STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **4**, September 2016, pp 243–251.
Published online in International Academic Press (www.IAPress.org)

Learning Unknown Structure in CRFs via Adaptive Gradient Projection Method

Wei Xue [1,*], Wensheng Zhang [1,2]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.*
[2] *Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.*

(Received: 22 April 2016; Accepted: 19 May 2016)

Abstract This paper focuses on learning unknown structure in conditional random fields (CRFs), especially learning both the structure and parameters of a CRF model simultaneously. By adding the l_2-regularization to node parameters and the group l_1-regularization to edge parameters, this structure learning problem can be cast as a convex minimization problem. Then an adaptive gradient method is proposed to solve the minimization problem. Extensive simulation experiments are presented to show the performance of the proposed approach for learning unknown structure.

Keywords Conditional random field, Structure learning, Constrained optimization, Two-point stepsize, Gradient projection, Nonmonotone line search

AMS 2010 subject classifications 90C30, 65K05

DOI: 10.19139/soic.v4i3.228

1. Introduction

In this paper, we consider the unknown structure learning problems in conditional random fields (CRFs) [11]. CRFs are undirected graphical models that can be used to represent conditional probability distributions $p(\mathbf{y}|\mathbf{x})$ compactly, where **y** represents the labels and **x** is the observed features. An important property of CRFs is their ability to cope with large and redundant features and to show the structural relationships between the output labels. Here, we consider CRFs with pairwise potentials:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\langle ij \rangle} \psi_{ij}(y_i, y_j, \mathbf{x}) \prod_i \psi_i(y_i, \mathbf{x}),$$

where $\langle ij \rangle$ is a product over all edges in the graph, ψ_{ij} is an edge potential, and ψ_i is a node potential. We focus on binary states, i.e., $y_i \in \{0, 1\}$, and assume that the node and edge potentials have the following form:

$$\psi_i(\cdot, \mathbf{x}) = \left(e^{\mathbf{v}_{i,1}^T \mathbf{x}_i}, e^{\mathbf{v}_{i,2}^T \mathbf{x}_i} \right) \text{ and } \psi_{ij}(\cdot, \cdot, \mathbf{x}) = \begin{pmatrix} e^{\mathbf{w}_{ij,11}^T \mathbf{x}_{ij}} & e^{\mathbf{w}_{ij,12}^T \mathbf{x}_{ij}} \\ e^{\mathbf{w}_{ij,21}^T \mathbf{x}_{ij}} & e^{\mathbf{w}_{ij,22}^T \mathbf{x}_{ij}} \end{pmatrix},$$

where $\mathbf{x}_i = [1, \mathbf{g}, \mathbf{f}_i]$ is the node feature, $\mathbf{x}_{ij} = [1, \mathbf{g}, \mathbf{f}_i, \mathbf{f}_j]$ is the edge feature, with **g** being global features shared across nodes and $\mathbf{f}_i$ being the node's local features, **v** denotes the node weights, and **w** denotes the edge weights.

[*] Correspondence to: Wei Xue (Email: mailweixue@163.com).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Here, we set $\mathbf{v}_{i,2} = 0$ and $\mathbf{w}_{ij,22} = 0$ to ensure identifiability, otherwise the model would be over-parameterized. Let $\boldsymbol{\theta} = [\mathbf{v}, \mathbf{w}]$ for all the parameters and $F(\mathbf{x}, \mathbf{y})$ for all the features, we can write the model more succinctly as

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}^T F(\mathbf{x}, \mathbf{y}))}{Z(\boldsymbol{\theta}, \mathbf{x})},$$

where $Z(\boldsymbol{\theta}, \mathbf{x}) = \sum_{\mathbf{y}'} \exp(\boldsymbol{\theta}^T F(\mathbf{x}, \mathbf{y}^{'}))$. Hence, the negative log-likelihood is

$$nll(\boldsymbol{\theta}) = \sum_{n=1}^{N} -\boldsymbol{\theta}^T F(\mathbf{x}_n, \mathbf{y}_n) + \sum_{n=1}^{N} \log Z(\boldsymbol{\theta}, \mathbf{x}_n).$$

However, computing the gradient of $nll(\boldsymbol{\theta})$ is expensive, since it requires an inference algorithm. To reduce the computational cost, we change the objective function to the pseudo likelihood [2] defined as

$$PL(\mathbf{y}^n|\mathbf{x}^n) = \prod_i \frac{\exp(\boldsymbol{\theta}_i^T F_i(\mathbf{x}, \mathbf{y}))}{Z_i},$$

where $\boldsymbol{\theta}_i = (\mathbf{v}_i, \{\mathbf{w}_{ij}\}_{j \in n_i})$ is the parameter for the i's Markov blanket, Z_i is the local partition function, $\mathbf{F}_i$ is the local feature vector, and n_i is the neighbor of i in the graph.

