Robustness of Regularized Linear Classification Methods in Text Categorization [*]

Jian Zhang
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, U.S.A.
jian.zhang@cs.cmu.edu

Yiming Yang
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, U.S.A.
yiming@cs.cmu.edu

ABSTRACT

Real-world applications often require the classification of documents under situations of small number of features, mis-labeled documents and rare positive examples. This paper investigates the robustness of three regularized linear classification methods (SVM, ridge regression and logistic regression) under above situations. We compare these methods in terms of their loss functions and score distributions, and establish the connection between their optimization problems and generalization error bounds. Several sets of controlled experiments on the Reuters-21578 corpus are conducted to investigate the robustness of these methods. Our results show that ridge regression seems to be the most promising candidate for rare class problems.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous; I.5.1 [**Pattern Recognition**]: Models–*Statistical*; I.5.4 [**Pattern Recognition**]: Applications–*Text processing*

General Terms

Algorithms, Performance, Reliability

Keywords

robustness, text categorization, SVM, ridge regression and logistic regression

1. INTRODUCTION

Many supervised learning methods have been applied to text categorization, including nearest neighbor classifiers, decision trees, Bayesian probabilistic classifiers, neural networks, regression methods, SVM, etc. And lots of empirical studies in text categorization have been done in recent years[9, 6, 15] which investigate different aspects of classification methods.

Text categorization problems can be characterized as dealing with high-dimensional and sparse data, and usually accompanied by skewly distributed categories. These characteristics together make text categorization different from classic pattern classification problems. Real-world applications often require the classification of documents under the following conditions:

1. Restrictions on space and time: Classifiers need less space and can be trained much faster with fewer features. And if vectorized files need to be stored and reused later, it will also reduce the storage and thus test time significantly since the most expensive part is to load the test data.

2. Mis-labeled training documents: The most crucial resources of classifiers are training documents, which are labeled by human. In cases there are many mis-labeled documents, candidate classifiers should be tolerant to labeling errors to some degree.

3. Small number of training documents: There are many important applications where only small number of positive training examples are available, like the filtering task in information retrieval. Candidate classifiers should perform reasonably well with rare positive training data.

Hastie et al. [3] gave insightful analysis on the robustness of classifiers based on their loss functions. However, the goodness of those analysis still depends on the intrinsic characteristics of the data. To our knowledge, no such study in text categorization has been done.

Yang & Liu [15] compared empirical results on common and rare categories. However, they only showed the performance degradation as the size of positive data decrease without further exploration, and their results are not comparable across categories.

Our work is mainly based on the work by Zhang [17], which studies several regularized linear classification methods and their applications in text categorization. However, we mainly focus on addressing the above issues, that is, we

[*] (Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
Copyright 2003 ACM 1-58113-646-3/03/0007 ...$5.00.

study three classifiers in text categorization under conditions of small number of features, noisy settings (mis-labeled data), and rare positive data.

We design several sets of controlled-experiments to investigate the behaviors of three regularized linear methods (ridge regression, regularized logistic regression and linear kernel SVM) under above conditions, as well as establish the connection between their optimization problems and generalization error bounds. Last but not the least, we analyze their different behaviors in case of rare positive data, which reveals another property in their loss functions.

The rest of this paper is organized as follows: Three linear methods together with their regularizations are introduced in Section 2. Section 3 discusses these methods in terms of loss functions and score distributions, generalization error bounds and implementations. Experimental setup is reported in Section 4, and results are reported in Section 5, where we compare three methods under above conditions and give our analysis. The last section summarizes the main results of the paper.

2. LINEAR METHODS AND REGULARIZATION

Among popular classification methods, linear classification methods are those methods that have linear decision boundaries between positive and negative classes, such as linear regression, logistic regression, linear kernel SVM, naive Bayes classifier, linear discriminant analysis, perceptron algorithm, etc. Compared with other methods, linear methods are simpler and the trained model is much easier to interpret. Even more important is that in text categorization they have been shown to be very effective and their performances are among the top classifiers [17].

We define the binary classification problem for linear methods as follows. Given $\chi = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$ the training set, where $y_i \in \{-1, +1\}$ and $\mathbf{x}_i$ is a vector of training instance. For linear classifier, it tries to find a weight vector $\mathbf{w}$, an intercept b and a threshold θ such that $f(\mathbf{w}^T\mathbf{x}+b) < \theta$ if the label is -1 and $f(\mathbf{w}^T\mathbf{x}+b) \geq \theta$ if the label is $+1$, where the function $f(.)$ is specified by the linear classifier. One can augment the input vector $\mathbf{x}$ to $[1, \mathbf{x}]$ and the weight vector $\mathbf{w}$ to $[b, \mathbf{w}]$ to absorb the intercept b.

