LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-235295

Improving Naive Bayes with Online Feature Selection for Quick Adaptation to Evolving Feature Usefulness

R. K. Pon, A. F. Cardenas, D. J. Buttler

October 5, 2007

SIAM Conference on Data Mining 2008
Atlanta, GA, United States
April 24, 2008 through April 26, 2008

Updated October 14, 2003

Improving Naive Bayes with Online Feature Selection for Quick Adaptation to Evolving Feature Usefulness *

Raymond K. Pon† Alfonso F. Cardenas† David J. Buttler‡

Abstract

The definition of what makes an article interesting varies from user to user and continually evolves even for a single user. As a result, for news recommendation systems, useless document features can not be determined *a priori* and all features are usually considered for interestingness classification. Consequently, the presence of currently useless features degrades classification performance [1], particularly over the initial set of news articles being classified. The initial set of document is critical for a user when considering which particular news recommendation system to adopt. To address these problems, we introduce an improved version of the naive Bayes classifier with online feature selection. We use correlation to determine the utility of each feature and take advantage of the conditional independence assumption used by naive Bayes for online feature selection and classification. The augmented naive Bayes classifier performs 28% better than the traditional naive Bayes classifier in recommending news articles from the Yahoo! RSS feeds.

1 Introduction

An explosive growth of online news has taken place in the last few years. Users are inundated with thousands of news articles, only some of which are interesting. A system to filter out uninteresting articles would aid users that need to read and analyze many news articles daily, such as financial analysts, government officials, and news reporters. In [2], iScore is introduced to address how interesting articles can be identified in a continuous stream of news articles by using a variety of interestingness-related features. Instead of applying the traditional approach for news filtering, which is to learn keywords of interest for a user [3, 4, 5], iScore tries to identify the multitude of characteristics that make an article interesting for a specific user. A variety of features are extracted from each article, ranging from topic relevancy to source reputation. The combination of multiple features yields higher quality results for identifying interesting articles for different users than traditional methods.

However, the definition of interestingness varies from user to user. For example, the writing style of an article may be important for one user; whereas, for another user it may be unimportant. As a result, it is not possible to predict which features are important for a specific user before constructing the system and so all features are included for classification. As a result, classification performance suffers initially and requires a significant amount of training to adapt to the presence of useless features. iScore in [2] suffers from this problem. And the definition of interestingness may even change for a single user over time. For example, the writing style of an article may not be important initially but may evolve to be becoming important later on. The traditional classifiers used by iScore, such as naive Bayes, can learn to adapt to the changing utility of features, but only with sufficient training. And because of the required large initial training period, the usefulness of the recommendation system suffers. Users of recommendation systems are less inclined to use a system if it requires a significant amount of training before it begins to give accurate recommendations.

To address these problems, we introduce online feature selection for naive Bayes. We use correlation to determine the utility of each feature and take advantage of the conditional independence assumption used by naive Bayes for online feature selection and classification. We make the following contributions: (1) Augmenting naive Bayes with online feature selection allows for the fast identification of useless features, significantly improving iScore's initial performance; (2) The continual learning of statistics about each feature allows for the invocation of any feature at any time if it has been determined to be useful, addressing the problem of the evolving definition of interestingness; (3) By only considering the top-k useful features, evaluation of all possible subsets of features is avoided, making our feature selection approach efficient.

*This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48. (UCRL-CONF-235295)

†University of California, Los Angeles

‡Lawrence Livermore National Laboratory

2 Related Works

2.1 News Recommendation and Online Filtering. iScore is a recommendation system in a limited user environment, so the only available information is the article's content and its metadata, disallowing the use of collaborative filtering for article recommendation. Several works use this information in a variety of ways. Some systems perform clustering or classification based on the article's content, computing such values as TF-IDF weights for tokens [5, 6]. We implement a variation of these methods as feature extractors in iScore. Work by [7] ranks news articles and new sources based on several properties, such as mutual reinforcement and freshness, in an online method. In contrast, we rank articles using user feedback along with a set of features that address the properties discussed in [7] among others. Also, [7] does not address the problem of personalized news filtering, but rather the identification of interesting articles for the general public. Another approach taken by [8] measures the interestingness of an article as the correlation between the article's content and the events that occur after the article's publication. Unfortunately, in most cases, these indicators are domain specific and are difficult to collect in advance for the online processing of articles.

