A Probabilistic Model for High Recall Feature Selection

Paper # 1253

Abstract

Feature selection is important in many practical supervised learning settings to address computational constraints or to prevent overfitting when the data may be high-dimensional but contain relatively few samples. While a wide variety of feature selection methods have been proposed in the literature, little research seems to focus on feature selection specifically targeted to improve recall in the case of binary classification. In this paper we propose a novel probabilistic voting model of feature selection and argue that choosing features so as to maximize likelihood objectives w.r.t. this model provides an effective method for high recall feature selection. For each objective, we derive an efficient formula for feature selection in the filtering framework (i.e., greedy forward selection) and we empirically compare the resulting feature selection criteria to a wide variety of existing methods. Results show that our high recall feature selection approach does indeed improve recall with improvements noticeable when there is a high feature to data ratio. Such results provide a novel and efficient feature selection algorithm to target high-recall classification tasks.

Introduction

Feature selection is important in many practical supervised learning settings to address computational constraints (e.g., in large-scale or online learning) or to prevent overfitting when the data may be high-dimensional but contain relatively few samples (e.g., as may occur in medical or bioinformatics domains where features are abundant but data is costly to obtain) Guyon and Elisseeff (2003).

While a wide variety of feature selection methods have been proposed in the literature, little research seems to focus on feature selection specifically targeted to improve recall (minimization of false negatives) in binary classification. One reason for this is simply that most feature selection methods are agnostic to the particular supervised learning task – applying to tasks from classification to regression – and hence are not focused on performance properties specific to binary classifiers. Yet, recall is an important aspect of many binary classification problems (e.g., minimizing false negatives in the identification of cancerous tumors) and it is critical to have feature selection algorithms that are targeted to maintain high recall.

Naturally though, it would not make sense for a classifier or feature selection method to focus on recall alone since the classifier which always predicts *true* (independent of the data) obtains optimal recall performance. To trade off precision with recall in order to maintain high accuracy, one often uses a geometric average of the two instead (known as F-score) but in practice most classifiers directly optimize accuracy or some surrogate, e.g., a convex surrogate of 0-1 loss as in the SVM or maximum (conditional) likelihood as in Naive Bayes or logistic regression. So how can one achieve a high recall SVM, Naive Bayes, or logistic regression classifier that already has a well-defined accuracy-focused optimization criterion? The idea we pursue in this paper is that feature selection can help modulate the recall performance of existing classifiers by encouraging selection of features that cover more of the true cases in the data (hence discouraging false negatives).

In this paper we propose a novel probabilistic voting model of feature selection with the intent of encouraging false negative reduction while still focusing on accuracy. We argue that choosing features so as to maximize likelihood objectives w.r.t. this model provides an effective method for high recall feature selection. Specifically, for each objective, we derive an efficient formula for feature selection in the filtering framework (i.e., greedy forward selection) and we empirically compare the resulting feature selection criteria to a wide variety of existing methods.

Our results demonstrate that our high recall feature selection approach does indeed improve recall with improvements noticeable when there is a high feature to data ratio. Such results provide a novel and efficient feature selection algorithm to target high-recall classification tasks.

Classification and Feature Selection

In this section, we briefly define the task of binary classification along with standard definitions of performance metrics we may wish to optimize. We then follow this by a discussion of feature selection and existing criteria proposed in the literature.

In the binary classification task, we assume we are given data $D = \{(\vec{x}^d, y^d)\}$ consisting of pairs of real-valued raw input vectors $\vec{x}^d \in \mathbb{R}^n$ of length n (e.g., the results of n different medical tests) and *actual* binary class label $y^d \in \{0(\text{false}), 1(\text{true})\}$ (where we often write F for false and T for true). A binary classifier is a function $C : \vec{x}^d \rightarrow y^d$ such that given a new unlabeled raw feature vector, $C(\vec{x}^d)$ produces a *predicted* classification.