Given a linear model and a task-specific loss function $l(f(\mathbf{x}), y)$, our goal is to minimize the expected loss:

$$\min E_D l(f(\mathbf{x}), y)$$

where D is the underlying and unknown distribution of our data. For classification problems, the loss function $L(f)$ is usually a convex upper bound of the classification error, which avoids the hardness of directly minimizing the classification error.

In order to fit the model, Empirical Risk Minimization (ERM) is usually used, which tries to minimize the above objective function over empirical data:

$$\min \frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{x}), y)$$

As we can see, ERM uses the uniform distribution over empirical data to replace the unknown distribution. Since empirical loss is the pure objective of ERM, ERM may overfit the data by favoring complex functions. Regularization [10, 11] is an effective way to prevent overfitting, and it has been successfully applied to many methods. The regularized version usually looks like:

$$\min \frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{x}), y) + \lambda J(f)$$

where $J(f)$ controls the learning complexity of the hypothesis family, and the coefficient λ controls balance between the model complexity and the empirical loss.

2.1 Linear SVM

SVM is based on statistical learning theory[13], which uses the principle of Structural Risk Minimization instead of Empirical Risk Minimization. It is regarded as high performance classifiers in many domains including text categorization. We limit our discussion to linear kernel SVM in this paper since it is the most popular kernel in text categorization [4, 1], and it is computationally much cheaper than other kernels. Linear SVM tries to find the hyperplane with maximum margin as the decision boundary in linear separable case, which is equivalent to minimizing the norm of the weight vector under some linear constraints[1]:

$$\begin{array}{rc} & \min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \textit{subject to}: & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i \end{array}$$

For linear non-separable case, by introducing slack variables, the optimization problem is augmented to:

$$\begin{array}{rc} & \min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \\ \textit{subject to}: & \xi_i \geq 0, \forall i \\ & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \forall i \end{array}$$

where C represents cost coefficient, and slack variable ξ_i measures how far away the corresponding data point $(\mathbf{x}_i, y_i)$ falls in the wrong side of the margin.

The dual form of the above optimization problem can be written as follows

$$\begin{array}{rc} & \max \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ \textit{subject to}: & 0 \leq \alpha_i \leq C, \forall i \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{array}$$

which is both theoretically and practically meaningful. From Karush-Kuhn-Tucker (KKT) conditions [7] we know that only those data points whose Lagrangian coefficients (α_i's) are not zeros contribute to the final decision boundary $\hat{\mathbf{w}} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i$. These data points are called "support vectors", while the remaining data points are called "non-support vectors". For text categorization people found that usually a small portion of the training data points are support vectors.

Note that SVM itself can be treated as a regularized method with the loss function rewritten as

$$\hat{\mathbf{w}} = \arg\min(\sum_{i=1}^{n}\frac{1}{n}\max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) + \lambda\mathbf{w}^T\mathbf{w})$$

where $\max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) = \xi_i$.

[1] We do not use augmented vector for SVM because people do not penalize the intercept in SVM. Otherwise, the constraints of SVM (in its dual form) will be changed, and so does its algorithm. Whether penalizing intercept or not will only bring in trivial modifications for ridge regression and logistic regression.

2.2 Linear Regression and Regularization

The problem of linear regression tries to find a linear function

$$f(x) = \mathbf{w}^T\mathbf{x}$$

that can fit the training data very well. Least squares algorithm is the most popular estimation method for linear regression, which is equivalent to the Maximum Likelihood Estimation when the y is influenced by Gaussian noise. Least squares algorithm computes a weight vector $\mathbf{w}$ based on the minimization of the squared loss between the model output $\mathbf{w}^T\mathbf{x}$ and y:

$$\begin{aligned}\hat{\mathbf{w}} &= \arg\min_{\mathbf{w}}\{\frac{1}{n}\sum_{i=1}^{n}(1 - y_i\mathbf{w}^T\mathbf{x}_i)^2\} \\ &= \arg\min_{\mathbf{w}}\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T\mathbf{x}_i)^2\}\end{aligned}$$

The solution is given by

$$\hat{\mathbf{w}} = (\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T)^{-1}\sum_{i=1}^{n}\mathbf{x}_i y_i$$

Though we can give the close form solution, the matrix $\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T$ can be singular, which means there are multiple minimizers for the objective function. Particularly, in text categorization the number of features is often larger than the number of training documents, and the matrix is singular in those cases. One solution is to use pseudo-inverse matrix[14], which is built on top of the computation of Singular Value Decomposition (SVD). Another solution is to use ridge regression [3, 17], which regularizes the original objective function by adding a penalizer $\lambda\mathbf{w}^T\mathbf{w}$. In this paper we only discuss the later case, whose objective function becomes the following:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 + \lambda\mathbf{w}^T\mathbf{w}\}$$

and now the close form solution becomes

$$\hat{\mathbf{w}} = (\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T + n\lambda\mathbf{I})^{-1}\sum_{i=1}^{n}\mathbf{x}_i y_i$$

Note that after this transformation, the matrix

$$\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T + n\lambda\mathbf{I}$$

is guaranteed to be non-singular provided $\lambda > 0$. As a result, we have a unique solution for ridge regression, and the optimization problem can be solved with simple algorithm.

