A Comparison of Techniques for Detecting Abnormal Change in Blogs

Paul Logasa Bogen II
Intelligent Computing Research Team
Computational Data Analytics Group
Oak Ridge National Laboratory
Oak Ridge, Tennessee
bogenpl@ornl.gov

Frank Shipman, Richard Furuta
Center for the Study of Digital Libraries and
Dept. of Computer Science and Engineering
Texas A&M University
College Station, Texas
dcm@csdl.tamu.edu

ABSTRACT

Distributed collections are made of metadata entries that contain references to artifacts not controlled by the collection curators. These collections often have limited forms of change; for digital distributed collections, primarily creation and deletion of additional resources. However, there exists a class of digital collection that undergoes additional kinds of change. These collections consist of resources that are distributed across the Internet and brought together via hyperlinking. Resources in these collections can be expected to change as time goes on. Part of the difficulty in maintaining these collections is determining if a changed page is still a valid member of the collection. Others have tried to address this by defining a maximum allowed threshold of change, however, these methods treat change as a potential problem and treat web content as static despite its intrinsic dynamicism. Instead we acknowledge change on the web as a normal part of a web document and determine the difference between what a maintainer expects a page to do and what it actually does. In this work we evaluate options for extractors and analyzers from a suite of techniques against a human-generated ground-truth set of blog changes. The results of this work show a statistically significant improvement over traditional threshold techniques for our collection.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: Collection, Systems issues, User issues; H.5.4 [**Hypertext/Hypermedia**]: User issues; I.5.2 [**Design Methodology**]: Feature evaluation and selection

General Terms

Management, Design, Experimentation, Human Factors

Keywords

Distributed Collection Manager, Ensemble, Pattern Classification, Change Analysis

JCDL '12 Washington, D.C., USA

1. INTRODUCTION

It is well known [19] that as distributed collections age, their constituents change, become obsolete, or disappear. The result of collection aging has been a diminishing value of the collection despite the significant effort required for collection creation. If the lifespan of collections can be extended, users may be more motivated to create them and thus decrease or prevent instances of rediscovery in the future. It is this problem that the Distributed Collection Manager (DCM)–the successor to the Walden's Paths Project's PathManager path maintenance utility [8]–was designed to address. DCM is motivated not only by the observation that the fluidity of web pages leads to collections becoming stale, requiring revisions and updates [14], but also by observations that the web as a whole has changed and that assumptions we made in the past may no longer hold [7]. Unlike PathManager, which was used to maintain paths for Walden's Paths, DCM supports distrbuted collections of web sites in general.

To help inform the design of DCM, we performed two studies. The first study recruited users of social news sites and asked them about their habits regarding their personal collections including tools they use, kinds of sites they visit, and issues they face in managing their collections [9]. This study was followed with a series of interviews where we examined personal collections themselves, primarily bookmarks. Our studies affirmed the existence of personal collections of highly dynamic web sites, the inadequacies of existing tools to maintain these collections, and the importance of content over presentation and interactive features of a site for user re-evaluation of web pages.

Since 2006, we have been caching a collection of a wide variety of popular blogs [7]. From these caches, we observed patterns in the behavior of blogs [7] and followed this with a qualatative evaluation of our Kalman Filter based modelling technique [8]. Kalman filters are are a technique that makes predictions of future states of noisy systems based on the history of observations. Thus, when provided a sequence of previous feature vectors of a resource, the filter will predict the feature vector, and thereby its changs, for the next observation. From these observations, we were able to subjectively match patterns in the filter's behavior to abnormal events in blogs. Given this subjective validation of our techniques we sought to objectively quantify our results. This paper follows up on our prior work by describing our work

building a ground truth set for blog change that we then use to evaluate a suite of Kalman filter based techniques to determine the best techniques in our suite. These best techniques are then compared to traditonal cosine similarity methods to show the improvement in performance over prior work.

The remainder of this paper will first discuss the problem of change management and some prior work surrounding change management and filter methods. This will be followed by a discussion of the construction of our ground truth set. Then we will discuss our selection of features. Next we will examine both the comparative evaluation between filter based techniques and cosine similarity based techniques. Finally we will discuss our conclusions and areas of future work.

2. PROBLEM

When a distibuted collection of web resources is created, the curator has certain expectations to how the collection will or won't change. For instance, when a user creates a collection of news sites as a reference aid, they most likely expect the sites not to remain static. Not only would a change in behvaior such as topic shift or abnormally frequent changes would be considered problematic, but so would an abnormal lack of change, i.e., the page ceasing to change at all. Examples of this case include 9/11, when CNN's normal constant flow of articles halted while they were trying to sort out what was going on, and after Michael Jackson's death when entertainment news sites were rapidly posting every bit of information they received out of a concern that they would be perceived as disconnected from an unfolding event. As we will discuss in the next section, previous systems working on thresholds of change cannot distinguish between normal change and the abnormal activities associated with problematic behaviors or important but non-problematic events. Additionally, these systems cannot recognize abnormally static behavior. The essential problem is these systems are not comparing change to expectation but to a pre-defined threshold amount.

Thus, our problem is defined as trying to identify changes that may be considered as abnormal in terms of what a user would expect when revisiting the page. Note that this is essentially the visceral reaction one would have when rechecking a site as part of a curator's maintenance task. The goal is to allow the first step of maintenance–re-visitation– to be done automatically either at a frequency more often than or to more resources than a curator normally would be able to. In order to better illustrate the issues with existing systems and the need for our approach, we will now discuss prior change management systems.

3. PRIOR WORK

Management of collections has been an important topic in many research communities even before the advent of information systems. In the past, change has been viewed from a perspective similar to that of a traditional library – new documents arrive and old documents are lost or are worn out, but the existing documents remain unchanged [26]. This perspective of change only contemplates types of change whose magnitude and frequency are very low since changes in the document contents are not possible except as a consequence of wear, damage, or annotation. As a result, the manual maintenance of the collection is possible in traditional libraries, even large ones.