Our work in iScore is closely related to the adaptive filtering task in TREC, which is the online identification of news articles that are most relevant to a set of topics. The task is different from identifying interesting articles for a user because an article that is relevant to a topic may not necessarily be interesting. The report by [9] summarizes the results of the last run of the TREC filtering task. Like much of the work in the task, we use adaptive thresholds and incremental profile updates.

2.2 Feature Selection. There has been a significant amount of work done in offline feature selection. The study by [10] surveys a variety of feature selection techniques, noting cases where feature selection would improve the results of classifiers. They show that noise reduction and better class separation may be obtained by adding features that are presumably redundant. Features that are independently and identically distributed are not truly redundant. Perfectly correlated features are truly redundant in the sense that no additional information is gained by adding them. However, very high feature correlation does not mean the absence of feature complementarity. A feature that is completely useless by itself can provide a significant performance improvement when taken with others. In other words,two features that are useless by themselves can be useful together.

There are three approaches to feature selection: wrappers, filters, and embedded methods. Wrappers use the learning machine of interest as a black box to score subsets of features according to their predictive power. An example of a wrapper approach is [11], which uses a hill-climbing approach to find a good set of features. Filters select subsets of features as a pre-preprocessing step, independently of the chosen predictor. Embedded methods perform feature selection during the process of training and are usually specific to given learning machines.

There has been some work done in embedding feature selection within classification algorithms, but they can not be applied directly to the features used within iScore. The work by [12] discusses feature-weighting methods such as Winnow [13]. However, the inputs and outputs of the Winnow algorithm are all binary and can not be applied directly to continuous inputs, such as the feature scores generated by iScore's feature extractors. Other Winnow variants and Winnow-based online feature selection techniques studied by [14] require that all inputs are weights of importance and must be values between 0 and 1, such as normalized term frequencies. However, in general, features may not necessarily be positive weights or even have the same semantic meaning. In the case of iScore, a feature's correlation to interestingness may be positively correlated; whereas, another feature maybe negatively correlated. Work by [15] helps address this problem with an incremental decision tree algorithm that makes use of an efficient tree restructuring algorithm. However, the drawback is that any numeric data must be stored and maintained in sorted order by value and the decision tree's storage requirements will continually grow.

Other work in online feature selection addresses a different problem. In [16], techniques are studied for selecting features from a set of features that grow over time. Instead of a fixed set of features and a growing number of training instances to work from, the set of features continues to grow as the number of training instances remains fixed. However, in the iScore framework, the set of features with varying degrees of utility is fixed while the number of training instances continues to grow.

Another method for feature selection is to reduce the number of redundant features, which is different from our goal of reducing the number of irrelevant features. In [17], redundant features are identified by performing pair-wise similarities measurements using the properties of time series data, which may not be directly applied to news articles. In our experiments, we assume a more general setup, where documents from different news sources that span multiple domains are aggregated together into a single document stream and are simply

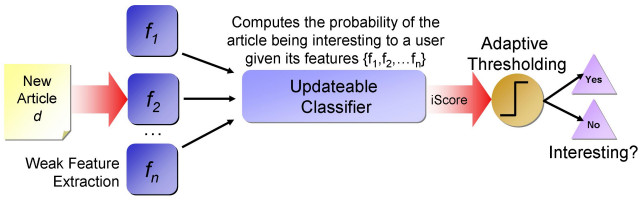

Figure 1: iScore architecture.

ordered by publication time. Consequently, an article in the document stream is not necessarily dependent upon the content of the article that immediately precedes it in the document stream.

Because the importance of features for what makes an article interesting varies among users, are unknown *a priori*, and may change over time, no features can be discarded when constructing the overall classifier. The usefulness of each feature must be learned in an online fashion. And current online feature selection approaches are not general or efficient enough to handle the general features used by iScore.