Given a trained classifier C and a dataset D, we can build the well-known contingency table

	Actual T	Actual F
Predicted T	TP	FP
Predicted F	FN	TN

where the four entries represent the counts of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) and sum to the total amount of data (i.e., $\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN} = |D|$). Each table entry represents the count of data for which the respective row matched the predicted classification $C(\vec{x}^d)$ and the respective column matched the actual label y^d. Given these definitions, we can easily define

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$

where accuracy represents the overall fraction of correct classification, but recall represents the overall fraction of true labeled data that has been classified as true (i.e., recalled). For somewhat rare events such as medical diagnosis of cancer, we certainly care about accuracy, but we may also want to place additional emphasis on recall performance so as to avoid the occurrence of false negatives (cases of cancer that were missed by the classifier). Of course false positives are also a problem, but additional tests would rule these out and hence not as critical of a classification failure as missing a potential cancer diagnosis.

Previously we did not specify exactly how $C(\vec{x}^d)$ learns from the raw data $\vec{x}^d$. In general, practitioners often process the raw input vectors $\vec{x}^d$ into a subset of features that we'll denote $f_1, \ldots, f_k$. Whereas the raw input may consist of the results of individual tests, a feature $f_i \in \mathbb{R}$ may represent some nonlinear function of one or more tests deemed to be relevant to the classification task. Hence we might more appropriately write a classifier as $C(\vec{x}^d, f_1, \ldots, f_k)$ to represent the raw input data and the features of the data that the classifier may use. In the case of high-dimensional data or few data samples, we may wish to limit the set of features generated to improve classifier performance and this is the task of feature selection — select $\{f_1, \ldots, f_k\}$ to optimize performance. There are already a variety of feature selection methods that we outline next.

Existing Feature Selection Algorithms

Existing feature selection algorithms fall into categories of filter methods and wrapper methods Guyon and Elisseeff (2003). While wrapper methods, such as SVM-RFE Guyon et al. (2002), select features by evaluating their usefulness to a given classifier, filter methods evaluate feature subsets according to certain properties of features themselves. Within filter methods, feature selection algorithms can be further classified into ranking methods and subset methods. The former evaluate each feature independently, while the latter evaluate a subset at a time Brown et al. (2012). Listed below is a brief introduction to popular feature selection algorithms that we later compare to.

Correlation based rank is a naïve ranking algorithm that ranks features according to their linear correlation with the output, i.e. Pearson's r. More sophisticated ranking methods include:

1. The conditional entropy of class Y given feature X, i.e. $H\left(Y|X\right)$, quantifies the amount of information in Y that is not provided by X. Subtracting $H\left(Y|X\right)$ from the entropy of class Y, the information gain of class Y given feature X is the mutual information between feature X and class Y, i.e. $I\left(X;Y\right)$.
2. The gain ratio of feature X is defined as the information gain of feature X normalized against the entropy of itself, i.e. $\frac{I(X;Y)}{H(X)}$.
3. The symmetric uncertainty between feature X and class Y measures the amount of redundancy between them. It is defined as $U\left(X,Y\right) = 2\frac{I(X;Y)}{H(X)+H(Y)}$.
4. The Relief method Kira and Rendell (1992) evaluates the relevance of features to the output class according to how well their values distinguish between nearest instances of the same and different classes.

Correlation-based subset Hall (1998) is an extension of Pearson's r to subset methods. It measures the linear correlation between each pair of selected features in addition to the correlation between selected features and the output class. The other subset algorithm to which we compared our work is MRMR Peng, Long, and Ding (2005). MRMR is an information-theoretic method that not only maximizes the relevance of selected features to the supervised output class, but also minimizes the redundancy among selected features. Different search strategies can be applied using these two algorithms as heuristics.

Although existing algorithms give various solutions, they are mostly ad-hoc, until Brown et al. (2012). Brown et al. derived a probabilistic model with information-theoretic motivations from first principles, and retro-fitted existing information-theory-based feature selection methods to the model. In this paper, we also take a derivational approach to specifying a probabilistic model and optimizing a feature selection objective w.r.t. that model, but in our case we focus on the specific task of optimizing for recall in binary classification, which it seems no prior method has focused on in previous work.

