L_1 AND L_2 REGULARIZATION FOR MULTICLASS HINGE LOSS MODELS

Robert C. Moore and John DeNero

Google Research

ABSTRACT

This paper investigates the relationship between the loss function, the type of regularization, and the resulting model sparsity of discriminatively-trained multiclass linear models. The effects on sparsity of optimizing log loss are straightforward: L_2 regularization produces very dense models while L_1 regularization produces much sparser models. However, optimizing hinge loss yields more nuanced behavior. We give experimental evidence and theoretical arguments that, for a class of problems that arises frequently in natural-language processing, both L_1- and L_2-regularized hinge loss lead to sparser models than L_2-regularized log loss, but less sparse models than L_1-regularized log loss. Furthermore, we give evidence and arguments that for models with only indicator features, there is a critical threshold on the weight of the regularizer below which L_1- and L_2-regularized hinge loss tends to produce models of similar sparsity.

Index Terms— regularization, hinge loss, support vector machines, SVMs, sparsity

1. INTRODUCTION

In this paper, we offer some observations concerning the relationship between model sparsity and the degree and form of regularization, for linear models trained by optimizing L_1- and L_2-regularized hinge loss. We describe the simple mathematical properties of the regularization and loss functions that govern this relationship, and present results of experiments on a typical natural language processing classification problem that demonstrate these properties empirically.

By model *sparsity* (*density*), we mean the proportion of feature weights in a statistical model that are zero (nonzero). In statistical natural language processing, we frequently train models over very large feature spaces; often the number of possible feature weights is much greater than the number of training examples. In such situations, model sparsity is highly desirable. The smaller the number of nonzero feature weights, the easier the model is to store, and the faster it is to apply.

Before starting to experiment with regularized hinge loss, we had expected that L_1 regularization would result in quite sparse models and L_2 regularization would result in very dense models, based on experience of the research community with L_1 and L_2 regularization of log loss for log-linear probabilistic models (e.g., [1]). However, when we compared L_1- and L_2-regularized hinge loss, the results surprised us. We found that, after optimizing the weight of the regularization penalty on development data, the sparsities of the models resulting from L_1 and L_2 regularization were remarkably similar. We then compared those results to the sparsities of models obtained by L_1 and L_2 regularization of log loss and found that the hinge-loss-based models were much sparser than those based on L_2-regularized log loss, and denser than those based on L_1-regularized log loss. Moreover, we found that for hinge loss there was a critical threshold of the regularization weight, below which the choice of regularizer had relatively little effect on model sparsity.

2. DEFINITIONS

We define a multiclass classification problem by a label set $\mathcal{L}$ and a feature set $\mathcal{F}$. For simplicity, we restrict our discussion to indicator features. Each training example consists of a set of features $\mathbf{f} \subseteq \mathcal{F}$ and a correct label $l \in \mathcal{L}$. A linear model is a vector $\mathbf{w} \in \Re^{|\mathcal{L}|\times|\mathcal{F}|}$ indexed by feature-label pairs. We refer to the coordinates $w_{(f,l)}$ of $\mathbf{w}$ as *feature weights*. A model $\mathbf{w}$ maximizes the sum of relevant feature weights to predict a label $l(\mathbf{f}, \mathbf{w})$:

$$l(\mathbf{f}, \mathbf{w}) = \arg\max_{l \in \mathcal{L}} \sum_{f \in \mathbf{f}} w_{(f,l)}$$

We learn a model $\mathbf{w}$ by minimizing a regularized loss function over a training set T

$$\mathbf{w} = \arg\min_{\mathbf{w}'} \sum_{(\mathbf{f},l) \in T} \ell(\mathbf{w}'; (\mathbf{f}, l)) \; + \; C\, r(\mathbf{w}')$$

where ℓ is the loss function, r is the regularizer, and C is a regularization weight that controls the trade-off between minimizing loss on the training data and regularization.

In the experiments described here, one of the loss functions we consider is the version of multiclass hinge loss introduced by Crammer and Singer [2]. For a linear model, the hinge loss on a single example can be expressed as

$$\ell_h(\mathbf{w}; (\mathbf{f}, l)) = \max\left(0,\, 1 + \max_{l' \neq l} \sum_{f \in \mathbf{f}} w_{(f,l')} - \sum_{f \in \mathbf{f}} w_{(f,l)}\right)$$

Hinge loss is 0 if the score of the correct label exceeds the score of every other label by by a margin of at least 1. Otherwise, it is the amount by which the score of the correct label falls short of exceeding the score of every other label by a margin of at least 1.