This perspective of change has carried over to digital libraries. This is a reasonable approach not only in the traditional library, but in curated digital libraries, since they both share a fairly tight control over their collections; sometimes even more so in the digital library where wear is not as great of an issue. However, when the contents of the members of a collection are loosely controlled, like a collection of web resources, this approach does not apply. The source of this incompatibility is the fact that many new types of change not only can occur, but often do. These changes are not only in membership to a category like in a curated collection, but can also be in what the individual documents contain or how they are presented. Additionally, the frequency of change can increase to unmanageable levels. Initially, the majority of responses from the community were to find ways to "deal with change" [26, 2, 15, 21, 6, 18]. These solutions essentially try to resist change. While more recently, the community has grown more aware that change is intrinsic to digital collections, more recent work still seeks to only deal with change, as illustrated by Askehave and Nielsen's work on digital genre where they saw change as a detrimental factor that could render their determination of characteristics of a genre invalid as time went on despite recognizing change as an intrinsic feature of web documents [3].

An early work to try an examine change on the web was the Do-I-Care Agent (DICA) [26]. DICA was primarily interested in finding new information for particular users. Changed documents were treated as if a new document had been entered in the user's collection. Later, Ashman described three classes of strategies for dealing with change on the web: preventative, corrective, and adaptive [2]. Preventative strategies are applied beforehand to avoid change from the beginning, such as caching. This makes the document static. Corrective strategies are applied after a problem and are designed to find replacements or equivalents. Adaptive methods are those that try to hedge the maintainer's bet by taking preventative actions that can then be used to form a corrective strategies. These classes still assume that change is a problem to be resolved and not the nature of the document. In 2001, the Walden's Paths project managed change by extending Johnson's approach for distance between page versions to determine how much a page had changed [15]. Further examples, can be seen in Koehler's change study [21], Boese and Howe's study that viewed change as a possible detriment to classification work [6], and Ivory and Megraw's study of Web Site Design Patterns [18]. In each of these cases the authors tried to eliminate or mitigate change in their collections. Their goal was either to find a underlying truth about the document without change getting in the way, or to restore the integrity of a collection *tainted by change*. While Askehave and Nielsen recognized change as an integral part of web documents, they admittedly did not fully account for their "view of web-mediated genres as dynamic documents" [3]. In fact, no change management systems have been built around the changing nature of web documents.

Results from the previously cited methods provide a two-point comparison, which in the past [6, 13, 14] was usually the extent of the comparison. However, when these comparisons continue over a large period of time, we found that patterns of magnitudes and frequencies of change emerge. These patterns provide a strong basis for identifying documents that may not be coherent with their original collection. After conducting a study of blogs from late 2006 to 2007 [7], we concluded that evidence of a commonality of change behavior across a given fine-grained genre, particularly blogs, indicates that associating a pattern of change to a genre enables pages to be grouped by observed change. This work was later confirmed by Saad and Gançarski's work on a collection of websites for French public television [4]. This better helps to determine an expectation of change. Despite the changing nature of web pages, existing genre-based approaches have treated them as a new class of *static* documents [6].

4. GROUND TRUTH BUILDING

In order to evaluate our methods, we created a corpus consisting of daily caches during the two month period surrounding the 2006 United States Congressional mid-term election. This period was picked both for its completeness, as there were no system or network outages, and since it was a politically charged time, it was expected to produce more interesting results in many of the blogs in the collection. A total of 62 days of 66 blogs were first checked for static instances–days where no addition occurred–this reduced the blogs to 22 interesting days. These 1452 caches were then processed with an HTML differencing engine to produce an easier to scan display of what had changed. Each of these caches was shown to three evaluators who selected from a variety of classifications of change to tag the newer version. The options we provided were: no change, change too slow, change too fast, topic shift, abnormal content, and normal change. Due to the size of the collection, evaluators were given as much time as they wanted to spend to tag as many as they could. While differences within a cache were shown consecutively in order of capture date, the order of sets of pages were randomly shuffled. Once an entry in the cache had been evaluated by three different judges, it was not shown to future judges. This corpus was then used to quantitatively compare different combinations of features, and analysis techniques.

5. FEATURE SELECTION

When revisiting a site, our surveyed social media users and interviewed graduate students indicated that their primary concerns when re-evaluating resources were in the areas of content, topic, target audience, and authorship. Because of these concerns we sought features that attempt to address these issues. For topic we included all 2-,3- grams in a document and the stemmed term vectors with stopwords removed. We used Porter's Snowball stemmer [24] for our stemming algorithm. For the target audience, we performed a Flesch-Kincaid readability index [20]. For authorship we included punctuation usage information and a structural analysis. Each of these features is used to create an ensemble feature vector that is then reduced using principal component analysis before being processed by our Kalman Filter.

Traditionally, feature selection is a fine-grained process that, due to the inherent computational complexity of calculating permutations, cannot be done exhaustively. Term vectors, in particular, have the potential to reach impractical running times even with small dictionaries. To avoid calculating every permutation, there are a variety of techniques to trim the decision space and test only options that are likely to provide an improvement. However, since DCM is built on coarse-grained feature extractors that each produce a number of features and because these features are reduced through principal components analysis, we chose to forgo the fine-grained analysis of each individual feature in preference of combinations of extractors. While better results may be found when selecting features individually, the results are more prone to overfitting [5] and don't help to provide insights into the usefulness of the coarse-grained extractors as a whole.

