Modified Logistic Regression: An Approximation to SVM and Its Applications in Large-Scale Text Categorization

Jian Zhang JIAN.ZHANG@CS.CMU.EDU
Rong Jin RONG@CS.CMU.EDU
Yiming Yang YIMING@CS.CMU.EDU
Alex G. Hauptmann ALEX@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 USA

Abstract

Logistic Regression (LR) has been widely used in statistics for many years, and has received extensive study in machine learning community recently due to its close relations to Support Vector Machines (SVM) and AdaBoost. In this paper, we use a modified version of LR to approximate the optimization of SVM by a sequence of unconstrained optimization problems. We prove that our approximation will converge to SVM, and propose an iterative algorithm called "MLR-CG" which uses Conjugate Gradient as its inner loop. Multiclass version "MMLR-CG" is also obtained after simple modifications. We compare the MLR-CG with SVM^{light} over different text categorization collections, and show that our algorithm is much more efficient than SVM^{light} when the number of training examples is very large. Results of the multiclass version MMLR-CG is also reported.

1. Introduction

Logistic Regression is a traditional statistical tool, and gained popularity recently in machine learning due to its close relations to SVM (Vapnik, 1999) and AdaBoost (Freund, 1996), which can be called "large margin classifiers" since they pursue the concept of margin either explicitly or implicitly. Large margin classifiers are not only supported by theoretical analysis, but also proved to be promising in practice.

Vapnik (1999) compared LR and SVM in terms of minimizing loss functional, and showed that the loss function of LR can be very well approximated by SVM loss with multiple knots (SVM$_n$). Friedman et al. (1998) discussed SVM, LR and boosting on top of their different loss functions. Collins et al. (2000) gave a general framework of boosting and LR in which each learning problem is cast in terms of optimization of Bregman distances. Lebanon et al. (2002) showed that the only difference between AdaBoost and LR is that the latter requires the model to be normalized to a probability distribution.

Here we show that by simple modifications, LR can also be used to approximate SVM. Specifically, we use a modified version of LR to approximate SVM by introducing a sequence of smooth functions, which can be shown to converge uniformly to the SVM objective function. Based on that, we further prove that the sequence of their solutions will converge to the solution of SVM. As a result, we can use simple unconstrained convex optimization techniques to solve the SVM optimization problem. Specifically, we propose a new algorithm named "MLR-CG" that is more efficient than SVM^{light} in case of large-scale text categorization collections. We also show that the MLR-CG algorithm can be easily extended to its multiclass [1] version "MMLR-CG".

The rest of this paper is arranged as follows: Section 2 briefly reviews LR and SVM, and discusses some related works. Section 3 introduces the modified version of LR, proves the convergence of our approximation, and proposes the MLR-CG algorithm. Section 4 extends our algorithm to a multiclass version, which shows that the multiclass SVM can be achieved with a simple algorithm. Section 5 introduces our datasets, and section 6 reports the experimental results on several text categorization collections. Section 7 concludes.

[1] In this paper, "multiclass" means that each data instance belongs to one and only one class.

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

2. Reviews and Related Works

2.1. Logistic Regression

Logistic Regression can be applied to both real and binary responses, and its output posterior probabilities can be conveniently processed and fed to other systems. It tries to model the conditional probability of the class label give its observation:

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

where $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ is the data vector[2], m is the number of features, $y \in \{+1, -1\}$ is the class label. $\mathbf{w} = (w_1, w_2, \ldots, w_m)$ and b are the weight vector and intercept of the decision hyperplane respectively.

The Regularized LR is trained by computing

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

Note the Hessian matrix of the above objective function $O(\mathbf{w})$ is:

$$\mathbf{H} = \frac{d^2O(\mathbf{w})}{d\mathbf{w}d\mathbf{w}^T} = \frac{1}{n}\sum_{i=1}^{n}\frac{\exp(-y_i\mathbf{w}^T\mathbf{x}_i)}{(1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i))^2}\mathbf{x}_i\mathbf{x}_i^T+2\lambda\mathbf{I}$$

where $\mathbf{I}$ is the identity matrix. Since $\lambda > 0$, the above Hessian matrix is positive definite, which implies the *strict* convexity of the objective function of regularized LR, and thus the uniqueness and globalness of its solution (Luenberger, 1973).

2.2. SVM

Support Vector Machine is a new generation learning system based on Structural Risk Minimization instead of Empirical Risk Minimization (Vapnik, 1999). It is not only theoretically well-founded, but also practically effective. It achieves very good performance in many real-world applications including text categorization (Joachims, 1998a), which is our research focus here.

