Efficient Online Feature Selection based on ℓ_1-Regularized Logistic Regression

Kengo Ooi and Takashi Ninomiya
Graduate School of Science and Engineering, Ehime University, 3, Bunkyo-cho, Matsuyama, Japan

Keywords: Machine Learning, Feature Selection, Grafting, ℓ_1-Regularized Logistic Regression.

Abstract: Finding features for classifiers is one of the most important concerns in various fields, such as information retrieval, speech recognition, bio-informatics and natural language processing, for improving classifier prediction performance. Online grafting is one solution for finding useful features from an extremely large feature set. Given a sequence of features, online grafting selects or discards each feature in the sequence of features one at a time. Online grafting is preferable in that it incrementally selects features, and it is defined as an optimization problem based on ℓ_1-regularized logistic regression. However, its learning is inefficient due to frequent parameter optimization. We propose two improved methods, in terms of efficiency, for online grafting that approximate original online grafting by testing multiple features simultaneously. The experiments have shown that our methods significantly improved efficiency of online grafting. Though our methods are approximation techniques, deterioration of prediction performance was negligibly small.

1 INTRODUCTION

Currently, many techniques developed in the field of machine learning are used in various fields including information retrieval, natural language processing, speech recognition and bio-informatics. Among these techniques, learning of classification is one of the most fundamental and general techniques that can be used for many applications. Given training samples with correct labels, a classifier that infers the correct label for an input is learned from the training samples. The classifiers infer the answer from features, which are extracted from an input by applying feature functions for the input. The accuracy of classifiers highly depends on the feature functions, but it is not easy to find good feature functions. In many applications, feature functions are still designed by human experts. Specifically, in natural language processing or information retrieval, combinations of local word/part-of-speech features are frequently used to detect the co-occurrence of words, part-of-speech, and phrases in text. However, it is obviously not easy even for human experts to find feature functions from exponentially many combination features.

Selecting useful features from a large feature set is called *feature selection* (Guyon and Elisseeff, 2003). Feature selection is studied in machine learning as a method for eliminating redundant or unnecessary features to reduce the learning and inference costs and improving a classifier's generalization ability to predict better answers for unseen data. With feature selection, we can consider a scenario for obtaining good features for improving a classifier's prediction performance, where we generate exponentially many combinations of features and then select useful features from them (Okanohara and Tsujii, 2009). ℓ_1-regularized logistic regression is preferable for such a purpose because its prediction performance is comparable to the state-of-the-art classifiers (Gao et al., 2007), and many features are eliminated to minimize its objective function. However, ℓ_1-regularized logistic regression must estimate its parameters for all possible features, that is, if we had an extremely large feature set as feature candidates, its parameter optimization would be extremely expensive or intractable. Perkins *et al.* proposed a method of incremental feature selection for ℓ_1-regularized logistic regression called *grafting* (Perkins et al., 2003). Grafting starts from an empty set of features and incrementally selects features from a set of candidate features. Unlike original ℓ_1-regularized logistic regression, grafting does not optimize parameters for the full set of features, but it requires frequent optimization of parameters for the currently selected features, which turns out to be extremely expensive. Perkins *et al.* also proposed an online version of grafting, which we

Ooi K. and Ninomiya T..
Efficient Online Feature Selection based on ℓ1-Regularized Logistic Regression.
DOI: 10.5220/0004255902770282
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 277-282
ISBN: 978-989-8565-39-6

call *online grafting* (Perkins and Theiler, 2003), but its computational cost is still high due to frequent parameter optimization.

We propose two improved methods for online grafting, online grafting with multiplicative division and online grafting with constant division, in which parameter optimization is performed less frequently than original online grafting. Our methods have trade-offs between efficiency and prediction performance. If we perform parameter optimization more frequently, prediction performance will be better but efficiency will decrease. If we have less frequent parameter optimization, it will be efficient but prediction performance will decrease. Online grafting with multiplicative division attempts to optimize parameters each time multiple numbers of features are tested. Online grafting with constant division attempts to optimize parameters each time constant numbers of features are tested. The experiments showed that our methods significantly improved efficiency of online grafting without deteriorating prediction performance. We also measured the performance of our methods for combination features in the experiments.

2 BACKGROUND

2.1 Logistic Regression with ℓ_1-Regularizer

Logistic regression, known as the maximum entropy model, is a well known discriminative probabilistic model for binary classification. Given an input x, the logistic regression classifier maps it into an output $y \in \{-1,+1\}$, where -1 is a negative label and $+1$ is a positive label. The logistic regression classifier learns its probabilistic model from annotated training data.

