Efficient Robust Conditional Random Fields

Dongjin Song, Wei Liu, Tianyi Zhou, Dacheng Tao, *Fellow, IEEE*, and David A. Meyer

***Abstract*— Conditional random fields (CRFs) are a flexible yet powerful probabilistic approach and have shown advantages for popular applications in various areas, including text analysis, bioinformatics, and computer vision. Traditional CRF models, however, are incapable of selecting relevant features as well as suppressing noise from noisy original features. Moreover, conventional optimization methods often converge slowly in solving the training procedure of CRFs, and will degrade significantly for tasks with a large number of samples and features. In this paper, we propose robust CRFs (RCRFs) to simultaneously select relevant features. An optimal gradient method (OGM) is further designed to train RCRFs efficiently. Specifically, the proposed RCRFs employ the ℓ_1 norm of the model parameters to regularize the objective used by traditional CRFs, therefore enabling discovery of the relevant unary features and pairwise features of CRFs. In each iteration of OGM, the gradient direction is determined jointly by the current gradient together with the historical gradients, and the Lipschitz constant is leveraged to specify the proper step size. We show that an OGM can tackle the RCRF model training very efficiently, achieving the optimal convergence rate $O(1/k^2)$ (where k is the number of iterations). This convergence rate is theoretically superior to the convergence rate $O(1/k)$ of previous first-order optimization methods. Extensive experiments performed on three practical image segmentation tasks demonstrate the efficacy of OGM in training our proposed RCRFs.**

***Index Terms*— Optimal gradient method, conditional random fields, robust conditional random fields, image segmentation.**

I. Introduction

CONDITIONAL random fields (CRFs) [9], [19], [20], [24], [37], [52] are a successful probabilistic approach for labeling structured data, and have been applied to various practical applications, *e.g.*, natural language processing [46], [47], text analysis [34], [41], bioinformatics [42], multi-view learning [53], [54], and computer vision [16], [25], [26], [40], [43], [44], [48], [56], [58]. Since CRFs directly model a label distribution conditioned on observations or input variables without explicitly considering the dependencies among the observations or variables, CRFs can flexibly tackle a variety of arbitrary and interdependent features of inputs.

Generally, CRFs can be categorized into three classes, *i.e.*, linear chain CRFs [24], 2D CRFs [19], [20], [49], and high-order tensor CRFs [5], based upon the observation formats and their associated label contextual structures, *e.g.*, text, images, and computed tomography (CT) images. This paper concentrates on the first two classes because these two are more common. Note that 2D CRFs are essentially different from linear chain CRFs due to the difference between their underlying inference problems. Specifically, since the objective function used by CRFs is convex, linear chain CRFs can achieve global optima via exact inference, such as message passing algorithms and combinatorial *min cut* or *max flow* algorithms. However, 2D CRFs cannot perform exact inference in general because of the high computational complexity. Besides inference, training of linear chain CRFs and 2D CRFs is another important issue.

In the past decade, a number of algorithms have been developed to attain the optimal parameters of linear chain CRFs. For example, Lafferty *et al.* [24] introduced a family of iterative scaling algorithms [4], [13] to train linear chain CRFs. Such algorithms seek the optimal solution of the lower bounded auxiliary function, which is an approximation to the globally optimal solution of the true log-likelihood function. Although these iterative scaling algorithms are simple and convergent, their convergence rate is lower than those of typical convex optimization algorithms when features of input observations or variables are highly correlated [29], [30]. To train CRFs more efficiently, Sha and Pereira [41] used the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method [31] to estimate the curvature of the likelihood function by taking advantage of previous gradients and updates. L-BFGS avoids the computation of the exact Hessian inverse and has shown its computational efficiency in shallow parsing [41]. Nevertheless, L-BFGS cannot directly work under batch and online settings. To this end, Vishwanathan *et al.* [49] employed the stochastic gradient method (SGD) to train CRFs in an online fashion, for the reason that SGD (or stochastic meta-descent in particular) is generally far more efficient than L-BFGS in training CRFs. Other alternative methods for training linear chain CRFs include Collins perceptron [10], Gibbs sampling [14], and

Manuscript received December 7, 2014; revised March 27, 2015; accepted May 9, 2015. Date of publication June 1, 2015; date of current version June 12, 2015. The work of D. Song and D. A. Meyer were supported by the U.S. Department of Defense Minerva Research Initiative/Army under Grant W911NF-09-1-0081 and in part by the National Science Foundation–Division of Mathematical Sciences under Grant 1223137. The work of D. Tao was supported by the Australian Research Council under Project DP-140102164, Project FT-130101457, and Project LP-140100569. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ling Shao.

D. Song is with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: dosong@ucsd.edu).

W. Liu is with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: weiliu@us.ibm.com).

T. Zhou is with the Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: tianyizh@uw.edu).

D. Tao is with the Centre for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, 81 Broadway Street, Ultimo, NSW 2007, Australia (e-mail: dacheng.tao@uts.edu.au).

D. A. Meyer is with the Department of Mathematics, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: dmeyer@math.ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2438553

contrastive divergence [17]. It is worth pointing out that most of the aforementioned approaches can naturally be applied to train 2D CRFs [49] which have been widely utilized in practical applications such as information extraction [57] and image labeling [19], [20].

To label image pixels/regions effectively, a variety of features, such as density scale invariant feature transform (SIFT) [22] and Gabor features [12], were extracted and adopted. Existing 2D CRFs approaches, however, cannot automatically determine which features are useful and which features are not. Although a preprocessing step such as sparse coding [23] and dimensionality reduction [36] may be used, it is suboptimal because the objective involved is inconsistent with that of CRFs. To address this issue, we propose robust conditional random fields (RCRFs), which regularize the objective of CRFs with the ℓ_1 norm of the model parameters and thus enable selection of relevant unary and pairwise features simultaneously. On one hand, RCRFs are related to sparse linear chain CRFs [45] because they both leverage the ℓ_1 norm to regularize the model parameters. On the other hand, RCRFs are different from sparse linear chain CRFs, because 1) we focus on 2D CRFs, 2) their underlying inference methods are distinct, and 3) their targeted applications are different.

The aforementioned conventional optimization approaches, however, cannot be directly applied to solve RCRFs. This is because the objective of RCRFs is convex but non-smooth, *i.e.*, non-differentiable. Although sub-gradient methods or projected sub-gradient methods [2] can be applied to optimize the non-smooth objective, they generally converge slowly and will degrade significantly for tasks with a large number of samples and features. To mitigate this disadvantage, in this paper we design and analyze an optimal gradient method (OGM), or equivalently Nesterov's gradient method [33], to solve RCRFs in an efficient manner. The OGM developed incorporates an extra term into the objective of RCRFs to smooth it, and exploits a gradient-based method with the proven optimal convergence rate to estimate the optimal parameters of RCRFs. In each iteration, two auxiliary optimization problems are solved, and a weighted combination of their solutions is set as the current parameters of RCRFs. In other words, the current parameters are determined jointly by the current gradient and historical gradients. The main computational cost of OGM is to calculate the first-order derivative of the objective function per iteration, so OGM is very efficient in computation and meanwhile alleviates the slow-convergence disadvantage of the conventional optimization methods.

To thoroughly examine the effectiveness and efficiency of OGM in training RCRFs, we perform RCRF-OGM on three real-world image segmentation tasks. Extensive experiments demonstrate that OGM outperforms competing optimization methods including the iterative shrinkage-thresholding algorithm (ISTA) [2] and the fast iterative shrinkage-thresholding algorithm (FISTA) [2] when all of them are used for training RCRFs.

The rest of this paper is organized as follows. Section II briefly introduces CRFs. Section III describes RCRFs and explains how to select relevant features of RCRFs. Section IV presents OGM for training RCRFs. Section V compares OGM against representative optimization methods in terms of training RCRFs on various applications. Finally, Section VI concludes the paper.

II. Conditional Random Fields

Conditional random fields (CRFs) [24] are undirected graphical models; let $\mathbf{x} = (x_1, x_2, \ldots, x_n;\ x_i \in \pi \equiv \mathbb{R})$ be a collection of random variables (observations), and $\mathbf{y} = (y_1, y_2, \ldots, y_n;\ y_i \in \gamma)$ (with $\gamma \equiv \{-1, 1\}$ in this paper) be the collection of associated labels. Each entry y_i is assumed to range over a finite label alphabet γ. CRFs construct a conditional model $p(\mathbf{y}|\mathbf{x})$ with a given set of features from the paired observations and labels. The definition of CRFs [24] can be given as:

Definition 1: Let $G = (V, E)$ be a graph such that $\mathbf{y}$ is indexed by the vertices (nodes) of G. Then $(\mathbf{x}, \mathbf{y})$ is said to be a conditional random field if, when conditioned on $\mathbf{x}$, the random variables y_i obey the Markov property with respect to the graph: $p(y_i|\mathbf{x}, y_{V\setminus\{i\}}) = p(y_i|\mathbf{x}, y_{N_i})$, where $V \setminus \{i\}$ is the set of all nodes in the graph except the node i, N_i is the set of neighbors of the node i in G.