A Probabilistic Model of Feature Selection

In our work, we wish to derive a model of feature selection in the filtering framework that focuses specifically on recall (we assume the classifier itself aims to maximize accuracy or some surrogate). To begin our derivation, we propose a graphical model of feature selection for the binary classification problem, as shown in Figure 1. Double-walled nodes

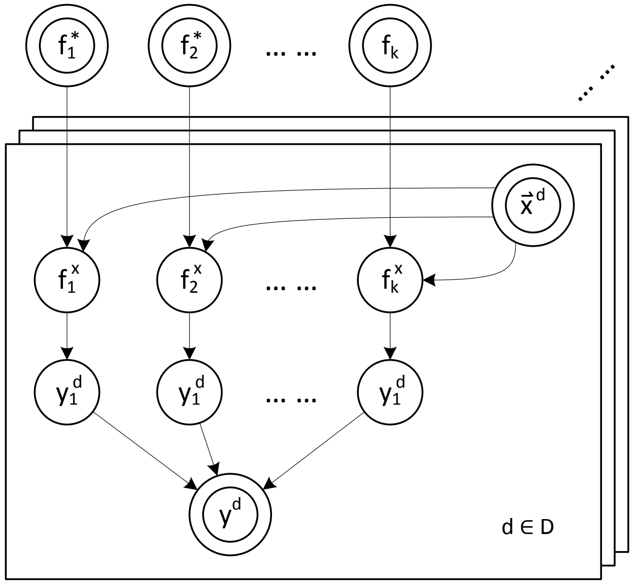

Figure 1: A probabilistic voting model of classification represented as a graphical model. Double-walled circles denote observed nodes, single-walled circles denote latent (unobserved) nodes.

represent observed variables, including the data point vector $\vec{x}^d$, selected features f_i^* (where $1 \leq i \leq k$, $f_i \in F$) and the supervised class label y^d. Single-walled circle nodes represent latent variables, including a selected attribute of data point vector f_i^x (i.e., the value resulting from applying a deterministic feature function f_i to data input $\vec{x}$) and the prediction from a feature $y_i^d \in \{0, 1\}$ (whose conditional probability is the maximum likelihood estimate w.r.t. the data D). The only part of this model that we have not yet specified is $P(y^d|\{y_i^d\})$ which determines how an observed classification prediction y^d is made given the latent predictions y_i^d of each selected feature f_i. We use a deterministic voting scheme to represent this conditional probability, which we will discuss in-depth shortly. First we focus on selecting features which optimize this model.

Since jointly optimizing this objective is NP-hard, we build F^* in a greedy manner by choosing the next optimal feature f_k^* given the previous set of optimal features F_{k-1}^* and recursively defining $F_k^* = F_{k-1}^* \cup f_k^*$ with $F_0^* = \emptyset$. The fitness measure of a selected feature set F_k^* can be expressed in terms of log or expected likelihood, discussed next.

Log Likelihood (L) Because Figure 1 essentially provides a probabilistic classifier model as a function of the input data and selected features, we can choose features as to maximize the log likelihood of the data given the model (i.e., features selected), i.e., $l(D|F_k^*)$ as follows:

$$f_k^* = \arg\max_{f_k} l(D|F_k^*)$$

$$\begin{aligned}
&= \arg\max_{f_k} \log \prod_{d \in D} P\left(y^d | \vec{x}^{\,d}, F_k\right) \\
&= \arg\max_{f_k} \sum_{d \in D} \log \left(\frac{P\left(y^d, \vec{x}^{\,d}, F_k\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \right) \\
&= \arg\max_{f_k} \sum_{d \in D} \log \sum_{\{y_i^d\}_{1 \leq i \leq k}} \frac{P\left(y^d, \vec{x}^{\,d}, F_k, \{y_i^d\}_{1 \leq i \leq k}\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \\
&= \arg\max_{f_k} \sum_{d \in D} \log \sum_{\{y_i^d\}_{1 \leq i \leq k}} P\left(y^d | \{y_i^d\}_{1 \leq i \leq k}\right) \prod_{i=1}^{k} P\left(y_i^d | \vec{x}^{\,d}, f_i\right)
\end{aligned} \tag{1}$$

Here, we rewrote the likelihood of a binary event as its probability, factorized the conditional probability in the joint probability divided by the marginal, marginalized over $\{y_i^d\}_{1 \leq i \leq k}$, factorized joint probability in conditional and prior following the graphical model and exploited D-separation to remove irrelevant conditions and to cancel terms in the equation.