The primal form of Support Vector Machine with linear kernel can be described as follows:

$$\min\left\{C\sum_{i=1}^{n}\xi_i+\frac{1}{2}\mathbf{w}^T\mathbf{w}\right\}$$
$$subject\,to: y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1-\xi_i; \xi_i\geq 0$$

By using implicit constraints, we can transform the above objective function into:

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

[2]We use column vector by default.

where $\max\{0, 1-y_i(\mathbf{w}^T\mathbf{x}+b)\}$ can be thought as the SVM loss for one instance, and the second term $\lambda\mathbf{w}^T\mathbf{w}$ is the regularization term.

Due to the non-differentiable of its loss function, the fitting of SVM is usually solved in its dual form:

$$\max\left\{\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\mathbf{x}_i^T\mathbf{x}_j\right\}$$
$$subject\,to: 0\leq\alpha_i\leq C; \sum_{i=1}^{n}\alpha_i y_i = 0$$

2.3. Related Works

Our approach is similar in spirit to penalty algorithms in optimization literature, especially to the "Exponential-Penalty" algorithm proposed by Cominetti & Dussault (1994). The focus is to solve the constrained optimization problem with a sequence of unconstrained ones.

For a general optimization problem

$$\min_{x\in C} f(x)$$

where the feasible set C is defined in terms of functional inequalities

$$C = \{x \in R^n : c_i(x) \leq 0, i = 1, \ldots, m\},$$

the exponential penalty method tries to solve the unconstrained problems

$$\min\left\{f(x)+r_k\sum_{i=1}^{m}\exp[\frac{c_i(x)}{r_k}]\right\}$$

as $r_k \to 0$, and it has been shown that the sequence of the unconstrained solutions will converge to the solution of the original constrained problem under certain conditions. Applying their method to the SVM primal problem, we need to minimize the following functional:

$$\begin{aligned}E(r_k) &= C\sum_{i=1}^{n}\xi_i+\frac{1}{2}\mathbf{w}^T\mathbf{w}+r_k\sum_{i=1}^{n}\exp(-\frac{\xi_i}{r_k})\\ &+ r_k\sum_{i=1}^{n}\exp(\frac{1-y_i(\mathbf{w}^T\mathbf{x}_i+b)-\xi_i}{r_k})\end{aligned}$$

By setting $\frac{\partial E(r_k)}{\partial\xi_i} = 0$ (Rätsch et al., 2000), we will get similar formula as equation 1. However, as we will show later, our algorithm does not need those exponential penalty terms, which may cause overflow problem when $r_k \to 0$, as pointed out by Cominetti & Dussault. Though this issue can be addressed in some ad-hoc way, it is considered as one drawback of their method. To sum up, we will show that our algorithm is very simple and suitable for SVM-like optimization problems, and its convergence is guaranteed.

3. Modified Logistic Regression

In this section we first show that by constructing a sequence of optimization problems whose solutions converge to the solution of SVM. Thus, SVM can be solved with simple unconstrained optimization techniques. Then we propose our simple MLR-CG algorithm which uses CG as its inner loop.

In order to simplify our derivations, we use the augmented weight vector $\mathbf{w} = (b, w_1, w_2, \ldots, w_m)$ and augmented data vector $\mathbf{x} = (1, x_1, x_2, \ldots, x_m)$ from now on unless otherwise specified. To keep the SVM optimization problem unchanged, its form becomes

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\} + \lambda \sum_{j=1}^{m} w_j^2 \right\}$$

so that the intercept $w_0 = b$ is not included in the regularization term. We also intend not to penalize the intercept w_0 in the regularized LR to approximate SVM:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)) + \lambda \sum_{j=1}^{m} w_j^2 \right\}$$

3.1. Approximating SVM Loss Function

From previous discussions we can see that loss functions play an important role in the SVM and LR.

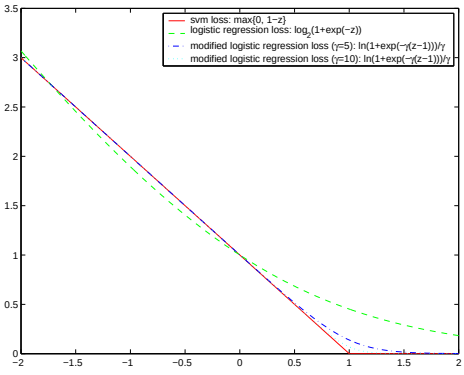

Figure 1. Approximation to SVM Loss ($z = y\mathbf{w}^T\mathbf{x}$)

Figure 1 shows the SVM loss function can be approximated by the loss of the following "modified LR" (Zhang, 2001):

$$g_\gamma(\mathbf{x}, y, \mathbf{w}) = \frac{1}{\gamma} \ln(1 + \exp(-\gamma(y\mathbf{w}^T\mathbf{x} - 1))) \qquad (1)$$

If we can approximate the SVM loss function

$$g_{svm}(\mathbf{x}, y, \mathbf{w}) = \max\{0, 1 - y\mathbf{w}^T\mathbf{x}\}$$

with the above sequence of functions $\{g_\gamma\}$, then the problem can be solved with simple unconstrained optimization techniques.