Unlike Markov random fields (MRFs) [11], [50], which model the prior and likelihood independently, CRFs directly model the conditional dependence of labels given the observations, *i.e.*, the posterior $p(\mathbf{y}|\mathbf{x})$. Based upon the Hammersley-Clifford theorem [18], the conditional probability distribution defined by the CRFs can be written as

$$p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta},\mathbf{x})}\prod_{i\in V}\exp\langle\Phi_i(\mathbf{x}, y_i), \boldsymbol{\theta}_1\rangle \times \prod_{(i,j)\in E}\exp\langle\Phi_{ij}(\mathbf{x}, y_i, y_j), \boldsymbol{\theta}_2\rangle \quad (1)$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1; \boldsymbol{\theta}_2]$ are model parameters with $\boldsymbol{\theta}_1 \in \mathbb{R}^{d_\mathbf{h}}$ and $\boldsymbol{\theta}_2 \in \mathbb{R}^{d_\mathbf{g}}$, $Z(\boldsymbol{\theta}, \mathbf{x})$ is known as the partition function which is a normalization factor over all the output values, and $\langle\cdot,\cdot\rangle$ denotes inner product. $\exp\langle\Phi_i(\mathbf{x}, y_i), \boldsymbol{\theta}_1\rangle$ is the unary potential and it encodes the compatibility of the ith label given the observation $\mathbf{x}$. $\exp\langle\Phi_{ij}(\mathbf{x}, y_i, y_j), \boldsymbol{\theta}_2\rangle$ is the pairwise potential and it encodes the pairwise ith and jth labels compatibility over the observation $\mathbf{x}$. The specific forms of $\langle\Phi_i(\mathbf{x}, y_i), \boldsymbol{\theta}_1\rangle$ and $\langle\Phi_{ij}(\mathbf{x}, y_i, y_j), \boldsymbol{\theta}_2\rangle$ are

$$\langle\Phi_i(\mathbf{x}, y_i), \boldsymbol{\theta}_1\rangle = y_i\boldsymbol{\theta}_1^T\mathbf{h}_i(\mathbf{x}), \quad (2)$$

and

$$\langle\Phi_{ij}(\mathbf{x}, y_i, y_j), \boldsymbol{\theta}_2\rangle = y_i y_j\boldsymbol{\theta}_2^T\mathbf{g}_{ij}(\mathbf{x}), \quad (3)$$

where $\mathbf{h}_i(\mathbf{x}) \in \mathbb{R}^{d_\mathbf{h}}$ and $\mathbf{g}_{ij}(\mathbf{x}) \in \mathbb{R}^{d_\mathbf{g}}$ represent the node feature vector and edge feature vector, respectively. $d_\mathbf{h}$ and $d_\mathbf{g}$ are the dimensions of node feature vector and edge feature vector, respectively.

For simplicity, we can use $\Phi(\mathbf{x}, \mathbf{y})$ to denote the sufficient statistics of a distribution, thus the clique potentials over all

nodes and edges can be encoded as:

$$\begin{aligned}\Phi(\mathbf{x}, \mathbf{y}) &= \sum_{i \in V} \Phi(\mathbf{x}, y_i) \\ &= \Big(\sum_{i \in V} \Phi_i(\mathbf{x}, y_i); \sum_{(i,j) \in E} \Phi_{ij}(\mathbf{x}, y_i, y_j) \Big),\end{aligned}$$

where V is the node set and E is the edge set.

A. Parameter Estimation

To estimate the parameters of CRFs, $\boldsymbol{\theta}$ are utilized to fit the training samples. According to Bayes's rule, the $\boldsymbol{\theta}$ can be inferred by $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y}) \propto p(\boldsymbol{\theta}) p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$. For convenience, the isotropic Gaussian prior is usually assumed over the parameters $\boldsymbol{\theta}$, *i.e.*, $p(\boldsymbol{\theta}) \propto \exp\left(-1/2\sigma^2 \parallel \boldsymbol{\theta} \parallel^2\right)$ for a fixed σ. Thus the log-posterior of the parameters gives the data and corresponding labels can be represented as:

$$\begin{aligned}\mathcal{O}_{\mathcal{CRF}}(\boldsymbol{\theta}) &= \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) - \frac{1}{2\sigma^2} \parallel \boldsymbol{\theta} \parallel^2 \\ &= \sum_{i \in V} \langle \Phi_i(\mathbf{x}, y_i), \boldsymbol{\theta}_1 \rangle + \sum_{(i,j) \in E} \langle \Phi_{ij}(\mathbf{x}, y_i, y_j), \boldsymbol{\theta}_2 \rangle \\ &\quad - \log(Z(\boldsymbol{\theta}, \mathbf{x})) - \frac{1}{2\sigma^2} \parallel \boldsymbol{\theta} \parallel^2 \\ &= \langle \Phi(\mathbf{x}, \mathbf{y}), [\boldsymbol{\theta}_1; \boldsymbol{\theta}_2] \rangle - \log(Z(\boldsymbol{\theta}, \mathbf{x})) - \frac{1}{2\sigma^2} \parallel \boldsymbol{\theta} \parallel^2\end{aligned} \tag{4}$$

where $Z(\boldsymbol{\theta}, \mathbf{x}) = \sum_{y' \in \gamma} \exp(\langle \sum_{i \in V} \Phi(\mathbf{x}, y_i = y'), [\boldsymbol{\theta}_1; \boldsymbol{\theta}_2] \rangle)$ is the partition function. Exact calculation of the partition function here is difficult since it ranges over all the possible labels associated to vertices of the graph G. Therefore, pseudo-likelihood or approximated inference methods can be used in practical applications to resolve this issue.

The optimized parameters $\boldsymbol{\theta}$ are obtained by maximizing *a posteriori* (MAP) estimate of $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})$, *i.e.*, $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{O}_{CRF}(\boldsymbol{\theta})$.

B. Approximate Inference

To obtain $Z(\boldsymbol{\theta}, \mathbf{x})$, probabilistic inference should be conducted. For simple structures (*e.g.*, chains and trees), exact inference is computationally tractable. For complex graphs such as grids and complete graphs (including loops), exact inference is computationally intractable [35]. For example, the treewidth of an $n \times n$ grid is $w = O(2n)$ [27], so exact inferences will take $O(|\gamma|^{2n})$ time which is infeasible for practical applications.

One solution to this problem is to train the model with the pseudo-likelihood (PL) [7], [20], which is a tractable approximation of the true likelihood. Another approach is to use approximate inference algorithms, *e.g.*, the loop belief propagation (LBP) [51], [55] and mean fields (MF) [51].

In this paper, we will compare different optimization methods of RCRFs based upon loopy belief propagation (LBP), mean fields (MF), and pseudo-likelihood (PL).

C. Optimization Methods

To obtain the MAP estimate of $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})$ w.r.t, $\boldsymbol{\theta}$, we need maximize $\mathcal{O}_{CRF}(\boldsymbol{\theta})$. For this purpose, we can calculate the gradient of $\mathcal{O}_{CRF}(\boldsymbol{\theta})$ as

$$\begin{aligned}&\frac{\partial \mathcal{O}_{CRF}}{\partial \boldsymbol{\theta}} \\ &= \sum_{i \in V} \Phi(\mathbf{x}, y_i) - \frac{1}{Z(\boldsymbol{\theta}, \mathbf{x})} \frac{\partial Z(\boldsymbol{\theta}, \mathbf{x})}{\partial \boldsymbol{\theta}} - \frac{\boldsymbol{\theta}}{\sigma^2} \\ &= \sum_{i \in V} \Phi(\mathbf{x}, y_i) - \frac{\boldsymbol{\theta}}{\sigma^2} \\ &\quad - \sum_{y' \in \gamma} \sum_{i \in V} \frac{\exp(\langle \sum_{i \in V} \Phi(\mathbf{x}, y_i = y'), \boldsymbol{\theta} \rangle)}{Z(\boldsymbol{\theta}, \mathbf{x})} \Phi(\mathbf{x}, y_i = y') \\ &= \sum_{i \in V} \Phi(\mathbf{x}, y_i) - \sum_{y' \in \gamma} \sum_{i \in V} p(y_i = y'|\mathbf{x}; \boldsymbol{\theta}) \Phi(\mathbf{x}, y_i = y') - \frac{\boldsymbol{\theta}}{\sigma^2} \\ &= \sum_{i \in V} \Big(\Phi(\mathbf{x}, y_i) - \sum_{y' \in \gamma} p(y_i = y'|\mathbf{x}; \boldsymbol{\theta}) \Phi(\mathbf{x}, y_i = y') \Big) - \frac{\boldsymbol{\theta}}{\sigma^2},\end{aligned} \tag{5}$$

where the second sum represents the expected feature vector for each clique. Based upon this gradient, first-order methods, *e.g.*, the gradient descent/ascent, and stochastic gradient descent/ascent, can be utilized to search for the optimal parameters. Since only the current gradient of $L(\boldsymbol{\theta})$ is considered, the objective converges relative slowly. To obtain the optimal parameters efficiently, more information about the gradient should be considered.

Generally, second-order methods, *e.g.*, Newton's method, converge much faster than first-order methods because they take the second order derivative of likelihood into account. The calculation of the Hessian matrix, however, is very expensive and can be achieve by:

$$\begin{aligned}H(\boldsymbol{\theta}) &= \frac{\partial^2 \mathcal{O}_{\mathcal{CRF}}}{\partial \boldsymbol{\theta}^2} \\ &= -\sum_{i \in V} \mathrm{Cov}_{p(y_i = y'|\mathbf{x}; \boldsymbol{\theta})} \Big(\Phi(\mathbf{x}, y_i = y') \Big) - \frac{\mathbf{I}}{\sigma^2},\end{aligned} \tag{6}$$

where $\mathrm{Cov}(\cdot)$ represents the covariance matrix. Since the size of the Hessian is quadratic with respect to the number of parameters and some practical applications (such as text analysis) often use tens of thousands parameters, it is impractical to store the full Hessian matrix, let alone computing it. Quasi-Newton methods, *e.g.*, BFGS [8], approximate the Hessian matrix by using the gradient of the objective function. A full $d \times d$ approximation to the Hessian matrix is still of quadratic size; if d is too large to afford, L-BFGS [31] can be used. Since the aforementioned algorithms are not suitable for online/batch setting, Vishwanathan *et al.* [49] presented the stochastic meta-gradient (SMD) method, which uses the second-order information to adapt the gradient step size. Their empirical study shows the efficiency of SMD in online and batch settings.