The other loss function we consider is log loss

$$\ell_l(\mathbf{w}; (\mathbf{f}, l)) = -\log \mathrm{P}_{\mathbf{w}}(l|\mathbf{f})$$

where

$$\mathrm{P}_{\mathbf{w}}(l|\mathbf{f}) = \frac{\exp \sum_{f \in \mathbf{f}} w_{(f,l)}}{\sum_{l' \in \mathcal{L}} \exp \sum_{f \in \mathbf{f}} w_{(f,l')}}$$

Log loss is the negative of the logarithm of the probability of the correct label given the features, according to an exponential model.

The two regularizers we consider are the L_1 regularizer

$$r_1(\mathbf{w}) = \sum_{f \in \mathcal{F}} \sum_{l \in \mathcal{L}} |w_{(f,l)}|$$

and the L_2 regularizer

$$r_2(\mathbf{w}) = \sum_{f \in \mathcal{F}} \sum_{l \in \mathcal{L}} w_{(f,l)}^2$$

which is actually the square of the L_2 norm of $\mathbf{w}$.

3. TASK AND DATA

In our experiments, we address the task of part-of-speech (POS) tagging by independent classifiers. That is, tagging each token is treated as an independent multi-class classification problem, in which observable features of neighboring tokens may be used in the classifier, but decisions about how neighboring tokens are tagged are not used. We chose this task because it is a multiclass classification task with clear relevance to NLP, and it has previously been shown that POS tagging by independent classifiers can perform nearly as well (96.8% accuracy [5]) as the state of the art for sequence models (97.33% accuracy [10]). [11] also reports very competitive results on this task using independent classifiers (96.57% accuracy).

For data, we use the usual Wall Street Journal (WSJ) corpus from Penn Treebank III [6], including what has come to be the standard split (for POS tagging) into training (sections 0–18), development (sections 19–21), and test (sections 22-24) data sets.

Our feature space consists of the following indicator functions, which express various aspects of the orthography and frequency of the tokens in the training data:

- Contains a digit
- Contains a hyphen
- Contains an upper-case character
- Lower-cased frequency class
- Original-case frequency class
- Lower-cased prefixes up to 4 characters
- Lower-cased suffixes up to 4 characters
- Lower-cased token
- Original-case token

For many possible features derived from orthography or frequency, we face the question of whether to normalize capitalization before feature extraction (e.g., by lower-casing all tokens). We resolve this by having a version of every feature derived from lower-casing each token, and an additional version of some features derived from the original casing of the token, if that differs from the lower-cased form. That is, "bush" and "Bush" will share a set of features derived from their lower-cased forms, but "Bush" will have some additional features, based on its original case.

The frequency class features have values equal to the $\log_2$ of the observed token count (lower-cased, or original case) in the training set, rounded up to the next integer, if not already an integer. Unknown tokens in unseen data are assigned the same frequency classes as singletons in the training set.

The features above are further annotated as belonging to the token being tagged, the preceding token, or the following token. We added two additional features for "follows sentence boundary," and "precedes sentence boundary." These feature were then multiplied by the 45 Penn Treebank POS tags to produce a set of 11,391,660 possible feature weights.

4. LEARNING ALGORITHMS

We trained four classifiers for the POS-tagging task, optimizing L_1-regularized hinge loss, L_2-regularized hinge loss, L_1-regularized log loss, and L_2-regularized log loss, tuning the regularization weight C on the development data to obtain the highest tagging accuracy for each objective.

To optimize L_1-regularized hinge loss, we used an iterative line search method we are currently developing that leverages the fact that L_1-regularized hinge loss is piecewise linear. To optimize L_2-regularized hinge loss we used the toolkit *SVM*^{multiclass} [3, 4], which applies a cutting plane method, incrementally adding constraints until a desired degree of convergence is reached. L_1- and L_2-regularized log loss were optimized using the MALLET toolkit [7]. MALLET employs the L-BFGS quasi-Newton method [8] to optimize the L_2-regularized log loss objective, and an orthant-wise version of L-BFGS [1] to optimize L_1-regularized log loss.