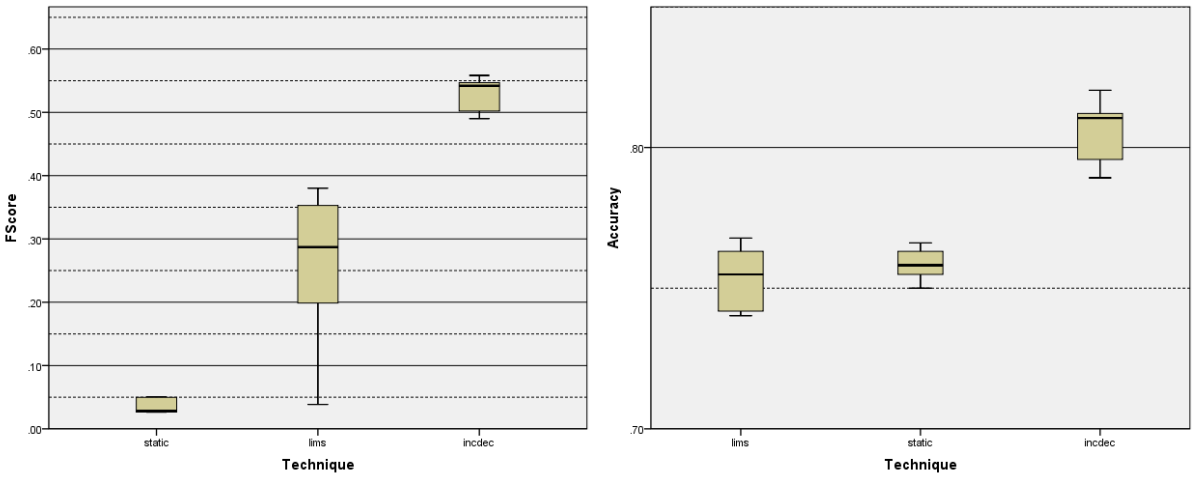
From these facts, we proceeded with a naive feature selection of all sixty-three combinations of the six features we currently have implemented. From the results of all feature combinations we performed an ANOVA on the F-Scores and the accuracies produced by each feature combination.

We determined that none of the features alone or any combination thereof provided a statistically significant improvement over any other. This is explained by the usage of principal component analysis performed by our analysis system before the Kalman filter is executed. Principal component analysis, by projecting the data into a lower dimensionality that maximizes the expression of the data's variance, would explain the wide variance we found for feature-wise analysis. This wide variance would result in a poor separation of feature-based classes thus resulting in needing to accept the null hypothesis (that the features are equivalent) in an ANOVA analysis.

6. ABNORMAL CHANGE DETECTION

The output of our Kalman Filter is a list of how far off each prediction made by the filter was in compared to the actual observed results. These error measurements when standardized and manually inspected seemed to indicate a correlation between the standard score and the likelihood that they were abnormal pages. To quantify this observation we devised a number of candidate methods to pick out the abnormal pages. These methods can be divided into three categories–heuristic methods, classifiers, and statistical outlier detectors. Each of these categories has several alternate methods while sharing a common perspective on how to treat this data. The theory behind most of these techniques is that users can provide feedback to help improve future attempts to determine if an error measurement is normal or abnormal. For the purpose of testing, we simulated the case where users always provide feedback. This is done by retraining the techniques by feeding the results of the ground truth study for each revision back into it after classification of a revision.

Due to the nature of results for machine learning type tasks, a single simple measure cannot capture the complexity of the data's results. For the remainder of this section we will present both the overall accuracy of the method and an F-Score, a commonly used technique that reports a bal-



(a) F-Score

(b) Accuracy

Figure 1: Results for Heuristic Techniques

anced combination of precision and recall. Some techniques perform very well in terms of F-Score and some in terms in Accuracy. A few techniques perform well in both measures, albeit not as well as the maximal techniques for each measure independently. Many measures perform poorly on both and are clearly inappropriate for the task at hand. In particular, we have found Spectral Regression Discriminant Analysis [10] and nearest neighbor techniques [1] to be the most interesting for deployment.

6.1 Heuristic Methods

The heuristic methods are often the simplest and are informed by our observations of the data. Currently, we have tested three different heuristic methods. A summary of these results can be seen in Figures 1b and 1a.

The first method and by far the simplest is a static error threshold placed at two standard deviations above and below the mean for past errors. If the new revision's prediction error is outside the range between the two thresholds then it is marked as abnormal, if it is between then it is normal. To provide an example of how the static technique behaves, please see Figure 2.

The purpose of a static error threshold was to provide a baseline to compare other heuristic techniques to,f as the static threshold is the most naive means of analysis–one that does not change in response to feedback. The static technique provided one of the best of overall accuracies (75.86%) at the cost of one of the worst overall F-Scores (.035). This reflects the overall dominance of negative (normal behavior) results in our collection. The static method tended to err on the side of more negative items, thus having a good accuracy (mainly by chance) with such a poor F-Score.

The second method increases the complexity by allowing the bounds to grow and shrink using constant steps from a starting point (two standard deviations) in response to false positives and false negatives. In a production system this information would be obtained from user feedback. Thus in the case of no feedback it will perform like the previous method, while with perfect feedback it will perform as well as the algorithm can be expected to perform. One thing to note is the threshold above and below the mean change independently of each other. To provide an example of the results from the constant step technique, please see Figure 3.

Allowing constant steps has the effect of dampening the effect of highly abnormal results–a revision with error far outside other revisions, or an abnormal cache near the mean. This can be viewed as a conservative heuristic. Surprisingly, this conservative method obtained the best over all accuracy (80.67%) and a slightly better than a random classifier would be expected to perform on F-Score (.531). Unfortunately in cases where the misclassified cache is distant from the threshold, repeated distant caches may still be misclassified.

The third method takes a potentially more aggressive approach, that again could be informed by feedback. In this case, instead of steps in set increments, we expand or contract the limits so that the most recent false positive is just inside the limits and a false negative is just outside the limits. This ensures that a similar value with a similar classification in subsequent caches is not misclassified, however, if a misclassified cache has an error contained in normally correctly classified caches, the technique would then misclassify results it would otherwise classify correctly. To provide an example of how the limits technique behaves, please see Figure 4.