III. Robust Conditional Random Fields

Unlike CRFs which assume an isotropic Gaussian prior over the model parameters to prevent over-fitting, robust conditional conditional random fields (RCRFs) consider a Laplacian prior which is robust to the outliers. Due to this prior, traditional optimization approaches cannot be used for training RCRFs. In particular, since the Laplacian prior puts more probability density at zero than the Gaussian prior, it will encourage sparsity of RCRFs, which is especially helpful in applications with a great number of samples as well as a large number of features. Specifically, the Laplacian prior takes the form $p(\boldsymbol{\theta}) \propto \exp\big(-\lambda \parallel \boldsymbol{\theta} \parallel_1\big)$ for a fixed λ. Thus the negative log-posterior of the parameters gives the data and corresponding labels can be represented as:

$$\begin{aligned}\mathcal{O}_{\mathcal{RCRF}}(\boldsymbol{\theta}) &= -\sum_{i\in V}\log p(y_i|\mathbf{x};\boldsymbol{\theta}) + \lambda \parallel \boldsymbol{\theta} \parallel_1 \\ &= f(\boldsymbol{\theta}) + r(\boldsymbol{\theta}),\end{aligned} \tag{7}$$

where $f(\boldsymbol{\theta}) = -\sum_{i\in V}\log p(y_i|\mathbf{x};\boldsymbol{\theta})$ and $r(\boldsymbol{\theta}) = \lambda \parallel \boldsymbol{\theta} \parallel_1 = \sum_{i=1}^{d_h+d_g}\lambda|\boldsymbol{\theta}_i|$. $\lambda > 0$ is a regularization parameter which controls the trade off of these two terms. Note that Eq. (1) is essentially a log-linear model taking a softmax form. Therefore, $f(\boldsymbol{\theta})$ is a differentiable (smooth) convex function. $r(\boldsymbol{\theta})$, however, is a convex but non-differentiable (non-smooth) regularization term.

To optimize the objective of RCRFs, we first introduce two classical methods, *i.e.*, iterative shrinkage-thresholding algorithm (ISTA) [2] and fast iterative shrinkage-thresholding algorithm (FISTA) [2] in this section and then show how to address this problem with optimal gradient method (OGM) in the next section.

Definition 2: Assuming $f : \mathbb{R}^{d_h+d_g} \to \mathbb{R}$ is a smooth convex function of type $C^{1,1}$, *i.e.*, continuously differentiable with Lipchitz continuous gradient L_f which satisfies:

$$\parallel \partial_{\boldsymbol{\theta}^1} f(\boldsymbol{\theta}^1) - \partial_{\boldsymbol{\theta}^2} f(\boldsymbol{\theta}^2) \parallel_2 \leq L_f \parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2, \tag{8}$$

where $\parallel \cdot \parallel_2$ is the Euclidean norm, $L_f > 0$ is the Lipschitz constant of $\partial f(\boldsymbol{\theta})$.

Based upon the definition of the Lipchitz constant, it can be calculated with

$$\begin{aligned}L_f &= \max \frac{\parallel \frac{\partial f(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}^1} - \frac{\partial f(\boldsymbol{\theta}^2)}{\partial \boldsymbol{\theta}^2} \parallel_2}{\parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2}, \\ &= \sigma_{\max}\Big[\frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\Big] \\ &= \sigma_{\max}\Big[\sum_{i\in V}\mathrm{Cov}_{p(y_i=y'|\mathbf{x};\boldsymbol{\theta})}\Big(\Phi(\mathbf{x}, y_i = y')\Big)\Big],\end{aligned} \tag{9}$$

where $\sigma_{\max}$ is the maximum eigenvalue of the target matrix.

A. ISTA for RCRFs

The iterative shrinkage-thresholding algorithm (ISTA) [2] is essentially a proximal gradient method for dealing with an unconstrained optimization problem with objective splitting into a convex differentiable term and a convex non-differentiable term. Therefore, ISTA is perfectly suitable for optimizing the objective of RCRFs in Eq. (7). Specially, the basic step of ISTA [2] is

$$\boldsymbol{\theta}^k = \mathrm{Prox}_{\frac{\lambda}{L_f}}(\boldsymbol{\theta}^{k-1} - \frac{1}{L_f}\partial f(\boldsymbol{\theta}^{k-1})) \tag{10}$$

where L_f is the Lipschitz constant which controls the step size and $\mathrm{Prox} : \mathbb{R}_n \to \mathbb{R}_n$ is the shrinkage operator defined as:

$$\mathrm{Prox}_{\frac{\lambda}{L_f}}(\boldsymbol{\theta}_i) = (|\boldsymbol{\theta}_i| - \frac{\lambda}{L_f})_+ \mathrm{sgn}(\boldsymbol{\theta}_i). \tag{11}$$

It has been shown that ISTA has a worst case convergence rate of $O(1/k)$ [2] where k is the number of iterations.

Algorithm 1 The Optimization Framework of FISTA [2] for RCRFs

Input: L_f (the Lipschiz constant of ∂f), T (the maximum number of iterations), $\boldsymbol{\theta}^0$, and ϵ.
Output: Optimal model parameters $\boldsymbol{\theta}^k$
Initialize: $\boldsymbol{\eta}^1 = \boldsymbol{\theta}^0 \in \mathbb{R}^{d_{\mathbf{f}}+d_{\mathbf{g}}}$, $a^1 = 1$.
repeat
Step1: Compute $\boldsymbol{\theta}^k = \mathrm{Prox}_{\frac{\lambda}{L_f}}(\boldsymbol{\eta}^k - \frac{1}{L_f}\partial f(\boldsymbol{\eta}^k))$
Step2: Compute $a^{k+1} = \big(1 + \sqrt{1 + 4(a^k)^2}\big)/2$
Step3: Compute $\boldsymbol{\eta}^{k+1} = \boldsymbol{\theta}^k + \frac{a^k - 1}{a^{k+1}}(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1})$
Step4: Update $k = k + 1$
until $\frac{|\mathcal{O}_{RCRF}(\boldsymbol{\theta}^{k+1}) - \mathcal{O}_{RCRF}(\boldsymbol{\theta}^k)|}{|\mathcal{O}_{RCRF}(\boldsymbol{\theta}^k)|} < \epsilon$ or $k \geq T$
return $\boldsymbol{\theta} = \boldsymbol{\theta}^{k+1}$

B. FISTA for RCRFs

The fast iterative shrinkage-thresholding algorithm (FISTA) [2] is an extension of Nesterov's work [32] and improves ISTA to achieve a worst case convergence rate of $O(1/k^2)$. The essential difference between FISTA and ISTA is that the proximate operator $\mathrm{Prox}(\cdot)$ is not directly employed on the previous point $\boldsymbol{\theta}^{k-1}$, but rather at point $\boldsymbol{\eta}^k$ which is a linear combination of the previous two points, *i.e.*, $\boldsymbol{\theta}^{k-1}$ and $\boldsymbol{\theta}^{k-2}$. The optimization framework of FISTA for RCRFs is provided in Algorithm 1. As we can notice, the additional computation involved for FISTA compared to ISTA is marginal.

For many practical applications, especially for image segmentation, we often utilize pseudo-likelihood (PL) or approximated inference approaches, *e.g.*, the loop belief propagation (LBP) [51], [55] and mean fields (MF) [51], to calculate the gradient of RCRFs' objective function. Since the gradient obtained by this way is inaccurate in general and may be particularly inaccurate at certain steps, it is unreasonable to update the gradient direction by only considering the current gradient and previous gradient. On the contrary, we should consider the current as well as all the historical points so as to determine the gradient direction smoothly and accurately.

IV. Optimal Gradient Method for RCRFs

In recent years, Nesterov's method [32] has shown its efficiency for image denoising, image restoration, and image classification [1]–[3], [15], [28], [59]. In a recent work [33],

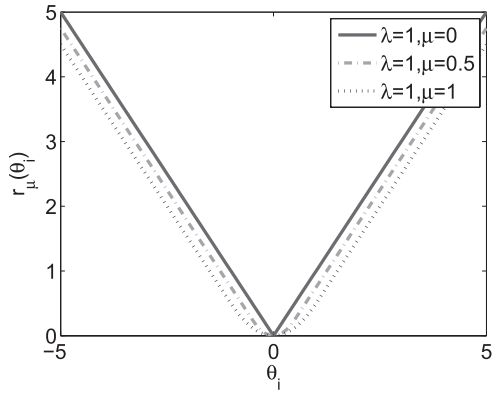


Fig. 1. The smooth approximated curves $r_\mu(\boldsymbol{\theta}_j)$ with $\lambda = 1$ and $\mu = 0$, 0.5, and 1 respectively. Note that when $\mu = 0$, we have $r(\boldsymbol{\theta}_j) = r_\mu(\boldsymbol{\theta}_j)$, *i.e.*, non-smooth is performed.

Nesterov developed a novel framework to deal with non-smooth convex functions. In this section, we extend Nesterov's smoothing technique to the objective of RCRFs, namely using the optimal gradient method (OGM) to perform the optimization. OGM is a fast gradient method with proved optimal convergence rate of $O(1/k^2)$ where k is the number of iterations. Moreover, in contrast to FISTA [2] which only considers the previous two points, the new gradient direction in OGM is not only determined by the current gradient, but also by all historical gradients.

