Grafting-Light: Fast, Incremental Feature Selection and Structure Learning of Markov Random Fields

Jun Zhu
Machine Learning Department
Carnegie Mellon University
Pittsburgh, 15213 PA
junzhu@cs.cmu.edu

Ni Lao
Language Technology Institute
Carnegie Mellon University
Pittsburgh, 15213 PA
nlao@cs.cmu.edu

Eric P. Xing
Machine Learning Department
Carnegie Mellon University
Pittsburgh, 15213 PA
epxing@cs.cmu.edu

ABSTRACT

Feature selection is an important task in order to achieve better generalizability in high dimensional learning, and structure learning of Markov random fields (MRFs) can automatically discover the inherent structures underlying complex data. Both problems can be cast as solving an ℓ_1-norm regularized parameter estimation problem. The existing Grafting [16] method can avoid doing inference on dense graphs in structure learning by incrementally selecting new features. However, Grafting performs a greedy step to optimize over free parameters once new features are included. This greedy strategy results in low efficiency when parameter learning is itself non-trivial, such as in MRFs, in which parameter learning depends on an expensive subroutine to calculate gradients. The complexity of calculating gradients in MRFs is typically exponential to the size of maximal cliques.

In this paper, we present a fast algorithm called *Grafting-Light* to solve the ℓ_1-norm regularized maximum likelihood estimation of MRFs for efficient feature selection and structure learning. Grafting-Light iteratively performs one-step of orthant-wise gradient descent over free parameters and selects new features. This lazy strategy is guaranteed to converge to the global optimum and can effectively select significant features. On both synthetic and real data sets, we show that Grafting-Light is much more efficient than Grafting for both feature selection and structure learning, and performs comparably with the optimal batch method that directly optimizes over all the features for feature selection but is much more efficient and accurate for structure learning of MRFs.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical

General Terms

Algorithms, Experimentation

Keywords

Feature Selection, Structure Learning, Markov Random Fields

1. INTRODUCTION

Markov random fields (MRFs) are undirected graphical models and have been widely used in an ever-growing variety of applications, including natural language processing [21], data mining [14], signal processing [29], etc. Conditional random fields (CRFs) [11] are special MRFs that are globally conditioned on inputs and have shown great promise in various applications [21, 14]. These models are based on composite features that explicitly exploit the structural dependencies among elements in high-dimensional inputs (e.g., text sequences) and structured interpretational outputs (e.g., part-of-speech tagging). Therefore, they usually have a complex and high-dimensional feature space. To achieve better generalizability and interpret complex data, it is desirable to do feature selection [7] and pursue a sparse representation of such models that leaves out irrelevant features. Since the problem of selecting an optimal subset of features is NP-hard [28], a popular solution is to use a convex relaxation of the non-convex feature selection problem. ℓ_1-norm regularized maximum likelihood estimation (MLE) is among the most popular approaches to selecting features in Markov random fields and has shown great promise [1, 16]. The sparsity of the ℓ_1-norm regularized MLE is due to the singularity of the ℓ_1-norm at the origin [24].

Another important problem we consider is the structure learning of MRFs. As the variety and scale of problems increase, hand-crafting MRFs become less applicable. Learning the structures of MRFs can automatically discover the inherent structures underlying complex data. Similar as in feature selection, structure learning of MRFs can be cast as solving an ℓ_1-norm regularized parameter estimation problem [12, 27], where the structures of MRFs are encoded by a set of features. However, solving the ℓ_1-norm regularized estimation problem is not easy, as we explain below, especially in MRFs, where the inference is typically exponential to the size of maximal cliques. In this paper, we focus on developing efficient algorithms to solve the ℓ_1-norm regularized estimation problem for both feature selection and structure learning of MRFs.

Two types of approaches have been successfully used to solve the ℓ_1-regularized MLE, that is, the *batch* methods (such as the OWL-QN algorithm [1] and the ℓ_1-ball projection-based method [20]) that directly optimize over all the candidate features and the *incremental* methods (such as Grafting [16]) that incrementally include new features. Although batch methods can deal with a large number (e.g., millions [1]) of features, there are several scenarios in which only the incremental methods can be applied. First, for online

feature selection [17], not all the features are available at the beginning. In this case, incremental methods can be easily applied. Second, even all the candidate features are available, a model with all the features can be extremely difficult to do inference. One typical example is the structure learning of MRFs, which has been formulated as a feature selection problem by defining features that encode the dependencies among random variables and performing the ℓ_1-regularized MLE [12]. In this case, including all the features could lead to an extremely dense (usually complete) graph structure, of which inference can be extremely hard and inaccurate. Therefore, we consider the incremental methods in this paper.

Existing incremental methods, such as Grafting [16], optimize the ℓ_1-regularized MLE by iteratively performing two steps, i.e., optimizing over all the free parameters and selecting new features. Although selecting features can be quickly done by using a gradient-based heuristic to assess and select new features that can improve the existing model much, optimization over the free parameters is usually an expensive step, especially in Markov random fields. In MRFs, finding the optimal parameters requires an iterative procedure, such as the quasi-Newton method [13, 21] or stochastic gradient descent [26], in which each iteration needs to compute the gradients. Computing gradients in MRFs is computationally expensive even for the models whose tree-width is small. Moreover, our empirical results show that this greedy strategy of Grafting tends to select fewer features than the batch method [1] and under-fit the data.

In this paper, we propose a fast incremental algorithm called Grafting-Light to solve the ℓ_1-regularized MLE for efficient feature selection and structure learning [12, 27] of Markov random fields. Grafting-Light fully integrates the feature selection and parameter learning together by alternating between one-step of gradient descent (instead of full optimization as in Grafting) over the free parameters and selecting new features. For gradient descent, we use the orthant-wise quasi-Newton step [1] as the search direction and perform a backtracking line search, and for selecting features, we apply the same gradient-based method as in Grafting [16]. This simple algorithm is guaranteed to converge to the global optimum. Although this lazy strategy can result in selecting some redundant features as compared to the greedy Grafting method, they can be effectively discarded when the ℓ_1-regularized MLE achieves its optimum. Empirical results on both synthetic and real data sets show that (1) Grafting-Light is much more efficient than Grafting for both feature selection and structure learning of MRFs; and (2) Grafting-Light can perform as well as the optimal batch method that optimizes over all the features on feature selection, but Grafting-Light is much more efficient and accurate than the batch method on structure learning of MRFs.