While the limits technique still maintained a decent accuracy at 75.30%, this accuracy is worse than the static method. The results for F-Score did improve over static (.255) but not as well as either the expected random results or the constant step method.

6.2 Classifier Methods

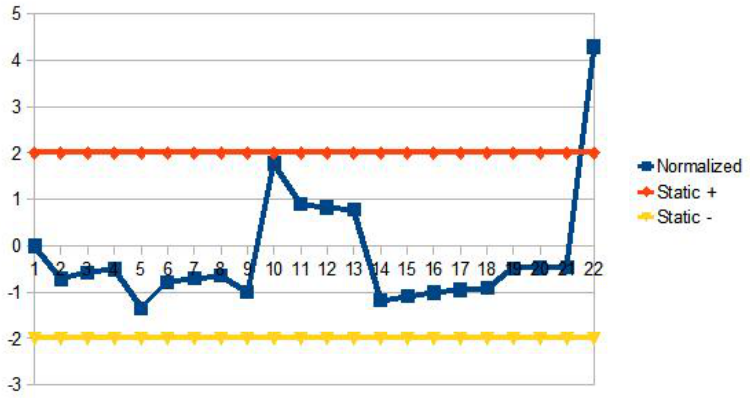


Figure 2: Static Technique on Outer Court Blog.

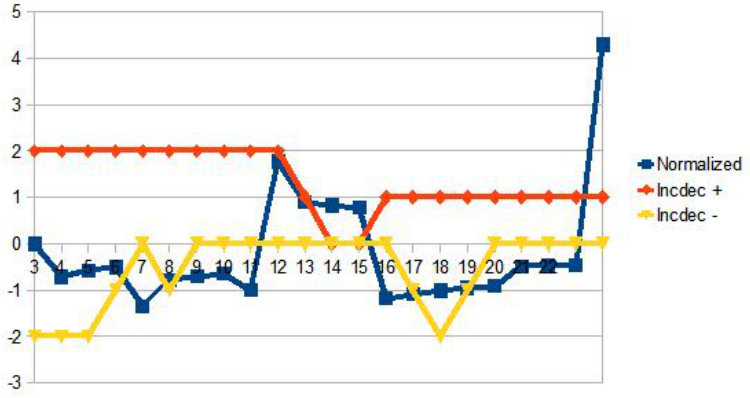


Figure 3: Constant Step Technique on Outer Court Blog.

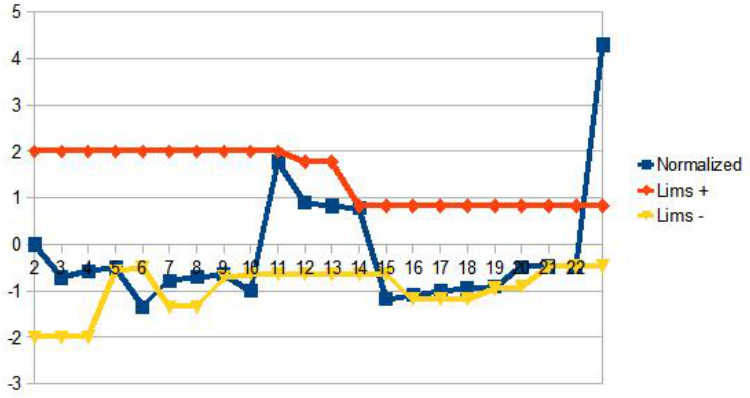
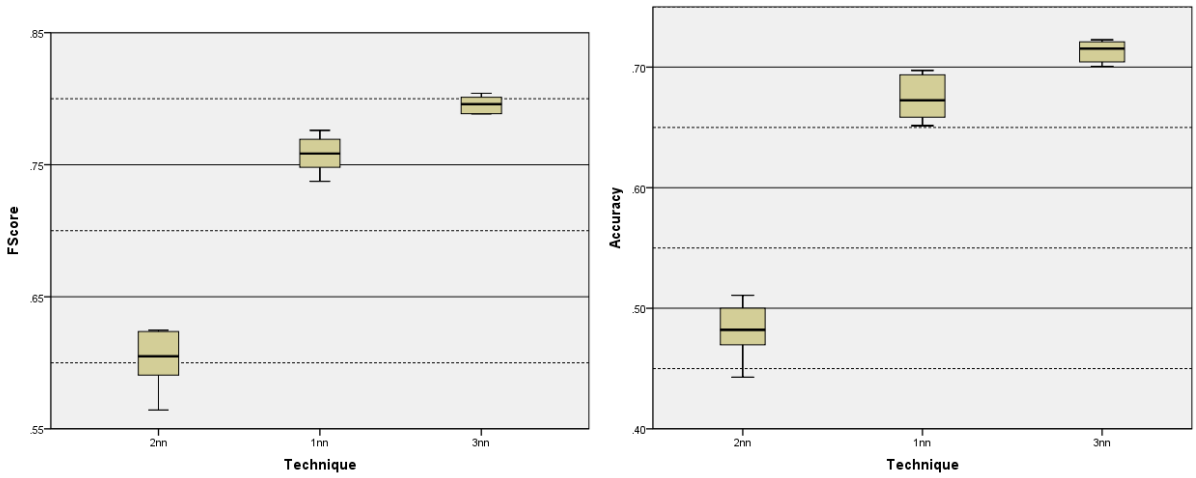


Figure 4: Limits Technique on Outer Court Blog.