2.3 Logistic Regression and Regularization

Logistic regression has been widely used in statistics for many years, and has received extensive studies in machine learning recently due to its close relation to SVM and AdaBoost. However, in text categorization, it has not been as widely used as least squares algorithm and SVM. Recently this method is applied to text categorization and compared with other linear classification methods [9, 17], which shows that its performance is comparable to that of SVM. Logistic regression tries to model the conditional probability $p(y|x)$, and the model is fitted by maximizing the conditional log-likelihood. For binary classification problems the conditional probability is modeled as:

$$p(y \mid \mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-y\mathbf{w}^T\mathbf{x})}$$

The Maximum Likelihood Estimation is equivalent to minimizing the following:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}\frac{1}{n}\sum_{i=1}^{n}\log(1 + \exp(-y_i\mathbf{w}^T\mathbf{x}_i))$$

The solution of the above problem may be infinite: Suppose training data are linear separable, and $\mathbf{w}_0$ is one separating weight vector. Then any weight vector $\alpha\mathbf{w}_0$ provided $\alpha > 1$ can separate training data with a smaller objective function value. So the solution is unbounded.

In order to solve this problem, we once again resort to regularized version:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}\{\frac{1}{n}\sum_{i=1}^{n}\log(1 + \exp(-y_i\mathbf{w}^T\mathbf{x}_i)) + \lambda\mathbf{w}^T\mathbf{w}\}$$

The Hessian matrix of the objective function L (w.r.t. $\mathbf{w}$) is defined as:

$$\begin{aligned}H &= \frac{\partial}{\partial\mathbf{w}^T}\frac{\partial L}{\partial\mathbf{w}} \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{\exp(-y_i\mathbf{w}^T\mathbf{x}_i)}{(1 + \exp(-y_i\mathbf{w}^T\mathbf{x}_i))^2}\mathbf{x}_i\mathbf{x}_i^T + 2\lambda\mathbf{I}\end{aligned}$$

where $\mathbf{I}$ is the identity matrix. It is straight forward to show that the Hessian matrix is positive definite (given $\lambda > 0$), which is equivalent to the convexity of the objective function[7]. After the regularization the Hessian matrix is bounded away from $\mathbf{0}$, which is a nice property for many numerical algorithms.

3. ANALYSIS

3.1 Loss Functions and Distributions

Based on previous discussions, we can unify the objective functions of all three linear classification algorithms as:

$$L = \frac{1}{n}\sum_{i=1}^{n}f(y_i\mathbf{w}^T\mathbf{x}_i) + \lambda\mathbf{w}^T\mathbf{w}$$

We can see that all these methods are using the 2-norm regularization term, and the difference is that each method is associated with a particular loss function $f(.)$:

1. Ridge Regression:
   $f(y_i\mathbf{w}^T\mathbf{x}_i) = (1 - y_i\mathbf{w}^T\mathbf{x}_i)^2$

2. Regularized Logistic Regression:
   $f(y_i\mathbf{w}^T\mathbf{x}_i) = \log(1 + \exp(-y_i\mathbf{w}^T\mathbf{x}_i))$

3. Linear SVM:
   $f(y_i(\mathbf{w}^T\mathbf{x}_i + b)) = \max\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\}$

It would be meaningful to compare their loss functions against the mis-classification error in Figure 1 [3], which can help us understand different behaviors among three methods.