Recently, a popular learning paradigm to generative structure learning is to impose an l_1-regularization on the parameters of the model, leading to a sparse graph. However, in more general scenarios many features are associated with each edge. In this situation, group l_1-regularization-based methods that jointly reduce groups of parameters to zero can be used to achieve sparsity. More specifically, in this paper we focus on the following group regularized structure learning function:

$$J(\boldsymbol{\theta}) = nll(\boldsymbol{\theta}) + \lambda_1 \|\mathbf{v}\|_2^2 + \lambda_2 \Omega(\mathbf{w}), \tag{1}$$

where we force an l_2-regularization on the node weights $\mathbf{v}$ and place an regularization $\Omega(\mathbf{w})$ on the edge weights $\mathbf{w}$, where $\Omega(\mathbf{w}) = \sum_{b=1}^{B} (\sum_{i \in b} |w_i|^{\alpha})^{1/\alpha} = \sum_b \|\mathbf{w}_b\|_\alpha$. If $\alpha = 1$, $\Omega(\mathbf{w})$ is the standard l_1-regularization, namely, $\Omega_1(\mathbf{w}) = \|\mathbf{w}\|_1$. A more computationally appealing alternative is to use $\alpha = \infty$, that is, $\Omega_\infty(\mathbf{w}) = \sum_b \|\mathbf{w}_b\|_\infty = \sum_b \max_{i \in b} |w_i|$. This choice of α often yields sparsity at the group level, see [16].

This paper proposes a nonmonotone adaptive gradient projection method to minimize the learning problem (1) with $\alpha = \infty$ in $\Omega(\mathbf{w})$. Section 2 describes the proposed method in detail. Numerical results are presented to illustrate the performance of our approach in Section 3. Finally, we have a conclusion section.

2. Algorithm

In this section, we show how to minimize the regularized function (1) with $\Omega_\infty(\mathbf{w})$. We begin with some preliminary results to introduce the proposed algorithm.

2.1. *Preliminary results*

2.1.1. ***Smooth bound constrained optimization.*** To solve the problem (1) efficiently, we first convert it to a smooth optimization problem with linear constraints by introducing auxiliary variables (one for each set) that are constrained to be the maximum value of a set. Note that minimizing the group l_1-regularization in the set $S = s_1, \ldots, s_n$ is equivalent to minimizing the ∞-norm $\|(s_1, \ldots, s_n)\|_\infty = \max_i\{|s_i|\}$, so we have the following smooth bound constrained minimization problem:

$$\min_{\boldsymbol{\alpha}, \mathbf{v}, \mathbf{w}} \mathrm{nll}(\boldsymbol{\theta}) + \lambda_1 \|\mathbf{v}\|_2^2 + \lambda_2 \sum_s \alpha_s, \tag{2}$$
$$\text{s.t.} \; -\alpha_s \leq w_{sk} \leq \alpha_s, \forall_s, \forall_k \in s,$$

where s indexes the edges.

2.1.2. ***Gradient-based projection methods.*** For convenience, we use a triple $\mathbf{x}_k = \{\boldsymbol{\alpha}, \mathbf{v}, \mathbf{w}\}$ to denote the concatenation of all variables and use $f(\mathbf{x}_k)$ to denote the value of the objective function at the k-th iteration. In general, we cannot compute the solution to problem (2) analytically and must adopt iterative-based algorithms. Among the existing bound constrained optimization techniques, projected gradient method is a simple and effective one.[†] A common variant of projected gradient methods computes a descent search direction at the k-th iteration by finding the Euclidean norm projection of a scaling steepest descent direction onto a feasible set, see for example [5, 14, 15]. Our approach belongs to the class of projected gradient methods.

Let $\mathcal{P}$ denote the projection operator, β be the scale factor for the steepest descent direction, and t be a stepsize chosen by a line search, then the iteration form can be expressed as $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k \mathbf{d}_k$, where $\mathbf{d}_k = \mathcal{P}(\mathbf{x}_k - \beta_k \nabla f(\mathbf{x}_k)) - \mathbf{x}_k$. However, the use of classical steepest descent direction may lead to slow convergence rate, because this procedure may result in a sawtooth phenomenon. In [1], Barzilai and Borwein (BB) proposed an ingenious gradient algorithm, two-point stepsize gradient method, in which β_k along the steepest descent direction is determined by

$$\beta_k^{(BB1)} = \frac{\mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}} \text{ or } \beta_k^{(BB2)} = \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^T \mathbf{y}_{k-1}}, \tag{3}$$