Expected Likelihood (E) In contrast to log likelihood which tends to give more weight to extreme probabilities, we may wish to instead maximize expected likelihood defined as follows:

$$\begin{aligned}
f_k^* &= \arg\max_{f_k} E_D \left[P\left(y^d | \vec{x}^{\,d}, F_k\right) \right] \\
&= \arg\max_{f_k} \frac{1}{|D|} \sum_{d \in D} P\left(y^d | \vec{x}^{\,d}, F_k\right) \\
&= \arg\max_{f_k} \sum_{d \in D} P\left(y^d | \vec{x}^{\,d}, F_k\right)
\end{aligned}$$

If we remove the logarithm from the previous derivation above, we easily derive the final result of the log likelihood in (1) without the log.

Voting Schemes for High Recall Critically, it is how we specify the model and objective that we believe should encourage feature selection with high recall. Specifically, we have complete license in Figure 1 to determine the conditional probability $P(y^d|\{y_i^d\})$, which represents how the class prediction y_d is determined from the class prediction of each feature denoted as y_i^d. Here we consider two different extreme voting schemes represent logical conjuntion and disjunction of the feature votes (defined shortly). If we use conjunction, we require strong agreement among all features for a datum to be classified as true while for the disjunctive case, only one feature need vote true for the classification to be true. Overall, we believe conjunctive is better in terms of cancelling effects of noisy features (hence leading to accuracy), but we derive the results for both conjunction and disjunction since they are symmetric and will allow us to test this hypothesis. We remark that a critical feature of our two objectives above based on log and expected likelihood is that each datum is weighted equally and the newly selected features f_k^* which leads to more correct classifications for more data (especially those not already classified correctly under the model of Figure 1) are more likely to be selected, hence overall targeted recall.

Conjunctive Voting (CV) In order to select the subset of features with conjunctive voting, we need the agreement of all features predictors y_i^d to predict true. That is, when y^d is true we need all of the predictors y_i^d equals to true, and if y^d were false, just one of the predictors y_i^d would have to be false. Then the probability $P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right)$ can be expressed as follows:

$$P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right) = I\left[y^d = \bigwedge_{i=1}^{k} y_i^d\right] = \begin{cases} y^d = 1 \text{ when } \{y_i^d = 1\}_{1\leq i\leq k} \\ y^d = 0 \text{ when } \{y_i^d\}_{1\leq i\leq k, \exists y_j^d \| y_j^d = 0} \end{cases} \tag{2}$$

Then, we combined equations (1) and (2), separated each term according to the actual label value y^d and used the probability sum rule to rewire the second term as follows:

$$\begin{aligned} f_k^* &= \arg\max_{f_k} \sum_{d\in D} I\left[y^d = \bigwedge_{i=1}^{k} y_i^d\right] \prod_{i=1}^{k} P\left(y_i^d|\vec{x}^{\,d}, f_i\right) \\ &= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 1 : \prod_{i=1}^{k} P\left(y_i^d = 1|\vec{x}^{\,d}, f_i\right) \\ y^d = 0 : \sum_{\substack{\{y_i^d\}_{1\leq i\leq k}, \\ \exists y_i^d \| y_i^d = 0}} \prod_{i=1}^{k} P\left(y_i^d|\vec{x}^{\,d}, f_i\right) \end{cases} \\ &= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 1 : \prod_{i=1}^{k} P\left(y_i^d = 1|\vec{x}^{\,d}, f_i\right) \\ y^d = 0 : 1 - \prod_{i=1}^{k} P\left(y_i^d = 1|\vec{x}^{\,d}, f_i\right) \end{cases} \end{aligned} \tag{3}$$

$$= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 1 : \prod_{i=1}^{k} P\left(y^d = 1|\vec{x}_i^{\,d}, f_i\right) \\ y^d = 0 : 1 - \prod_{i=1}^{k} P\left(y^d = 1|\vec{x}_i^{\,d}, f_i\right) \end{cases}$$

From (3) we can intuitively describe how this equation is related to the precision metric. In the confusion matrix, Precision is calculated dividing the numbers of true positives by the number of all examples predicted as positive (true positives and false positives). When y^d is true, (3) gives higher score to feature that have higher probability to predict true, encouraging true positives. Meanwhile, if y^d is false, (3) gives lower score to features that have higher probability to predict true, penalising false positives.

