This paper was submitted as a final project report for CS6424/ECE6424 *Probabilistic Graphical Models and Structured Prediction* in the spring semester of 2016.

The work presented here is done by students on short time constraints, so it may be preliminary or have inconclusive results. I encouraged the students to choose projects that complemented or integrated with their own research, so it is possible the project has continued since this report was submitted. If you are interested in the topic, I encourage you to contact the authors to see the latest developments on these ideas.

Bert Huang
Department of Computer Science
Virginia Tech

Priority Grafting: Prioritizing edge selection with structure mining

Walid Chaabene
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
walidch@vt.edu

Bert Huang
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
bhuang@vt.edu

Abstract

One of the most popular approaches to pairwise Markov Random Fields structure learning is through optimizing the $L1$-regularized log-likelihood. At the end of the optimization task, parameters of irrelevant edges are reduced to zero. The main challenge when using this particular method is the fact that the space of possible edges is extremely large which makes it infeasible for large number of variables. To work around this issue, incremental methods such as `Grafting` were introduced. This method constructs a working set by activating one edge at a time and hence avoids performing inference at each optimization step over the set of all possible edges. The activation step is done by computing the gradient for each feature in the model and activate the edge that possesses the feature with the highest gradient that is above the defined $L1$ coefficient. This however comes at the price of testing every possible edge for activation which grows exponentially with the number of variables and their corresponding features.

This paper introduces our method, `Priority Grafting`, that works around this issue by mining the partially constructed structure of the Markov network, using efficient heuristics, and assigns a priority value to each edge. `Priority Grafting` tests edges for activation in order of decreasing priority. We use a priority queue, an optimized data structure, to store edges and their relevant priorities. Unlike `Grafting`, our method selects the first encountered edge that has a feature gradient that is above the $L1$ coefficient. Inspired by the results of `Grafting Light`, we run the intermediate optimization tasks for a limited number of steps. This provides gain in computational time but might result, in some cases, in some spurious edges that are later removed using a pruning procedure. `Priority Grafting` converges faster than `Grafting` and produces exactly the same structure as shown in our preliminary results.

Introduction

Current structure learning methods of Markov random fields [5] fall into two main categories: constraint-based and score-based methods. Constraint based methods haven't been very popular. On one hand, they lack robustness to statistical noise in the empirical distribution which can give rise to incorrect independence assumptions.On the other hand, they only produce the structure of the model without learning any parameters [6]. Score-based methods construct an objective function by combining a likelihood term with a penalty term to avoid spurious edges. They then search for structures that optimize the objective. One well established approach is based on the $L1$ regularized likelihood [1], which allows the objective to be formulated as a cost function over the set of parameters. At the end of the optimization task, parameters of irrelevant edges are reduced to zero. The main challenge when using this particular methods is the fact that the space of possible edges is extremely

Probabilistic Graphical Models and Structured Prediction Final Project Report. Spring 2016, Virginia Tech.

large which makes it infeasible. To work around this issue incremental methods were introduced. The most popular one is based on `Grafting` [10, 7]. The main idea is to define a set of active features on which we run optimization and a set describing the search space. The active set is initiated as an empty set, then at each iteration, a greedy test is performed to select the feature that would introduce the best improvement to the cost function and we add. The active set is then updated by inserting the selected feature into it. In our implementation of Grafting we activate the edge that contains the selected feature. In other words we include all related features to the edge containing the selected feature. The procedure is performed until an optimal active set is constructed, i.e. no inactive feature is able to significantly improve the cost function in case it is selected. One drawback of `Grafting` is the fact that the selection task is greedy and might become infeasible in models with extremely large number of variables.

A modified version called `Grafting Light` [13] has been introduced as a faster extension to `Grafting`. It aims at reducing the computational cost by running one gradient step each time a new feature is selected. This however lacks a solid theoretical proof that the optimal solution is reached and does not solve the time complexity of the activation test.

The main drawback that we notice in all scoring-based methods is that they do not exploit the partially constructed structure that is being built in the process of incremental structure learning. This presents a valuable source of information that provides information about the importance of candidate edges.

We introduce `Priority Grafting`, a method that mines the partially constructed structure of a given Markov network, using efficient heuristics, and assigns a priority value to each edge. `Priority Grafting` tests edges for activation in order of decreasing priority. We use a priority queue [11], an optimized data structure, to store edges and their relevant priorities. Unlike `Grafting`, our method selects the first encountered edge that has a feature gradient above the $L1$ coefficient. Although, this might result in some extra edges, our method removes them using a pruning procedure.

`Priority Grafting` converges faster than `Grafting` and produces exactly the same structure as shown in our preliminary results.