where $\mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$, and $\mathbf{y}_{k-1} = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$. For unconstrained optimization, the two-point stepsize gradient (BB) method outperforms the classical steepest descent method both in theory and in real computations. Due to its simplicity and numerical efficiency, this method is very useful in solving large-scale smooth unconstrained minimization problems, and several modified versions are proposed. In [8], Dai et al. interpreted the choice for the stepsize β_k in the BB method from the angle of interpolation and proposed two variants. Numerical results reported in [8] suggest that improvements have been achieved. Along this line, Xiao et al. [19] proposed another two choices of β_k by introducing two modified quasi-Newton secant equations developed in [22, 18]. Numerical experiments show that their method seems to be better. Based on the work in [3], Biglari and Solimanpur [4] recently proposed four choices of β_k by considering a fourth order conic model applied to the objective function:

$$\beta_k^{(BS1)} = \frac{\mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^T \hat{\mathbf{y}}_{k-1}}, \ \beta_k^{(BS2)} = \frac{\mathbf{s}_{k-1}^T \hat{\mathbf{y}}_{k-1}}{\hat{\mathbf{y}}_{k-1}^T \hat{\mathbf{y}}_{k-1}},$$

$$\beta_k^{(BS3)} = \frac{\mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}{\xi \mathbf{s}_{k-1}^T \mathbf{y}_{k-1}} \text{ and } \beta_k^{(BS4)} = \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\xi \mathbf{y}_{k-1}^T \mathbf{y}_{k-1}},$$

where $\hat{\mathbf{y}}_{k-1} = \mathbf{y}_{k-1} + \frac{\phi \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}$, $\phi = 4(f(\mathbf{x})_{k-1} - f(\mathbf{x})_k) + 2(\mathbf{g}_k + \mathbf{g}_{k-1})^T \mathbf{s}_{k-1}$, and $\xi = 1 + \phi/(\mathbf{s}_{k-1}^T \mathbf{y}_{k-1})$. It is not difficult to see that in essence $\beta_k^{(BS3)}$ and $\beta_k^{(BS4)}$ are the same with $\beta_k^{(BB1)}$ and $\beta_k^{(BB2)}$, respectively, which can be viewed as scaling BB methods.

2.1.3. ***Adaptive stepsize selection rules.*** In the last decades, stepsize selection rules in gradient-type methods have received an increasing interest from both the theoretical and the practical points of view. In [23], Zhou et al. proposed an adaptive selection strategy which takes the form of

$$\beta_k^{(Z)} = \begin{cases} \beta_k^{(BB2)} & \text{if } \frac{\beta_k^{(BB2)}}{\beta_k^{(BB1)}} < \kappa_1, \\ \beta_k^{(BB1)} & \text{otherwise,} \end{cases} \tag{4}$$

where $\kappa_1 \in (0, 1)$. Their numerical results show that this strategy can improve its practical performance. Based on (4), Yu et al. [20] suggested a different one:

$$\beta_k^{(Y)} = \begin{cases} \beta_k^{(BB1)} & \text{if } k \text{ is odd or } \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\|\mathbf{s}_{k-1}\| \|\mathbf{y}_{k-1}\|} \geq \kappa_2, \\ \beta_k^{(BB2)} & \text{otherwise,} \end{cases} \tag{5}$$

[†] For bound constrained optimization, the projection under the Euclidean norm is the standard orthogonal projection.

where $\kappa_2 < 1$ is close to 1. It is clear that $\frac{\beta_k^{(BB2)}}{\beta_k^{(BB1)}} = \frac{(\mathbf{s}_{k-1}^T \mathbf{y}_{k-1})^2}{(\|\mathbf{s}_{k-1}\| \|\mathbf{y}_{k-1}\|)^2}$. Even so, the switch condition in Eq. (5) is different to that in Eq. (4).

2.2. Proposed algorithm

Note that $\beta_k^{(BS1)}/\beta_k^{(BS2)} \geq 1$, and based on the previous work in [23, 20], we propose the following stepsize selection strategy which can adaptively choose a small stepsize or a large one at each iteration:

$$\beta_k^{(New)} = \begin{cases} \beta_k^{(BS1)} & \text{if } k \text{ is odd or } \frac{\|\mathbf{s}_{k-1}\| \|\hat{\mathbf{y}}_{k-1}\|}{\mathbf{s}_{k-1}^T \hat{\mathbf{y}}_{k-1}} \geq \kappa, \\ \beta_k^{(BS2)} & \text{otherwise}, \end{cases} \tag{6}$$

where $\kappa \in [0.5, 1)$. Obviously, the judgment criterion in Eq. (6) is different to those in Eqs. (4) and (5).