3 iScore Architecture

In iScore, news articles are processed in a streaming fashion, much like the document processing done in the TREC adaptive filter task [9]. Articles are introduced to the system in chronological order of their publication time. Once the system classifies an article, an interestingness judgment is made available to the system by the user. The article classification pipeline consists of four phases, shown in Figure 1. In the first phase, for an article d, a set of feature extractors generate a set of feature scores $F(d) = f_1(d), f_2(d), \ldots, f_n(d)$. Several topic relevancy features, uniqueness measurements and other features, such as source reputation, freshness, subjectivity, and polarity of news articles are discussed and implemented in [2]. The feature values are continuous real numbers. Then a classifier C generates an overall classification score, or an iScore $I(d)$:

$$I(d) = C(f_1(d), f_2(d), \ldots, f_n(d)) \quad (1)$$

The study of classifiers in [2] show that a naive Bayes classifier can identify interesting articles well.

Following the generation of an iScore, an the adaptive thresholder thresholds the iScore to generate a binary classification, indicating the interestingness of the article to the user. The adaptive thresholder tries to find the optimal threshold that yields the best metric result, such as $f_{0.5}$-measure. In the final phase, the user examines the article and provides his own binary classification of interestingness (i.e., tagging) $I'(d)$. This feedback is used to update the feature extractors, the classifier, and the thresholder. The process continues similarly for the next document in the pipeline. In this study, we focus on the overall classifier, comparing a naive Bayes classifier against an augmented naive Bayes classifier with online feature selection.

4 Correlation

The usefulness of features for determining the interestingness of articles are evaluated in [2]. The features are evaluated using a collection of 35,256 news articles from all the Yahoo! News RSS feeds [18], collected between June and August 2006. The classification task is to identify which articles come from which RSS feed. RSS feeds considered for labeling are feeds of the form: "Top Stories *category*," "Most Viewed *category*," "Most Emailed *category*," and "Most Highly Rated *category*." Because user evaluation is difficult to collect and such data is often sparse, the Yahoo! news articles and their source feeds are used for their resemblance to user labeled articles. For example, RSS feeds such as "Most Viewed Technology" is a good proxy of what the most interesting articles are for technologists. Other categories, such as Top Stories Politics, are a collection of news stories that the Yahoo! political news editors deem to be of interest to their audience, so the feed also would serve well as a proxy for interestingness.

Figure 2 shows the Pearsons correlation of the features (from [2]) with interestingness in each of the RSS feeds. For most feeds, the topic relevancy and source reputation features are significantly directly correlated with interestingness. Other features, such as writing style, speech events, anomaly detection, and subjectivity have varying correlation magnitudes and directions with interestingness, depending on the RSS feed. A variety of criterion that users may use when evaluating the interestingness of an article are shown.

Correlation is not necessarily the best metric for measuring the utility of a feature in document classification since the actual usefulness of a feature can not be determined by studying a single feature in isolation. There are certainly cases where two features that are useless by themselves can be useful when combined together [10]. However, correlation is a useful guide if the features were designed to be directly or indirectly correlated with interestingness in mind, as they were for the iScore features. And by coupling this independent correlation metric with a classifier that assumes that each feature is independent, such as naive Bayes, performance of the classifier should improve. In [10], information gain and correlation are suggested for feature ranking. Information gain is difficult to compute in an online fashion because the appropriate discretization is difficult to determine if the entire data is not available