A. Smoothing ℓ_1 Norm

In the objective of RCRFs, $f(\boldsymbol{\theta})$ is a smooth convex term and $r(\boldsymbol{\theta}) = \lambda \parallel \boldsymbol{\theta} \parallel_1 = \sum_{i=1}^{d_h+d_g} |\boldsymbol{\theta}_i|$ is a non-smooth convex term. Recall that $r(\boldsymbol{\theta})$ can be written as the dual form, *i.e.*,

$$r(\boldsymbol{\theta}) = \lambda \parallel \boldsymbol{\theta} \parallel_1 = \lambda \max_{\mathbf{u}\in\mathcal{Q}} \langle \mathbf{u}, \boldsymbol{\theta} \rangle, \tag{12}$$

where the dual feasible set is an ℓ_∞

$$\mathcal{Q} = \{\mathbf{u} : \parallel \mathbf{u} \parallel_\infty \leq 1, \mathbf{u} \in \mathbb{R}^{d_h+d_g}\}. \tag{13}$$

A natural smooth approximation for $r(\boldsymbol{\theta})$ is

$$r_\mu(\boldsymbol{\theta}) = \lambda \parallel \boldsymbol{\theta} \parallel_1 = \lambda\big(\max_{\mathbf{u}\in\mathcal{Q}} \langle \mathbf{u}, \boldsymbol{\theta} \rangle - \mu d(\mathbf{u})\big) \tag{14}$$

where $d(\mathbf{u})$ is the dual proximity function and $\mu > 0$ is a parameter to control the balance of approximation accuracy and smoothness given fixed λ. A convenient choice is $d(\mathbf{u}) = \frac{1}{2} \parallel \mathbf{u} \parallel_2^2$ which gives the Huber penalty

$$\begin{aligned} r_\mu(\boldsymbol{\theta}_j) &= \lambda \sup_{-1\leq \mathbf{u}_j \leq 1} (\mathbf{u}_j \boldsymbol{\theta}_j - \frac{1}{2}\mu \mathbf{u}_j^2) \\ &= \begin{cases} \dfrac{\lambda \boldsymbol{\theta}_j^2}{2\mu}, & |\boldsymbol{\theta}_j| \leq \mu; \\ \lambda(|\boldsymbol{\theta}_j| - \dfrac{\mu}{2}), & |\boldsymbol{\theta}_j| > \mu. \end{cases} \end{aligned} \tag{15}$$

Figure 1 shows smooth approximated curves, *i.e.*, $r_\mu(\boldsymbol{\theta}_j)$ with $\lambda = 1$ and $\mu = 0$, 0.5, and 1 respectively. The following theorem shows $r_\mu(\boldsymbol{\theta}_j)$'s theoretical bound on the approximation.

Theorem 1: $r(\boldsymbol{\theta})$ is bounded by its smooth approximation $r_\mu(\boldsymbol{\theta})$ and the approximation error is controlled with the smoothing parameter μ given fixed λ, *i.e.*, we have

$$\lambda\big(|\boldsymbol{\theta}_j| - \frac{\mu}{2}\big) \leq r_\mu(\boldsymbol{\theta}_j) \leq \lambda |\boldsymbol{\theta}|_j. \tag{16}$$

To perform optimization with Eq. (15), we expect to calculate its gradient:

$$\frac{\partial r_\mu(\boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = \begin{cases} \dfrac{\lambda \boldsymbol{\theta}_j}{\mu}, & |\boldsymbol{\theta}_j| \leq \mu; \\ \lambda \mathbf{sgn}(\boldsymbol{\theta}_j), & |\boldsymbol{\theta}_j| > \mu, \end{cases} \tag{17}$$

where $\mathbf{sgn}(\cdot)$ takes the sign of its argument.

B. Determining the Gradient Direction

Based upon the aforementioned smoothing technique over the ℓ_1 norm, the objective of RCRFs in Eq. (7) can be transformed into an approximated convex and smooth function, *i.e.*, $\mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta})$, which can be optimized conveniently with the optimal gradient method (OGM).

In OGM, the Lipschitz constant is utilized to determine the step size in each iteration. According to the definition, the Lipschitz constant of RCRFs' smoothed objective is given by:

$$\parallel \partial_{\boldsymbol{\theta}^1} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^1) - \partial_{\boldsymbol{\theta}^2} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^2) \parallel_2 \leq L \parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2 . \tag{18}$$

Therefore, the Lipschitz constant can be calculated with

$$L = \max \frac{\parallel \frac{\partial \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}^1} - \frac{\partial \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^2)}{\partial \boldsymbol{\theta}^2} \parallel_2}{\parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2}. \tag{19}$$

Since we have

$$\begin{aligned} \frac{\partial \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}_j^1} &= \frac{\partial f(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}_j^1} + \frac{\partial r_\mu(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}_j^1} \\ &= \Big[\sum_{i\in V} \big(\Phi(\mathbf{x}, y_i) \\ &\quad - \sum_{y'\in\gamma} p(y_i = y' | \mathbf{x}; \boldsymbol{\theta}^1) \Phi(\mathbf{x}, y_i = y')\big) \Big]_j \\ &\quad + \begin{cases} \dfrac{\lambda \boldsymbol{\theta}_j}{\mu}, & |\boldsymbol{\theta}_j| \leq \mu; \\ \lambda \mathbf{sgn}(\boldsymbol{\theta}_j), & |\boldsymbol{\theta}_j| > \mu, \end{cases} \end{aligned} \tag{20}$$

Eq. (19) can be expressed as

$$\begin{aligned} L &= \max \frac{\parallel \frac{\partial \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}^1} - \frac{\partial \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^2)}{\partial \boldsymbol{\theta}^2} \parallel_2}{\parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2} \\ &= \max \frac{\parallel \frac{\partial f(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}^1} - \frac{\partial f(\boldsymbol{\theta}^2)}{\partial \boldsymbol{\theta}^2} + \frac{\partial r_\mu(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}^1} - \frac{\partial r_\mu(\boldsymbol{\theta}^2)}{\partial \boldsymbol{\theta}^2} \parallel_2}{\parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2} \\ &\leq \max \frac{\parallel \frac{\partial f(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}^1} - \frac{\partial f(\boldsymbol{\theta}^2)}{\partial \boldsymbol{\theta}^2} \parallel_2}{\parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2} + \max \frac{\parallel \frac{\partial r_\mu(\boldsymbol{\theta}^1)}{\partial \boldsymbol{\theta}^1} - \frac{\partial r_\mu(\boldsymbol{\theta}^2)}{\partial \boldsymbol{\theta}^2} \parallel_2}{\parallel \boldsymbol{\theta}^1 - \boldsymbol{\theta}^2 \parallel_2} \\ &= \sigma_{\max}\Big[\frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\Big] + \frac{\lambda}{\mu} \\ &= L_f + \frac{\lambda}{\mu}, \end{aligned} \tag{21}$$

which illustrates that the Lipschitz constant of RCRFs is upper bounded by $L_f + \frac{\lambda}{\mu}$.

In each iteration of OGM, two auxiliary functions are constructed for optimization and their solutions are used to update the parameters within the same iteration round. We use $\boldsymbol{\theta}^k$, $\mathbf{s}^k$, and $\mathbf{t}^k$ to represent the parameters of RCRFs and the two auxiliary functions at the k^{th} iteration round, respectively. Specifically, the two auxiliary functions are given by

$$\min_{\mathbf{s}\in\mathbb{R}^{D_\mathbf{f}+D_\mathbf{g}}} \left\langle \partial_{\boldsymbol{\theta}^k} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^k), \mathbf{s}-\boldsymbol{\theta}^k \right\rangle + \frac{L}{2} \parallel \mathbf{s}-\boldsymbol{\theta}^k \parallel_2^2, \quad (22)$$

and

$$\min_{\mathbf{t}\in\mathbb{R}^{D_\mathbf{f}+D_\mathbf{g}}} \frac{L}{\sigma_1} d_1(t) + \sum_{m=0}^{k} \frac{m+1}{2} \Big[\mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m) + \left\langle \partial_{\boldsymbol{\theta}^\mathbf{m}} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m), \mathbf{t}^m - \boldsymbol{\theta}^m \right\rangle \Big]. \quad (23)$$

We choose the prox-function $d_1(t) = \parallel \mathbf{t} - \boldsymbol{\theta}^* \parallel_2^2 /2$, whose strong convexity parameter is σ_1, wherein $\boldsymbol{\theta}^*$ is the prox-center and $\sigma_1 = 1$. The $\boldsymbol{\theta}^*$ is usually selected as a guessed solution of $\boldsymbol{\theta}$.

By setting the gradients of the two auxiliary functions to 0, we can obtain their solutions $\mathbf{s}^k$ and $\mathbf{t}^k$, respectively, *i.e.*,

$$\mathbf{s}^k = \boldsymbol{\theta}^k - \frac{1}{L} \partial_{\boldsymbol{\theta}^k} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^k), \quad (24)$$

and

$$\mathbf{t}^k = \boldsymbol{\theta}^* - \frac{\sigma_1}{L} \sum_{m=0}^{k} \frac{m+1}{2} \partial_{\boldsymbol{\theta}^m} \mathcal{O}_{RCRF,\,\mu}(\boldsymbol{\theta}^m). \quad (25)$$

These two equations are interpreted as follows. The $\mathbf{s}^k$ is the model parameter obtained by standard gradient descent with step size $1/L$ at the k^{th} iteration round. The $\mathbf{t}^k$ is obtained based upon a gradient descent step that starts from the guessed solution $\boldsymbol{\theta}^*$ and proceeds along a descent direction determined by the weighted sum of the gradients in all previous iteration rounds. The weights of gradients at later iteration rounds are larger than those at earlier iteration rounds. Therefore, $\mathbf{s}^k$ and $\mathbf{t}^k$ encode the current gradient and historical gradients information, respectively. In OGM, their weighted sum determines the model parameters of RCRFs after the k^{th} iteration round, *i.e.*,

$$\boldsymbol{\theta}^{k+1} = \frac{2}{k+3}\mathbf{t}^k + \frac{k+1}{k+3}\mathbf{s}^k. \quad (26)$$

The intuition here is that the current gradient should be weighted more than historical gradients as the iteration index k increases.