The intuition behind the defined heuristics is to limit the interaction between two sets of variables to flow over only one edge relating their respective most central nodes. In case we test that edge for relevancy, any relevant connection between the two sets of nodes would result in a positive result. The opposite case informs us that it is very unlikely that there is any relevant relation between the two sets of nodes. `Priority Grafting`, assigns edge priorities to all possible edges between both sets of nodes accordingly.

The paper is structured as follows. The first section introduces the `Grafting`. In the second section we present `Priority Grafting` and summarize the algorithm. Experimental results and future work are presented in the third section.

1 Structure learning using Grafting

In this section we introduce notations for the rest of the paper and we define the problem $L1$-regularized pseudo-likelihood structure learning problem. Throughout this paper we consider the case of the exponential family pairwise Markov Random fields. In this case, the probability of a set of variables X is:

$$p(x) = \frac{1}{Z} \prod_c \phi_c(x) \tag{1}$$

Where, $Z = \sum_x \prod_c \phi_c(x)$, $\phi(x) = \sum_k w_k f_k(x)$, $f_k(x)$ are feature functions and w_k are their corresponding weights. In our case he c denotes cliques consisting of at most two variables.

1.1 $L1$-regularized likelihood

Given a set of labeled training data $D = (x_i, y_i)$, $i = 1 \dots N$, the likelihood based parameter learning of MRF structure is to find the best parameter vector that has the maximum log-likelihood or minimal negative log-likelihood $L(w)$ value, where

Algorithm 1 Grafting based structure learning

repeat
  Select the k^{th} feature using the selection criteria C_1.
  Activate the edge corresponding to the k^{th} feature.
  Optimize the $L1$-regularized L using gradient descent algorithm over the active set.
until no feature is activated

$$L(w) = -\sum_{i=1}^{N} \log p(xi) = -\sum_{i=1}^{N} \left(w^\top f(x_i) - \log Z \right) \tag{2}$$

w and f correspond to the vectors of w_k and f_k respectively. This can be cast as a structure learning problem. In fact, structure learning consists in detecting edges between variables. These edge have corresponding features and weight. An edge is created whenever at least one corresponding weight is learned to be non zero.

The most natural way to approach this problem is to create a set of weights corresponding to all possible edges and minimize $L(w)$ using methods like quasi-Newton [8], stochastic [12], or exponentiated gradient methods [2]. In either case we are required to compute the gradient of L, ∇L. The k^{th} coordinate of ∇L is:

$$\delta_k L = \frac{\partial L}{\partial w_k} = -\sum_{i=1}^{N} f(x_i) + \sum_{i=1}^{N} E_p[f(x_i)] \tag{3}$$

To promote sparsity of the learned weights and hence of the learned Markov network, an $L1$ regularization term is added to ∇L as follows:

$$\mathbb{L}(w) = L(w) + \lambda ||w||_1 \tag{4}$$

This can be seen as introducing a prior for the model parameters using a Gaussian distribution of the form $\mathbb{N}(0, 1/\lambda^2)$. This prior will force irrelevant parameters to have zero values at the solution.

However the set of parameters w_k is combinatorial and grows drastically with the number of variables. Hence, this approach to structure learning can be infeasible for a large number of variables.

1.2 Grafting

Unlike traditional methods, `Grafting` does not runs optimization over the whole set of weights. Instead, it performs a selection step where it activates one weight at a time. After each activation Grafting runs optimization on the active set and learn the corresponding parameters. This can be seen as an incremental feature selection task [9, 4]. Using these parameters `Grafting` runs the next selection test. This method converges when no new parameter is activated. The power of `Grafting` comes from the fact that selection task does not require inference, which is the most expensive operation in the structure learning framework as it has to run at each optimization iteration.

The selection step is performed by selecting the j^{th} feature using the following selection criteria C_1:

$$C_1 : \begin{cases} j = \arg\max_k |\delta_k L| \\ |\delta_j L| > \lambda \end{cases} \tag{5}$$

Grafting-based structure learning steps are summarized in algorithm 1.

The main drawback with `Grafting` is the fact that the search space is large and hence the activation step becomes expensive with large number of variables. On the other hand `Grafting` gradually constructs the structure of the MRF by activating edges at each iteration and hence grows a partial structure at each iteration. Although this partial structure contains rich information about the dependencies and independencies between variables, it is not exploited in the process.

2 Priority Grafting

In this section we present `Priority Grafting`. This method is an extension of `Grafting` that performs efficient partial structure mining heuristics. These heuristics are used to assign edge priorities to be followed when running the activation step. Unlike `Grafting`, this method selects the first encountered feature that has a gradient above the $L1$ coefficient. This takes advantage of the priority ordering and results in an early stopping of the selection task.