The paper is organized as follows. Section 2 presents some related work. Section 3 introduces some preliminaries. Section 4 formally describes the two problems of feature selection and structure learning in MRFs. Section 5 presents the Grafting-Light algorithm, and section 6 presents our empirical results. Finally, section 7 concludes this paper.

2. RELATED WORK

Feature selection is an important problem and has become the focus of much research in many areas where data sam-ples can have tens of thousands of variables, e.g., genomic microarray data analysis [30]. Feature selection can help interpret complex data and reduce the risk of over-fitting. Early approaches including the *Filter* [9] and *Wrapper* [10] often treat feature selection as a separate or weakly correlated task with the learning. Recently, feature selection has been viewed as an integrated step during learning within the framework of *regularized risk minimization*, i.e., minimizing a regularized empirical risk. By using the ℓ_1-norm regularizer, irrelevant features can be effectively discarded when the minimization problem obtains its optimum [24, 33, 34].

The approaches to structure learning of Markov random fields typically use greedy local heuristic search that incrementally changes the model structure by adding or deleting edges. The adding or deleting operation is guided towards an improvement of some objective function, such as marginal likelihood [15]. As the search is local and greedy, the learned network is (at best) a local optimum of a penalized likelihood score. Recently, structure learning of MRFs has been formulated as a convex program that maximizes an ℓ_1-norm regularized log-likelihood [12]. One advantage of this formulation is that it admits a unique global optimal solution.

Many methods have been developed to solve the ℓ_1-norm regularized estimation problem for feature selection or structure learning, including the batch and incremental methods as we have discussed in the introduction. The Gauss-Seidel method [22] is another batch method that applies a coordinate-descent strategy and optimizes over one subset of features at each step while keeping the weights of all other features fixed. Like Grafting, this method relies on a greedy sub-step of fully optimizing over free parameters, and thus is inefficient for MRFs and may under-fit the data. Other incremental methods like [18, 14] are even less efficient than Grafting even when some heuristics are used as in [14], because at each iteration they need to estimate the likelihood gain for each candidate feature, which depends on a step of estimating the weights of newly added features. An empirical comparison of several approaches to solving the ℓ_1-regularization problem is provided in [20].

Finally, for learning structures of the special Gaussian Markov random fields (GMRFs), inverse covariance estimation methods [2, 32, 3] have been developed based on the ℓ_1-norm penalized maximum likelihood estimation. For the structure learning of directed Bayesian networks, the structural EM (SEM) algorithm [4, 5] has a similar procedure as Grafting, that is, alternatively performing structural search to find new model structures and parametric search to obtain the optimal model parameters. Therefore, SEM is greedy in nature. See Section 5.3.2 for more comparison between SEM and Grafting-Light.

3. PRELIMINARIES

Without loss of generality, we consider the conditional random fields (CRFs) [11] which are special Markov random fields that are globally conditioned on observations. Our algorithm can be applied to any MRFs. Let $G = (V, E)$ be an undirected model over a set of random variables $\mathbf{X}$ and $\mathbf{Y}$. $\mathbf{X}$ are variables over the observations (e.g., text sentences) to be labeled and $\mathbf{Y}$ are variables over the corresponding labels (e.g., part-of-speech tag sequences). The variables $\mathbf{Y}$ could have a non-trivial structure, such as a linear-chain [11] or 2D grid. Each component Y_i takes values from a set of possible class labels $\mathcal{Y}_i$ (e.g. part-of-speech tags). The con-

ditional distribution of the label $\mathbf{y}$ (an instance of $\mathbf{Y}$) given the observation $\mathbf{x}$ (an instance of $\mathbf{X}$) is

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \phi(\mathbf{x}, \mathbf{y}|_c),$$

where $\mathcal{C}$ is the set of cliques on G; $\mathbf{y}|_c$ are the components of $\mathbf{y}$ associated with the clique c; ϕ is a potential function taking non-negative real values; $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in \mathcal{C}} \phi(\mathbf{x}, \mathbf{y}|_c)$ is the normalization factor. Usually, the potential functions are of a log-linear form, i.e., $\phi(\mathbf{x}, \mathbf{y}|_c) = \exp\{\sum_k w_k f_k(\mathbf{x}, \mathbf{y}|_c)\}$, where $f_k(\mathbf{x}, \mathbf{y}|_c)$ are feature functions and w_k are their weights. We use $\mathbf{f}$ to denote the vector of f_k and $\mathbf{w}$ to denote the corresponding vector of weights.

Given a set of labeled training data $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^N$, the standard parameter learning of CRFs is a task to find the best parameter vector that has the maximal log-likelihood or minimal negative log-likelihood $L(\mathbf{w})$, where

$$L(\mathbf{w}) = -\sum_{i=1}^N \log p(\mathbf{y}^i|\mathbf{x}^i) = -\sum_{i=1}^N (\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \log Z(\mathbf{x}^i)),$$

where $f_k(\mathbf{x}, \mathbf{y})$ to denote the summation of f_k over the sample $(\mathbf{x}, \mathbf{y})$. At least in principle, the parameter learning problem can be solved with gradient descent methods, such as quasi-Newton [13], stochastic [26], or exponentiated gradient [6] methods. Each component of the gradient is

$$\frac{\partial L(\mathbf{w})}{\partial w_k} = -\sum_{i=1}^N f_k(\mathbf{x}^i, \mathbf{y}^i) + \sum_{i=1}^N E_{p(\mathbf{y}|\mathbf{x}^i)}[f_k(\mathbf{x}^i, \mathbf{y})]. \quad (1)$$

However, from Eq. (1), we can see that the gradient depends on the marginal probabilities of the variables associated with cliques. For a model (e.g., linear chain CRFs [11]) whose graph structure has a small tree-width (i.e., the size of the maximum cliques minus one), inferring the marginal probabilities (and the gradients) can be efficiently done, e.g., by doing forward-backward message passing or using the general junction tree algorithm. The complexity of these exact methods is exponential to the size of the maximum cliques. But for those models whose graph structures contain large loops, we have to turn to approximation methods, either deterministic variational methods [8] and belief propagation [31]), or stochastic Markov chain Monte Carlo (MCMC) methods. In general, the accuracy and time-efficiency of these approximation methods depend largely on the graph structures. For dense graphs (e.g., a complete graph), approximation methods can be extremely inaccurate.

In this paper, we present a fast algorithm for efficient feature selection and structure learning of CRFs (MRFs in general). Generally, our algorithm improves in the following two key aspects: (1) it significantly reduces the number of gradient computation in feature selection and structure learning, which is of exponential complexity; and (2) it performs inference only on sparse graphs in the structure learning problem.