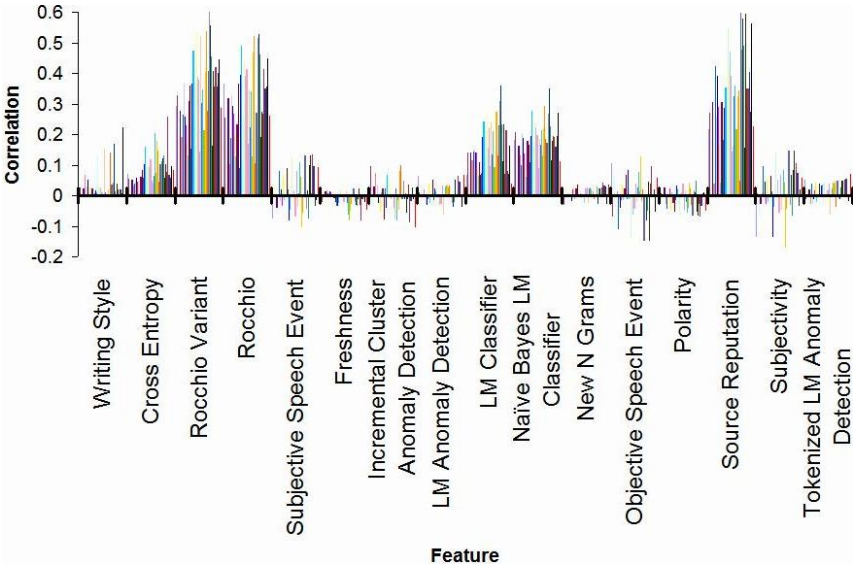

Figure 2: Feature correlation with interestingness. Each color represents a different proxy user/RSS feed.

during evaluation (as in an online streaming environment). Consequently, we use correlation instead due to its simple online computability and its lack of a need for discretization.

5 Online Feature Selection with Naive Bayes

Based on Bayes' theorem, a naive Bayes classifier is a simple and fast probabilistic classifier that assumes that features are conditionally independent [19]. In the context of classifying articles, the probability of an article being interesting is defined by a naive Bayes classifier as:

$$p(Int|f_1,\ldots,f_n) = \frac{1}{Z}p(Int)\prod_{i=1}^{n} p(f_i|Int) \tag{2}$$

where Z is a scaling factor dependent on $f_1,\ldots,f_n$, and Int is the interesting article class. The probability $p(f_i|Int)$ is estimated using kernel estimators [20]. During classification, when a feature is unavailable, it is simply ignored, which is equivalent to marginalizing over them.

Ideally, we would like to classify an article using only the most useful features for a specific user. Thus, given a set of n features, the features are ordered by their current absolute Pearsons correlation to interestingness. We then take the top-k most highly correlated features for classification, where $k = 1,\ldots,n$. Thus, for every document, we generate n classification scores (each referred to as a subset score); one score for each subset. The overall score is the subset score associated with the subset of features with the highest $f_{0.5}$-measure statistic. Because of the conditional independence of the features, we only need to maintain a single set of statistics (in the form of kernel estimators) related to $p(f_i|Int)$ and $p(f_i)$ even though we are generating n classification scores for each document. For a subset of features of size less than n, features not in the subset are essentially ignored when generating a classification score from the naive Bayes classifier.

After a document is classified, the classifier's kernel estimators for each feature are updated given the actual interestingness of the article. Also, the $f_{0.5}$-measure statistic for each feature subset considered is updated as well as the correlation with interestingness for each feature.

Because statistics about each feature are continually maintained, a feature that was deemed useless early on can be invoked for classification later. This allows for an evolving definition of interestingness for a specific user. Although irrelevant features are ignored for the overall document classification, statistics learned about the features are never forgotten.

Since only subsets of features with the highest correlations are considered for each document, as opposed to all possible subsets, our feature selection solution is tractable. Sets consisting of only features with low correlation with interestingness would be expected to be very low performing for document classification; whereas, sets of features with high correlation would be expected to be higher performing. Because we consider only the top-k most highly correlated features, subsets consisting of only lowly correlated features are never considered. And from document to document, we would

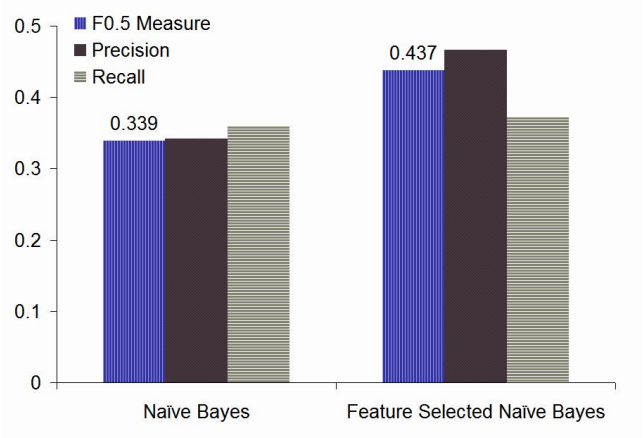

Figure 3: Mean average performance of the classifiers evaluating all documents in the set.

