other.
Figure B.8: Web Document P-R values with K-Means set to 6 Clusters: Full

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,279 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). For the unsupervised data, which on average actually produced lower feature counts than the supervised, lower did not mean better. On average, as one increased the number of features there early levels of precision also increased. Versions with over 100 features performed fairly equivalent to each other.
The figure illustrates Precision versus Recall values only using the full feature set (red, 12,279 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). For the unsupervised data, which on average actually produced lower feature counts than the supervised, lower did not mean better. On average, as one increased the number of features there early levels of precision also increased. Versions with over 100 features performed fairly equivalent to each other.
Figure B.10: Web Document PvR values with K-Means set to 24 Clusters: Full

The figure illustrates Precision versus Recall values only using the full feature set (red, 12,279 features) and the unsupervised trained with four clusters ReliefF selected feature sets. Shown are all three of the threshold points (0.1, 0.01 and 0.001) and the three fixed feature counts (100, 1000 and 2500). For the unsupervised data, which on average actually produced lower feature counts than the supervised, lower did not mean better. On average, as one increased the number of features there early levels of precision also increased. Versions with over 100 features performed fairly equivalent to each other.
REFERENCES