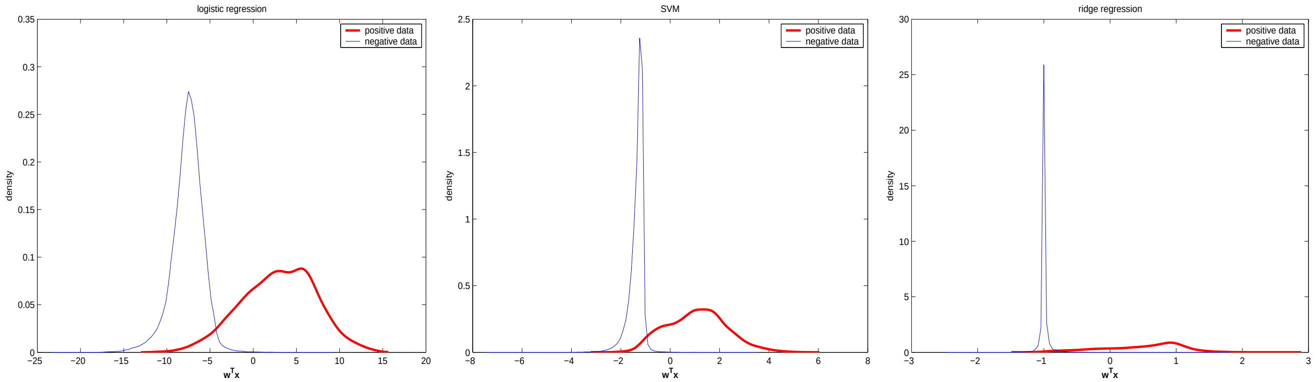

Figure 2: Distributions of $\mathbf{w}^T\mathbf{x}$ for positive and negative data (over test data, feature size=3000)

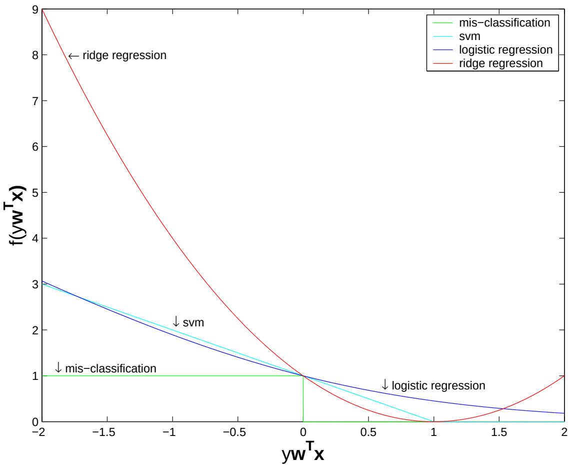

Figure 1: Loss Functions

We can see from the graphs (or simply derive from their formulas) that all three loss functions are convex functions, and they are also upper bounds of the mis-classification error.

It is believed that squared loss is not as robust as the other two loss functions since its loss function will be more influenced by extreme data, as we can see from the graph. Another disadvantage of squared loss is that it also penalizes those correctly classified data points if their output values are larger than 1. Meanwhile, it has the good property that both the first and second derivatives of its objective function are well-behaved, and it can be solved with simple optimization techniques.

SVM loss is linear, and logistic loss is close to linear. Both loss functions are less sensitive to extreme data points compared with squared loss. The non-differentiable characteristic of SVM loss function makes it harder to solve than the other two methods.

To show how effective these loss functions are, we plot the score distributions ($\mathbf{w}^T\mathbf{x}$) for three methods over test data in figure 2. We can see that most of the dense of the distributions has loss close to zero, which shows that our data is linear-separable to a great degree. Also notice that the overlapping of ridge regression is small compared with the other two methods, which is consistent with its good performance reported in section 5.

3.2 Generalization Error

We first refer to two theorems of Vapnik[12, 13]:
Theorem 1 Suppose $f(\mathbf{x}, \alpha)$ is a set of learning functions for binary classification with adjustable parameters α, then the following bound holds with a probability of at least $1-\eta$:

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

where $R(\alpha) = \int L(y, f(\mathbf{x}, \alpha))dF(\mathbf{x}, y)$ is the expected risk, $R_{emp}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i, y_i))$ is the empirical risk, n is the size of training data, h is the VC dimension of the learning model, $L(.)$ is the mis-classification error loss, and $F(.)$ is the underlying data distribution which is unknown.

Theorem 2 A subset of separating hyperplane defined on $\chi = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$ $(\forall i : \mathbf{x_i}^T\mathbf{x_i} \leq D^2)$ satisfying $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 : \forall i$ and $\mathbf{w}^T\mathbf{w} \leq A^2$ has the VC dimension h bounded by

$$h \leq \min([D^2A^2], n) + 1$$

From the above theorems we know that for linear-separable data, if we shrink the hypothesis space by putting limit on the 2-norm of the weight vector $\mathbf{w}$ of linear classification methods, we can potentially reduce the VC dimension, thus reduce the generalization error bound.

On the other hand, we know that the optimization problem[8]

$$\min f(\mathbf{w}) \ \mathit{subject\ to} : \mathbf{w}^T\mathbf{w} \leq A^2$$

is equivalent to the problem

$$\min f(\mathbf{w}) + \lambda \mathbf{w}^T\mathbf{w}$$

given the function $f(.)$ and the constraint are convex. So, by appropriately choosing λ, we are actually limiting our hypothesis space by the constraint $\mathbf{w}^T\mathbf{w} \leq A^2$. This establishes the connection between all three regularized methods and the above generalization error bound, which has been regarded as one major theoretical advantage of SVM.