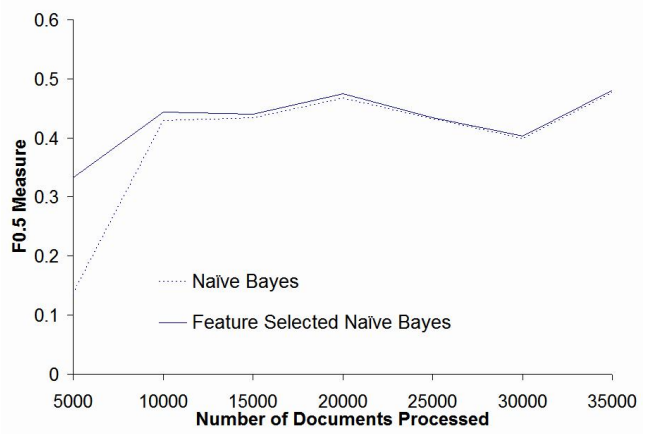

Figure 4: The f-measure performance of the classifiers evaluating the last 5000 documents over time.

expect to see very similar top-k subsets and so it may be sufficient to only update the $f_{0.5}$-measure statistics for each top-k subsets considered for that document.

6 Experimental Results

Following a similar experimental setup as in the TREC11's adaptive filter task [9], we evaluate a naive Bayes classifier and a naive Bayes classifier augmented with online feature selection as the overall classifier in the iScore framework. The results in this section show the mean average performance of the classifiers over the 43 different RSS feeds. The feeds serve as proxies for users in the Yahoo! News collection.

Figure 3 shows the overall performance of the classifiers evaluating all documents in the set. The figure shows that the feature-selected naive Bayes classifier yields significantly higher precision while maintaining a similar recall level as the traditional naive Bayes classifier. Consequently, the mean average $f_{0.5}$-measure for the feature-selected naive Bayes is 28.9% better.

Figure 4 shows the mean average $f_{0.5}$-measure performance in classifying the previous 5,000 documents at different time periods. The number of articles in each time period roughly follows the number of documents evaluated in each period in the TREC11 evaluation. We believe the window size is sufficiently large to give accurate results yet small enough to give results for the early documents of the document stream. The figure indicates that the majority of improvement over the traditional naive Bayes is attributed to the better classification of the initial 5,000 documents. From the figure, we can conclude that correlation can be used to determine which useless features to discard very quickly. A traditional naive Bayes classifier can learn after sometime that a feature is useless when its kernel estimators determine that the distribution of a feature is uniform. However, the figure shows that using correlation is much faster in identifying uniform distributions for useless features. The initial performance by the feature-selected naive Bayes is very important for news recommendation systems. A system is only successful if it begins to accurately recommend interesting articles early on. A system that requires a significant amount of training and incorrectly recommends articles initially would discourage users from adopting the system.

Given all the scores generated during the online classification process for each feed, we evaluate the two classifiers in an offline context. For each recall level, we determine the minimum threshold for each feed and classifier pair to attain the desired recall. We compute the precision achieved for that threshold. Figure 5

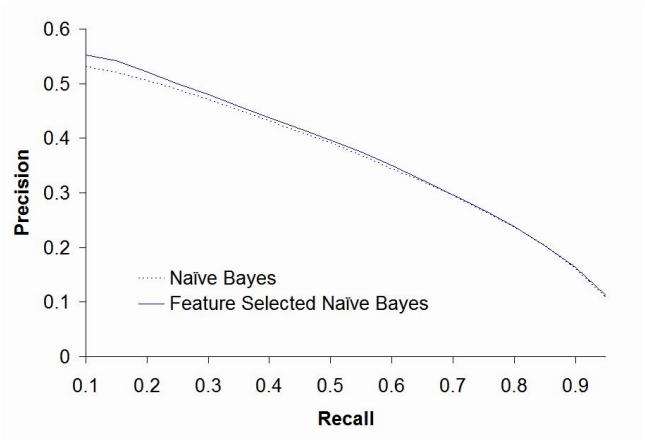

Figure 5: Precision recall curve of the classifiers.