2.1 Priority assignment

In this section we treat the partially constructed MRF structure as a graph $G(E, V)$, where E refers to the set of constructed edges and V denotes the set of all model variables.

All candidate edges, i.e. edges that are in the search space are initiated with the same priority. At each priority assignment iteration we aim at reducing the priority of a set of edges that are proved to be less relevant than what they were assumed to be at earlier iterations. This is done by mining the partially constructed structure of the MRF.

The first step is to detect central nodes V_c in G. A central node is a node that has a number of neighbors that exceeds a predefined threshold τ_n. We then construct all possible edges between the set of central nodes that are not in E and we denote it by E_c (not a subset of E). Each edge (i, k) in E_c is scored by the cardinality of the set $\hat{E}_{(i,k)}$ that contains all possible inactive edges between N_i and N_k referring respectively to the neighbors of i, and the neighbors of k, i.e.:

$$score_{(i,k)} = \#(\hat{E}_{(i,k)}) \tag{6}$$

The next step is to select the highest scoring edge from E_c, i.e.

$$(i, j) = \arg \max_{(i,j) \in E_c} score_{i,j} \tag{7}$$

Assuming that (i, j) is selected and that the set of features K refers to the edge (i, j). Then (i, j) is considered not relevant if all features f in K satisfies the following criteria:

$$C_2 : |\delta_f L| < \lambda \tag{8}$$

In case, the edge (i, j) is found to be non relevant then we reduce the priority of all edges in $\hat{E}_{(i,k)}$ by 1.

The intuition behind these heuristics is the fact that we limit the flow of influence between variables in the set N_i and the set N_k to pass only via the the edge (i, j). Hence in case there is a relevant relation between these two sets of variables it will force (i, j) to be relevant.

2.2 Structure learning with Priority Grafting

`Pruning to graft` algorithm is to be viewed as an extension of `Grafting`. The key difference is the structure mining heuristics that assign priority values to candidate edges and provides a priority ordering for the iterations of the selection task. This is meant to boost the speed of the selection task by down scoring irrelevant edges. This is most effective when using a priority queue where we iterate over the edges following a decreasing order of their priority. However the priority reassignment procedure comes at a computational cost and hence it is unrecommended to perform it at the first grafting steps, mainly because the partially constructed structure is not large enough to be informative. Hence we consider the density of the partially constructed graph which is given by:

$$d = \frac{\#E}{n^2} \tag{9}$$

We define a threshold τ_d above which we start performing the priority reassignment tasks.

Algorithm 2 Priority Grafting

Initialize priority queue using same priorities for all edge.
repeat
  Optimize the $L1$-regularized L using gradient descent algorithm over the active set for a limited number of iterations
  if Graph density above threshold **then**
    Perform priority queue update.
  end if
  Iterate over the priority queue and select the first feature f satisfying C_2.
  Activate the edge corresponding to feature f.
until no feature is activated
Optimize the $L1$-regularized L using gradient descent algorithm over the active set until convergence.
Perform pruning of non relevant edges.

Inspired by `Grafting Light`, we run the intermediate optimization tasks for a limitted number of iterations, this might result in extra edges. After the last optimization operation which is run until convergence, the weights of these edges are reduced to very small values. We use a pruning step in which we deactivate edges that have corresponding weights that less than a predefined threshold ϵ. The steps of `Priority Grafting` are shown in algorithm 2.

3 Experiments

Our aim is to show that `Priority Grafting` learns the same structure as the traditional baseline method of `Grafting` but does it noticeably faster.

3.1 Data

Our experiments were performed on the Mushroom data set made public on UCI website [1]. The dataset consists of 5644 data points each containing 23 variables. Each variable has a different number of possible states that range from 2 to 12. We assume that the dataset is a sequence of random variables that are independent and identically distributed.

3.2 Pseudo negative log likelihood as performance metric

The pseudo-likelihood [3] method has been very popular in replacing computing the likelihood either in the learning or the testing step. The goal is to replace the likelihood by a more tractable objective. Although we use the likelihood formulation for the training of the model to get accurately trained models, we use the pseudo-log likelihood for testing. The pseudo likelihood formulation is based on the following approximation:

$$p(x) = \prod_i p(x_i | x \setminus \{x_i\}) = p(x_i | N_i) \tag{10}$$

Where N_i is the set of variables in the Markov Blanket (direct neighbors in the Markov network) of x_i. Since each variable has a limited number of neighbors, this yields a drastic gain in running time complexity compared with the the classic likelihood formulation. In fact, taking the above approximation into consideration we get:

$$p(x) = \prod_i \frac{\exp\left(\phi(x_i) + \sum_{j \in N_i} \phi(x_j, x_i)\right)}{\sum_{\hat{x}_i} \exp\left(\phi(x_i) + \sum_{j \in N_i} \phi(x_j, \hat{x}_i)\right)} \tag{11}$$