3.3 Implementation Issue

For real-world applications, computational efficiency is another important issue. In our experiments, we use SVM^{light}

[5] for the linear kernel SVM model, and use iterative algorithms[17] that are variants of Gauss-Seidel[2] for solving both the ridge regression and regularized logistic regression.

The algorithm of ridge regression is very simple and more efficient than the other two methods. Another advantage of ridge regression (though we did not apply in this paper) is that for collections that contain large number of categories (like Ohsumed), we can first compute the matrix inverse

$$\mathbf{M} = (\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T + n\lambda\mathbf{I})^{-1}$$

which is independent of category[2], followed by the computation of $\mathbf{M}\sum_{i=1}^{n} \mathbf{x}_i y_i$ for each individual category.

4. EXPERIMENTAL SETUP

Reuters-21578 (ModApte split) is used as our data collection, which is a standard testbed for text categorization. Since every document in the collection can have multiple labels, we split the classification problem into multiple binary classification problems with respect to each category. All numbers and stopwords are removed, and words are converted into lowercase without stemming. Infrequent features (occur less than 3 times) are filtered out. Feature selection is done using Information Gain[16] per category, binary term weighting is applied, and words in document titles are treated as different words in document bodies.

Precision (p) and recall (r) are used to evaluate methods in their combined form $F1$, which is defined as

$$F1 = \frac{2rp}{(r+p)}$$

In order to compare the global performance of different methods, $Macro_{avg}F1$ and $Micro_{avg}F1$ are also used. As we know, $Macro_{avg}F1$ gives the same weight to all categories, and thus it will be mainly influenced by the performance of rare categories for our data collection due to the skewed category distribution of the Reuters-21578 collection. On the contrary, $Micro_{avg}F1$ will be dominated by the performance of common categories.

We conduct our experiments under three sets of conditions:

1. Robustness in terms of small number of features: We compare classifiers' performance as the number of features varies. Particularly, we examine their performance in case of very few features, which are top-ranked features by Information Gain.

2. Robustness in terms of "noise" level (mis-labeled data): We randomly pick up some portion of training examples and flip their labels. Performance is measured with respect to the percentage of flipped training examples.

3. Robustness in terms of "rare" positive training data: In order to study different behaviors of three classifiers in case of rare positive training data, we use the top 12 common categories in our data collection and simulate the process by reducing the available percent of positive training data.

5. RESULTS AND DISCUSSIONS

5.1 Performance vs. Feature Size

In this subsection we show how well those methods can perform with relatively small number of features.

Both thresholds and the regularization coefficient λ ($\lambda = 10^{-k}, k = 0, 1, \cdots, 5$) are chosen by maximizing F1 over training data with 5-fold cross-validation.

Our results in both $Micro_{avg}F1$ and $Macro_{avg}F1$ are shown in figure 3. And $F1$ results of the most common 12 categories (with 3000 features) are also listed in table 1, which are consistent with previous published results [17]. From the figure 3 we can see that even with 30 features, the $Micro_{avg}F1$ of three methods can still be above 80%, which is an acceptable performance for some applications. On the other hand, the $Macro_{avg}F1$s of three methods behave differently with ridge regression the best and logistic regression the worst. The difference of $MacroF1$ results (right graph in Figure 3) between ridge regression and SVM (and logistic regression) are significant[3].

Since the $Macro_{avg}F1$ performance is mainly influenced by rare categories, we believe that three methods must have different behaviors in case of rare categories. Our later experiments about "rare positive data" will further explore the differences.

Table 1: $F1$ performance of SVM, ridge regression and regularized logistic regression (Reuters-21578, top 12 categories, feature size=3000)

	# of positive training examples	SVM	RR	LR
earn	2877	0.978	0.976	**0.980**
acq	1650	**0.962**	0.956	0.960
money-fx	538	0.762	0.728	**0.779**
grain	433	0.912	**0.929**	0.902
crude	389	**0.886**	0.865	0.879
trade	368	0.741	0.700	**0.754**
interest	347	0.764	**0.785**	0.743
wheat	212	0.902	**0.903**	0.895
ship	197	0.830	**0.846**	0.834
corn	180	**0.920**	0.917	0.889
money-supply	140	0.820	**0.836**	0.825
dlr	131	**0.791**	0.727	**0.791**

5.2 Performance vs. Noise Level

In order to conduct experiments under noisy settings, we randomly pick up 1%, 3%, 5%, 10%, 15%, 20% and 30% training data respectively and flip their labels. Thresholds and the regularization coefficient λ are tuned with 5-fold cross-validation under each condition to make sure that the term $\lambda\mathbf{w}^T\mathbf{w}$ can play an active role for every method to resist noisy data.

We only reported $Micro_{avg}F1$ versus noise level in figure 4. $Macro_{avg}F1$ results of all three methods drop dramatically (below 0.05) after noise level is greater than 3%. This can be explained as follows: Most of the 90 categories in

[2] Suppose features are independent of categories.