shows the average precision-recall performance of the two classifiers for all the feeds. Although the chart shows a less dramatic improvement introduced by the feature-selected naive Bayes than that shown in our evaluations for online classifiers discussed earlier, it does show that at all recall levels, the feature-selected naive Bayes has a higher mean average precision, especially at lower recall levels. At low recall levels, the difference in precision between the two classifiers is much higher. In other words, the feature-selected naive Bayes can yield higher recall at higher precision levels. It is important to note that this evaluation, typical for offline classifiers, differs from the earlier evaluations for online classifiers. In an online setting, the scores generated by the classifiers are thresholded using a dynamic threshold that tries to maximize the $f_{0.5}$-measure. Consequently, the threshold is allowed to change as documents are processed, adjusting with the evolving accuracy of the classifier. In contrast, in an offline setting, a single static threshold is used.

7 Conclusion

Online feature selection for naive Bayes significantly improves the accuracy in recommending news articles, particularly, when there is very little training data. By learning which features are useful and useless for identifying interesting articles for a specific user in an online setting, the augmented naive Bayes can adapt quickly to changes in the definition of what makes an article interesting with little training data. By considering only useful subsets of features, our online feature selection approach is efficient while yielding higher quality results that are 28% better than the traditional naive Bayes classifier.

References

[1] G. Forman, "A pitfall and solution in multi-class feature selection for text classification," in *Proc. 21st International Conference on Machine Learning*, p. 38, 2004.

[2] R. K. Pon, A. F. Cardenas, D. Buttler, and T. Critchlow, "iscore: Measuring the interestingness of articles in a limited user environment," in *IEEE Symposium on Computational Intelligence and Data Mining 2007*, (Honolulu, HI), April 2007.

[3] R. Carreira, J. M. Crato, D. Goncalves, and J. A. Jorge, "Evaluating adaptive user profiles for news classification," in *IUI '04: Proceedings of the 9th international conference on Intelligent user interface*, (New York, NY, USA), pp. 206–212, ACM Press, 2004.

[4] H.-J. Lai, T.-P. Liang, and Y. C. Ku, "Customized internet news services based on customer profiles," in *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, (New York, NY, USA), pp. 225–229, ACM Press, 2003.

[5] D. Billsus, M. J. Pazzani, and J. Chen, "A learning agent for wireless news access," in *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, (New York, NY, USA), pp. 33–36, ACM Press, 2000.

[6] D. Radev, W. Fan, and Z. Zhang, "Webinessence: A personalised web-based multi-document summarisation and recommendation system," in *Proceedings of the NAACL-01*, pp. 79–88, 2001.

[7] G. M. D. Corso, A. Gulli, and F. Romani, "Ranking a stream of news," in *WWW '05: Proceedings of the 14th international conference on World Wide Web*, (New York, NY, USA), pp. 97–106, ACM Press, 2005.

[8] S. A. Macskassy and F. Provost, "Intelligent information triage," in *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 318–326, ACM Press, 2001.

[9] S. Robertson and I. Soboroff, "The trec 2002 filtering track report," in *TREC 2002*, 2002.

[10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[11] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1-2, pp. 273–324, 1997.

[12] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 1-2, pp. 245–271, 1997.

[13] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Mach. Learn.*, vol. 2, no. 4, pp. 285–318, 1988.

[14] V. R. Carvalho and W. W. Cohen, "Single-pass online learning: performance, voting schemes and online feature selection," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 548–553, ACM Press, 2006.

[15] P. E. Utgoff, N. C. Berkman, and J. A. Clouse, "Decision tree induction based on efficient tree restructuring," *Machine Learning*, vol. 29, October 1997.

[16] S. Perkins and J. Theiler, "Online feature selection using grafting," in *ICML*, pp. 592–599, 2003.

[17] P. Nurmi and P. Floreen, "Online feature selection for contextual time series data," in *PASCAL Subspace, Latent Structure and Feature Selection Workshop*, (Bohinj, Slovenia), February 2005.

[18] Yahoo, "Yahoo news rss feeds." [Online] http://news.yahoo.com/rss, 2007.

[19] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann, 2nd edition ed., 2004.

[20] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, 1995.