Once β_k is determined, the search direction $\mathbf{d}_k$ can be fixed. And then a suitable learning rate t_k along $\mathbf{d}_k$ should be found to compute the next iteration $\mathbf{x}_{k+1}$. As argued by [14], due to its use of the steepest descent direction, the nonmonotone two-point stepsize strategy can also be used to significantly speed up the convergence of gradient projection algorithms. Here, we consider a nonmonotone line search that differs from the standard Armijo line search [12]. The standard Armijo line search requires the objective function values to decrease monotonically at each iteration. This requirement may cause the sequence of iterations to fall into the bottom of a curved narrow valley, resulting in a slow convergence. To overcome this difficulty, an alternative is to allow an occasional increase in the objective function values at each iteration. In other words, we can adopt a nonmonotone line search. The earliest nonmonotone line search was developed by Grippo, Lampariello, and Lucidi (GLL) [9] which permits some growth in the function values as the iteration process, and the BB method has received increasing attention since its global convergence was proved under the GLL line search [13]. As pointed out in [21], although the GLL line search works well in many cases, there exist some drawbacks. For example, some good function values may be discarded, and the numerical performance depends very much on the choice of a predefined memory parameter. For this, Zhang and Hager [21] proposed an improved nonmonotone line search which requires that the average of the successive function values decreases. Their numerical results show that the new line search technique outperforms the monotone line search and the GLL. In this paper, we pay attention to the Zhang-Hager line search. To clarify further our method, we recall this line search briefly. The Zhang-Hager line search is used to find a stepsize t that satisfies the following Armijo-type condition:

$$f(\mathbf{x}_k + t\mathbf{d}_k) \leq C_k + \nu t \nabla f(\mathbf{x}_k)^T \mathbf{d}_k, \tag{7}$$

where $\nu \in (0, 1)$, and C_k is updated by

$$Q_{k+1} = \eta_k Q_k + 1,\ C_{k+1} = \frac{\eta_k Q_k C_k + f(\mathbf{x}_{k+1})}{Q_{k+1}},$$

with $\eta_k \in (0, 1)$, $Q_0 = 1$, and $C_0 = f(\mathbf{x}_0)$.

Remark 1

If $\eta_k = 0$ for each k, then the Zhang-Hager line search reduces to the standard (monotone) Armijo line search. As η_k approaches 1, this scheme becomes more nonmonotone, treating all the previous objective function values with equal weight when we compute C_k.

In our problem, the projection operator $\mathcal{P}$ onto the convex feasible set $\mathcal{F} = \{\{\boldsymbol{\alpha}, \mathbf{v}, \mathbf{w}\} | \forall_k \in s, -\alpha_s \leq w_{sk} \leq \alpha_s\}$ is defined as $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \mathcal{F}} \|x - u\|_2^2$, which may be very expensive to solve at each iteration for large scale optimization. Note that the projection is separable across groups, which implies that we just have to solve

$$\min_{\mathbf{w}, \alpha'} \|(\mathbf{w}', \alpha') - (\mathbf{w}_s, \alpha_s)\|_2^2, \tag{8}$$
$$\text{s.t. } \forall_i, -\alpha' \leq w_i' \leq \alpha'$$

Algorithm 1 Adaptive Gradient Projection Method (AGPM)

```
 1: Input: $\mathbf{x}_0$, regularization parameters $\lambda_1$ and $\lambda_2$, threshold parameters $\kappa$ and $\epsilon$, sufficient decrease parameter $\nu$,
    $\beta_{\max}$, $\beta_{\min}$, $C_0$, $Q_0$, and $\eta_k$.
 2: while “not converged” do
 3:    Compute $f(\mathbf{x}_k)$ and $\nabla f(\mathbf{x}_k)$.
 4:    if $k = 0$ then
 5:       $\beta_k = 1$.
 6:    else
 7:       Compute $\beta_k$ by Eq.(6).
 8:    end if
 9:    Update $\beta_k = \min(\beta_{\max}, \max(\beta_k, \beta_{\min}))$.
10:    $\bar{\mathbf{x}}_k = \mathbf{x}_k - \beta_k \nabla f(\mathbf{x}_k)$.
11:    for each group $s$ do
12:       Compute $\mathcal{P}(\bar{\mathbf{w}}_s, \bar{\boldsymbol{\alpha}}_s)$ by solving (8).
13:    end for
14:    Compute $\mathbf{d}_k = \mathcal{P}(\bar{\mathbf{x}}_k) - \mathbf{x}_k$.
15:    Compute $t_k$ by Eq.(7).
16:    Update $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k \mathbf{d}_k$.
17:    Let $k = k + 1$.
18: end while
```

for each $(\mathbf{w}_s, \alpha_s)$ independently. Hence, we can compute the optimal projection by solving a small-scale linear constrained minimization problem for each group.

Based on the analysis above, we now describe our adaptive gradient projection method with the (nonmonotone) Zhang-Hager line search for solving problem (2) in Algorithm 1.