[3] We use Macro t-test [15] with significant level 0.05.

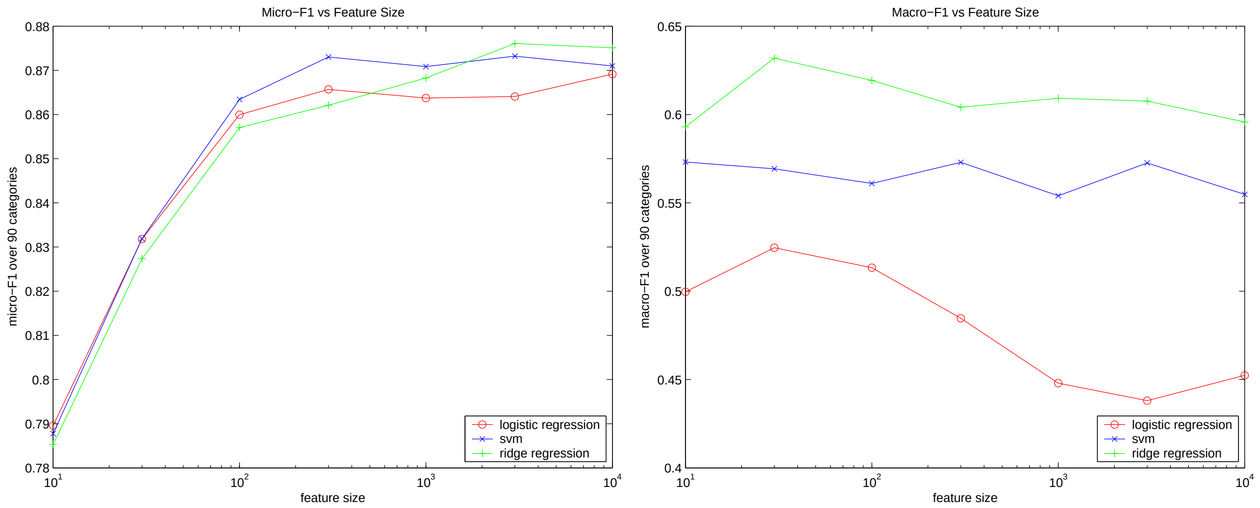

Figure 3: Micro-F1, Macro-F1 vs. Feature Size

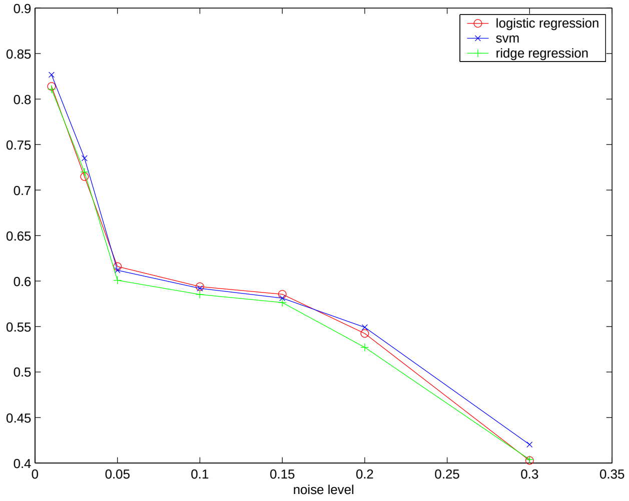

Figure 4: $Micro_{avg}$ $F1$ vs. noise level (percentage of mis-labeled data)

our data collection are relatively small, and due to the way we manipulate our data, a $p\%$ flip of negative data (which become "positive" now) will overwhelm the amount of remaining positive data which are correct. Thus, it is very hard to learn the target correctly for small categories, which leads to the poor performance in $Macro_{avg}F1$.

Our results show that SVM is superior to the other two methods as the noise level initially goes up, as we anticipated [4]. Ridge regression works reasonably well though it failed in the significant test compared with SVM, if we consider the large number of decisions it made (number of document times number of categories). This tells us that for text collection, squared loss is acceptable even with certain amount of mis-labeled documents.

5.3 Performance vs. Rare Positive Data

There are many cases where only small number of positive examples are available to our classifiers. Here we want to examine the different behaviors of the three methods under such cases, which can help us further explain the performance differences of $Macro_{avg}F1$ in figure 3.