4. PROBLEM SETUPS

In this section, we formally present the problems we want to solve.

4.1 ℓ_1-regularized Feature Selection

As we have stated, feature selection is an important task to learn a sparse model representation that has a better generalization ability and can interpret complex data. For discriminative models (e.g., CRFs [11]), since in principle they can use arbitrary and overlapping features, the dimension of their feature spaces is usually very high and sparse, and

discarding redundant features will not hurt the performance much. For example, in the NP-chunking task more than 3 million (much larger than the number of training data) features were used in [21]. However, as we shall see, most (e.g., 99.9%) of these features can be discarded without decreasing the performance more than 1% in F1 score.

Traditional feature selection methods like Wrapper [10] and Filter [9] adopt an ineffective strategy that treats feature selection and learning as two independent or weakly correlated (as in Wrapper) tasks. Thus the information gleaned from the data by the learning system may be ignored when selecting features. We consider the integrated approach of ℓ_1-regularized maximum (conditional) likelihood estimation as in [16]. Our objective function to minimize is:

$$\mathcal{L}(\mathbf{w}) \triangleq L(\mathbf{w}) + \lambda \|\mathbf{w}\|. \quad (2)$$

where $\|\mathbf{w}\| \triangleq \sum_k |w_k|$ is the ℓ_1-norm.

Due to the singularity at the origin, the ℓ_1-norm has been widely used as a regularizer to achieve sparse estimates [24, 16, 3] by setting some feature functions' weights to exact zeros. In this generic formulation, we can use other loss functions $L(\mathbf{w})$, such as the hinge loss in SVMs [33] or structured hinge loss in max-margin Markov networks [34]. We can also introduce an additional differentiable ℓ_2-norm[1] as in [16] without changing the algorithm as presented below.

4.2 ℓ_1-regularized Structure Learning

Structure learning is a task to learn the graph topology of MRFs. Recently, the structure learning problem of MRFs has been formulated as a feature selection problem by defining the feature functions f to encode the model structures and performing the ℓ_1-regularized MLE [12, 27]. We consider both the pure feature selection problem (when model structures are kept fixed) and learning structures of MRFs in our experiments.

Two major types of approaches have been used to minimize $\mathcal{L}(\mathbf{w})$, that is, batch methods and incremental methods. Batch methods (e.g., OWL-QN [1] and the ℓ_1-ball projection-based approach [20]) optimize $\mathcal{L}(\mathbf{w})$ over *all* the features from the very beginning. In contrast, incremental methods (e.g., Grafting [16]) maintain a working set of active features and iteratively optimize $\mathcal{L}(\mathbf{w})$ over the *active* features and select new features to add into the working set. As we have stated, although batch methods can handle a large set of features, there are many scenarios in which only the incremental methods can be applied, such as structure learning of graphical models [12] and online feature selection [17]. Therefore, we focus on the incremental methods and present a fast algorithm for selecting features and learning structures of Markov random fields.

5. THE GRAFTING-LIGHT ALGORITHM

In this section, we present the fast Grafting-Light algorithm with comparisons with existing methods and convergence analysis. The basic idea is that we begin with a model whose weights are almost all zeros. Then, we iteratively perform two steps, which are similar to but fundamentally different from those of Grafting [16]. At each iteration, we use a fast gradient-based heuristic to decide which features

---

[1]The composite regularizer of ℓ_1 and ℓ_2 norms is known as an elastic net regularizer, which has nice properties as discussed in [35].

Algorithm 1 Grafting-Light

Input: data $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^N$, constant λ, candidate feature set $\mathcal{F}$, and Select Unit M ($M \geq 1$)
Output: a subset $\mathcal{S} \subseteq \mathcal{F}$ and weights $\mathbf{w}$
Initialize $\mathcal{S} \leftarrow \emptyset$ and $\mathcal{U} = \mathcal{F}$.
repeat
    Step 1: Perform one-step of orthant-wise gradient descent of $\mathcal{L}(\mathbf{w})$ over the working set S.
    Step 2: Select top M features from the set $\{f_k : f_k \in \mathcal{U}, \text{ and } |\partial_k L(\mathbf{w})| > \lambda\}$ with large absolute subgradients. Add selected features into S and remove them from $\mathcal{U}$.
until convergence.
---

should be included in order to decrease the objective function by the maximum amount. Then, we perform *one-step* (or several steps) of gradient descent over all the active features that have been selected.

5.1 Optimality Conditions

Before describing the algorithm, we note that the optimality conditions of minimizing $\mathcal{L}(\mathbf{w})$ are:

$$\forall k, \quad \begin{cases} \partial_k L(\mathbf{w}) + \lambda sgn(w_k) = 0, & w_k \neq 0 \\ |\partial_k L(\mathbf{w})| \leq \lambda, & \text{otherwise} \end{cases} \quad (3)$$

where we have used $\partial_k L(\mathbf{w})$ to denote the partial derivative $\frac{\partial L(\mathbf{w})}{\partial w_k}$ as computed in Eq. (1), and the signum function $sgn(w)$ takes values from $\{-1, 0, 1\}$ according to whether w is negative, zero, or positive. From the optimality conditions, we can define the sub-gradient of $\mathcal{L}(\mathbf{w})$ as:

$$\partial_k \mathcal{L}(\mathbf{w}) = \begin{cases} \partial_k L(\mathbf{w}) + \lambda sgn(w_k), & w_k \neq 0 \\ \partial_k L(\mathbf{w}) + \lambda, & w_k = 0, \partial_k L(\mathbf{w}) < -\lambda \\ \partial_k L(\mathbf{w}) - \lambda, & w_k = 0, \partial_k L(\mathbf{w}) > \lambda \\ 0, & w_k = 0, |\partial_k L(\mathbf{w})| \leq \lambda \end{cases} \quad (4)$$

Then, the negative sub-gradient as defined in Eq. (4) is the direction of maximum descent and the optimality conditions (3) are equivalent to the condition that $\partial_k \mathcal{L}(\mathbf{w}) = 0, \forall k$. Both Grafting and Grafting-Light are iterative procedures that incrementally include new features until the above optimality conditions are satisfied for all the features.

5.2 The Basic Algorithm

The Grafting-Light algorithm maintains a working set S of selected features, and alternates between two steps. The procedure is outlined in Algorithm 1, and detailed below.
Step 1: one-step of orthant-wise gradient descent over the working set S. At each iteration, Grafting-Light performs one step of gradient descent over the features in S. Although the objective function $\mathcal{L}(\mathbf{w})$ is not differentiable everywhere, it is if $\mathbf{w}$ takes values within one orthant. Based on this observation, we can apply the fast quasi-Newton gradient descent step within a particular orthant. This idea has been explored in the OWL-QN algorithm [1].