Given input x and output y, the probabilistic model of logistic regression is defined as follows.

$$P(y=+1|\mathbf{x}) = \frac{exp(f(\mathbf{x}))}{1+exp(f(\mathbf{x}))} \tag{1}$$

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + b \tag{2}$$

where d is the number of features, b is a bias term, x_j is a j-th feature of x, and w_j is a weight for the j-th feature.

The logistic regression classifier is trained by maximizing the log-likelihood of the training data, or equivalently minimizing the binomial negative log-likelihood (BNLL) (Hastie et al., 2001) loss function for the training data. Logistic regression is often trained with regularization to prevent overfitting to the training data. Typically, ℓ_1-norm or ℓ_2-norm is used as a regularization term, which is called ℓ_1-regularizer for ℓ_1-norm and ℓ_2-regularizer for ℓ_2-norm. ℓ_1-regularizer can also be used for feature selection because many of the weights become zeros for useless features as a result of training with ℓ_1-regularizer (Tibshirani, 1994). Both grafting and online grafting select features in the framework of ℓ_1-regularized logistic regression.

Given a data set D, the objective function C for ℓ_1-regularized logistic regression is defined as follows:

$$C(\mathbf{w},D) = \frac{1}{|D|} \sum_{\langle \mathbf{x},y \rangle \in D} \log(1+e^{-yf(\mathbf{x})}) + \lambda \sum_{j=1}^{d} |w_j|, \tag{3}$$

where λ is a hyper-parameter, that is tuned manually. This is the sum of the BNLL loss function and ℓ_1-regularizer. The objective function is a convex optimization problem without constraints.

2.2 Grafting

Grafting is a learning method that provides both parameter optimization and feature selection in the framework of ℓ_1-regularized logistic regression. Given a set of features F, grafting incrementally selects features from F. For each iteration of feature selection, grafting optimizes weight parameters using a general-purpose gradient descent method and selects a feature that has the greatest effect on reducing the objective function C, i.e., a feature that has the greatest magnitude of gradient $\frac{\partial C}{\partial w_i}$. Grafting stops iteration when the magnitude of the gradient of the loss function $\frac{\partial L}{\partial w_i}$ is below λ (the hyper-parameter for the ℓ_1-regularizer) for all remaining features. The obtained weights are guaranteed to be locally optimum, so grafting is guaranteed to find the global optimum because the objective function is convex. This means that grafting finds the optimum for the problem of ℓ_1-regularized logistic regression.

2.3 Online Grafting

Online grafting is an online feature selection algorithm that provides both parameter optimization and online feature selection. Non-online grafting selects a feature from the set of features, that is, each feature is tested many times so as to find the best feature in the feature set. Online grafting is a feature selection scheme where we assume that a sequence of features is given, and each feature is tested only once along the sequence of features.

Algorithm 1: ℓ_1-reduction.

```
Input: feature vector $\mathbf{f}$ and weight vector $\mathbf{w}$
for all $w_i \in \mathbf{w}$ do
    if $w_i = 0$ then
        Remove $f_i$ from $\mathbf{f}$
        Remove $w_i$ from $\mathbf{w}$
    end if
end for
```

The algorithm for online grafting is as follows. Let F be the feature sequence, D the data set, and λ the hyper-parameter for online grafting. Assume that we have feature $\mathbf{f}$ and weight $\mathbf{w}$ vectors. First, we retrieve a feature f from the head of F, and f is removed from the sequence. Then, f is added to $\mathbf{f}$, and 0 is added to $\mathbf{w}$. Let the added feature vector be $\mathbf{f}^{(test)}$ and the added weight vector $\mathbf{w}^{(test)}$. The online grafting algorithm tests the added f as follows.

$$\left|\frac{\partial \bar{L}}{\partial w_{j+1}}\right| > \lambda, \tag{4}$$

where $\bar{L}$ is the average loss function for D. If f satisfies the above condition, it is selected as a new feature in the model. If it is selected, $\mathbf{f}$ and $\mathbf{w}$ are updated to $\mathbf{f}^{(test)}$ and $\mathbf{w}^{(test)}$, then the new $\mathbf{w}$ is re-estimated by solving the ℓ_1-regularized logistic regression problem for the new $\mathbf{f}$ and D by using general-purpose numerical optimization methods such as quasi-Newton methods. If f is not selected, $\mathbf{w}$ and $\mathbf{f}$ remain unchanged. Online grafting repeats this procedure until F becomes empty.