One way to deal with rare class classification problems is to re-weight the training data (or change the cost function to be asymmetric) so that the same amount of positive data play more important roles than negative data. All three methods can be adapted in to this version by slightly modifying the original optimization problem into:

$$\hat{\mathbf{w}} = \arg\min\{\frac{C_{+/-}\sum_{i\in D_+} f(y_i\mathbf{w}^T\mathbf{x}) + \sum_{i\in D_-} f(y_i\mathbf{w}^T\mathbf{x})}{(|D_+|C_{+/-}+|D_-|)} + \lambda\mathbf{w}^T\mathbf{w}\}$$

where D_+ and D_- are sets of positive and negative examples, $|D_+|$ and $|D_-|$ are their cardinalities, $C_{+/-}$ measures the relative importance between positive and negative data, and $(|D_+|C_{+/-}+|D_-|)$ is the normalization factor so that λ is set to be independent of number of examples.

We did not apply this approach in our experiments for several reasons: First of all, all three methods can be adapted in this way, which does not help to reflect different characteristics among three methods. Second, we want to examine the capabilities of dealing with rare class in a natural way, while this adaptation changes the data prior distribution. And the weight ratio $C_{+/-}$ between positive and negative data need to be chosen empirically per category by cross-validation, which may be unstable for rare class.

Instead, we design our experiments to directly examine different behaviors of the three methods without the changing the optimization objective function.

In order to examine performance under this condition, we choose the most 12 common categories (in terms of number of positive training examples, see table 1) in our data collection. Since these categories are relatively common, we can randomly hold some portion of the positive examples, thus

[4] For example, the results between SVM and ridge regression at noise level 3% is significant at level 0.05 with Micro s-test [15], since SVM wins 211 out of 365 different decisions.

investigate the behaviors as the available amount of positive data changes. In order to make our results stable, results are averaged from 5 to 20 times (the less the percent of positive data used, the more times it is averaged over).

Results here are reported in terms of "best possible" $F1$ over test data, which avoids the tuning of thresholds. Figure 5 shows the $F1$ results of 12 common categories (as shown in table 1) as the available percent of positive data changes (1%, 3%, 5%, 10%, 30%, and 50%).

From the results we can see that though results vary from category to category, logistic regression is much worse than both SVM and ridge regression. Ridge regression performs slightly better than SVM for small categories, which further confirms our results in figure 3.

Note that for all three methods, when the amount of negative data is much larger than positive data, the objective function's value will be initially dominated by negative data. However, since logistic loss is *strictly* greater than 0 even for correctly classified data, the optimization process will keep pushing the majority (correctly classified negative data) further down the loss function as long as their role in the objective function is still larger than (or comparable to) that of rare positive data, thus sacrifice the performance of positive data, which will lead to poor $F1$ score. Both SVM loss and squared loss will have some point(s) that have exactly zero loss with finite $y\mathbf{w}^T\mathbf{x}$ value (contrast to logistic loss which goes to 0 as $y\mathbf{w}^T\mathbf{x}$ goes to ∞), and thus their performance for rare class will not drop as dramatic as logistic regression.

6. CONCLUDING REMARKS

In this paper we presented a controlled study on the robustness of three regularized linear classification methods in text categorization. We discussed their loss functions and related score distributions, as well as establishing the connection between their optimization targets and the generalization error bounds. In our experiments, we investigated their performance under conditions of small number of features, noisy settings and rare positive data. Their performance differences are compared and analyzed. Our concluding remarks are:

- Theoretically, all three methods can be treated as shrinking the hypothesis space when performing the optimization, thus they have similar generalization error bounds. Practically, they all perform very well in $Micro_{avg}F1$ even with very few selected features.
- Under noisy settings, SVM is better than logistic regression and ridge regression. Ridge regression, as indicated by its loss function, performs the worst.
- Ridge regression is better than SVM when only small number of positive examples are available. We show that logistic regression performs very badly in this case, as well as give explanations.

7. ACKNOWLEDGMENTS

We thank anonymous reviewers for their helpful comments. This research is sponsored in part by National Science Foundation (NSF) under the grants EIA-9873009 and IIS-9982226, and in part by DoD under the award 114008-N66001992891808. However, any opinions or conclusions in this paper are the authors' and do not necessarily reflect those of the sponsors.

8. REFERENCES

[1] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM'98*, 1998.

[2] G. Golub and C. V. Loan. *Matrix Computations (3rd edition)*. Johns Hopkins University Press, Baltimore, MD, 1996.

[3] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: Data mining, inference, and prediction. New York, 2001. Springer.

[4] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *23, Universit Dortmund, LSVIII-Report*, 1997.

[5] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *European Conference on Machine Learning (ECML)*, pages 137–142, Berlin, 1998. Springer.

[6] D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, Nevada, Las Vegas, 1994. University of Nevada, Las Vegas.

[7] D. Luenberger. *Linear and nonlinear programming.* Addison-Wesley, New York, 1989.

[8] D. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, New York, 1997.

[9] H. Schütze, D. Hull, and J. Pedersen. A comparison of classifiers and document representations for the routing problem. In *18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 229–237, 1995.