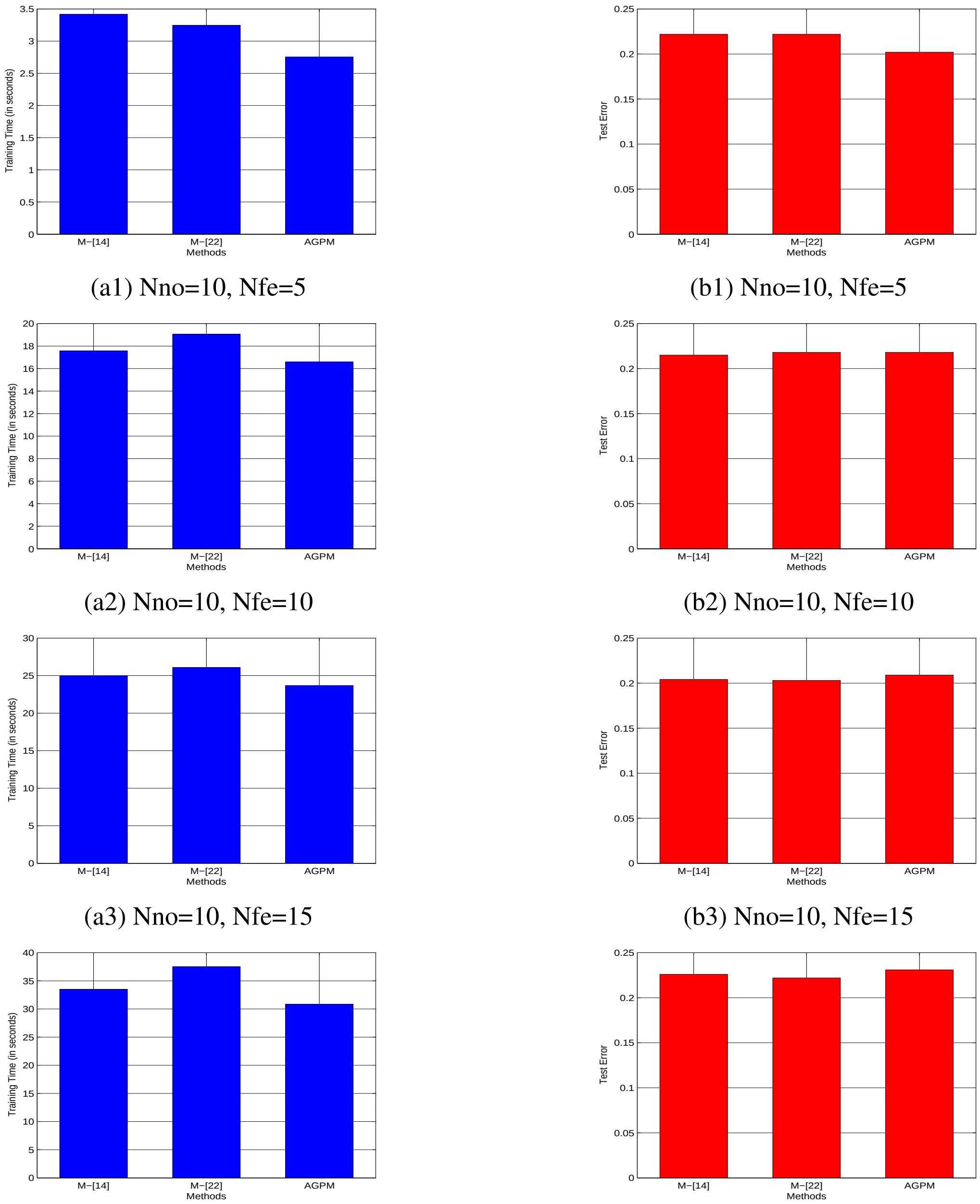
Remark 2
At line 2 in Algorithm 1, we judge whether the current point is stationary. If $\|P(\mathbf{x}_k - \nabla f(\mathbf{x}_k)) - \mathbf{x}_k\| < \epsilon$, stop, declaring that $\mathbf{x}_k$ is stationary.

Remark 3
If $\beta_k > 0$, then the generated search direction $\beta_k \nabla f(\mathbf{x}_k)$ is descent. However, the condition $\beta_k > 0$ may not be always fulfilled, and the descent property can no longer be guaranteed. To deal with this issue, we should keep the sequence $\{\beta_k\}$ uniformly bounded, that is to say for sufficiently large $\beta_{\max} > 0$ and sufficiently small $\beta_{\min} > 0$, β_k is forced as $\beta_k = \min(\beta_{\max}, \max(\beta_k, \beta_{\min}))$, see line 9 in Algorithm 1. This strategy is also adopted in [17, 19].