Let Ψ_k be the optimal objective value of the second auxiliary optimization; according to [33], we have the following theorem.

Theorem 2: For any k and the corresponding $\mathbf{s}^k$, $\mathbf{t}^k$, and $\boldsymbol{\theta}^{k+1}$ defined by Eq. (24), Eq. (25), and Eq. (26), respectively, we have

$$\frac{(k+1)(k+2)}{4} \mathcal{O}_{RCRF,\mu}(\mathbf{s}^k) \leq \Psi_k. \quad (27)$$

Theorem 2 results from [33, Lemma 2] and it can be directly applied to analyze the convergence rate of OGM for training RCRFs.

Algorithm 2 The Optimization Framework of OGM for RCRFs

Input: $\mathbf{h}_i(\mathbf{x}) \in \mathbb{R}^{d_\mathbf{h}}$, $\mathbf{g}_{ij}(\mathbf{x}) \in \mathbb{R}^{d_\mathbf{g}}$, T (the maximum number of iterations), $\boldsymbol{\theta}^*$, $\boldsymbol{\theta}^0$, L and ϵ.
Output: Optimal model parameter $\boldsymbol{\theta}$
Initialize: Random initialize $\boldsymbol{\theta}^0$.
repeat
  Step1: Compute $\partial_{\boldsymbol{\theta}^k} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^k)$.
  Step2: Compute $\mathbf{s}^k$ and $\mathbf{t}^k$ by Eq. (24) and Eq. (25).
  Step3: Update RCRFs parameter $\boldsymbol{\theta}^{k+1}$ by Eq. (26)
  Step4: $k = k+1$
until $\frac{|\mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^{k+1}) - \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^k)|}{|\mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^k)|} < \epsilon$ or $k \geq T$
return $\boldsymbol{\theta}$

Fig. 2. Sample images from three datasets. The images of first row are from the TU Darmstadt car side dataset, the images of second row are from the Weizmann horse dataset, and the images of third row are from the TU Darmstadt cow side dataset.

C. Optimization Framework of OGM

The optimization framework of OGM for training RCRFs is presented in Algorithm 2. The input variables are node features $\mathbf{h}_i(\mathbf{x}) \in \mathbb{R}^{d_\mathbf{h}}$ and the edge features $\mathbf{g}_{ij}(\mathbf{x}) \in \mathbb{R}^{d_\mathbf{g}}$, the maximum number of iterations T, the initial model parameter $\boldsymbol{\theta}^0$, the guessed model parameter $\boldsymbol{\theta}^*$, the Lipschitz constant L, and the tolerance of the termination criterion ϵ. In each iteration, the gradient of the smoothed objective function of RCRFs, *i.e.*, $\partial_{\boldsymbol{\theta}^k} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^k)$, is first calculated, then $\mathbf{s}^k$ and $\mathbf{t}^k$ are calculated based upon the gradient, and finally $\boldsymbol{\theta}^{k+1}$ is updated at the end of the iteration round. The main time costs are the computation of $\partial_{\boldsymbol{\theta}^k} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^k)$ in Step 1 (with time complexity at most $O(|\gamma|^w)$ where $w = O(2n)$ is the treewidth for a $n \times n$ grid).

D. Convergence Rate Analysis

The following theorem shows the convergence rate for training RCRFs.

Theorem 3: For a fixed μ, the convergence rate of OGM for training RCRFs is $O\left(1/k^2\right)$. It requires $O\left(1/\sqrt{\varepsilon}\right)$ iteration rounds to reach an ε accurate solution.

The detailed proof of Theorem 3 is provided in Appendix.

V. Experiments

In this section, we study the efficiency and effectiveness of the proposed OGM by comparing it against ISTA and FISTA

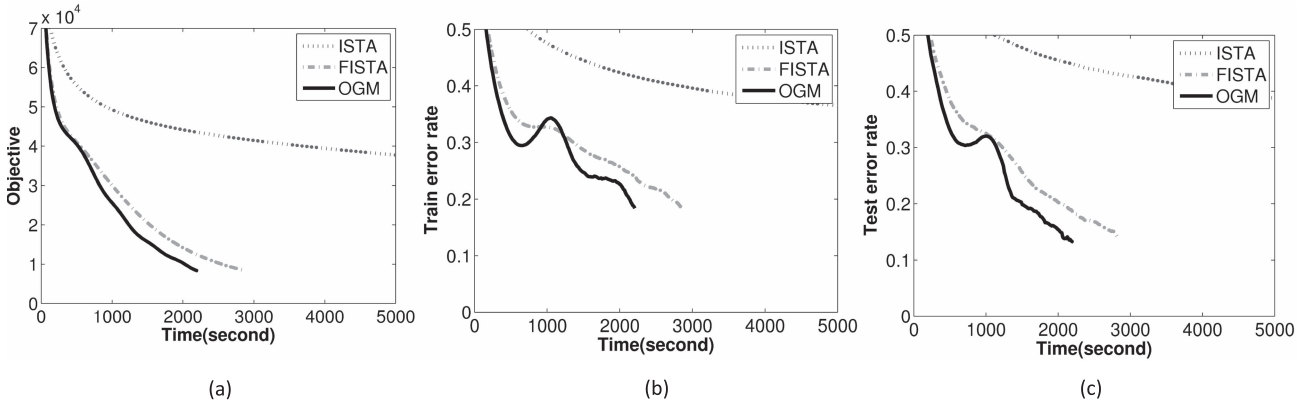


Fig. 3. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the car segmentation task with LBP inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

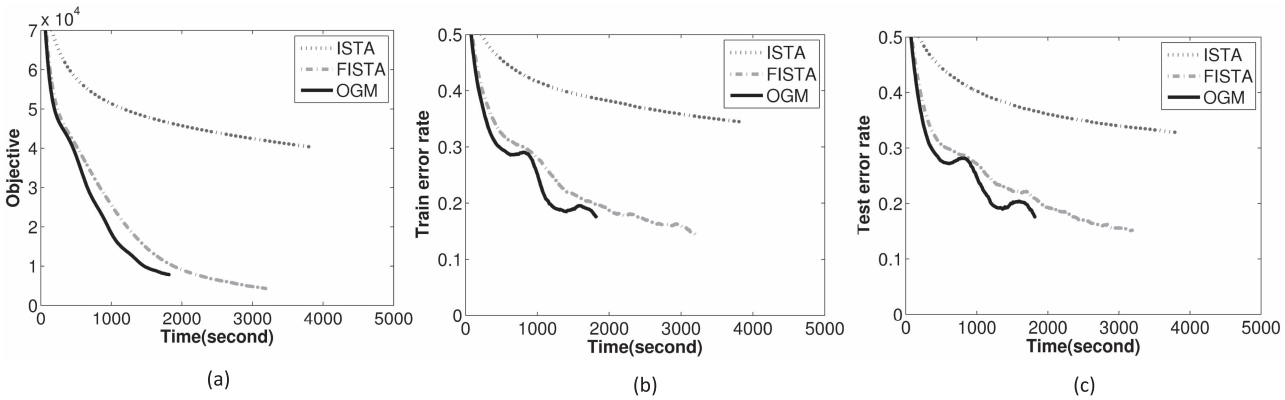


Fig. 4. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the car segmentation task with MF inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

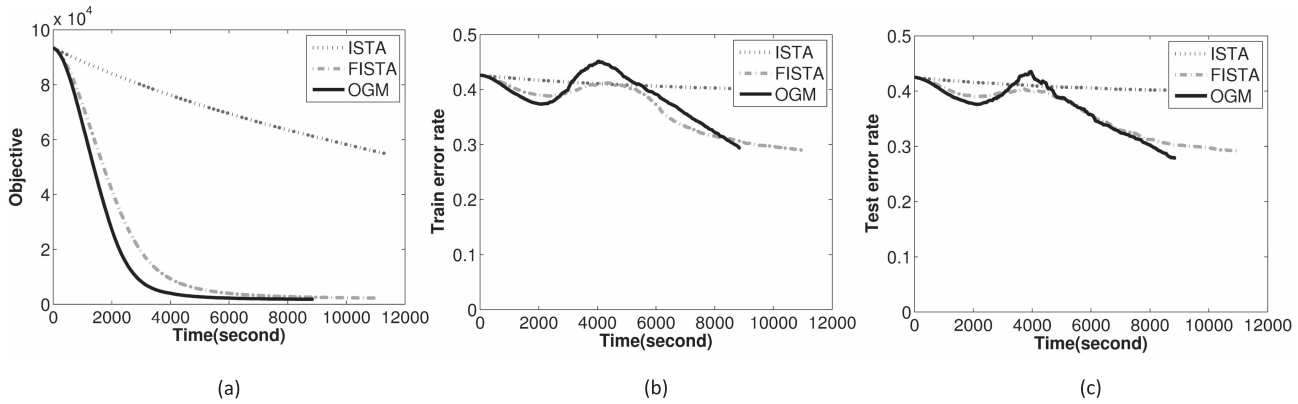


Fig. 5. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the car segmentation task with PL inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

based upon three different image segmentation tasks. In our study, three datasets are used for empirical study, which are the TU Darmstadt car side dataset (containing 50 images which are resized them to 80×100) [21], the Weizmann horse dataset [6] (we random sampled 50 images and resized to 80×100), and the TU Darmstadt cow side dataset [21] (consisting of 111 images of size 100×140). The sample images are shown in Figure 2.