At the tth iteration, according to the definition of the subgradient as in Eq. (4), a reasonable orthant is the one that contains $\mathbf{w}^t$ and into which $\partial \mathcal{L}(\mathbf{w}^t)$ leads, that is,

$$\forall k, \quad e_k = \begin{cases} sgn(w_k), & w_k \neq 0 \\ sgn(-\partial_k \mathcal{L}(\mathbf{w})), & w_k = 0 \end{cases}.$$

Let $\mathbf{H}_t$ denote the approximate hessian matrix at the point

$\mathbf{w}^t$. This matrix can be efficiently computed as in the limited memory BFGS (L-BFGS) algorithm [13], which only uses the first-order information gathered from previously explored points within several steps. We refer the interested readers to [13] for more details. Given $\mathbf{H}_t$, the quasi-Newton search direction is:

$$d^t = \Pi(\mathbf{H}_t p^t, \mathbf{e}),$$

where $p^t = \Pi(-\partial \mathcal{L}(\mathbf{w}^t), \mathbf{e})$ is the projection of the negative sub-gradient into the subspace associated with the orthant $\mathbf{e}$, and

$$\forall k, \quad \Pi_k(\mu, v) = \begin{cases} \mu_k, & sgn(\mu_k) = sgn(v_k) \\ 0, & \text{otherwise} \end{cases}$$

Given d^t, we do backtracking line search to select a step size α. During this procedure, we need to keep all the explored points $\mathbf{w}$ within the orthant $\mathbf{e}$. This can be done by using the projection operator Π. Specifically, the backtracking line search looks for the first step size α such that:

$$L(\Pi(\mathbf{w}^t + \alpha d^t, \mathbf{e})) \leq L(\mathbf{w}^t) - \gamma \mathbf{e}^\top [\Pi(\mathbf{w}^t + \alpha d^t, \mathbf{e}) - \mathbf{w}^t],$$

where $\alpha = \beta^n$ ($n = 0, 1, 2, \cdots$) and $\gamma, \beta \in (0, 1)$ are constants. Then, the new model parameter is $\mathbf{w}^{t+1} = \Pi(\mathbf{w}^t + \alpha d^t, \mathbf{e})$.
Step 2: select new features. Grafting-Light selects new features to add into the working set S. Like Grafting, Grafting-Light selects features that have largest subgradients (in magnitude). For those inactive features which satisfy the optimality condition (3), the coordinate-wise subgradients are zeros, and the weights will stay at zero at the next iteration. Therefore, Grafting-Light selects M ($M \geq 1$) features that satisfy $|\partial_k L(\mathbf{w})| > \lambda$ and have the largest sub-gradients in magnitude[2]. The selected new features are added to the working set S. We refer to M as the *Select Unit*, that is, the number of features selected at each iteration.

Grafting-Light alternates between the above two steps until convergence. We set the stopping criterion as the average change of the $\mathcal{L}(\mathbf{w})$ within several steps is less than a threshold ϵ. Note that the working set S may not be changed at Step 2 (when no features satisfy the condition $|\partial_k L(\mathbf{w})| > \lambda$).

5.3 Connections to Other Methods

The procedure of Grafting-Light is very simple as in Algorithm 1, and one can easily implement it. To better understand the idea of Grafting-Light, we discuss the relationships between Grafting-Light and existing methods.

5.3.1 Comparison with Incremental Methods

The Grafting [16] method takes the similar iterative procedure to incrementally select new features. But the key difference is that Grafting aggressively optimizes over the free parameters at each iteration after new features have been selected, while Grafting-Light just performs one step of gradient descent. Figure 1 illustrates the basic idea. Suppose the working set S is initialized to contain the feature weighted by μ. Then, Grafting-Light performs one step of gradient descent along the coordinate μ to μ_1, while Grafting optimizes over the free parameter μ to arrive at the local optimum

---
[2]If the number of features that satisfy $|\partial_k L(\mathbf{w})| > \lambda$ is smaller than M, all these feature are selected.
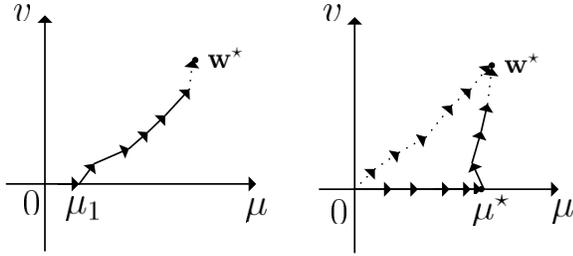
Figure 1: 2D illustration of the algorithms: (Left) Grafting-Light; (Right) Grafting.

point $\mu^\star$. After this step, both Grafting and Grafting-Light select new features. Suppose the feature weighted by v is selected by both algorithms. Then, Grafting-Light starts performing gradient-descent in the two dimensional space from $(\mu_1, 0)$ and keep going until convergence, while Grafting again optimizes over the free parameters (i.e., μ and v), starting from the previous local optimum point $(\mu^\star, 0)$ (solid line) or the origin (dashed line). Similar to Grafting, the information gain-based methods [18, 14] also fully optimize over free parameters at each iteration. These greedy incremental methods are inefficient for the feature selection or structure learning of MRFs because the optimization over free parameters can require many times of calculating the gradients, which is an expensive subroutine in MRFs. Moreover, they may result in a too sparse model that under-fits the data.

5.3.2 Comparison with Structural EM

As we have stated, both the Grafting and Grafting-Light can be used to learn the structures of MRFs. The key idea of Grafting and Grafting-Light is to integrate the structure learning of MRFs into the procedure of parameter estimation, which is performed with gradient descent methods. The basic procedure alternatively performs the *structural search* (i.e., including new features in Grafting or Grafting-Light) and *parametric search* (i.e., parameter estimation with gradient descents) in the joint structure-parameter space. Since the parametric search step is very computationally demanding, Grafting-Light adopts a lazy strategy, which performs one or several steps of gradient descents. This lazy strategy can significantly reduce the number of gradient evaluation, which can be exponentially expensive in MRFs, as we shall see in the experiments.