3. Experiments

In this section, we conduct numerical experiments to evaluate the performance of the proposed method for solving structure learning problems by using artificial data from different CRFs. We compare AGPM with the method used in [14] and the method using $\beta_k^{(Z)}$ [23]. For convenience, we denote the two methods as M-[14] and M-[23], respectively. All experiments are implemented under Matlab on a PC with Windows 7 operating system.

We chose the graph structures randomly, and the probability of each edge is 0.5. We sample random node weights $\mathbf{v}_i \sim \mathcal{N}(0, 1)$ and edge weights $\mathbf{w}_{i,j} \sim U(-b, b)$, where $b \sim \mathcal{N}(0, 1)$ for each edge. We draw 100/500/1000 training samples and 100 test samples from the exact distribution $p(\mathbf{y}|\mathbf{x})$, respectively, and set $\lambda_1 = \lambda_2 = 0.5$, $M = 10$, $\nu = 10^{-4}$, $\epsilon = 10^{-4}$, $\beta_{\min} = 1/\beta_{\max} = 10^{-10}$, $\kappa = \kappa_1 = 0.5$, $\kappa_2 = 0.9$, $\eta = 0.7$. In the following, “Nno” refers to the number of nodes, “Nfe” refers to the number of features for each node, “Ntr” refers to the number of examples to use for training, and “Nte” refers to the number of examples to use for test. To evaluate the learning performance, we measure the training time in seconds (Time) and the test error (Err).