[1] www.uci.edu

For M different data points $x^{(m)}$, the negative pseudo log likelihood is hence given by:

$$\begin{aligned} \hat{nll} &= \frac{1}{M} \sum_m \left(\log \prod_i p(x_i^{(m)} | N_i) \right) \\ &= \frac{1}{M} \sum_m \left(\sum_i \log p(x_i^{(m)} | N_i) \right) \\ &= \frac{1}{M} \sum_m \sum_i \left(\phi(x_i) + \sum_{j \in N_i} \phi(x_j, x_i) - \log \sum_{\hat{x}_i} \exp \left(\phi(x_j, \hat{x}_i) + \sum_{j \in N_i} \phi(x_j, \hat{x}_i) \right) \right) \end{aligned} \tag{12}$$

3.3 Settings and Results

We split the data set into training and testing sets. Each containing half of the total data points. We set τ_d to 0.05 and τ_n to 4. Figures 1 and 2 show that `Priority Grafting` learns the same structure as `Grafting`. It is shown in tables 1 and 2 that both methods yield the same negative pseudo log likelihood values. However `Priority Grafting` comes with less computational time for edge selection.

((a)) Priority Grafting ((b)) Grafting

Figure 1: Learned structure ($\lambda = 0.7$)

((a)) Priority Grafting ((b)) Grafting

Figure 2: Learned structure ($\lambda = 0.6$)

4 Conclusion and future work

In this paper we present preliminary work of the development of `Priority Grafting`, an extension of the previously introduced grafting method. `Priority Grafting` aims at reducing the running time of `Grafting` using edge priority done by exploiting the partially constructed structure of MRF

Table 1: $\lambda = 0.7$

Method	Avg. edge selection time (ms)	$\hat{nll}$	number of edges
Grafting	17.04	32.4519	12
Priority Grafting	3.64	32.4519	12

Table 2: $\lambda = 0.6$

Method	Avg. edge selection time (ms)	$\hat{nll}$	number of edges
Grafting	15.24	31.2475	17
Priority Grafting	3.21	31.2475	17

and assigning different priority values to the inactive edges. The priority ordering is exploited with a priority queue data structure. Since `Priority Grafting` selects the first feature having a gradient above the $L1$ coeficient, iterating over the priority queue in a decreasing priority order makes feature selection noticeably faster than in the case of `Grafting`.

Experimental results on the Mushroom data showed that `Priority Grafting` produces the same structure as `Priority Grafting`. It also learns the exact same weights and produces the same negative pseudo log likelihood measures. We have also showed that selection time is noticeably lower in the case of `Priority Grafting`.

Future work will aim at refining the heuristics for graph mining and optimizing it to reduce its computational cost. We will also apply `Priority Grafting` on other datasets that have more important dependency relations between their variables. To further prove the advantage of `Priority Grafting` we will focus on datasets with an important variable dimension where the cost of feature selection for `Grafting` is very high.

References

[1] G. Andrew and J. Gao. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40. ACM, 2007.

[2] A. Globerson, T. Y. Koo, X. Carreras, and M. Collins. Exponentiated gradient algorithms for log-linear structured prediction. In *Proceedings of the 24th international conference on Machine learning*, pages 305–312. ACM, 2007.

[3] C. Gourieroux, A. Monfort, and A. Trognon. Pseudo maximum likelihood methods: Theory. *Econometrica: Journal of the Econometric Society*, pages 681–700, 1984.

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

[5] R. Kindermann, J. L. Snell, et al. *Markov random fields and their applications*, volume 1. American Mathematical Society Providence, RI, 1980.

[6] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[7] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using l_1-regularization. In *Advances in neural Information processing systems*, pages 817–824, 2006.

[8] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[9] A. Y. Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[10] S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *The Journal of Machine Learning Research*, 3:1333–1356, 2003.

Table 3: $\lambda = 0.6$

Method	Avg. edge selection time (s)	$\hat{nll}$	number of edges
Grafting	0.01524 s	200.38	
Priority Grafting	0.00321 s	199.44	

[11] P. van Emde Boas, R. Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Mathematical Systems Theory*, 10(1):99–127, 1976.

[12] S. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd international conference on Machine learning*, pages 969–976. ACM, 2006.

[13] J. Zhu, N. Lao, and E. P. Xing. Grafting-light: fast, incremental feature selection and structure learning of markov random fields. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 303–312. ACM, 2010.