3.2. Convergence

Let

$$O_{svm}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} g_{svm}(\mathbf{x}_i, y_i, \mathbf{w}) + \lambda \sum_{j=1}^{m} w_j^2$$

$$O_\gamma(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} g_\gamma(\mathbf{x}_i, y_i, \mathbf{w}) + \lambda \sum_{j=1}^{m} w_j^2$$

represent optimization objective functions for SVM and modified LR respectively, we now prove that by computing the optimal solutions of $O_{\gamma=k}(\mathbf{w})$ ($k = 1, 2, \ldots$), we are able to find the solution of $O_{svm}(\mathbf{w})$.

In the following we use $g_k(\mathbf{x}, y, \mathbf{w})$ and $O_k(\mathbf{w})$ to denote $g_{\gamma=k}(\mathbf{x}, y, \mathbf{w})$ and $O_{\gamma=k}(\mathbf{w})$, and use $\hat{\mathbf{w}}_k$ and $\bar{\mathbf{w}}$ to denote optimal solutions of $O_{svm}(\mathbf{w})$ and $O_k(\mathbf{w})$ respectively.

Proposition 1 *The sequence of functions $\{g_k(\mathbf{x}, y, \mathbf{w})\}$ ($k = 1, 2, \ldots$) is monotonically decreasing and converges uniformly to the SVM loss $g_{svm}(\mathbf{x}, y, \mathbf{w})$; the sequence of functions $\{O_k(\mathbf{w})\}$ also monotonically converges to the SVM objective function $O_{svm}(\mathbf{w})$. Furthermore, for any $\gamma > 0$, we have*

$$\ln 2/\gamma = \max_{\mathbf{x},y,\mathbf{w}} \{g_\gamma(\mathbf{x}, y, \mathbf{w}) - g_{svm}(\mathbf{x}, y, \mathbf{w})\} \text{ and}$$

$$\ln 2/\gamma \geq \max_{\mathbf{w}} \{O_\gamma(\mathbf{w}) - O_{svm}(\mathbf{w})\}$$

The proof is provided in Appendix.

Theorem 2 *(1) Solutions of objective functions $\{O_k(\mathbf{w})\}$ are unique. (2) Suppose $\{\hat{\mathbf{w}}_k\}$ are the solutions of objective functions $\{O_k(\mathbf{w})\}$, then the sequence $\{O_k(\hat{\mathbf{w}}_k)\}$ converges to the minimum of objective function $O_{svm}(\mathbf{w})$.*

Sketch of Proof
(1) The Hessian matrix of the objective function $O_k(\mathbf{w})$ is:

$$\mathbf{H}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\gamma \exp(-\gamma(y_i \mathbf{w}^T \mathbf{x}_i - 1))}{(1 + \exp(-\gamma(y_i \mathbf{w}^T \mathbf{x}_i - 1)))^2} \mathbf{x}_i \mathbf{x}_i^T + 2\lambda \mathbf{I}^*$$

where $\mathbf{I}^*$ is the same as $(m+1) \times (m+1)$ identity matrix except that its element at the first row and first column is zero, which results from the non-regularized intercept w_0. For any given non-zero column vector $\mathbf{v} = (v_0, v_1, \ldots, v_m)$, it is easy to show that $\mathbf{v}^T\mathbf{H}\mathbf{v} > 0$ since $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{im})$ is an augmented vector

with the first element constant 1. Thus, the Hessian matrix is positive definite given $\lambda > 0$, which implies the strict convexity of the objective function $O_k(\mathbf{w})$. As a result, the solution will be both global and unique.

(2) Suppose $\bar{\mathbf{w}}$ is a solution of $O_{svm}(\mathbf{w})$, then by the uniform convergence of $\{O_k(\mathbf{w})\}$ to $O_{svm}(\mathbf{w})$ (Proposition 1) it directly follows that the sequence $\{O_k(\bar{\mathbf{w}})\}$ converges to $O_{svm}(\bar{\mathbf{w}})$. Since

$$O_k(\bar{\mathbf{w}}) \geq O_k(\hat{\mathbf{w}}_k) \geq O_{svm}(\hat{\mathbf{w}}_k) \geq O_{svm}(\bar{\mathbf{w}})$$

we conclude that

$$\lim_{k\to\infty} O_k(\hat{\mathbf{w}}_k) \leq \lim_{k\to\infty} O_k(\bar{\mathbf{w}}) = O_{svm}(\bar{\mathbf{w}}) \text{ and}$$

$$\lim_{k\to\infty} O_k(\hat{\mathbf{w}}_k) \geq O_{svm}(\bar{\mathbf{w}})$$

Thus $\lim_{k\to\infty} O_k(\hat{\mathbf{w}}_k) = O_{svm}(\bar{\mathbf{w}})$. ■

In the following theorem, we will use the unaugmented weight vector $\hat{\mathbf{w}}_k^u$ and $\bar{\mathbf{w}}^u$. That is, $\hat{\mathbf{w}}_k = (\hat{b}_k, \hat{\mathbf{w}}_k^u)$ and $\bar{\mathbf{w}} = (\bar{b}, \bar{\mathbf{w}}^u)$.