(a1) Nno=10, Nfe=5

(b1) Nno=10, Nfe=5

(a2) Nno=10, Nfe=10

(b2) Nno=10, Nfe=10

(a3) Nno=10, Nfe=15

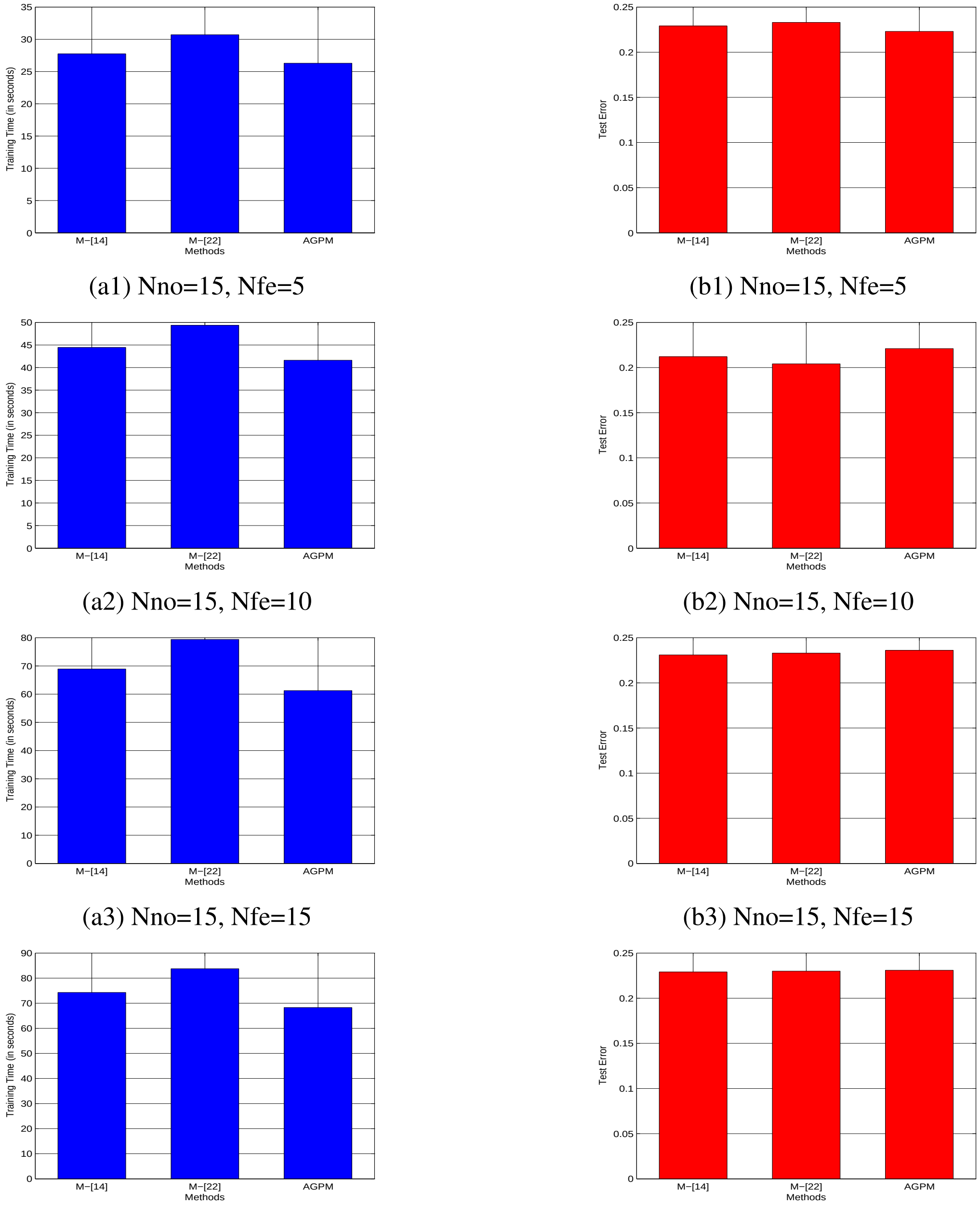
(b3) Nno=10, Nfe=15

(a4) Nno=10, Nfe=20

(b4) Nno=10, Nfe=20

Figure 1. Numerical results of “Time” and “Err” with Ntr=500 and Nte=100.

We first create the data from a 5/10/15-node CRF and use 5/10/15/20 local features for each node sampled from a standard normal distribution, and take Ntr=100, Nte=100. The numerical results are presented in Table 1. As shown in Table 1, the obtained test errors of all methods nearly coincide. While from the view of training time, we can see that the proposed method AGPM is the fastest among those compared in this set of test. Additionally, it is observed that the performance of the proposed adaptive stepsize selection strategy outperforms that of [23].



(a1) Nno=15, Nfe=5

(b1) Nno=15, Nfe=5

(a2) Nno=15, Nfe=10

(b2) Nno=15, Nfe=10

(a3) Nno=15, Nfe=15

(b3) Nno=15, Nfe=15

(a4) Nno=15, Nfe=20

(b4) Nno=15, Nfe=20

Figure 1 (cont'd): Numerical results of "Time" and "Err" with Ntr=500 and Nte=100.

In the second test, we draw 500 training samples and 100 test samples. Figure 1 reports the experimental results of training time and test error of all methods in the form of histograms. Comparing the blue histograms in Figure 1, we clearly see that our method AGPM is superior to the others for all situations.

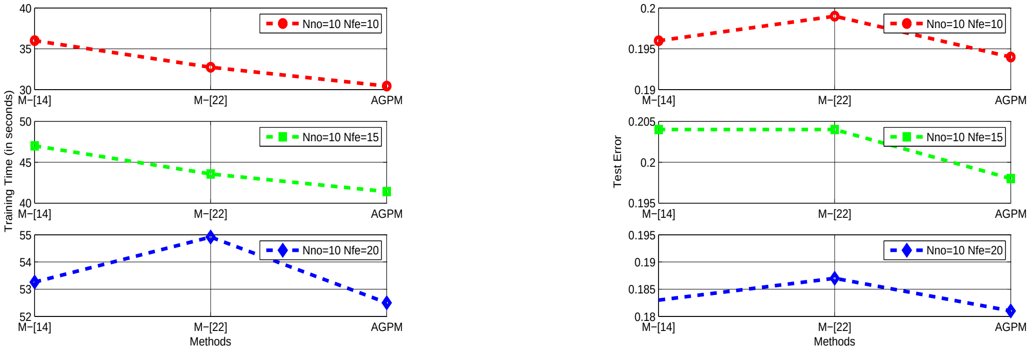
Finally, we consider a larger training date including 1000 training samples, and the data sets are generated from a 10-node CRF and use 10/15/20 local features for each node. The results of AGPM, M-[14] and M-[23] are presented in Figure 2. The left plot in Figure 2 shows that the proposed method AGPM requires less time than

Table 1. Numerical results of “Time” and “Err” with Ntr=100 and Nte=100.

Datasets	M-[14]		M-[22]		AGPM	
Nno/Nfe	Time	Err	Time	Err	Time	Err
5/5	1.111452	0.282	1.299522	0.276	**0.832011**	**0.272**
5/10	2.395456	0.250	2.006264	0.250	**1.154688**	**0.244**
5/15	2.393800	**0.272**	2.772199	0.278	**1.538309**	0.278
5/20	2.640080	**0.302**	3.165659	0.304	**1.981395**	**0.302**
10/5	4.564721	**0.273**	5.745670	**0.273**	**3.423104**	**0.273**
10/10	6.382067	**0.286**	8.548289	0.287	**4.999761**	0.288
10/15	8.194089	0.331	11.791843	**0.329**	**6.779315**	0.331
10/20	11.860101	0.323	16.150115	**0.320**	**9.378568**	0.323
15/5	11.257542	**0.281**	15.223453	**0.281**	**8.943181**	0.282
15/10	17.566537	0.281	24.840462	0.279	**13.923254**	**0.276**
15/15	25.650972	**0.323**	36.425686	0.325	**20.123700**	0.325
15/20	30.545750	0.304	45.989224	**0.301**	**25.622546**	0.302

M-[14] and M-[23]. In addition, although each method obtains comparable errors, AGPM is slightly better than the others, see the right plot in Figure 2.