A similar idea has been investigated in the well-known structural EM (SEM) (a.k.a, model selection EM: MS-EM) algorithm [4, 5], which is used for the structure learning of directed Bayesian networks (BNs) in the presence of missing data or hidden variables. The basic procedure of SEM is the similar to that of Grafting or Grafting-Light, that is, alternatively performing the *structural search* and *parametric search*. For Bayesian networks with hidden variables, the parametric search is performed with an EM algorithm, which is greedy in the sense that the EM algorithm finds the optimal model parameters. This greedy strategy is acceptable for Bayesian networks because in BNs the EM procedure is much cheaper than the structural search. However, a greedy strategy of parametric search is not acceptable in the structure learning of MRFs, because of the expensive and inaccurate gradient evaluation, as we have discussed.

The alternating MS-EM algorithm [4] is also greedy in nature, and thereby essentially different from Grafting-Light.

5.3.3 Comparison with OWL-QN

The orthant-wise quasi-Newton (OWL-QN) [1] method for solving ℓ_1-regularized CRFs is a batch method that optimizes over all the features from the very beginning. As we have stated, this batch method has its disadvantages in several scenarios, like structure learning of MRFs and online feature selection. The Grafting-Light can be seen as an incremental version of the OWL-QN. As we shall see, Grafting-Light can work as well as the OWL-QN when model structures are kept fixed, and is much more efficient and accurate on learning the structures of MRFs.

5.4 Convergence

Based on the above connections, we can get the following convergence theorem of Grafting-Light:

Theorem 1 (Convergence). *When the loss function $L(\mathbf{w})$ is convex, bounded below, and continuously differentiable, the Grafting-Light converges to the global optimum.*

The convergence can be derived from the convergence theorem of OWL-QN [1]. We present some intuitive insights here. The stopping criterion guarantees that when Grafting-Light stops, the optimality conditions (3) are satisfied. Thus, the solution of Grafting-Light is a local optimum of $\mathcal{L}(\mathbf{w})$, because the points explored at each iteration are constrained in a subspace. Since L is convex, $\mathcal{L}$ is also convex and the local optimum is the global optimum.

6. EXPERIMENTS

In this section, we report our empirical results of Grafting-Light on both feature selection and structure learning of MRFs [12]. We compare with Grafting [16], Gauss-Seidel [22], and the full optimization with an ℓ_1-regularizer (*Full-Opt.-L1*) [1]. The information-gain methods [18, 14] are too expensive and we do not report their results here. All the algorithms are implemented in the C++ language on a standard Intel 2.00 GHz processor. For Grafting, we use the same select unit M, and for Gauss-Seidel, M is the number of features that are optimized over at each step.

6.1 Feature Selection

Our first set of experiments are on selecting features of MRFs, whose model structures are kept fixed during the learning. We report results on both synthetic and real NP-Chunking data sets. In this case, the batch method *Full-Opt.-L1* is optimal in the sense of obtaining the best subset of features, achieving the best predictive performance, etc. Although stochastic gradient methods [26, 25] may achieve better time efficiency than Full-Opt.-L1, they usually select many irrelevant features because of the approximate stochastic gradients they are using. The main observation in these experiments is that Grafting-Light performs comparably with the optimal Full-Opt.-L1, and is much more efficient than Grafting and Gauss-Seidel.

6.1.1 Evaluation on Synthetic Data

We generate sequence data sets, i.e., each input $\mathbf{x}$ is a sequence $(x_1, \cdots, x_L)$, and each component x_l is a d-dimensional vector of input features. The synthetic data are generated from pre-specified CRF models with i.i.d. instantiations of
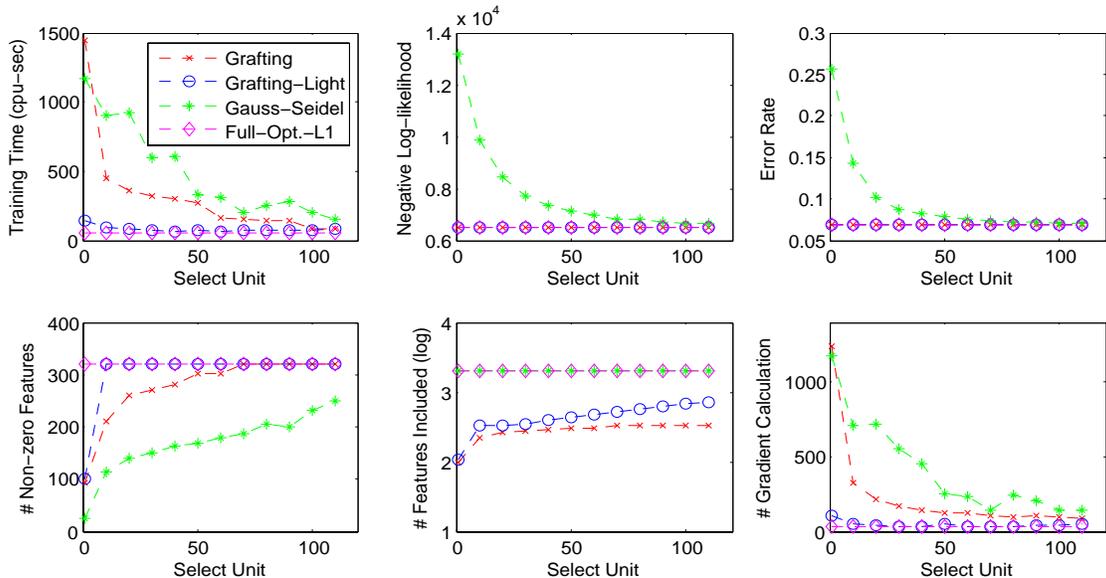
Figure 2: The training time (in cpu-seconds), negative log-likelihood, error rate, number of non-zero features, number of selected features (in base 10 logarithm) during training, and number of gradient calculation on the synthetic data by different algorithms when they use different Select Unit M.

the input features, from which samples of the structured output $\mathbf{y}$, i.e., a sequence $(y_1, \cdots, y_L)$, can be drawn from the conditional distribution $p(\mathbf{y}|\mathbf{x})$ defined by the CRF based on a Gibbs sampler. Specifically, we set $d = 1000$ and 100 input features are relevant to the output. The i.i.d input features are randomly drawn from a standard Gaussian distribution. We randomly generate a linear-chain CRFs with 8 binary states (i.e., $L = 8$ and $\mathcal{Y}_l = \{0, 1\}$). The feature functions include: 2000 real valued state-feature functions, of which each is over a one-dimensional input feature and a class label; and 4 (2×2) transition feature functions capturing pairwise label dependencies. We generate a data set that contains 5000 instances of which 3000 are randomly selected as training data and the rest are for testing.