5. EXPERIMENTAL RESULTS

For each of the four classifiers we trained, Table 1 shows the value of the regularization weight C[1] that maximizes

[1]The regularization weight actually used by *SVM*^{multiclass} is combined with per-example-loss, so we have rescaled values of C for this algorithm to be comparable to those for the total-corpus-loss used by the other algo-

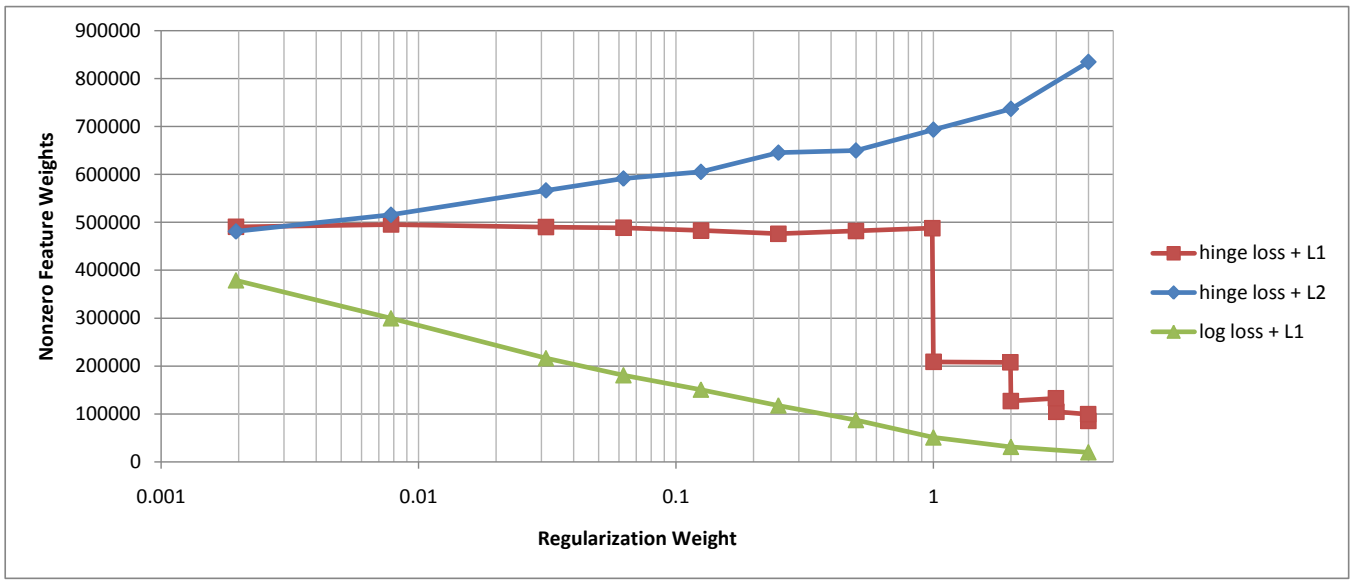

Fig. 1. Nonzero feature weight count vs. regularization weight C

Objective	C	Nonzero weights	Accuracy
hinge loss + L_1	0.125	482,933	96.85%
hinge loss + L_2	0.125	605,347	96.92%
log loss + L_1	1.000	50,864	96.79%
log loss + L_2	1.000	11,391,705	96.84%

Table 1. Model size and test set accuracy for different objectives

development set accuracy, along with the resulting number of nonzero feature weights and test set tagging accuracy. The accuracies of the four models are fairly close, but the model sparsities differ dramatically. The model optimizing L_2-regularized log loss is maximally dense; every possible feature weight[2] receives a nonzero value. The model optimizing L_1-regularized log loss is extremely sparse, with nonzero values for only 0.45% of the possible feature weights. The numbers of nonzero feature weights for the two hinge loss models lie in between, with approximately the same number of nonzero feature weights for either L_1 or L_2 regularization (4.2% and 5.3% of the possible feature weights).

Figure 1 shows how the number of nonzero feature weights varies with the regularization weight C, for L_1- and L_2-regularized hinge loss and for L_1-regularized log loss. For L_2-regularized log loss, the models were maximally dense for all values of C and are not included in Figure 1. It seems striking that the model sparsity for L_1-regularized hinge loss changes significantly only at integer values of C, and is remarkably close to the model sparsity for L_2-regularized hinge loss when $C < 1$.

rithms.