In a word, preliminary experiment shows that the proposed method AGPM is effective for learning structure problems.



(a) (b)

Figure 2. Numerical results of “Time” and “Err” with Ntr=1000 and Nte=100.

4. Conclusion

In this paper, we studied the structure learning in conditional random fields. We first formulated this problem as a convex optimization model combined l_1-regularization and l_2-regularization, and then a new stepsize selection strategy combined a nonmonotone line search was proposed to solve this model. Experimental results demonstrated the efficiency and stability of our approach.

We think that there are two issues that could lead to improvements. The first point that should be considered is to use other improved nonmonotone line searches, such as the two proposed in [7, 10]. Another important point worth considering is the use of algorithms to solve problem (1) directly. For example, since the objection function in (1) has the separable structure, it can be solved by the so-called alternating direction method of multipliers (ADMM) [6].

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants 61432008, 61472423 and 61262026, and the Postgraduate Scientific Research Innovation Program of Jiangsu Province (KYZZ15_0123).

REFERENCES

1. J. Barzilai and J. M. Borwein, Two-point step size gradient methods, IMA J. Numer. Anal., vol. 8, no. 1, pp. 141-148, 1988.
2. J. Besag, Efficiency of pseudo-likelihood estimation for simple Gaussian fields, Biometrika, vol. 64, pp. 616-618, 1977
3. F. Biglari, M. A. Hassan and W. J. Leong, New quasi-Newton methods via higher order tensor models, J. Comput. Appl. Math., vol. 235, pp. 2412-2422, 2011.
4. F. Biglari and M. Solimanpur, Scaling on the spectral gradient method, J. Optim. Theory Appl., vol. 158, no. 2, pp. 626-635, 2013.
5. E. G. Birgin, J. M. Martínez and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, SIAM J. Optim., vol.10, no. 4, pp. 1196-1211, 2000.
6. S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1-122, 2011.
7. Y. Dai and R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming, Numer. Math., vol. 100, no. 1, pp. 21-47, 2005.
8. Y. Dai, J. Yuan and Y. Yuan, Modified two-point stepsize gradient methods for unconstrained optimization, Comput. Optim. Appl., vol.22, no. 1, pp. 103-109, 2002.
9. L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM J. Numer. Anal., vol. 23, pp. 707-716, 1986.
10. S. Huang, Z. Wan and X. Chen, A new nonmonotone line search technique for unconstrained optimization, Numer. Algorithms, vol. 68, no.4, pp. 671-689, 2015.
11. J. Lafferty, A. McCallum and F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in ICML, pp. 282-289, 2011.
12. J. Nocedal and S. J. Wright, Convex Optimization, New York: Springer Science & Business Media, 2006.
13. M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM J. Optim., vol. 7, no. 1, pp. 26-33, 1997.
14. M. Schmidt, K. Murphy, G. Fung and R. Rosales, Structure learning in random fields for heart motion abnormality detection, in CVPR, 2008.
15. M. Schmidt, E. van den Berg, M. P. Friedlander and K. Murphy, Optimizing costly functions with simple constraints: a limited-memory projected quasi-Newton algorithm, in AISTATS, 2009.
16. B. A. Turlach, W. N. Venables and S. J. Wright, Simultaneous variable selection, Technometrics, vol. 47, no. 3, pp. 349-363, 2005.
17. W. Wang and Q. Wang, Approximated function based spectral gradient algorithm for sparse signal recovery, Stat., Optim. Inf. Comput., vol. 2, no. 1, pp. 10-20, 2014.
18. Z. Wei, G. Yu, G. Yuan and Z. Lian, The superlinear convergence of a modified BFGS-type method for unconstrained optimization, Comput. Optim. Appl., vol. 29, no. 3, pp. 315-332, 2004.
19. Y. Xiao, Q. Wang and D. Wang, Notes on the Dai-Yuan-Yuan modified spectral gradient method, J. Comput. Appl. Math., vol. 234, no. 10, pp. 2986-2992, 2010.
20. G. Yu, L. Qi, Y. Sun and Y. Zhou, Impulse noise removal by a nonmonotone adaptive gradient method, Signal Process., vol. 90, pp. 2891-2897, 2010.
21. H. Zhang and W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim., vol. 14, pp. 1043-1056, 2004.
22. J. Zhang, N. Deng and L. Chen, New quasi-Newton equation and related methods for unconstrained optimization, J. Optim. Theory Appl., vol. 102, pp. 147-167, 1999.
23. B. Zhou, L. Gao and Y. Dai, Gradient methods with adaptive stepsizes, Comput. Optim. Appl., vol. 35, pp. 69-86, 2006.