Disjunctive Voting (DV) In order to select the subset of features with disjunctive voting, we need at least one y_i^d to predict true. Therefore, we need a disjunction operation between the predictors y_i^d. That is, when y^d is true, we need at least one predictor y_i^d to equals to true, and if y^d were false, all of the predictors y_i^d would have to be false. Then the probability $P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right)$ can be expressed as follows:

$$P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right) = I\left[y^d = \bigvee_{i=1}^{k} y_i^d\right] = \begin{cases} y^d = 1 \text{ when } \{y_i^d\}_{1\leq i\leq k, \exists y_j^d \| y_j^d = 1} \\ y^d = 0 \text{ when } \{y_i^d = 0\}_{1\leq i\leq k} \end{cases} \tag{4}$$

Here, we combined equations (1) and (4), separated each term according to the actual label value y^d and used the probability sum rule to rewrite the second term as follows:

$$\begin{aligned} f_k^* &= \arg\max_{f_k} \sum_{d\in D} I\left[y^d = \bigvee_{i=1}^{k} y_i^d\right] \prod_{i=1}^{k} P\left(y_i^d|\vec{x}^{\,d}, f_i\right) \\ &= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 0 : \prod_{i=1}^{k} P\left(y_i^d = 0|\vec{x}^{\,d}, f_i\right) \\ y^d = 1 : \sum_{\substack{\{y_i^d\}_{1\leq i\leq k}, \\ \exists y_i^d \| y_i^d = 1}} \prod_{i=1}^{k} P\left(y_i^d|\vec{x}^{\,d}, f_i\right) \end{cases} \\ &= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 0 : \prod_{i=1}^{k} P\left(y_i^d = 0|\vec{x}^{\,d}, f_i\right) \\ y^d = 1 : 1 - \prod_{i=1}^{k} P\left(y_i^d = 0|\vec{x}^{\,d}, f_i\right) \end{cases} \end{aligned} \tag{5}$$

$$= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 0 : \prod_{i=1}^{k} P\left(y^d = 0|\vec{x}_i^{\,d}, f_i\right) \\ y^d = 1 : 1 - \prod_{i=1}^{k} P\left(y^d = 0|\vec{x}_i^{\,d}, f_i\right) \end{cases}$$

From (5) we can intuitively describe how this equation is related to the precision metric. In the confusion matrix, Recall is calculated dividing the numbers of actual true positives by the number of all true positives (true positives and false negatives). When y^d is false, (5) gives higher score to feature that have higher probability to predict false, encouraging true negatives. Meanwhile, if y^d is true, (5) gives lower score to features that have higher probability to predict false, penalizing false negatives.

Empirical

In this section, we conduct experiments to study the feature selection performances of several well known methods such as Symmetrical Uncertainty Rank (SUR), Gain Ratio Rank (GRR), Mutual Information Rank (MIR), Correlation Based Rank (CR), Conditional Entropy Rank (CER) Guyon and Elisseeff (2003), Correlation-based Feature Subset Selection (CS) Hall (1998), Reliff (RF) Robnik-Sikonja and Kononenko (2003) , Minimum Redundancy Maximum Relevancem(MRMR) Peng, Long, and Ding (2005) and our proposed methods: Disjunction Voting Expectation(DVE) and Disjunction Voting likelihood (DVL), Conjunction Voting Expectation (CVE), Conjunction Voting likelihood (CVL). The first five methods are variable rank methods and the last seven are subset rank methods. The first five methods are variable rank methods that select variables by ranking them with some metric and the last seven are subset rank methods that assess subsets of variables and consider previous selections to decide the next selection.