(a) F-Score

(b) Accuracy

Figure 5: Results for kNN Techniques

Essentially, the problem of separating normal caches and abnormal caches using the error of prediction from a Kalman Filter is a binary classification task. As such, there are a wide variety of techniques that can be used. One difference from a traditional classification task where data is split into testing and training sets, data from an online predictive method should not be simply split. Every sample becomes a new piece of testing data using the prior data as a training set. After the new point is classified and a user provides feedback on the classification, the point can be folded into the training set to generate a new classifier. We tried a variety of supervised classifier methods which produced a variety of results. Some of these results provide a trade-off of overall accuracy with increased precision and recall when compared to the heuristic methods. The classifiers utilized were variations of three categories of binary classifiers–nearest-neighbor methods [1], support vector machine methods [12], and discriminant analysis methods [23].

Nearest-neighbor methods work by the assumption that the k nearest known samples to the test sample are likely to be of the same classification as the test sample [1]. In the case of mixed results, a majority vote of the neighbors is used to select the class. For the purposes of classifying errors, we tried 1-, 2-, and 3- nearest-neighbors variants. A summary of kNN results can be seen in Figures 5a and 5b.

While for accuracy all methods perform worse than the static technique, two methods still perform better than random chance. 3NN is the most accurate (71.35%), followed by 1NN (67.49%), and finally 2NN is worse than random at 48.19%. Conversely, by F-Score, all methods perform better than random and the static technique for all cases. 2NN is again the worse method (.602), followed again by 1NN (.758), and finally 3NN performed the best at (.796). While the constant steps technique is still the most accurate technique, all three techniques improved on the F-Score from the heuristic methods.

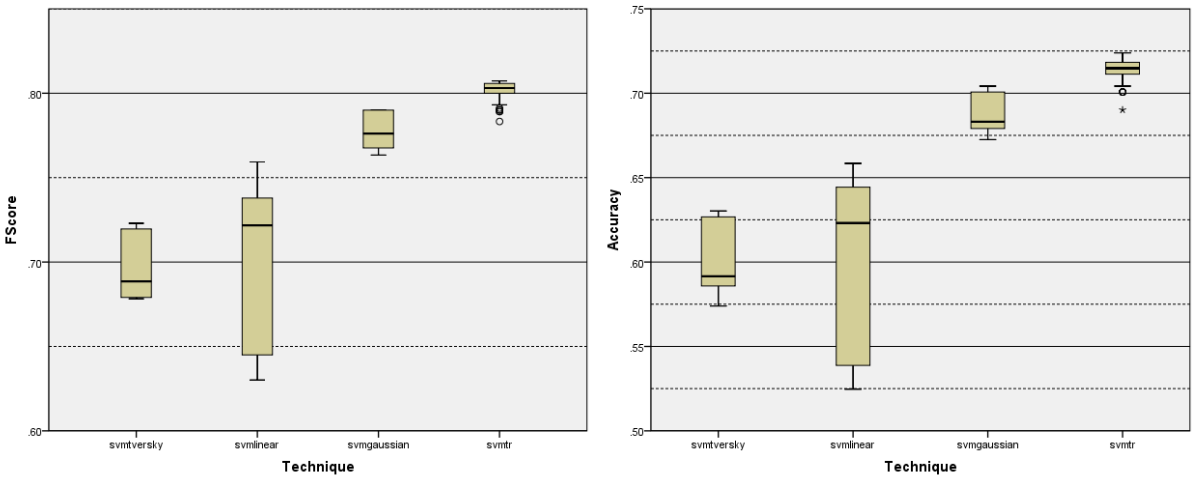
Support vector machines are a broad set of techniques that can be used to find a hyperplane that maximally separates two classes of data by projecting the data in to a higher dimensionality [12]. The shape of the hyperplane is described by a kernel which can be easily replaced to provide separators of different complexities. For our tests, we tried four kernels – linear, Gaussian, terminated ramp, and the Tversky kernel. A summary of SVM results can be seen in Figures 6a and 6b.

The linear kernel is the simplest kernel in that it uses a linear hyperplane to separate the two classes [12]. While it did perform better than random for accuracy, with an accuracy of 60.56%, the linear kernel did not out-perform a static measure. While the linear kernel does provide a better F-Score than heuristic methods and 2NN at .705, 1NN and 3NN still have better F-Scores.

The next kernel in order of complexity uses a Gaussian surface to separate the classes [12]. While Gaussian does provide an improvement for both accuracy (68.76%) and F-Score (.777) over the Linear kernel, it did not perform statistically better than 1NN, nor worse than 3NN

Terminated ramp uses a series of linear models to improve on linear separation for oppositely labelled points that lay close to the global separating hyperplane [22]. It does this by refining the seperation with local optimal linear hyperplanes between the opposing points. Terminated ramp, while proving a much better than random accuracy (71.36%), still is less accurate than heuristic techniques. However, the F-Score for terminated ramp is the best we obtained for any method (.802). Yet, it was not statistically better than 3NN and the Gaussian kernel.

The final kernel method we used is unique from the others in that it is a non-geometric asymmetrical technique devised by Amos Tversky to match his observations of human behavior when classifying objects [28]. Despite the method being designed to better model how humans perform classification tasks, Tversky's method was the worst perform-



(a) F-Score

(b) Accuracy

Figure 6: Results for SVM Techniques.

ing SVM method for both accuracy (69.49%) and F-Score (.599), albeit still better than random selection and 2NN for both measures in addition to having an F-Score better than all of the heuristic methods. This is most likely due to Tversky's dependence on feature vectors with different keys. However, the preprocessing steps taken before the application of the Kalman Filter eliminate keys with a low document frequency thus homogenizing the key set.

Discriminant analysis is similar to support vector machines in that the goal is to separate two classes of data. However, discriminant analysis reduces the dimensionality of data to a single dimension with a maximal separation between the class means. While SVM methods have been shown to be more powerful, adaptable, and generally usable for binary classification tasks, they incur a complexity penalty that makes their advantages only preferable when discriminant analysis cannot provide sufficiently accurate results. For this reason we decided to evaluate the use of discriminant analysis techniques for the classification of errors. The versions of discriminant analysis we tested were Fisher Discriminant Analysis [23], Spectral Regression Discriminant Analysis [10], Penalized Discriminant Analysis [16], and Diagonal Linear Discriminant Analysis [25]. Each version uses a different discriminant function to find the maximized separator between classes. A summary of discriminant analysis results can be seen in Figures 7a and 7b.