We compare different methods on six criteria, that is, training time, negative log-likelihood on the training set, error rate, number of non-zero features in the final estimate, number of selected features, and the number of computing gradients. Figure 2 shows the results of different methods with respect to the select unit M $(1, 10, 20, \cdots, 110)$. For easy comparison, we use the same regularization constant $\lambda = 64$, which is the best parameter for the *Full Opt. L1* method. From the results, we can see that: (1) Grafting-Light is much more efficient than Grafting and Gauss-Seidel, especially when the select unit M is small. Also, compared to Grafting, the Grafting-Light is more robust with respect to the select unit. This robustness is important for online feature selection [17], where only a few features come at one time. Moreover, as shown in the fourth and fifth plots, Grafting-Light usually includes more features than the greedy Grafting during training, but the ℓ_1-regularization can effectively discard redundant features when Grafting-Light converges and result in almost the same numbers of non-zero features as the optimal Full-Opt.-L1; (2) Grafting-Light performs comparably with the optimal batch method (Full-Opt.-L1) on all the six evaluation criteria, except the number of features included during training. Since the training time is mainly spent on performing forward-backward

message passing on the linear-chain CRFs, Full-Opt.-L1 is always the most efficient one because of the fewest number of computing gradients as shown in the last plot, although both Grafting and Gauss-Seidel use much fewer features than Full-Opt.-L1 during training; (3) when the select unit M is fixed, the greedy Grafting and Gauss-Seidel usually select fewer features, as compared to Grafting-Light and the optimal Full-Opt.-L1. A too sparse model may under-fit the data and result in a large error rate. For example, when M is very small, Gauss-Seidel selects much fewer features than Full-Opt.-L1 and leads to a model that has a larger error rate and larger negative log-likelihood.

6.1.2 Evaluation on NP Chunking

We perform feature selection on real NP chunking, which is a sequence labeling task. In NP Chunking, the input is a word sequence and each word has an automatically annotated part-of-speech (POS) tag. The output is a corresponding label sequence, in which each label indicates whether a word is outside a chunk (O), starts a chunk (B), or continues a chunk (I). We use the CoNLL-2000 data set [19].

We use the same method as in [21] to define the labels y for a second-order Markov dependency between chunk tags, and the feature functions that encode the pairwise dependency between labels and the dependency between a label and input features (e.g., unigram words, bigram word pairs, unigram POS tags, bigram POS tag pairs, and trigram POS tag tuples). See Table 1 in [21] for detailed definition. The total number of supporting feature functions whose predicate is on at least once in the training set is larger than 3 millions. We compare different methods to select features from all these candidate feature functions. We choose the regularization constant λ of Full-Opt.-L1 by doing 5-fold cross validation, and compare all the methods using the same λ.

Figure 3 shows the results on the six criteria, that is, training time, negative log-likelihood on the training set, F1 (i.e., the harmonic mean of precision and recall), number of non-zero features in the final estimate, number of selected
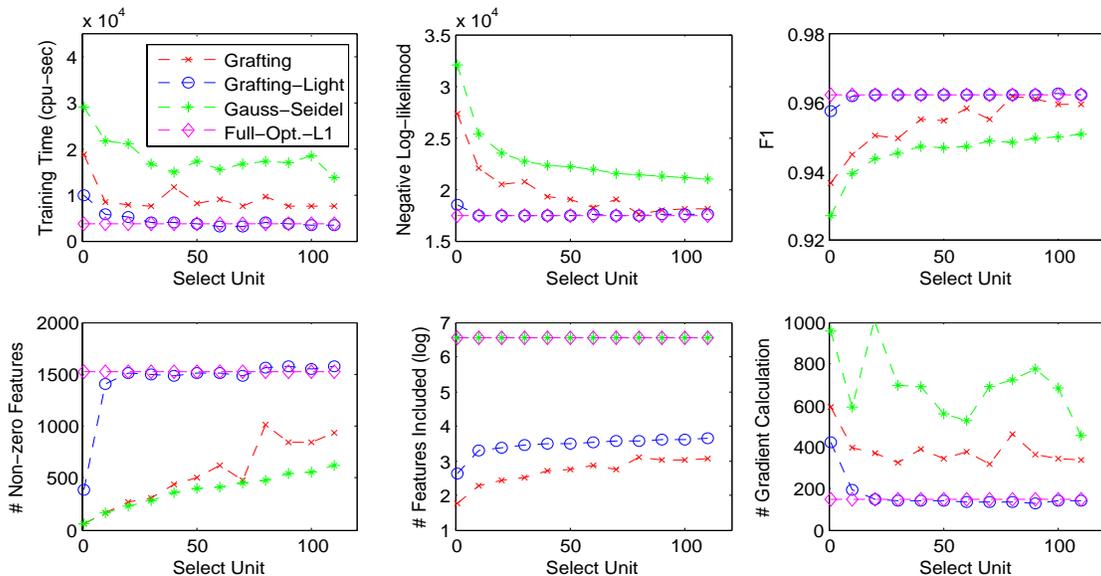
Figure 3: The training time (in cpu-seconds), negative log-likelihood, F1 score, number of non-zero features, number of selected features (in base 10 logarithm) during training, and number of gradient calculation on the NP-chunking task by different algorithms when they use different Select Unit M.

features, and the number of gradient calculation. Surprisingly, we get better results than those reported in [21]. The F1 score of Full-Opt.-L1 is less than 1 percent off than the CRF models using all the features. From the results, we can get almost the same conclusions as on the synthetic data. Firstly, Grafting-Light is much more efficient than Grafting and Gauss-Seidel when using different select units. The efficiency is due to the fact that Grafting-Light usually needs fewer numbers of gradient calculation (see the last plot), which is the most expensive step in MRFs, i.e., exponential to the size of maximum cliques. Secondly, Grafting-Light is much more robust than Grafting and Gauss-Seidel with respect to the select unit M in terms of time efficiency, modeling fitting (i.e., likelihood), predictive accuracy (i.e., F1), the number of non-zero features, and the number of gradient computation. Thirdly, similar to the results on synthetic data, Grafting-Light usually includes more features than Grafting during training, but the ℓ_1-norm regularizer can effectively discard redundant features when Grafting-Light converges and result in almost the same numbers of non-zero features as the optimal Full-Opt.-L1. Finally, on all the six criteria, except the number of features included, Grafting-Light performs as well as the Full-Opt.-L1, which is the optimal solution we can achieve as all features are presented from the beginning. Since the training time is mainly spent on computing gradients, Full-Opt.-L1 is always the most efficient one because of the fewest number of computing gradients as shown in the last plot, although both Grafting and Grating-Light use much fewer features than the Full-Opt.-L1 during training. Moreover, when the select unit is fixed, the greedy Grafting and Gauss-Seidel methods usually select much fewer features, as compared to Grafting-Light and Full-Opt.-L1, and they tend to under-fit the data when the select unit is small, which leads to a lower F1 and higher negative log-likelihood.