In this experimental study, we evaluated each feature selection method on a variety of binary classification problems

Dataset	# Features	# Data	# Features/# Data	% True Labels	Feature Type
Breast Cancer(BC)	9	699	0.013	34 %	All Numeric
Diabetes(D)	8	768	0.010	35 %	All Numeric
Heart Statlog(HS)	13	270	0.048	44 %	All Numeric
Spect(S)	22	80	0.275	33 %	Categorical
Vote(V)	16	435	0.037	39 %	Categorical
Newsgroup(NG)	500	1963	0.255	49 %	All Binary
Horse Colic(HC)	22	368	0.060	37 %	Categorical (16) and numeric (6)
Credit-American(CR)	15	690	0.022	44 %	Categorical(9) and numeric (6)
Credit-German(GC)	20	1000	0.020	30 %	Categorical (13) and Numeric (7)
Hepatitis(H)	19	155	0.123	21 %	Categorical (13) and Numeric (6)
Ionosphere(I)	34	351	0.097	36 %	All Numeric

Table 1: Descriptive statistics of the various datasets evaluated in this work.

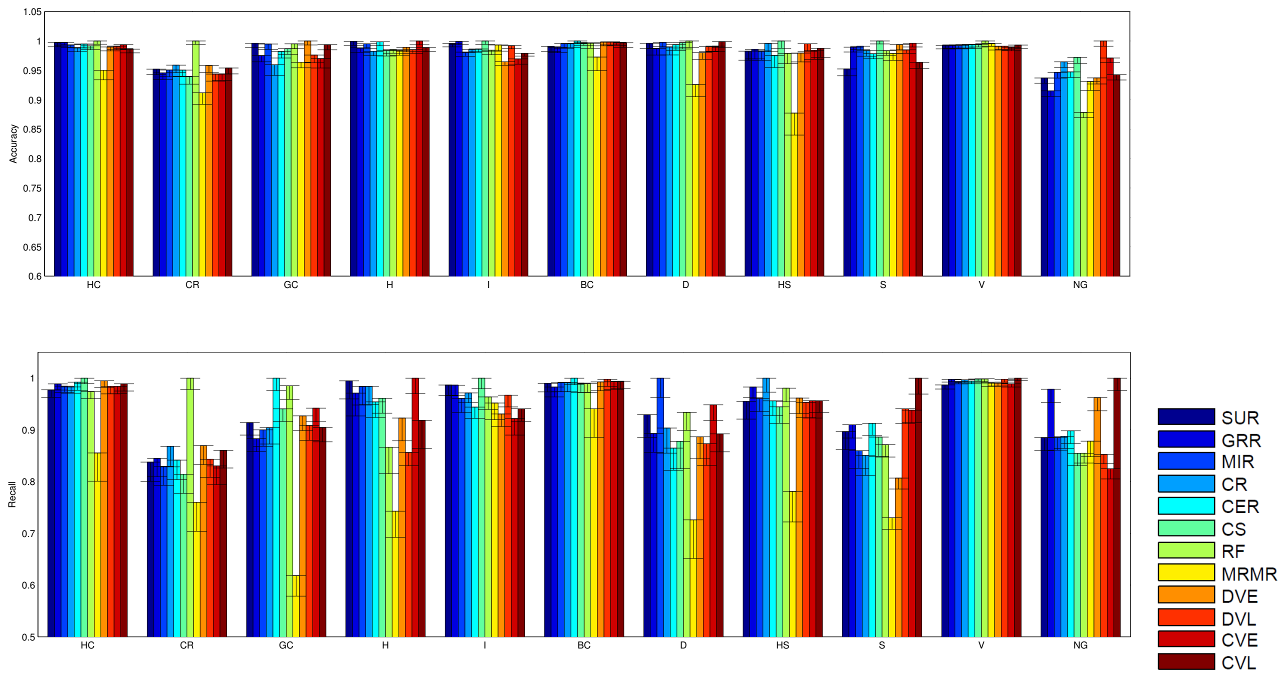

Figure 2: Performance of various feature selection algorithms per dataset, averaged across classifier type and each stage of feature selection. Overall the newly proposed CVL method performs well on Recall on most datasets, while the Recall performance of other feature selection algorithms vary; interestingly the best performance is achieved for datasets a high ratio of features to data indicating that the strong conjunctive voting scheme of CVL may help reduce noise in feature selection for these datasets.