For each dataset, 70% of the images are randomly selected for model training. The rest of the images are evenly divided into a validation set and a test set. We determine the model parameters $\lambda \in \{0.01, 0.1, 1, 10, 100\}$ and $\mu \in \{0.01, 0.1, 1\}$ based upon the validation set and evaluate the model with the test set. Three trials are performed and the average running time as well as average test error rate are reported for comparison. We set $T = 400$, $\epsilon = 10^{-4}$, $\boldsymbol{\theta}^* = \boldsymbol{\theta}^k$, and randomly initialize $\boldsymbol{\theta}^0$ for all experiments.

Given an image, a 162D feature vector (*i.e.*, a concatenation of the 34D feature vector in [38] and [39] and the 128D feature vector in [22]) $\mathbf{h}_i$ is extracted on

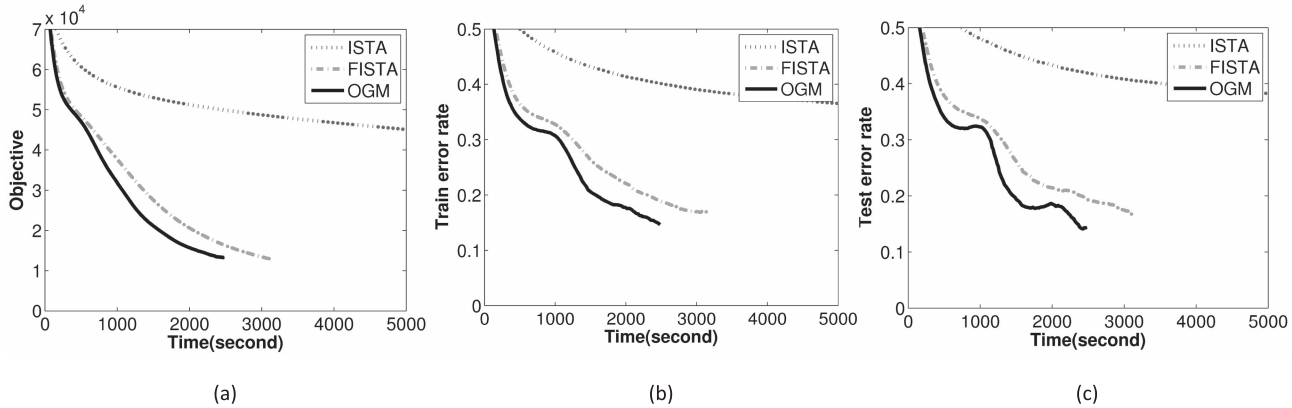


Fig. 6. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the horse segmentation task with LBP inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

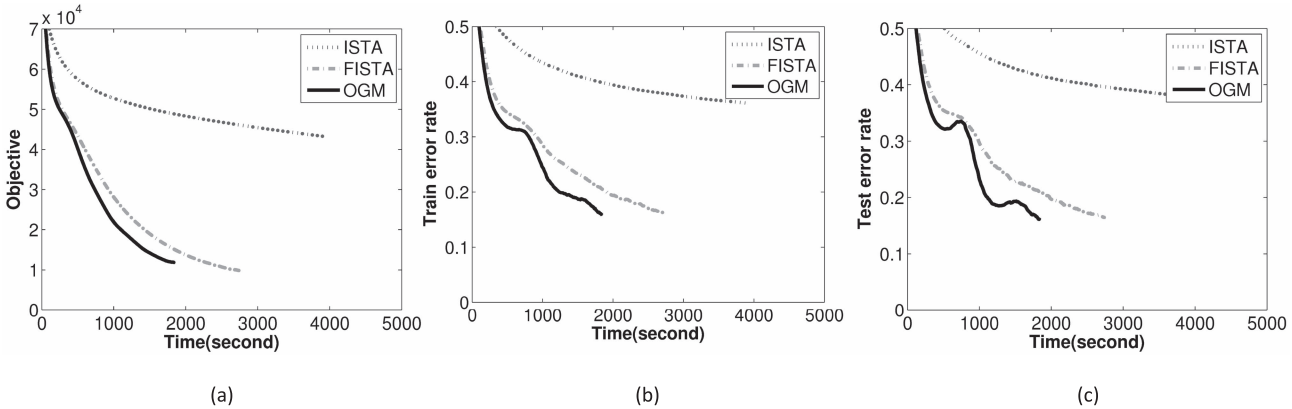


Fig. 7. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the horse segmentation task with MF inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

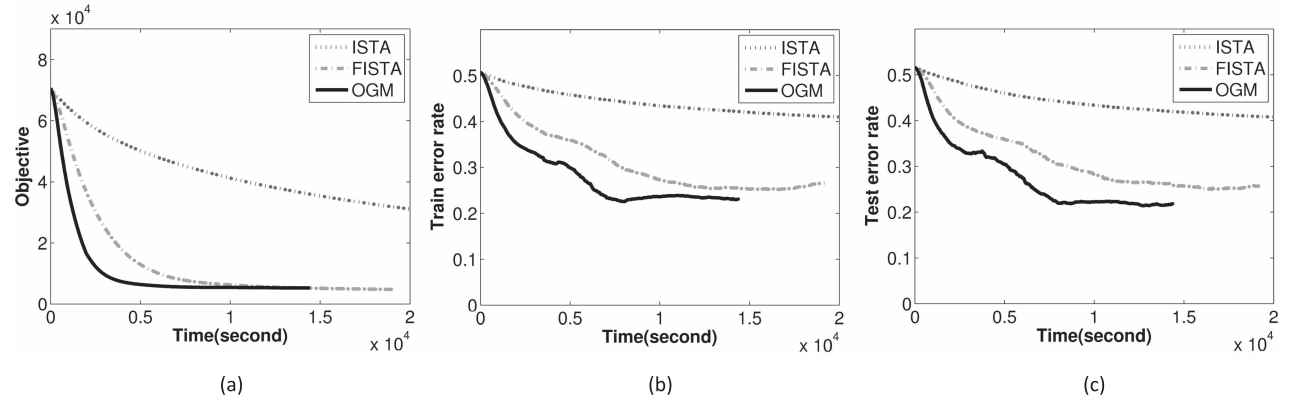


Fig. 8. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the horse segmentation task with PL inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

each 2×2 block. The node feature is denoted as $\mathbf{h}_i(\mathbf{x}) = [1, \mathbf{h}_i]$, while the edge feature is denoted as $\mathbf{g}_{ij}(\mathbf{x}) = [1, |\mathbf{h}_i - \mathbf{h}_j|]$.

Because exact inference is intractable for these three tasks, we train and evaluate ISTA, FISTA, and OGM based on three different methods, *i.e.*, the pseudo-likelihood (PL), the loopy belief propagation (LBP), and mean field (MF), to approximate the true log-likelihood function.

A. Efficiency

To demonstrate the efficiency of OGM for training RCRFs, we plot the objective function versus time for ISTA, FISTA, and OGM based upon three various approximated inference methods (*i.e.*, LBP, MF, and PL) as well as three different datasets in Figure 3a, 4a, 5a, 6a, 7a, 8a, 9a, 10a, and 11a.

We observe that given each inference method, OGM and FISTA generally converge much faster than ISTA. This is because OGM and FISTA can achieve convergence rate $O(1/k^2)$ while ISTA is only a first order method with convergence rate $O(1/k)$ and it updates the gradient direction only based upon the current gradient which may not be reliable with pseudo-likelihood (PL) and approximate inferences (LBP and MF). Moreover, we observe that OGM consistently consumes less time than

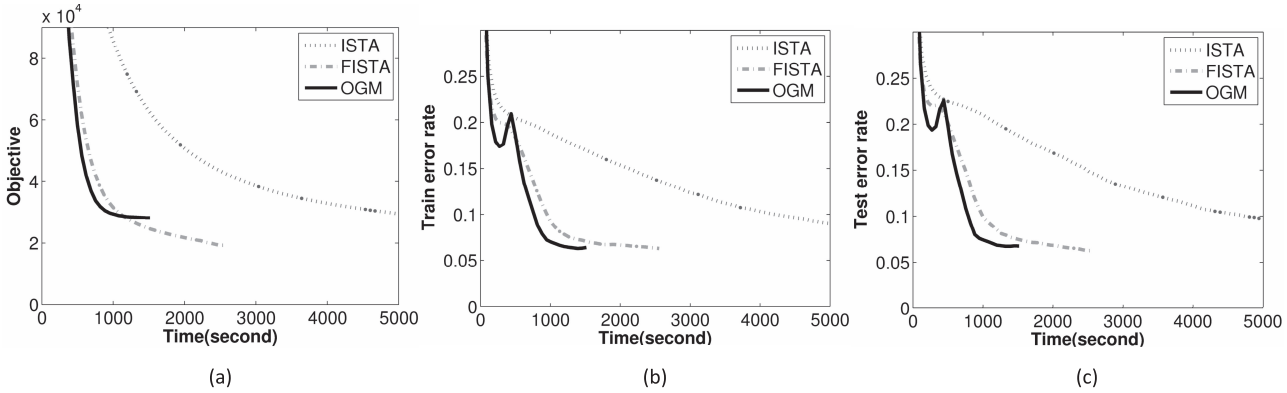


Fig. 9. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the cow segmentation task with LBP inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