6.2 Structure Learning

As we have stated, structure learning of MRFs can be for-

mulated as a feature selection problem, by defining feature functions to encode the structural dependencies among random variables and performing the ℓ_1-regularized MLE [12]. We evaluate the Grafting-Light on learning the structures of pairwise MRFs. We use the OCR data set [23], but treat characters independently. We get 20×20 binary images by placing the original 16×8 characters in the centers of 20×20 black squares. We compare Grafting-Light with the Grafting and Full-Opt.-L1. We use the loopy belief propagation (BP) [31] to do approximate inference for the gradient calculation as in Eq. (1).

Figure 4 shows the results of the six characters of the phrase "acm sig". The sample sizes of these characters are 4033, 2114, 1602, 1394, 4913, and 2472, respectively. For each data set, we use a half to learn the structure. We can see that Grafting-Light is much more efficient than the other two methods. For Full-Opt.-L1, which optimizes over all the features, the model structure is a complete graph and the inference is very slow. Therefore, although the number of gradient calculation is (in most cases) fewer than those of the other two methods, the total training time is much larger. Moreover, the approximation inference of the Loopy BP algorithm is very inaccurate on a complete model structure, which leads to a larger negative average log-likelihood (over the pixels) of the Full-Opt.-L1. Similar to the previous results, Grafting needs more steps of computing the gradients and thus has a slower convergence, as compared to Grafting-Light. We also show the average images learned by the three methods. Clearly, Full-Opt.-L1 under-fits the data and produces blurry average images, because of the inaccurate inference (i.e., inaccurate gradients) on a complete model graph.

Figure 5 shows the changes of the training time, negative log-likelihood, number of non-zero features and number of gradient calculation of the three different methods on the data set of character "c". We can see that the training time and the number of gradient calculation of Grafting-Light are reasonably stable, with small jumps when $M = 30$.

For Grafting, the training time and the number of gradient calculation decrease, but both are larger than those of Grafting-Light. Due to the inaccurate gradient calculation, the Full-Opt.-L1 does not achieve a sparse enough model structure. Therefore, the negative log-likelihood is higher than those of Grafting and Grafting-Light.

7. CONCLUSIONS

We present a fast incremental algorithm called Grafting-Light for feature selection and structure learning of Markov random fields, in which computing the gradients is usually very expensive. Grafting-Light iteratively performs one step of orthant-wise quasi-Newton gradient descent and selects new features. The algorithm is guaranteed to converge to the global optimum and can effectively select significant features. On both synthetic and real data sets, we show that (1) Grafting-Light is much more efficient than Grafting for both feature selection and structure learning; and (2) Grafting-Light can work comparably with the optimal batch method that optimizes over all the features for feature selection, but Grafting-Light is much more efficient and accurate than the batch method for structure learning of MRFs.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] G. Andrew and J.-F. Gao. Scalable training of ℓ_1-regularized log-linear models. In *ICML*, 2007.

[2] O. Banerjee, L. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *JMLR*, (9):485–516, 2008.

[3] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 0(0):1–10, 2007.

[4] N. Friedman. Learning bayesian networks in the presence of missing values and hidden variables. In *ML*, 1997.

[5] N. Friedman. The bayesian structural em algorithm. In *UAI*, 1998.

[6] A. Globerson, T. Koo, X. Carreras, and M. Collins. Exponentiated gradient algorithms for log-linear structured prediction. In *ICML*, 2007.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, (3):1157–1182, 2003.

[8] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. *An introduction to variational methods for graphical models*. M. I. Jordan (Ed.), Learning in Graphical Models, MIT Press, Cambridge, MA, 1999.

[9] K. Kira and L. Rendell. A practical approach to feature selection. In *ICML*, 1992.

[10] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[12] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using ℓ_1-regularization. In *NIPS*, 2006.

[13] D.-C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, (45):503–528, 1989.

[14] A. McCallum. Efficient inducing features of conditional random fields. In *UAI*, 2003.

[15] S. Parise and M. Welling. Structure learning in markov random fields. In *NIPS*, 2006.

[16] S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function spaces. *JMLR*, (3):1333–1356, 2003.

[17] S. Perkins and J. Theiler. Online feature selection using grafting. In *ICML*, 2003.

[18] S. D. Pietra, V. D. Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. on PAMI*, 19(4):380–393, 1997.

[19] T. Sang and S. Buchholz. Introduction to the conll-2000 shared task: Chunking. In *CoNLL*, 2000.

[20] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for ℓ_1 regularization: A comparative study and two new approaches. In *ECML*, 2007.

[21] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *HLT/NAACL*, 2003.

[22] S. Shevade and S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

[23] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*, 2003.

[24] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Royal. Statist. Soc.*, B(58):267–288, 1996.

[25] Y. Tsuruoka, J. Tsujii, and S. Ananiadou. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *ACL*, 2009.

[26] S. Vishvanathan, N. N. Shraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *ICML*, 2006.

[27] M. Wainwright, P. Ravikumar, and J. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *NIPS*, 2006.

[28] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *JMLR*, (3):1439–1461, 2003.

[29] A. Willsky. Multiresolution Markov models for signal and image processing. In *Proc. of the IEEE*, 2002.

[30] E. P. Xing, M. I. Jordan, and R. M. Karp. Feature selection for high-dimensional genomic microarray data. In *ICML*, 2001.

[31] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, 2000.

[32] M. Yuan and Y. Lin. Model selection and estimation in gaussian graphical model. *Biometrika*, 1(94):19–35, 2007.

[33] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. In *NIPS*, 2003.

[34] J. Zhu, E. Xing, and B. Zhang. Primal sparse maximum margin Markov networks. In *SIGKDD*, 2009.