Fisher's Discriminant (FDA) measures the ratio of the variance between classes to the variance within classes [23]. The results in terms of accuracy for FDA were better than random chance (51.60%), but not as good as heuristic methods, SVM-based methods, 1NN, or 3NN.

The second discriminant, Spectral Regression Discriminant Analysis (SRDA) [10] takes advantage of spectral analysis to reduce the complexity of the calculation of the discriminant in addition to improving overall accuracy. SRDA proved to be the best discriminant analysis technique with an accuracy of 67.41% and an F-Score of .777. While for accuracy, several methods, most notably the constant step heuristic and the static heuristic out-perform SRDA. The F-Score was statistically equivalent to 3NN and terminated ramp, the best overall methods.

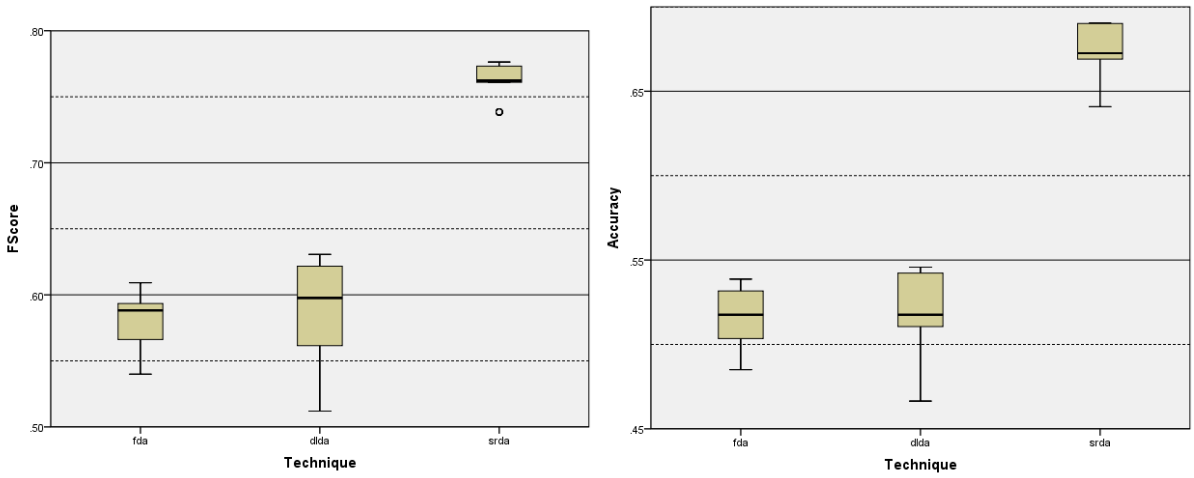
The Penalized Discriminant (PDA) [16], is a modification in Fisher's that regularizes covariance matrices to overcome deficiencies when the covariance matrix is degenerate. This can be used to overcome overfitting in large numbers of highly correlated variables or to prevent underfitting for non-linear or complex class separations. Unfortunately, PDA not only proved to be the absolute worst method, but it managed to get 0% accuracy and a 0 F-Score. Most likely this is caused by having a covariance matrix that is not degenerate.

The Diagonal Linear Discriminant Analysis (DLDA) [25], is a simplified case of Fisher's where the covariance matrix is found to be diagonal. This allows assumptions to be made that enable a less complex algorithm to be used to find a discriminant. While DLDA proved to be more accurate (51.63%) and have a better F-Score (.588) than FDA, it was not a statistically significant improvement.

6.3 Statistical Outlier Detection

In previous sections we viewed the problem in terms of, first, threshold tuning, and, second, as binary classification, in this section, instead, we cast the problem as detecting statistical outliers. Outlier detection is an important area of study for mathematical statistics. Undetected outliers can skew data and in some cases can help statisticians identify a multimodal distribution in an assumed unimodal one.

To address this, statisticians have sought algorithms and heuristics to identify outliers for centuries. Each method has its own drawbacks and caveats. Usually these warnings are concerning the results of overly liberal data point elimination. In our case since we are not aiming to eliminate these points, but to simply identify them for human inspection, we can proceed without fear of negative repercussions. For this class, we tested two outlier detection techniques–



(a) F-Score

(b) Accuracy

Figure 7: Results for Discriminant Analysis

Chauvenet's criterion and Grubbs' test. A summary of outlier detection results can be seen in Figures 8a and 8b.

One of the earliest techniques created was Chauvenet's criterion [11] for detecting outliers. This method rejects values where the probability of normalized data is multiplied by the number of data points. This value is rejected as an outlier if the result is less than 0.5. The results of our application of Chauvenet's criterion produced results that were not significantly different than the static method for both accuracy (75.75%) and F-Score (.578).

Grubbs' test [17] is a more recent method for outlier detection that relies on a variation of the t-statistic compared to the absolute value of the standardized minimum and maximum scores in a set and runs iteratively until neither maximum nor minimum is greater than the critical value. While Grubbs' performed the second best by accuracy (76.63%), it had the second worse F-Score (.010) which was not significantly any better than PDA. Additionally, Grubbs' accuracy was not significantly any better than the static heuristic.