In online grafting, weights that become zero as a result of optimization are explicitly pruned by ℓ_1-reduction each time after optimization is performed. That is, in online grafting, ℓ_1-reduction is frequently performed.

3 TESTING MULTIPLE FEATURES

Original online grafting is not practical for real data sets, e.g., a data set consisting of millions of features and several tens of thousands of data points, due to high computational cost of original online grafting. There are two reasons for this. First, parameter optimization is performed using a general-purpose numerical optimization method, which is computationally expensive. Second, original online grafting must apply expensive parameter optimization each time a feature passes the test. For example, if we have one million features and one tenth of features pass the test, we have to apply parameter optimization one thousand times.

Algorithm 2: Online Grafting (Multiplicative Division).

```
Input: $F$, $D$ and $\lambda$
$\mathbf{f} := ()$, $\mathbf{w} := ()$, $i := 0$, $j := 0$
for all $f \in F$ do
    $i := i + 1$
    let $\mathbf{f} = (f_1, \cdots, f_j)$ and $\mathbf{w} = (w_1, \cdots, w_j)$
    $\mathbf{f}^{(test)} := (f_1, \cdots, f_j, f)$, $\mathbf{w}^{(test)} := (w_1, \cdots, w_j, 0)$
    if $\left|\frac{\partial L_D(\mathbf{w}^{(test)})}{\partial w_{j+1}}\right| > \lambda$ then
        $\mathbf{f} := \mathbf{f}^{(test)}$, $\mathbf{w} := \mathbf{w}^{(test)}$
    end if
    if $i \geq 2^j$ then
        Optimize $\mathbf{w}$ for $D$            ........(*)
        $\ell_1$-reduction($\mathbf{f}$, $\mathbf{w}$)
        $i := 0$, $j := j + 1$
    end if
end for
return f and w
```

Our two proposed methods approximate online grafting by testing multiple features simultaneously, i.e., multiple features are tested successively without optimization, and the parameters are optimized only after the multiple feature test.

3.1 Multiplicative Division

The first method, *multiplicative division*, tests multiple features in which the number of tested features increases in multiple. First, one feature is tested then parameter optimization is performed. Next, two features are tested and parameter optimization is performed. Next, four features are tested and parameter optimization is performed. We continue this procedure until we reach the end of the feature sequence. With this method, we test 2^{i-1} features before we perform i-th parameter optimization.

Algorithm 2 is that for online grafting with multiplicative division.

This method frequently optimizes the weights in the beginning of the procedure. The trained model has a small number of features in the beginning; hence, we consider that the ability of selecting good features is weak in the beginning. After having a sufficient number of features, we assume that online grafting can select good features with less frequent parameter optimization.

This strategy can significantly reduce the total number of optimizations, but online grafting might wrongly select useless features or discard useful features. We examined this trade-off by evaluating the accuracy of online grafting through experiments.

In the experiments, we also evaluated the cumulative number of optimized weights, which is the cumulative number of weights that are optimized by parameter optimization. As the number of data points

Algorithm 3: Online Grafting (Constant Division).

```
Input: $F$, $D$, $\lambda$ and $C$
$\mathbf{f} := ()$, $\mathbf{w} := ()$, $i := 1$
for all $f \in F$ do
    $i := i + 1$
    let $\mathbf{f} = (f_1, \cdots, f_j)$ and $\mathbf{w} = (w_1, \cdots, w_j)$
    $\mathbf{f}^{(test)} := (f_1, \cdots, f_j, f)$, $\mathbf{w}^{(test)} := (w_1, \cdots, w_j, 0)$
    if $\left| \frac{\partial L_D(\mathbf{w}^{(test)})}{\partial w_{j+1}} \right| > \lambda$ then
        $\mathbf{f} := \mathbf{f}^{(test)}$, $\mathbf{w} := \mathbf{w}^{(test)}$
    end if
    if $i \geq |F|/C$ then
        Optimize $\mathbf{w}$ for $D$            ........(*)
        $\ell_1$-reduction($\mathbf{f}$, $\mathbf{w}$)
        $i := 1$
    end if
end for
return $\mathbf{f}$ and $\mathbf{w}$
```

is fixed in online feature selection, the computational cost for learning is related to the number of weights that are estimated by parameter optimization. We compare the cumulative number of optimized weights between original online grafting, our online grafting methods, and ℓ_1-regularized logistic regression. In principle, the accuracy of ℓ_1-regularized logistic regression is better than online grafting because online grafting is a method of approximating ℓ_1-regularized logistic regression. However, ℓ_1-regularized logistic regression requires full features and weights as inputs for the algorithm, which means that it requires significant time and space for parameter optimization. Therefore, there is also a trade-off between accuracy and cumulative number of optimized weights. We also evaluated this trade-off in the experiments.