[10] A. N. Tikhonov. Solutions of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.*, volume 4, pages 1035–1038, 1963.

[11] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems.* W. H. Winston, Washington, D.C., 1977.

[12] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[13] V. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, New York, 1998.

[14] Y. Yang and C. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transaction on Information Systems (TOIS)*, 12(3):252–277, 1994.

[15] Y. Yang and X. Liu. A re-examination of text categorization methods. In *The 22th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 42–49, 1999.

[16] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In J. D. H. Fisher, editor, *The Fourteenth International Conference on Machine Learning (ICML'97)*, pages 412–420. Morgan Kaufmann, 1997.

[17] T. Zhang and F. J. Oles. Text categorization based on regularized linear classificatin methods. In *Information Retrieval*, volume 4, pages 5–31, 2001.

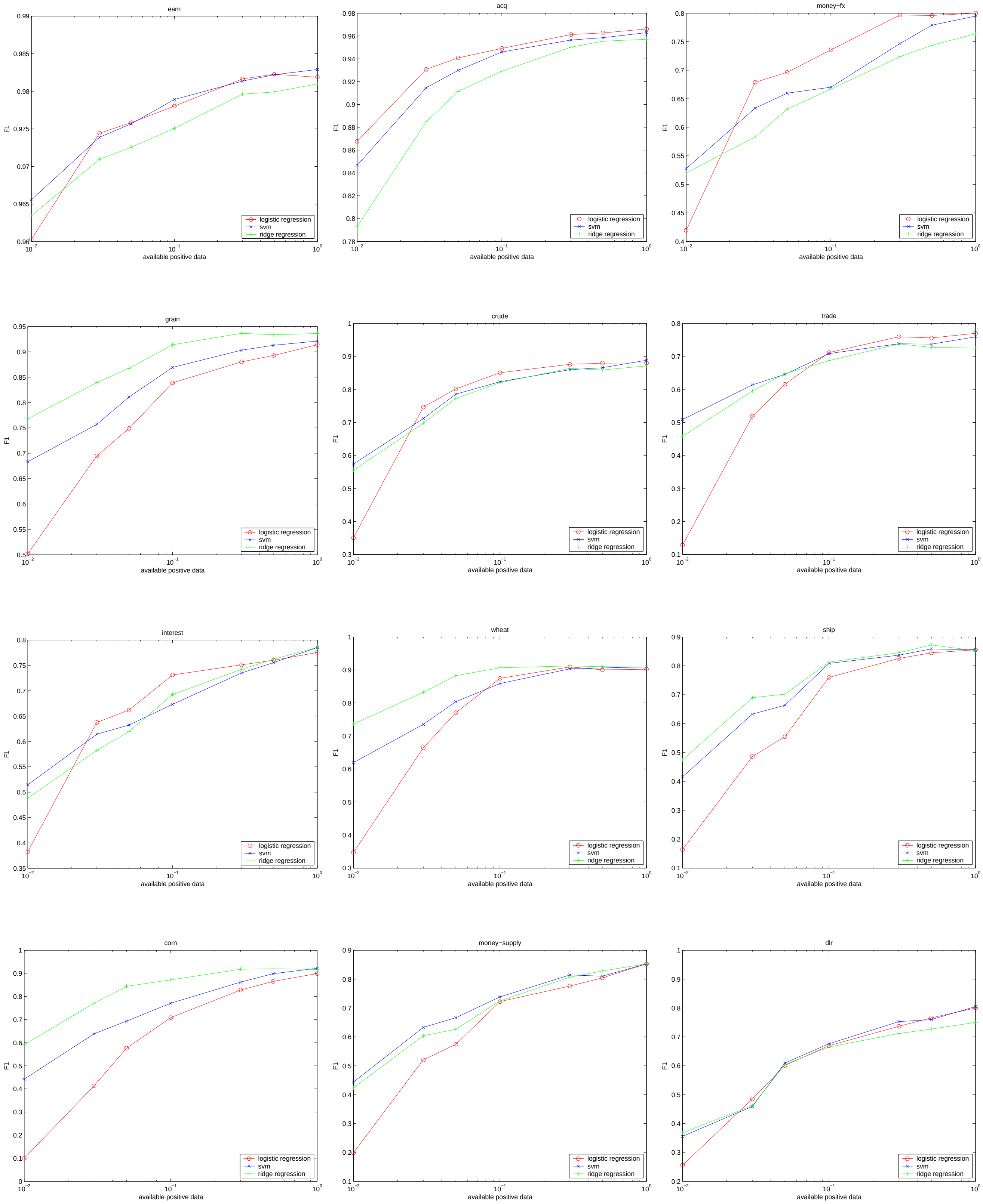

Figure 5: F1 vs. available amount of positive data (feature size=3000)