Theorem 3 *The (unaugmented) vector sequence $\{\hat{\mathbf{w}}_k^u\}$ converges to the unique $\bar{\mathbf{w}}^u$.*

Sketch of Proof [3]
*Suppose $\bar{\mathbf{w}}^u$ is the unaugmented part of **a** SVM solution. Given that the gradient of $O_k(\mathbf{w})$ at $\hat{\mathbf{w}}_k$ is zero, by the multivariate Taylor's Theorem, there is a $0 \leq \theta \leq 1$ such that*

$$\begin{aligned} & O_k(\bar{\mathbf{w}}) - O_k(\hat{\mathbf{w}}_k) \\ = \; & \frac{1}{2}(\bar{\mathbf{w}} - \hat{\mathbf{w}}_k)^T \mathbf{H}(\theta\hat{\mathbf{w}}_k + (1-\theta)\bar{\mathbf{w}})(\bar{\mathbf{w}} - \hat{\mathbf{w}}_k) \\ \geq \; & \frac{1}{2}(\bar{\mathbf{w}} - \hat{\mathbf{w}}_k)^T (2\lambda \mathbf{I}^*)(\bar{\mathbf{w}} - \hat{\mathbf{w}}_k) \\ = \; & \lambda(\bar{\mathbf{w}^u} - \hat{\mathbf{w}}_k^u)^T(\bar{\mathbf{w}}^u - \hat{\mathbf{w}}_k^u) \end{aligned}$$

If we take limit on both ends, we get

$$\lim_{k\to\infty} \|\bar{\mathbf{w}}^u - \hat{\mathbf{w}}_k^u\|^2 \leq \lim_{k\to\infty} \frac{(O_k(\bar{\mathbf{w}}) - O_k(\hat{\mathbf{w}}_k))}{\lambda} = 0$$

Thus $\lim_{k\to\infty} \hat{\mathbf{w}}_k^u = \bar{\mathbf{w}}^u$*, which also implies the uniqueness of* $\bar{\mathbf{w}}^u$. ■

Note the $\bar{b}$ of SVM solutions may not be unique, and Burges and Crisp (1999) gave more detailed discussions about the uniqueness of SVM solutions. We can see that the regularization coefficient λ also influences the convergence speed of our approximation.

[3] General results about the convergence of convex function's solutions can be found in Rochafellar (1970), such as theorem 27.2. Specifically, the minimum distance between $\hat{\mathbf{w}}_k$ and the set of SVM solutions (which is convex) converges to zero.

3.3. MLR-CG Algorithm

Our MLR-CG algorithm exactly follows the above convergence proof. That is, we compute the solution of SVM by solving a sequence of sub-optimization problems. In particular, we use the Conjugate Gradient (CG) to solve each sub-optimization problem $O_\gamma(\mathbf{w})$.

CG (Nocedal & Wright, 1999) is one of the most popular methods for solving large-scale nonlinear optimization problems. More importantly, Minka (2001) compared it with other methods in fitting LR, and found that it is more efficient than others.

Three best known conjugate directions are Fletcher-Reeves (FR), Polak-Rbiere (PR), and Hestenes-Stiefel (HS). In our experiment we found that HS direction is more efficient than the other two directions.

We list our MLR-CG algorithm below, which is an iterative algorithm with CG as its inner loop.

Algorithm 1 : MLR-CG

1. Start with $\mathbf{w} = \mathbf{0}$, $\gamma = 1.0$, $l = 10$ and $\delta = 10$
2. Repeat until convergence:
   (a) (Re-)Initialize CG by setting its search direction to minus gradient
   (b) Minimize O_γ with l steps CG
   (c) Increase $\gamma \leftarrow \gamma + \delta$

In practice, we should start from small γ and do not increase γ to infinity for a couple of reasons. One reason is that when γ is big the Hessian matrix is ill-conditioned, which will lead to the unsuitability of our algorithm. Starting from small γ and increase it gradually will lead to stable solution. Another reason is that, based on Proposition 1, $|O_\gamma(\mathbf{w}) - O_{svm}(\mathbf{w})| \leq \frac{\ln 2}{\gamma}$ for any $\mathbf{w}$. So we have $|O_\gamma(\hat{\mathbf{w}}_\gamma) - O_{svm}(\bar{\mathbf{w}})| \leq |O_\gamma(\bar{\mathbf{w}}) - O_{svm}(\bar{\mathbf{w}})| \leq \frac{\ln 2}{\gamma}$. For example, when $\gamma = 200$, it is at most 0.003, which is already not influential for our problems. And later we will show in our experiments that this approximation will not degrade the performance of our trained classifier. We do not increase γ after each CG step because we should let CG run at least several steps to fully utilize its power in finding conjugate directions; and also we do not need to wait until the $O_\gamma(.)$ converged before we change γ. In our experiments, we set both δ and l to be 10. And every time when γ is changed, CG should be re-initialized.