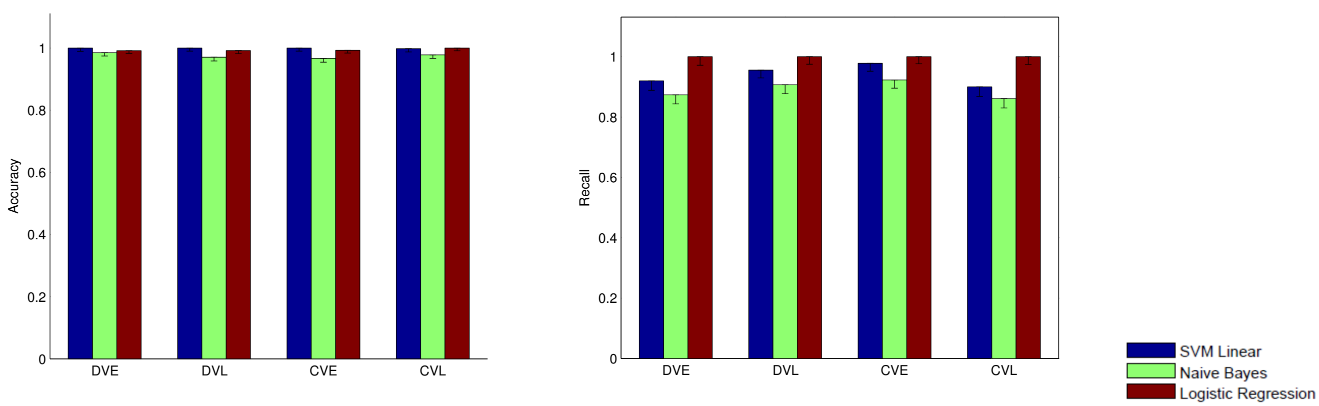

Figure 3: Performance of various feature selection algorithms per classifier, averaged across dataset and each stage of feature selection. DVE, DVL, CVE, and CVL align with the independence assumptions of Naive Bayes and the maximum (conditional) likelihood framework of Naive Bayes and Logistic Regression, which may explain the better excellent performance of these algorithms with these classifiers.