[2]The MALLET tooklkit used to optimize log loss adds a bias weight for each label, yielding an additional 45 weights.

6. ANALYSIS

Some of the questions raised by our results are:

- Why are the models for L_2-regularized hinge loss relatively sparse, rather than dense like the models for L_2-regularized log loss?
- Why are the models for L_1- and L_2-regularized hinge loss so similar in sparsity for $C < 1$?
- Why does model sparsity for L_1-regularized hinge loss change so abruptly at integer values of C?

These questions can be answered by considering the behavior of the multiclass hinge loss objective. Unregularized hinge loss is continuous, convex, and piecewise linear. Due to piecewise linearity, as we vary a feature weight $w_{(f,l)}$, holding other feature weights fixed, unregularized hinge loss ℓ_h may have a minimum, not just at a single point, but over an extended region where $\partial \ell_h / \partial w_{(f,l)} = 0$. If the minimum of regularized hinge loss and the point where $w_{(f,l)} = 0$ both fall in the region where $\partial \ell_h / \partial w_{(f,l)} = 0$, then regularized hinge loss must be minimal at the point where $w_{(f,l)} = 0$. Otherwise, we could reduce regularized hinge loss by setting $w_{(f,l)} = 0$, because the regularization penalty would be reduced, but the unregularized hinge loss would not change.

From inspection of the definition of unregularized multiclass hinge loss, we can see that $\partial \ell_h / \partial w_{(f,l)} = 0$ for a feature weight $w_{(f,l)}$ with respect to a model $\mathbf{w}$, if (but not only if), for every training example $(\mathbf{f}, l')$ one of the following holds:

1. $f \notin \mathbf{f}$,

2. $(\mathbf{f}, l')$ is correctly classified by a margin > 1, or

3. l is neither the correct label l' nor the incorrect label with the highest model score.

If $\mathbf{w}$ is a highly accurate model with low hinge loss, there may be many such feature weights, because most examples will be correctly classified by a margin > 1, and the other two conditions will capture many other feature weights on the small number of examples that are not correctly classified by a margin > 1. For such a feature weight $w_{(f,l)}$ that regularization has kept close to 0, the point where $w_{(f,l)} = 0$ may fall in the region where $\partial \ell_h / \partial w_{(f,l)} = 0$, in which case $w_{(f,l)} = 0$ will hold at the optimium.

This pattern is consistent with the standard analysis of hinge loss classifiers in terms of support vectors. Considering the training set as a whole, we know that an optimized SVM will tend to select only a few support vectors from the training set. Moreover, many features are associated only with a handful of training examples. If none of those examples are chosen as support vectors (i.e., condition 2 holds for all such examples), then the corresponding feature weight will be 0.

We can see empirically that this situation arises very frequently with our data and feature set. Examining the best models we found by optimizing L_1-regularized hinge loss, we observe that for more than 95% of the possible feature weights, $w_{(f,l)} = 0$ and $\partial \ell_h / \partial w_{(f,l)} = 0$, because at least one of the conditions listed above is met for every training example.

This source of sparsity is independent of the fact that all our features are 0-1 valued. However, additional examples of $w_{(f,l)} = 0$ and $\partial \ell_h / \partial w_{(f,l)} = 0$ coinciding arise because we use indicator features. Sometimes $\partial \ell_h / \partial w_{(f,l)} = 0$ over a region because the examples adding to and subtracting from $\partial \ell_h / \partial w_{(f,l)}$ exactly balance within the region. This happens easily with indicator features, which always change $\partial \ell_h / \partial w_{(f,l)}$ by exactly 1 as $w_{(f,l)}$ passes through a point where an example is classified correctly by a margin of exactly 1. This happens with about 4% of the remaining possible feature weights in the best models we found in our experiments.

We can now answer the question of why L_2-regularized hinge loss models are relatively sparse by observing that none of the above depends on the exact form of regularization. As long as the point $w_{(f,l)} = 0$ falls in the region where $\partial \ell_h / \partial w_{(f,l)} = 0$, $w_{(f,l)}$ will tend to go to 0 with any push at all from regularization.