In our experiments, we use 200 CG steps (that is, $200/l = 20$ iterations for the outer loop) as the stopping criteria. Other criteria like the change of weight

vector or objective value can also be used.

4. Multiclass Version

Real world applications often require the classification of $C > 2$ classes. One way is to consider the multiclass classification problem as a set of binary classification problem, using 1-versus-rest method (or more complicated, like ECOC) to construct C classifiers. Alternatively, we can construct a C-way classifier directly by choosing the appropriate model. The latter is usually regarded as more natural, and it can be solved within a single optimization. For SVM, there are cases (Weston & Watkins, 1999) where multiclass SVM can perfectly classify the training data, while the 1-versus-rest method cannot classify without error.

The multiclass SVM (Weston & Watkins, 1999; Vapnik, 1998) tries to model the C-way classification problem directly with the same margin concept. For the kth class, it tries to construct a linear function $f_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x}$ so that for a given data vector $\mathbf{x}$ the prediction is made by choosing the class corresponding to the maximal value of functions $\{f_k(\mathbf{x})\}$: $y = \arg\max_k\{f_k(\mathbf{x})\}$, (k=1,2,...,C).

Follow the notations by Weston & Watkins, we can get the primal form of multiclass SVM as follows:

$$\min\left\{C\sum_{i=1}^{n}\sum_{k\neq y_i}\xi_i^k + \frac{1}{2}\sum_{k=1}^{C}\sum_{j=1}^{m}(w_{k,j})^2\right\}$$

$$\begin{aligned} subject\,to: \quad & \mathbf{w}_{y_i}^T\mathbf{x}_i \geq \mathbf{w}_k^T\mathbf{x}_i + 2 - \xi_i^k \\ & \xi_i^k \geq 0 (i = 1, 2, \ldots, n; k \neq y_i) \end{aligned}$$

And as we did before, it can be transformed into

$$\begin{aligned} O_{m-svm} &= \frac{1}{n}\sum_{i=1}^{n}\sum_{k\neq y_i}\max\{0, 2 - (\mathbf{w}_{y_i}^T\mathbf{x}_i - \mathbf{w}_k^T\mathbf{x}_i)\} \\ &+ \lambda\sum_{k=1}^{C}\sum_{j=1}^{m}(w_{k,j})^2 \end{aligned}$$

Similar to what we did in Section 3, we use the following to approximate SVM:

$$\begin{aligned} O_{m-\gamma} &= \lambda\sum_{k=1}^{C}\sum_{j=1}^{m}(w_{k,j})^2 + \frac{1}{n}\sum_{i=1}^{n}\sum_{k\neq y_i}\frac{1}{\gamma} \\ &\times \ln(1 + \exp(-\gamma((\mathbf{w}_{y_i}^T\mathbf{x} - \mathbf{w}_k^T\mathbf{x}) - 2))) \end{aligned}$$

It can be shown that the above objective function is convex w.r.t. its variable $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_C) \in R^{C(m+1)\times 1}$. It is not strict convex, i.e., if a constant is added to the intercepts of all classes, its value will not be changed. This is also true for the multiclass SVM. However, Theorem 2.2 and 3 still hold, and thus our MLR-CG algorithm is extended to its multiclass version "MMLR-CG".

5. Datasets and Preprocessing

Our research focus here is the task of text categorization. For binary classification, We use two benchmark datasets in text categorization, namely the Reuters-21578 [4] (ModApte split) and the Reuters Corpus Volume I (RCV1) (Lewis et al., 2003) in our experiments. The size of training set and test set of the former are 7,769 and 3,019 respectively; while the latter is an archive of over 800,000 documents. In order to examine how efficient MLR-CG is in case of very large collections, we use the same split as Lewis did, but exchange the training set and test set. So our resulting collection contains 781,265 training documents and 23,149 test documents. For the Reuters-21578 collection, we report the training time and test errors of four categories ("earn", "acq", "money-fx" and "grain"), and F1 performance of the whole collection. For the RCV1 we choose the highest four topic codes (CCAT, ECAT, GCAT, and MCAT) in the "Topic Codes" hierarchy and report their training time and test errors.