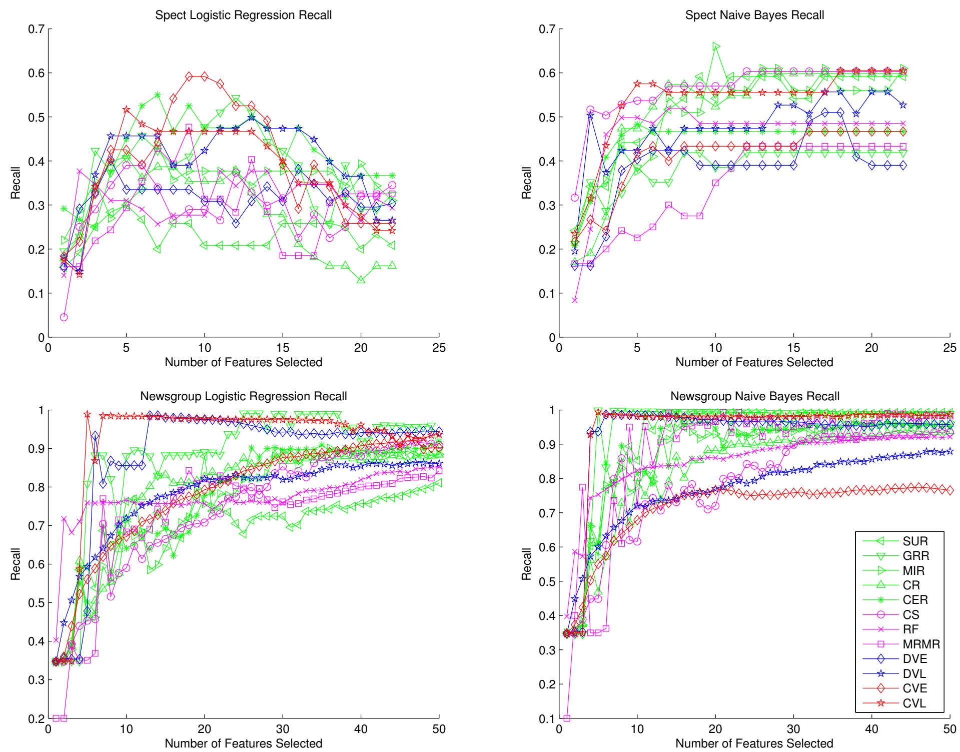

Figure 4: Performance distribution of various feature selection algorithms for two datasets and classifier types at each stage of feature selection. The conjunctive voting methods do best overall in these datasets.

from the UCI machine learning repository Bache and Lichman (2013). These datasets along with their properties are outlined in Table 1.

In addition, we used three different classifiers to solve each of the binary classifier problem. Logistic Regression, SVM Linear and Naive Bayes were used in the experiments and the first two are implemented in the LibLinear library Fan et al. (2008) and the last one is implemented in the Weka software Hall et al. (2009).

To evaluate feature selection algorithm performance, we perform 10-fold nested cross-validation (nesting for tuning hyperparameters of each classification algorithm) and evaluate accuracy, precision, and recall at each stage of feature selection from the first selected feature through to the maximum (either 50 or the number of features in the dataset, whichever is smaller).

In our evaluation, we wish to answer the following questions: Do different feature selection methods perform better with each classifier or on each dataset in comparison to others? How reliably do the different feature selection methods perform overall in terms of their performance distribution?

To answer these questions, we first start with Figure 2, where we evaluate the performance of various feature selection algorithms per dataset, averaged across classifier type and each stage of feature selection. We can observe that our approaches improve recall and accuracy in many of the binary classification problem.

In Figure 3, we examine how well our novel feature selection algorithms DVE, DVL, CVE, CVL perform vs. classifier, noting that overall performance is best for Naive Bayes followed by Logistic Regression and noticeably worse for the SVM Linear — it seems simply that the probabilistic nature of our feature selection dovetail best with classification methods that are themselves probabilistic, and overall best with Naive Bayes which makes the same independence assumptions.

Next we examine Figure 4, where we evaluate the performance distribution of various feature selection algorithms with samples taken for each dataset, classifier type, and stage of feature selection. This graph shows that our CVL high recall feature selection approach improves recall with the most improvement noticeable when there is a high feature to data ratio such as in the Newsgroup (NG) and Spect (S) data sets. We conjecture the reason for this stems from the strong agreement of features provided by the conjunctive voting scheme that helps reduce noise in the feature selection process.

Conclusion

In this paper we argued that no feature selection methods have previously targeted recall for the task of binary classification and we proposed a model and objectives to address this task. Empirically, our results show that our high recall feature selection approach shows strong performance on recall with significant improvements over other methods in cases where there is a high feature to data ratio. Future extensions of this work may examine tighter integration of the feature selection method with the algorithm to make an *embedded* approach based on our model since at it's core is a voting-based classifier.

References

Bache, K., and Lichman, M. 2013. UCI machine learning repository.

Brown, G.; Pocock, A.; Zhao, M.-J.; and Luján, M. 2012. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research* 13:27–66.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9:1871–1874.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3:1157–1182.

Guyon, I.; Weston, J.; Barnhill, S.; and Vapnik, V. 2002. Gene selection for cancer classification using support vector machines. *Machine learning* 46(1-3):389–422.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The weka data mining software: An update. *SIGKDD Explorations* 11.

Hall, M. A. 1998. *Correlation-based Feature Subset Selection for Machine Learning*. Ph.D. Dissertation, University of Waikato, Hamilton, New Zealand.

Kira, K., and Rendell, L. A. 1992. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, 129–134.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions* 27:1226–1238.

Robnik-Sikonja, M., and Kononenko, I. 2003. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning* 53:23–69.