[35] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. Royal. Statist. Soc.*, B(67):301–320, 2005.
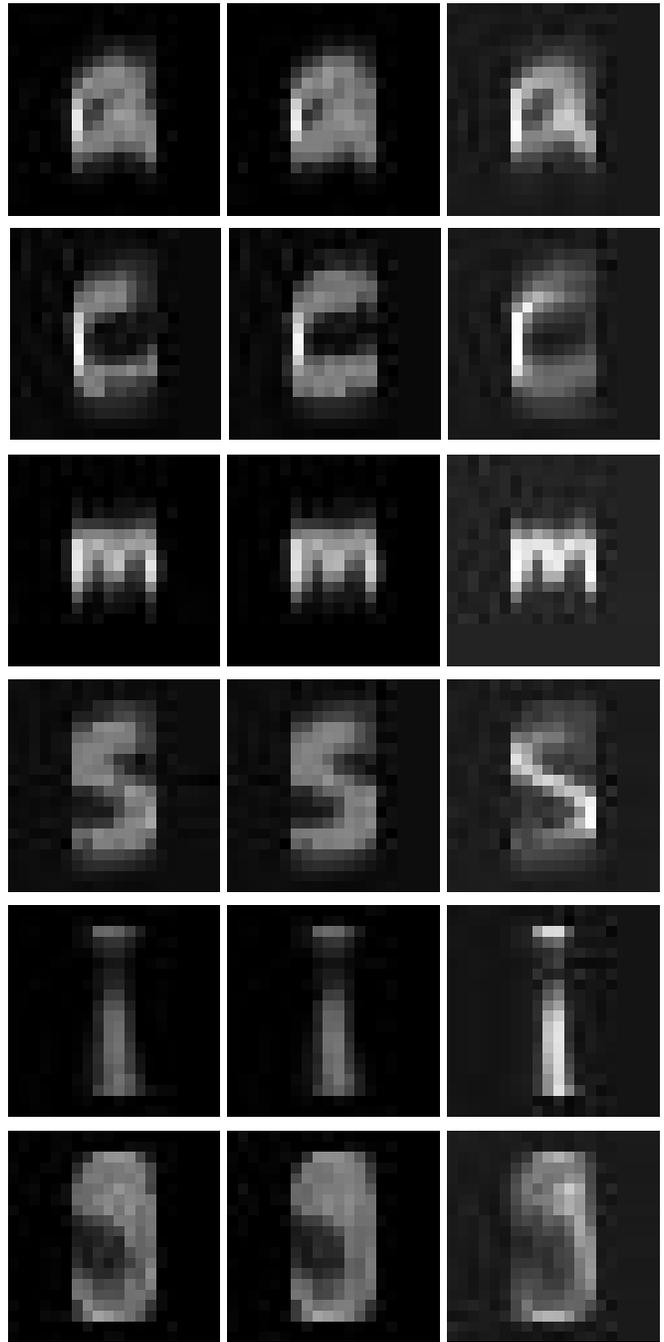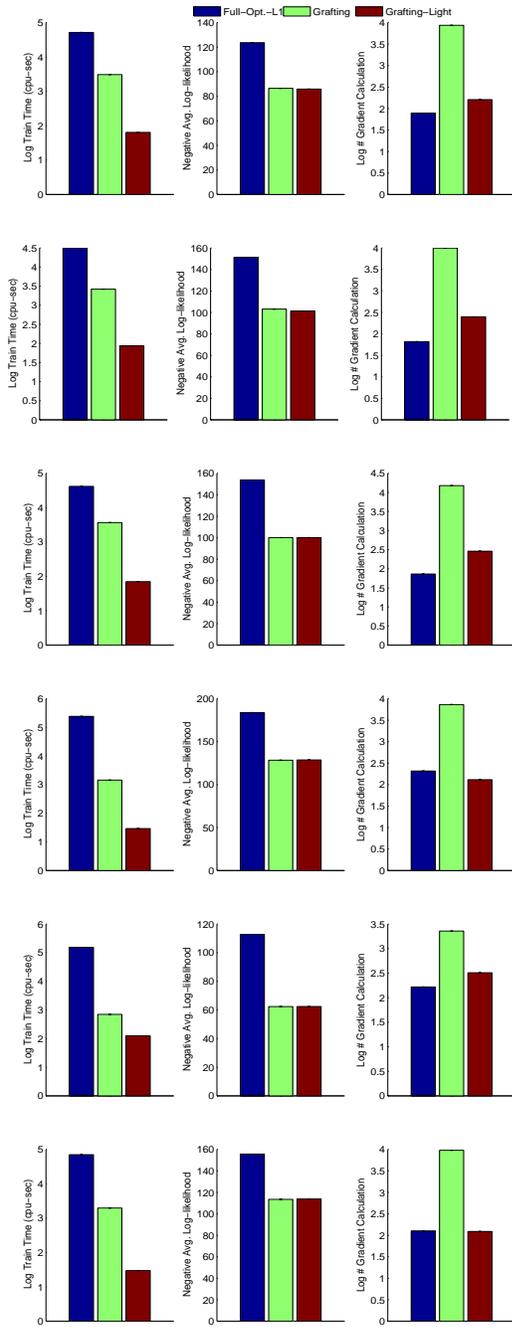
Figure 4: From left to right, the bars in different groups are the training time (in base 10 logarithm), negative per-pixel log-likelihood, and number of gradient calculation (in base 10 logarithm) of three methods. The images are the average images learned by (Left) Grafting-Light; (Middle) Grafting; and (Right) Full-Opt.-L1. From top to down, the rows are for the characters "a", "c", "m", "s", "i", and "g", respectively.
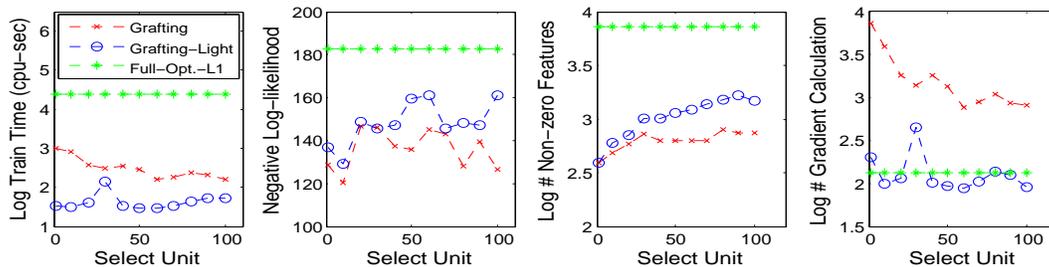

Figure 5: The training time (in base 10 logarithm), negative per-pixel log-likelihood, number of non-zero features (in base 10 logarithm), and number of gradient calculation (in base 10 logarithm) of the three different methods on the character "c".