For multiclass classification, we use three webpage collections (see Yang et al., 2002 for details about those collections), namely Hoovers28, Hoovers255 and Univ6 (aka WebKB). We only use the words appeared on pages, which correspond to the "Page Only" in Yang's work. For all three collections, we split them into 5 equal-sized subsets, and every time 4 of them are used as training set and the remaining one as test set. Our results in table 3 are the combination of the five runs.

Descriptions of all datasets are listed in table 1. We use binary-valued features for all datasets mentioned above, and 500 features are selected with Information Gain. The regularization coefficient λ is choosen from cross validation, and we use $\lambda = 0.001$ for all algorithms. Note that for SVM^{light}, its parameter $c = \frac{1}{2\lambda n}$.

6. Experimental Results

In this section we mainly examine two things for text categorization tasks: First, how efficient is the MLR-CG algorithm compared with the existing SVM algorithm; Second, whether our resulting model (with finite γ) performs similarly to SVM.

[4] A formatted version of this collection is electronically available at http://moscow.mt.cs.cmu.edu:8081/reuters_21578/.

Table 1. Dataset Descriptions

Collection	# of training examples	# of test examples	# of classes used	# of features
Reuters-21578	7,769	3,019	4 & 90 (binary)	500
RCV1	781,265	23,149	4 (binary)	500
Univ-6	4,165	4,165	6 (multiclass)	500
Hoovers-28	4,285	4,285	28 (multiclass)	500
Hoovers-255	4,285	4,285	255 (multiclass)	500

Table 2. Reuters-21578 and RCV1 Datasets: Training Time & Test Error Rate

Class	SVM^{light} Training	MLR-CG Training (200 CG steps)	SVM^{light} Test Error	MLR-CG Test Error
earn	0.88 sec	7 sec (795%)	1.66%	1.72% (103%)
acq	0.45 sec	6 sec (1333%)	2.45%	2.42% (98.7%)
money-fx	0.33 sec	6 sec (1818%)	3.01%	3.01% (100%)
grain	0.24 sec	5 sec (2083%)	0.762%	0.762% (100%)
CCAT	24,662 sec	3,696 sec (14.9%)	13.63%	13.62% (99.9%)
ECAT	13,953 sec	2,136 sec (15.3%)	8.92%	8.93% (100.1%)
GCAT*	2,356 sec	2,602 sec (110%)	29.01%	7.15% (24.6%)
MCAT	16,131 sec	3,244 sec (20.1%)	8.07%	8.07% (100%)
Note: SVM^{light} stops its training for GCAT class due to some error, which is also reflected by its abnormal test error.				

For binary text categorization, we compare our MLR-CG algorithm with the SVM^{light} 5.00 package[5] since it is one of the most popular and efficient SVM implementation used in text categorization, and it also supports the sparse matrix representation, which is an advantage of processing text collections. Besides, previous studies (Joachims, 1998b) show that algorithms like SMO (Platt, 1999) appears to share the similar scaling pattern with SVM^{light} in terms of number of training examples.

All our algorithms are implemented with C++, STL data structure "vector¡double¿", which is about two times slower in our setting than standard C array operations, as is done in SVM^{light}. For multiclass text categorization, we report the results of the MMLR-CG algorithm over those webpage collections.

Our machine is Pentium 4 Xeon with 2GB RAM and 200GB harddisk, Redhat Linux 7.0.

6.1. Binary Classification

Here we mainly compare the training time and test errors between our MLR-CG and SVM^{light} over the eight categories of two different text collections. Since during the training phase both algorithms use the objective O_{svm} as their final goals, we plot the $O_{svm}(\mathbf{w}_t)$ versus training time t of our algorithm MLR-CG in figure 2. From the graph we can see that our MLR-CG converges very fast to its final goal. In figure 2 we also plot the final objective value of SVM^{light} as a straight line[6] , and its actual training time can be found in table 2.

From table 2 we can infer that SVM^{light} is very efficient compared with our MLR-CG algorithm when the size of training set is limited, like Reuters-21578. However, our MLR-CG algorithm catches up SVM^{light} when the training set becomes very large, like RCV1. Though our algorithm is much slower than SVM^{light} in the former case, its training time is still the same magnitude as needed by preprocessing (tokenization, feature selection, etc). Their classification error rates (over eight categories) are also reported, which further show that 200 CG steps are enough for our text collections.

Besides, we run SVM and MLR-CG on the whole Reuter-21578 collection, and our results are[7] : For SVM, Micro-F1 and Macro-F1 are 87.21% and 56.74% respectively; for MLR-CG, Micro-F1 and Macro-F1 are 87.17% and 56.89%. Our SVM results are comparable to previous ones (Yang & Liu, 1999).

6.2. Multiclass Classification

We report the performance of the MMLR-CG algorithm over those multiclass webpage collections in table 3. In particular, for collection with small number of categories, MMLR-CG performs well, while for collections with more categories it overfits poorly, since we

[5]http://svmlight.joachims.org.