For our second question, why L_1- and L_2-regularized hinge loss models are so similar in sparsity for $C < 1$, the answer depends on the observation we mentioned above that, if only indicator features are used, $\partial \ell_h / \partial w_{(f,l)}$ can increase or decrease only by integer amounts. Thus, if $\partial \ell_h / \partial w_{(f,l)} \neq 0$, then $|\partial \ell_h / \partial w_{(f,l)}| \geq 1$. Note also that the derivative of the L_1 regularizer $\partial r_1 / \partial w_{(f,l)} = C$ if $w_{(f,l)} > 0$ and $\partial r_1 / \partial w_{(f,l)} = -C$ if $w_{(f,l)} < 0$. This means that if $C < 1$, the regularizer derivative can never outweigh a nonzero hinge loss derivative to keep a feature weight at 0. If $w_{(f,l)} = 0$, but $\partial \ell_h / \partial w_{(f,l)} \neq 0$, regularized hinge loss can always be reduced either by increasing or by decreasing $w_{(f,l)}$. Thus, with L_1 regularization and $C < 1$, $w_{(f,l)}$ tends to go to 0 only if $w_{(f,l)} = 0$ falls in the region where $\partial \ell_h / \partial w_{(f,l)} = 0$, the condition under which any reglarization will tend to push $w_{(f,l)}$ to 0. Hence, with $C < 1$, L_1 and L_2 regularization tend to produce hinge loss models of similar sparsity.

The observation that $\partial \ell_h / \partial w_{(f,l)}$ can increase or decrease only by integer amounts when only indicator features are used also helps explain why model sparsity for L_1-regularized hinge loss changes abruptly at integer values of C. At any nonzero value of a feature weight $w_{(f,l)}$ either $\partial r_1 / \partial w_{(f,l)} = C$ or $\partial r_1 / \partial w_{(f,l)} = -C$. Hence if $|\partial \ell_h / \partial w_{(f,l)}| < C$ on both sides[3] of the optimum of regularized loss, it must be the case that $w_{(f,l)} = 0$ at the optimum. Since any single example can increase or decrease $\partial \ell_h / \partial w_{(f,l)}$ only by 0 or 1, there must be at least C examples of f in the training corpus for $|\partial \ell_h / \partial w_{(f,l)}| \geq C$ to be possible. Thus C acts as a count cut-off on f, and whenever C is reduced past an integer value, a new set of low frequency features become candidates to have nonzero weights in an optimized, regularized model.

7. SUMMARY

We have shown empirically that L_1- and L_2-regularized hinge loss lead to sparser models than L_2-regularized log loss, but less sparse models than L_1-regularized log loss, and that, with only indicator features, model sparsity for L_1-regularized hinge loss changes abruptly at integer values of C, and is remarkably close to the model sparsity for L_2-regularized hinge loss when $C < 1$. These observations are explained by analyzing how the hinge loss derivative interacts with the derivatives of the two regularizers.

8. REFERENCES

[1] Galen Andrew and Jianfeng Gao, 2007. Scalable training of L_1-regularized log linear models. In *Proceedings of the 24th International Conference on Machine Learning*, June 20–24, Corvallis, Oregon, USA, 33–40.

[2] Koby Crammer and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

[3] Thorsten Joachims. 2006. Training linear SVMs in linear time. In *KDD '06, Proceedings of the 12th ACM*

[3] The derivative may be undefined at the optimium itself.

SIGKDD International Conference on Knowledge Discovery and Data Mining, August 20–23, Philadelpha, Pennsylvania, USA, 217–216.

[4] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural SVMs. *Journal of Machine Learning*, 77(1):27–59.

[5] Percy Liang, Hal Daumé III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *Proceedings of the 24th International Conference on Machine Learning*, July 5–9, Helsinki, Finland, 592–599.

[6] Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313-330.

[7] Andrew McCallum. 2002. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[8] Jorge Nocedal and Stephen J. Wright. 1999. *Numerical Optimization*. Springer.

[9] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, July 7–12, Sapporo, Japan, 160–167.

[10] Libin Shen, Giorgio Satta, and Aravind K. Joshi. 2007. Guided learning for bidirectional sequence classification. in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, June 23–30, Prague, Czech Republic, 760–767.

[11] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, May 27–June 1, Edmonton, Alberta, Canada, 173–180.