3.2 Constant Division

The second method, *constant division*, tests multiple features in which the number of tested features is fixed. Given a constant C, $|F|/C$ features are tested then parameter optimization is performed. We continue this procedure until we reach the end of feature sequence. With this method, we always test $|F|/C$ features before we perform parameter optimization. Algorithm 3 shows the algorithm for online grafting with constant division.

The advantage of this strategy is that we can control the frequency of parameter optimization. If we set the constant C as $|F|$, then online grafting with constant division is the same as original online grafting.

4 COMBINING FEATURES

We also examine a method for generating a new feature set by combining features in the framework of online feature selection. A new feature can be generated by multiplying the feature elements. As there are $2^{|F|}$ combinations for F, the feature combination method can generate an extremely long sequence of features. Let $F_1 = \{f_1, f_2, \cdots, f_K\}$. A new feature set F_2 is generated as $\{f_1 f_1, f_1 f_2, \cdots, f_1 f_K, f_2 f_1, \cdots, f_K f_K\}$ by multiplying two elements in F_1. For the combination of order L, we generate a new feature set $F_1 \cup F_2 \cup \cdots \cup F_L$. We call the new feature set 'L-combination feature set'.

5 EXPERIMENT

5.1 The Datasets

We used four data sets in the experiments; a9a (Frank and Asuncion, 2010), w8a (Platt, 1999), IJCNN1 (Prokhorov, 2001), and news20.binary (Keerthi and DeCoste, 2005). The prediction task of a9a is to determine that a person makes over 50,000 USD a year. w8a and news20.binary are for text categorization, and IJCNN1 is used in the IJCNN 2001 competition. Each data set was composed of a training set, a development set, and a test set. The number of features for each data set is listed in Table 1. Since a9a, w8a, and IJCNN1 do not have large feature sets, we used the 2-combination feature set. Table 1 also lists the number of combination features. These data sets were used for binary classification, i.e, the output labels were binary-label $\{-1, +1\}$ in all data sets.

5.2 Evaluation

We compared our methods (multiplicative division and constant division), a current method (original online grafting), and ℓ_1-regularized logistic regression (LR+L1). The division number C for constant division and the hyper-parameter λ for regularization were determined using the development sets. We implemented the online grafting algorithms in Python. We used the Python package of LIBLINEAR (Fan et al., 2008) for parameter optimization ((*) in Algorithm 2, 3). We used InTrigger[1], a distributed computing platform consisting of more than 1,900 CPU cores in 14 sites, for conducting the experiments. We evaluated precision, training time, the number of weights optimized by LIBLINEAR (optimized weights), and the number of features that remain after ℓ_1-reduction (active features) for the test data sets.

[1] http://www.intrigger.jp/

The experimental results are listed in Tables 2, 3, 4, and 5. The results from the tables indicate that the multiplicative division method achieved the least cumulative number of optimized weights and the shortest training time of the two proposed methods. That is, it successfully reduced the computational cost. When comparing the two proposed methods with original online grafting. The proposed methods dramatically reduced both training time and the cumulative number of optimized weights. The difference in training time between LR+L1 and the multiplicative division method was rather small, but the multiplicative division method was slightly faster than LR+L1, and the cumulative number of optimized weights with the multiplicative division method was smaller than LR+L1. These tables also show that the difference in precision was negligibly small among the proposed methods, original online grafting, and LR+L1 in a9a, w8a, and IJCNN1. These results indicate that our methods are good approximations of LR+L1 in terms of precision.

6 CONCLUSIONS

We proposed two improved methods, in terms of efficiency, for online grafting. Online grafting is an incremental gradient-based method for feature selection, which incrementally estimates features that should be assigned exactly zero weights in ℓ_1-regularized logistic regression, and eliminates them one at a time. Online grafting was preferable as a feature selection method but its learning was inefficient due to frequent parameter optimization. We approximated original online grafting by testing multiple features simultaneously, i.e., multiple features were tested successively without optimization.