[6]We did not plot the O_{svm} of SVM^{light} mainly because it is not always decreasing, and sometimes fluctuates a lot.

[7]Thresholds are tuned by 5-fold cross-validation.

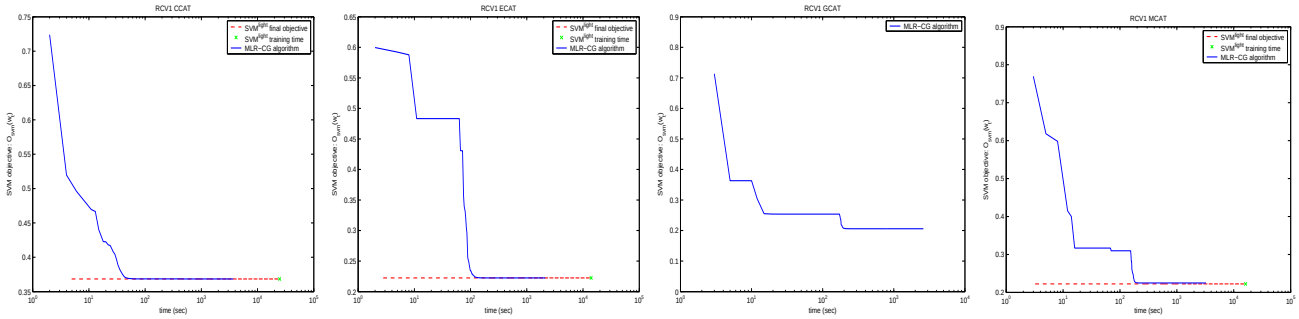

Figure 2. Convergence of MLR-CG algorithm over RCV1 collection in 200 CG steps (We only plot the training time of MLR-CG for category GCAT due to SVM^{light}'s failure on that category.)

observed that its training Micro-F1 and Macro-F1 are both above 90% even for Hoovers-255. We also report the results of SVM^{light} using 1-versus-rest method in table 3 under the same condition as the MMLR-CG algorithm.

Table 3. Performance of MMLR-CG and SVM^{light} with 1-versus-rest

	$(Micro-F1, Macro-F1)$	
Collection	MMLR-CG	SVM
Univ-6	(86.04%, 57.27%)	(85.84%, 57.56%)
Hoovers-28	(46.58%, 44.26%)	(44.02%, 42.59%)
Hoovers-255	(16.91%, 11.08%)	(19.57%, 12.36%)

7. Concluding Remarks

In this paper we use a modified version of LR to approximate the optimization of SVM, and prove that it will converge to SVM. Meanwhile, we propose an iterative algorithm called "MLR-CG" which is very simple and robust. We compare the MLR-CG algorithm with the algorithm implemented in SVM^{light} on several text categorization collections, and our results show that our algorithm is particularly efficient when the training set is very large, and it can have very close objective values as obtained by SVM^{light}. Our method can be easily extended to multiclass version, and the resulting MMLR-CG is tested over several webpage collections.

The time complexity of one step CG is $O(nm)$ (n and m are the numbers of training examples and features respectively). Theoretically, it is hard to get the time complexity of both MLR-CG and SVM^{light}. But empirically, if the number of CG steps is constant (which is true for all our experiments), the total time complexity of our MLR-CG algorithm will also be $O(nm)$. For SVM^{light} the number of iterations vary quite a lot, and empirically it has been shown to be super linear in terms of number of training examples n (Joachims, 1998b).

We should point out that since the dual form of LR does exist, and our methods can be modified to nonlinear kernel version. We did not investigate it in this paper since linear kernel SVM is one of the state-of-the-art classifiers in text categorization, and has been shown to be at least as effective as other kernel SVMs for this task.

To sum up, the MLR-CG algorithm is very efficient in case of very large training collection, which makes the training of millions of documents (like Yahoo! webpages) more applicable. Further investigations are needed to explore the application of multiclass SVM to text categorization.

Acknowledgements

We thank Reuters Ltd. for making Reuters Corpus Volume 1 (RCV1) available. We also thank Reha Tütüncü and Victor J. Mizel for helpful discussions, and anonymous reviewers for pointing out related works. This research is sponsored in part by 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.

References

Burges, C., & Crisp, D. (1999) Uniqueness of the SVM Solution. *NIPS 1999*

Collins, M., Schapire, R., & Singer, Y. (2000). Logistic regression, AdaBoost, and Bregman distances. *Proc. 13th COLT.*

Cominetti, R., & Dussault J.P. (1994). Stable Exponential-Penalty Algorithm with Superlinear Convergence. *Journal of Optimization Theory and*

Applications, 83(2).