6.4 Ensemble Method

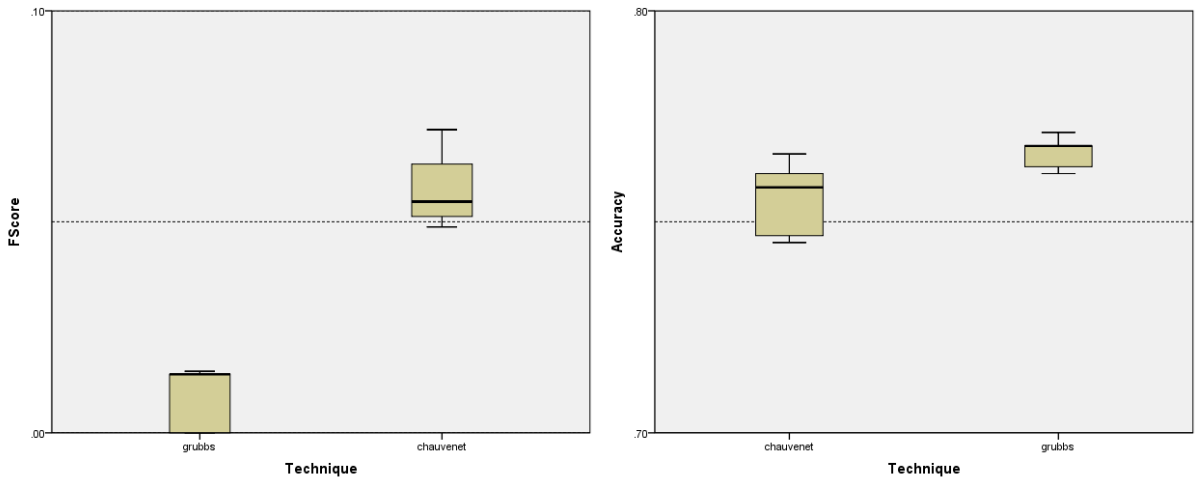
Finally we selected the methods with the best accuracy and F-scores and used them as votes for each cache where a majority classification ruled. The methods we used for our votes were 3NN, SVM with the Terminated Ramp kernel, SVM with the Gaussian Kernel, the constant step heuristic and Grubbs' outlier detection method. Unfortunately we were not able to improve performance and only obtained a 75.32% accuracy and an F-Score of .229. We suspect that a deeper analysis of results and a weighted ensemble method could yield better results.

6.5 Versus Cosine Similarity

From the combined results of feature selection and technique selection, we identified two pairs of feature set and technique that can be objectively called the best for each of our vital methods of importance–accuracy and F-Score. For optimal accuracy we selected the static step technique using the structural change algorithm with an accuracy of 82.02%. While many other combinations of features performed equivalently to this, this single feature represented the simplest of the best accuracy techniques. For F-Score, we picked the SVM technique using the terminated ramp kernel and a feature set consisting of structural changes and punctuation counts. This combination achieved and F-Score of .807. However, one portion of our analysis has not been objectively measured – the effectiveness of the Kalman filter technique over traditional threshold techniques. In order to address this issue, we performed cosine similarity tests both between the first and each subsequent revision and between each pair of adjacent revisions. These results are then evaluated at a variety of threshold levels to determine if the Kalman technique is better or worse than traditional techniques for every combination of feature set. Similarity is defined depending on the dimensionality of the data. If the data is one-dimensional, we use a linear distance between revisions, if it is multi-dimensional, we use the standard cosine similarity metric defined for term vectors.

Given the usage of of both absolute and relative measures to calculate threshold, the first question is–are these different measures different and if so which one is better? We performed an ANOVA with a post-hoc Tukey's Test [27] to determine if any pair of absolute and relative measures for each threshold level were significantly different. We found no statistically significant difference between measures at any threshold level for accuracy. For the F-Scores, we found that only at the 0.1 and 0.0 threshold levels was there a significant difference between measures. In both cases the absolute measure out-performed the relative one.

We did find that different threshold levels provided an advantage over others. For accuracy, the level for the relative measure provided the best results (71.77%), while it was not statistically better than those obtained from levels at or above 0.5, it did out perform all of the levels below 0.5. For F-Score, however, the results were the opposite with the ab-



(a) F-Score (b) Accuracy

Figure 8: Results for Outlier Detection

solute method at a threshold level of 0.0 obtaining a better F-Score (.307) than all other threshold levels.

The final step in our analysis is the comparison between threshold methods and our Kalman filter based method. As stated before, we selected the best results for each measure–SVM with a truncated ramp kernel for F-Score and the constant step heuristic for accuracy–as a point of comparison to threshold methods. In both cases, the optimal Kalman-filter based methods out-performed all of the threshold options. Just as with the comparisons between threshold options, we again used ANOVA with a post-hoc Tukey's Test to make the comparison. For accuracy, the constant step heuristic statistically out-performed the best threshold method– with a relative measure ($p = .001$). For F-Score, the SVM method with the terminated ramp kernel statistically out-performed the best threshold method–0.0 with the absolute measure ($p = .000$). What was more surprising was that while the SVM method with the terminated ramp kernel was statistically equivalent on accuracy to the most accurate threshold method ($p = 1.000$), the constant step heuristic did statistically out perform all of the threshold tests for F-Score also ($p = .000$).

7. CONCLUSIONS

From these results we are able to quantatively select the best options for classification of blog changes for accuracy and F-Score. For accuracy our constant step heuristic using structual change features alone provided the best accuracy (82.02%) over all other Kalman filter based techniques and statistically out-performed all of the threshold based techniques. For F-Score, our SVM with the truncated ramp kernel using both structural change and punctuation count features provided the best F-Score (.807). This technique also statistically out-performed all threshold methods.

This shows that the Kalman-based technique is capable of outperforming threshold based techniques for finding unexpected change in blogs. For practical considerations, while accuracy is theoretically interesting, since the collection is dominated by expected changes, it is not of high concern for designing systems to help collection managers. Instead the manager is more concerned with the precision and recall of the unexpected changes. Thus, F-Score is the preferred measure to optimize for. Therefore we can suggest the usage of SVM with Terminated Ramp as the preferred technique.