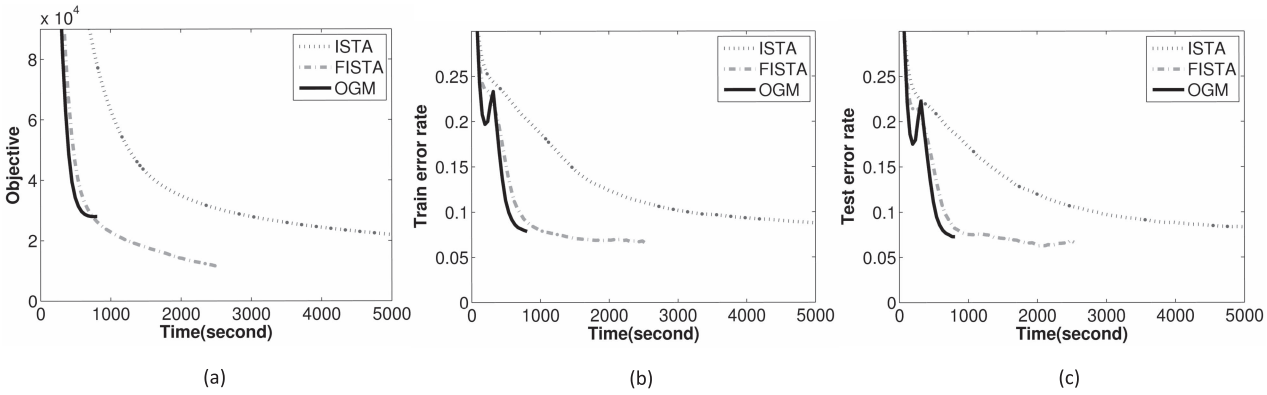


Fig. 10. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the cow segmentation task with MF inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

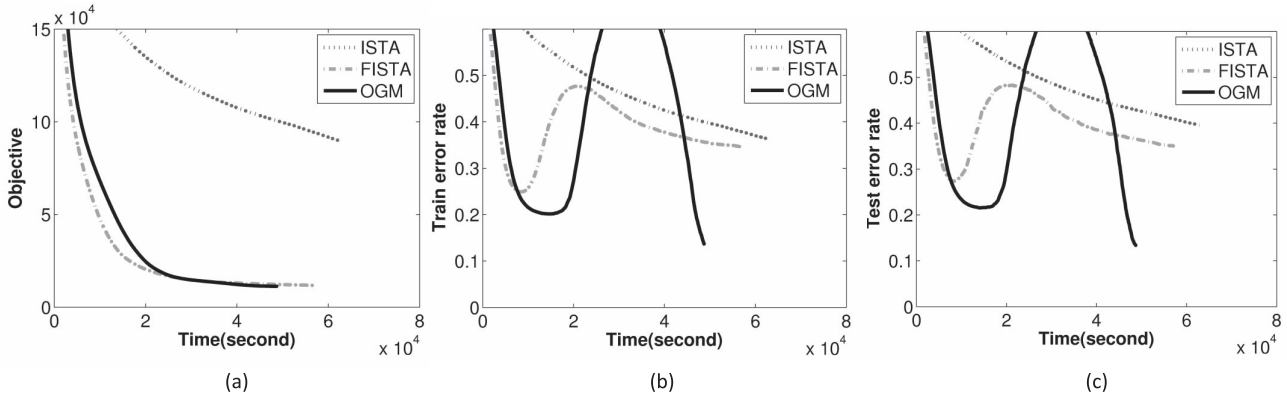


Fig. 11. Comparison of different optimization methods, *i.e.*, ISTA, FISTA, and OGM, on the cow segmentation task with PL inference. (a) Objective vs. time. (b) Training error rate vs. time. (c) Test error rate vs. time.

FISTA to achieve convergence. This is because in each step, the gradient is estimated based upon pseudo-likelihood or approximated inference approaches and thus is inaccurate in general and may be particularly inaccurate in certain steps. Therefore, rather than updating the gradient direction by only considering the current gradient and previous gradient as FISTA does, OGM provides a more robust/accurate gradient update direction by considering the current as well as all the historical points.

For each given dataset, we notice that LBP and MF spend much less time than PL to achieve convergence. This indicates that LBP and MF are more efficient than PL to approximate the true likelihood. Moreover, we observe that MF generally converges faster than LBP.

The detailed statistics of running time and the associated standard derivations for ISTA, FISTA, and OGM over these three datasets are shown in Table I. In particular for OGM in Table I, we observe that MF consistently consumes less time than LBP and PL. Therefore, MF should be utilized if efficiency is the primary concern.

B. Effectiveness

We examine the effectiveness of the OGM for training RCRFs by comparing it with ISTA and FISTA over

TABLE I

THE DETAILED STATISTICS OF RUNNING TIME AND THE ASSOCIATED STANDARD DERIVATIONS FOR ISTA, FISTA, AND OGM OVER THREE DATASETS BASED UPON LBP, MF, AND PL. (TIME: SECOND (S))

Tasks and methods		ISTA	FISTA	OGM
car segmentation	LBP	5513.67(±424.83)	3039.89(±612.33)	**2505.00(±259.85)**
	MF	4040.17(±401.73)	2945.60(±267.35)	**1941.54(±113.10)**
	PL	11315.86(±1036.63)	10974.47(±869.44)	**8871.30(±851.15)**
horse segmentation	LBP	5438.36(±276.48)	3392.03(±405.10)	**2418.48(±58.96)**
	MF	3887.54(±228.26)	3151.10(±497.62)	**2370.22(±661.57)**
	PL	25126.93(±832.84)	20588.54(±1981.91)	**12854.96(±2207.36)**
cow segmentation	LBP	7570.60(±2873.15)	2062.85(±696.79)	**1494.78(±128.79)**
	MF	7402.97(±124.27)	2830.93(±428.27)	**857.85(±72.47)**
	PL	62892.08(±2873.15)	57244.77(±696.79)	**48837.21(±428.79)**

TABLE II

THE DETAILED STATISTICS OF TEST ERROR RATE AND THE ASSOCIATED STANDARD DERIVATIONS FOR ISTA, FISTA, AND OGM OVER THREE DATASETS BASED UPON LBP, MF, AND PL. (TEST ERROR RATE: %)

Tasks and methods		ISTA	FISTA	OGM
car segmentation	LBP	35.94(±3.24)	16.43(±3.96)	**13.16(±0.23)**
	MF	30.47(±4.82)	**13.84(±1.65)**	15.36(±2.02)
	PL	39.99(±3.94)	29.17(±2.48)	**27.87**(±1.86)
horse segmentation	LBP	35.48(±3.86)	16.69(±0.82)	**16.49(±2.07)**
	MF	35.67(±4.52)	16.19(±0.56)	**15.51(±0.51)**
	PL	40.67(±2.16)	27.59(±2.61)	**20.23(±2.31)**
cow segmentation	LBP	7.57(±0.12)	**6.34(±0.07)**	6.62(±0.21)
	MF	7.66(±0.12)	**6.50(±0.24)**	7.13(±0.22)
	PL	39.58(±0.12)	35.00(±0.07)	**13.27(±0.21)**

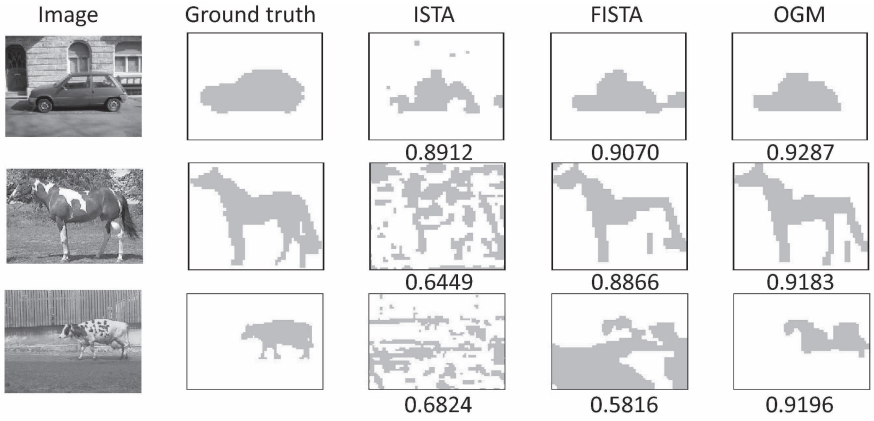


Fig. 12. Examples of segmentation results. The first row is car segmentation results based upon LPB from the TU Darmstadt car side dataset; the second row is horse segmentation results based upon MF from the Weizmann horse dataset; the third row is cow segmentation results based upon PL from the TU Darmstadt cow side dataset. The figure below each image represents test accuracy.

three datasets as well as three approximated inference methods. In particular, we plot the training error rate versus time of ISTA, FISTA, and OGM in Figure Figure 3b, 4b, 5b, 6b, 7b, 8b, 9b, 10b, and 11b; and we plot the test error rate versus time of ISTA, FISTA, and OGM in Figure 3c, 4c, 5c, 6c, 7c, 8c, 9c, 10c, and 11c.

For each given pair of dataset and inference method, we observe that when the training time increases, both the training error rate and test error rate of ISTA, FISTA, and OGM decrease in general. In particular, the training error rate and test error rate of OGM generally decrease faster than the other two. This is because (1) the gradient direction of OGM is determined by the current gradient as well as all historical gradients and thus may be more reliable than those of ISTA and FISTA; (2) OGM utilizes a smooth technique to approximate the ℓ_1 norm and the approximation error can be well controlled (Theorem 1).