Freund, Y., & Schapire, R.E. (1996). Experiments with a new boosting algorithm. *International Conference on Machine Learning.*

Friedman, J., Hastie, T., & Tibshirani, R. (1998). Additive logistic regression: a statistical view of boosting. *Dept. of Statistics, Stanford University Technical Report.*

Joachims, T. (1998a). Text categorization with support vector machines: Learning with many relevant features. *Proceedings of ECML.*

Joachims, T. (1998b). Making Large-Scale SVM Learning Practical. *Advances in Kernel Methods: Support Vector Machines.*

Lebanon, G., & Lafferty, J. (2002). Boosting and Maximum Likelihood for Exponential Models. *Proc. NIPS 14.*

Lewis, D., Yang, Y., Rose, T., & Li, F. (2003, to appear). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research.*

Luenberger, D. (1973). *Introduction to Linear and Nonlinear Programming.* Addison Wesley, Massachusetts.

Minka, T. (2001). Algorithms for maximum-likelihood logistic regression. *Carnegie Mellon University, Statistics Technical Report 758.*

Nocedal, J., & Wright, S. (1999). *Numerical Optimization.* Springer, New York.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning*, 185-208, Cambridge, MA, MIT Press.

Rätsch, G., Schölkopf, B., Mika, S., & Müller K.-R. (2000). SVM and Boosting: One Class. *Technical Report 119, GMD FIRST, Berlin, November 2000.*

Rockafellar, R. (1970). *Convex Analysis*, Princeton University Press.

Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley & Sons, New York.

Vapnik, V. (1999). *The Nature of Statistical Learning Theory, 2nd edition.* Springer Verlag.

Weston, J., & Watkins, C. (1999) Support Vector Machines for Multi-Class Pattern Classification. *Proceedings of the Seventh European Symposium on Artificial Neural Networks.*

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *SIGIR'99.*

Yang, Y., Slattery, S., & Ghani, R. (2002). A Study of Approaches to Hypertext Categorization. *Journal of Intelligent Information Systems*, vol 18:2, 2002.

Zhang, T., & Oles, F. (2001). Text Categorization based on regularized linear classification methods. *Information Retrieval*, vol 4:5-31.

Appendix

Proof of Proposition 1:
For any given $\mathbf{x}, y, \mathbf{w}$*, let* $z = y\mathbf{w}^T\mathbf{x}$*. First, we prove that* $g_\gamma(z) \geq g_{svm}(z), \forall z, \gamma$*. Since*

$$\begin{aligned} g_\gamma(z) &\geq 0 \quad \textit{and} \\ g_\gamma(z) &= \frac{1}{\gamma}\ln(1+\exp(-\gamma(z-1))) \\ &\geq \frac{1}{\gamma}\ln(\exp(-\gamma(z-1))) \\ &= 1-z \end{aligned}$$

Thus, $g_\gamma(z) \geq \max\{0, 1-z\} = g_{svm}(z)$.
Second, for any fixed $z \leq 1$

$$\begin{aligned} \frac{\partial g_\gamma}{\partial \gamma} &= -\frac{1}{\gamma^2}\ln(1+\exp(-\gamma(z-1))) \\ &+ \frac{1}{\gamma}\frac{(1-z)\exp(-\gamma(z-1))}{1+\exp(-\gamma(z-1))} \\ &< -\frac{1}{\gamma^2}(-\gamma(z-1)) + \frac{1}{\gamma}\frac{(1-z)\exp(-\gamma(z-1))}{1+\exp(-\gamma(z-1))} \\ &\leq 0 \end{aligned}$$

and for any fixed $z > 1$

$$\begin{aligned} \frac{\partial g_\gamma}{\partial \gamma} &= -\frac{1}{\gamma^2}\ln(1+\exp(-\gamma(z-1))) \\ &+ \frac{1}{\gamma}\frac{(1-z)\exp(-\gamma(z-1))}{1+\exp(-\gamma(z-1))} \\ &< 0 \end{aligned}$$

Then we conclude that the set of functions $\{g_k(z)\}$ *(*$k = 1, 2, \ldots$*) is monotonically decreasing.*
Finally, it is not hard to show (for $z \leq 1$ *and* $z > 1$*) that* $\frac{\ln 2}{\gamma} = \max_z \{g_\gamma(z) - g_{svm}(z)\}$*. Thus, for any given* $\epsilon > 0$*, we can make* γ *sufficiently large (independent on* z*) such that* $|g_\gamma(z) - g_{svm}(z)| < \epsilon$*. This finishes the uniform convergence proof. If we sum over all data points, we will get the inequality* $\ln 2/\gamma \geq \max_{\mathbf{w}}\{O_\gamma(\mathbf{w}) - O_{svm}(\mathbf{w})\}$*, which again implies the uniform convergence of* $O_k(.)$ *to* $O_{svm}(.)$*.* ■