While our work has shown good results with the blog collection, there are four areas in which further inquiry can be done. First, our feature set could be expanded upon to include more sophisticated techniques, perhaps from topic tracking and sentiment analysis work. Second, the substitution of other filters, such as particle filters, instead of the Kalman filter may provide better results. Third, while we conducted classification of Kalman filter results with a broad set of techniques, there are still more classification techniques that we could experiment with, such as self-organizing maps and fuzzy clustering techniques. And fourth, moving to a more general ground-truth set would provide us with the ability to compare results based on genre and potentially infer generalized strategies.

In conclusion, blogs are a type of resource that exhibits relatively frequent change. The results with blogs may apply to other resources that exhibit patterns of change–news sites, RSS feeds, pages with recent events / news, and project / department web pages. Future work should examine application to other genres.

8. ACKNOWLEDGEMENTS

The Distributed Collection Manager is being developed as part of the Ensemble Computing Pathways Project. This work has been funded in part by the National Science Foundation under grants DUE-0840715 and DUE-1044212.

This document was prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285; managed by UT-Battelle, LLC, for the US Department of Energy under contract number DE-AC05-00OR22725.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

9. REFERENCES

[1] Cover, T. and Hart, P. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1), 1967.

[2] Ashman, H. Electronic Document Addressing: Dealing with Change. *ACM Comp. Surv.*, 32(3), 2000.

[3] Askehave, I and Nielsen, A. E. What are the Characteristics of Digital Genres? - Genre Theory from a Multi-Modal Perspective. In *Proc. of HICSS 2005*, 2005.

[4] M. Ben Saad and S. Gançarski. Archiving the web using page changes patterns: a case study. In *Proc. of 11th JCDL*, JCDL '11, pages 113–122, 2011.

[5] Bishop, C. M. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.

[6] Boese, E. S. and Howe, A. E. Effects of Web Document Evolution on Genre Classification. In *Proc. of 14th ACM Int. Conf. on Information and Knowledge Management*, 2005.

[7] Bogen, P. L., Francisco-Revilla, L., Furuta, R., Hubbard, T., Karadkar, U. P. and Shipman, F. Longitudinal Study of Changes in Blogs. In *Proc. of JCDL 2007*.

[8] Bogen, P. L., Johnston, J., Karadkar, U. P., Furuta, R. and Shipman, F. Application of Kalman Filters to Identify Unexpected Change in Blogs. In *Proc. of JCDL 2008*.

[9] Bogen, P. L., Shipman, F. M. and Furuta, R. An analysis of personal collections among users of social media. In *Proc. of JCDL 2011*.

[10] Cai, D. and He, X. and Han, J. SRDA: An Efficient Algorithm for Large-Scale Discriminant Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 20, 2008.

[11] Chavuenet, W. *A Manual of Spherical and Practical Astronomy*. Lippincott, 1871.

[12] Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines: and Other Kernel-Based Learning Methods*. Cambridge UP, 2000.

[13] Dalal, Z. and Dash, S. and Dave, P. and Francisco-Revilla, L. and Furuta, R. and Karadkar, U. and Shipman, F. Managing Distributed Collections: Evaluating Web Page Changes, Movement, and Replacement. In *Proc. of 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2004.

[14] Francisco-Revilla, L., Shipman, F., Furuta, R., Karadkar, U., Arora, A. Managing Change on the Web. In *Proc. of JCDL 2001*.

[15] Furuta, R. and Shipman, F. M. and Marshall, C. C. and Brenner, D. and Hsieh, H.-W. Hypertext Paths and the World-Wide Web: Experiences with Walden's Paths. In *Proc. of 8th ACM Conference on Hypertext*, 1997.

[16] Ghosh, D. Penalized Discriminant Methods for the Classification of Tumors from Gene Expression Data. *Biometrics*, 59(4), 2003.

[17] Grubbs, F. E. Procedures for Detecting Outlying Observations in Samples. *Technometrics*, 11(1), 1969.

[18] Ivory, M. Y. and Megraw, R. Evolution of Web Site Design Patterns. *ACM Transactions on Information Systems*, 23(4), 2005.

[19] Jones, W. and Bruce, H. and Dumais, S. Keeping Found Things Found on the Web. In *Proc. of 10th Int. Conf. on Information and Knowledge Management*, 2001.

[20] Kincaid, J. and Fishburne, R. and Rodgers, R. and Chissom, B. Derivation of New Readability Formulas for Navy Enlisted Personnel. Technical Report 8-75, National Technical Information Service, 1975.

[21] Koehler, W. Web Page Change and Persistence—a Four-Year Longitudinal Study. *Journal of the American Society for Information Science and Technology*, 53(2), 2002.

[22] Merler, S. and Jurman, G. Terminated Ramp-Support Vector Machines: A Nonparametric Data Dependent Kernel. *Neural Networks*, 19, December 2006.

[23] Mika, S. and Smola, A. J. and Schölkopf, B. An Improved Training Algorithm for Kernel Fisher Discriminants. In *Proc. of Eighth International Workshop on Artificial Intelligence and Statistics*, volume 2001, 2001.

[24] Porter, M. Snowball Stemmers and Resources Page. On line http://www.snowball.tartarus.org. [Visited 8/22/2011].

[25] Simon, R. Supervised Analysis When the Number of Candidate Features (p) Greatly Exceeds the Number of Cases (n). *ACM SIGKDD Explorations Newsletter*, 5, December 2003.

[26] Starr, B. and Ackerman, M. S. and Pazzani, M. Do-I-Care: A Collaborative Web Agent. In *Conference Companion on Human Factors in Computing Systems*, 1996.

[27] Tukey, J. W. *Exploratory Data Analysis*, volume xvi of *Behavioral Science*. Addison-Wesley, 1977.

[28] Tversky, A. Features of Similarity. *Psychological Review*, 84(4), 1977.