We evaluated our two methods. They attempt to optimize parameters each time multiple/constant numbers of features are tested. Though our methods have trade-offs between efficiency and prediction accuracy, the experimental results showed that our methods worked efficiently with negligibly small loss of prediction accuracy, and in some cases prediction accuracy was better than original online grafting and ℓ_1-regularized logistic regression.

ACKNOWLEDGEMENTS

We would like to thank the InTrigger team for operating and offering the computing resources consisting of more than 1,900 CPU cores in 14 sites. This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (C) Grant Number 22500121.

REFERENCES

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.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository.

Gao, J., Andrew, G., Johnson, M., and Toutanova, K. (2007). A comparative study of parameter estimation methods for statistical natural language processing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*, pages 824–831, Prague, Czech Republic. The Association for Computational Linguistics.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The Elements of Statistical Learning*. Springer, New York: Springer-Verlag.

Keerthi, S. S. and DeCoste, D. (2005). A modified finite newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6:341–361.

Okanohara, D. and Tsujii, J. (2009). Learning combination features with L1 regularization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers (NAACL-short'09)*, pages 97–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Perkins, S., Lacker, K., Theiler, J., Guyon, I., and Elisseeff, A. (2003). Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3:1333–1356.

Perkins, S. and Theiler, J. (2003). Online feature selection using grafting. In *International Conference on Machine Learning (ICML 2003)*, pages 592–599. ACM Press.

Platt, J. C. (1999). Advances in kernel methods. chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press.

Prokhorov, D. (2001). IJCNN 2001 neural network competition. In *IJCNN'01*, Ford Research Laboratory.

Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288.

Table 1: Specifications of data sets.

	a9a	w8a	IJCNN1	news20.binary
# of features	123	300	22	1,355,191
# of combination features	15,252	90,300	506	-
# of training data	26,048	36,624	39,992	10,000
# of development data	6,513	8,922	9,998	4,998
# of test data	16,281	13,699	91,701	4,998

Table 2: Experimental results for a9a.

	Precision (%)	Cumulative number of optimized weights	Time (s)	Max number of optimized weights	Active features
Multiplicative division	85.25	6,772	10.45	2,146	451
Constant division C = 5	85.24	6,970	23.42	1,863	329
Constant division C = 10	85.19	8,058	34.07	1,199	379
Constant division C = 50	85.24	13,265	99.52	451	135
Original online grafting (Perkins and Theiler, 2003)	85.19	505,822	5,952.71	116	102
LR+L1	85.19	15,252	10.78	15,252	643

Table 3: Experimental results for w8a.

	Precision (%)	Cumulative number of optimized weights	Time (s)	Max number of optimized weights	Active features
Multiplicative	99.04	38,209	17.26	12,491	673
Constant division C = 5	99.04	37,713	37.77	9,429	673
Constant division C = 10	99.07	39,807	47.03	5,083	596
Constant division C = 50	99.06	53,424	131.52	1,512	442
Original online grafting (Perkins and Theiler, 2003)	99.05	8,833,804	56,029.97	278	269
LR+L1	99.11	90,300	24.4	90,300	958

Table 4: Experimental results for IJCNN1.

	Precision (%)	Cumulative number of optimized weights	Time (s)	Max number of optimized weights	Active features
Multiplicative division	97.64	742	20.44	394	391
Constant division C = 5	97.60	1,080	41.07	388	386
Constant division C = 10	97.58	1,962	67.92	380	380
Constant division C = 50	97.61	8,810	332.75	371	371
Original online grafting (Perkins and Theiler, 2003)	97.61	68,438	2,633.61	317	314
LR+L1	97.62	506	15.67	506	414

Table 5: Experimental results for news20.binary.

	Precision (%)	Cumulative number of optimized weights	Time (s)	Max number of optimized weights	Active features
Multiplicative division	94.90	37,176	11.77	5,007	2,327
Constant division C = 5	94.96	221,574	20.56	205,210	2,615
Constant division C = 10	94.94	145,456	34.44	115,802	2,201
Constant division C = 50	94.76	128,409	141.62	24,419	1,756
Original online grafting (Perkins and Theiler, 2003)	95.00	16,302,937	15,994.70	1,851	1,412
LR+L1	96.22	1,355,191	12.05	1,355,191	11,224