For each dataset, we examine the effectiveness of different approximate inference methods. We notice that LBP and MF generally achieve lower test error rate than PL. This is because PL tends to overestimate the edge features [20] and thus cannot provide an accurate approximation for the true likelihood as LBP and MF do.

The detailed statistics of test error rate and the associated standard derivations for ISTA, FISTA, and OGM over three datasets are shown in Table II. In particular, for OGM in Table II, we notice that LBP in general achieves better performance than MF and PL. In this case, LBP is preferred.

C. Segmentation Results

We examine three examples of segmentation results in Figure 12. For car and cow segmentation results based upon LBP and PL, respectively, we observe that both the results of ISTA and FISTA contain more false positive

foreground than OGM. For horse segmentation results based upon MF, we notice that the horse contour labeled with OGM is more accurate than ISTA and FISTA. These results indicate that OGM can achieve better segmentation results than ISTA and FISTA.

VI. Conclusion

In this paper, we proposed robust conditional random fields (RCRFs) which are capable of selecting relevant features to perform image segmentation or labeling automatically. To train RCRFs efficiently, we presented an optimal gradient method (OGM), achieving the optimal convergence rate $O(1/k^2)$ that is superior to the convergence rate $O(1/k)$ of previously used first-order optimization methods. OGM works effectively in the sense that the gradient direction in each iteration is determined jointly by the current gradient and historical gradients. Moreover, the step size is completely controlled by the Lipschitz constant, and the approximation accuracy can be well controlled. Empirical studies based upon three real-world image segmentation tasks demonstrated the efficacy of OGM in training RCRFs.

Although OGM has shown its efficiency and effectiveness, it still converges slowly or even is infeasible for scalable data. Therefore, in the future, it would be interesting to investigate how to conduct distributed optimization or online learning based upon OGM.

Appendix
Proof of Theorem 3

Proof: Let the optimal solution be $\boldsymbol{\theta}^*$. Since $\mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta})$ is strongly convex, its second-order derivative is nonnegative and thus its second-order Taylor expansion is larger than its first-order Taylor expansion, *i.e.*,

$$\begin{aligned}\mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^*) &= \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m) \\ &\quad + \langle \partial_{\boldsymbol{\theta}^m} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m), \boldsymbol{\theta}^* - \boldsymbol{\theta}^m \rangle \\ &\quad + \frac{1}{2}(\boldsymbol{\theta}^* - \boldsymbol{\theta}^m)^T \partial^2_{(\boldsymbol{\theta}^m)^2} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m)(\boldsymbol{\theta}^* - \boldsymbol{\theta}^m) \\ &\quad + O^3(\boldsymbol{\theta}^*), \\ \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^*) &\geq \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m) \\ &\quad + \langle \partial_{\boldsymbol{\theta}^m} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m), \boldsymbol{\theta}^* - \boldsymbol{\theta}^m \rangle.\end{aligned}$$

Therefore,

$$\begin{aligned}\Psi_k &\leq \frac{L}{\sigma_1} d_1(\boldsymbol{\theta}^*) + \sum_{m=0}^{k} \frac{m+1}{2} \Big[\mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m) \\ &\quad + \Big\langle \partial_{\boldsymbol{\theta}^m} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^m), \boldsymbol{\theta}^* - \boldsymbol{\theta}^m \Big\rangle \Big] \\ &\leq \frac{L}{\sigma_1} d_1(\boldsymbol{\theta}^*) + \sum_{m=0}^{k} \frac{m+1}{2} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^*). \\ &= \frac{L}{\sigma_1} d_1(\boldsymbol{\theta}^*) + \frac{(k+1)(k+2)}{4} \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^*).\end{aligned}$$

According to Theorem 2, we have

$$\begin{aligned}\alpha \mathcal{O}_{RCRF,\mu}(\mathbf{s}^k) &\leq \frac{L}{\sigma_1} d_1(\boldsymbol{\theta}^*) + \alpha \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^*), \\ \alpha &= (k+1)(k+2)/4.\end{aligned}$$

Therefore, the accuracy at the k^{th} iteration round is

$$\mathcal{O}_{RCRF,\mu}(\mathbf{s}^k) - \mathcal{O}_{RCRF,\mu}(\boldsymbol{\theta}^*) \leq \frac{4L d_1(\boldsymbol{\theta}^*)}{\sigma_1 (k+1)(k+2)}.$$

Therefore, OGM for training CRFs converges at rate $O\left(1/k^2\right)$, and the minimum iteration number to reach a ε accurate solution is $O\left(1/\sqrt{\varepsilon}\right)$.

Acknowledgment

The authors greatly thank the handling Associate Editor and all anonymous reviewers for their positive support and constructive comments for improving the quality of this paper.

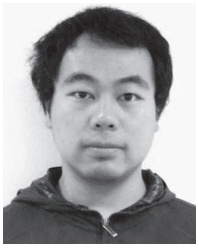
References

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.
[2] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Image Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[3] J. M. Bioucas-Dias and M. A. T. Figueiredo, "Multiplicative noise removal using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1720–1730, Jul. 2010.
[4] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguistics*, vol. 22, no. 1, pp. 39–71, Mar. 1996.
[5] C. Bhole, N. Morsillo, and C. Pal, "3D segmentation in CT imagery with conditional random fields and histograms of oriented gradients," in *Proc. 2nd Int. Conf. Mach. Learn. Med. Imag.*, 2011, pp. 326–334.
[6] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop (CVPRW)*, Jun. 2004, p. 46.
[7] J. Besag, "Statistical analysis of non-lattice data," *J. Roy. Statist. Soc. D*, vol. 24, no. 3, pp. 179–195, Sep. 1975.
[8] P. D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.
[9] S. Beigpour, C. Riess, J. van de Weijer, and E. Angelopoulou, "Multi-illuminant estimation with conditional random fields," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 83–96, Jan. 2014.
[10] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *Proc. Empirical Methods Natural Lang. Process.*, vol. 10. 2002, pp. 1–8.
[11] Y. Chen, R. Ranftl, and T. Pock, "Insights into analysis operator learning: From patch-based sparse models to higher order MRFs," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1060–1072, Mar. 2014.
[12] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vis. Res.*, vol. 20, no. 10, pp. 847–856, 1980.
[13] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 380–393, Apr. 1997.
[14] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2005, pp. 363–370.
[15] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–3145, Dec. 2010.
[16] Y. Fu, M. Liu, and T. S. Huang, "Conformal embedding analysis with local graph modeling on the unit hypersphere," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–6.
[17] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.
[18] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," unpublished.
[19] X. He, R. S. Zemel, and M. A. Carreira-Perpindn, "Multiscale conditional random fields for image labeling," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun./Jul. 2004, pp. II-695–II-702.
[20] S. Kumar and M. Hebert, "Discriminative fields for modeling spatial dependencies in natural images," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. 2004.
[21] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. ECCV Workshop Statist. Learn. Comput. Vis.*, 2004, pp. 17–32.

[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[23] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19. 2006, pp. 801–808.
[24] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.
[25] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 286–299.
[26] L. Liu, M. Yu, and L. Shao, "Multiview alignment hashing for efficient image search," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 956–966, Mar. 2015.
[27] R. J. Lipton and R. E. Tarjan, "A separator theorem for planar graphs," *SIAM J. Appl. Math.*, vol. 36, no. 2, pp. 177–189, 1979.
[28] S. Li and Y. Fu, "Low-rank coding with b-matching constraint for semi-supervised classification," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1472–1478.
[29] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proc. COLING*, 2002, pp. 1–7.
[30] T. P. Minka, "Algorithms for maximum-likelihood logistic regression," Dept. Statist., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. 758, 2001.
[31] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer-Verlag, 1999.
[32] Y. Nesterov, "A method for solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Math. Doklady*, vol. 27, no. 2, pp. 372–376, 1983.
[33] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.
[34] F. Peng and A. McCallum, "Accurate information extraction from research papers using conditional random fields," in *Proc. Human Lang. Technol. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2004, pp. 329–336.
[35] P. Pletscher, C. S. Ong, and J. M. Buhmann, "Spanning tree approximations for conditional random fields," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, 2006, pp. 408–415.
[36] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[37] X. Qian, X. Jiang, Q. Zhang, X. Huang, and L. Wu, "Sparse higher order conditional random fields for improved sequence labeling," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 849–856.
[38] D. Song and D. Tao, "C1 units for scene classification," in *Proc. 19th IEEE Int. Conf. Pattern Recognit.*, Tampa, FL, USA, Dec. 2008, pp. 1–4.
[39] D. Song and D. Tao, "Biologically inspired feature manifold for scene classification," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 174–184, Jan. 2010.
[40] D. Song, W. Liu, D. A. Meyer, R. Ji, and D. Tao, "Rank preserving hashing for rapid image search," in *Proc. Data Compress. Conf.*, Snowbird, UT, USA, 2015, pp. 353–362.
[41] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proc. Human Lang. Technol. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2003, pp. 134–141.
[42] K. Sato and Y. Sakakibara, "RNA secondary structural alignment with conditional random fields," *Bioinformatics*, vol. 21, no. 2, pp. 237–242, 2005.
[43] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1001–1013, Jul. 2014.
[44] L. Shao, D. Wu, and X. Li, "Learning deep and wide: A spectral method for learning deep networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2303–2308, Dec. 2014.
[45] N. Sokolovska, T. Lavergne, O. Cappé, and F. Yvon, "Efficient learning of sparse conditional random fields for supervised sequence labeling," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 6, pp. 953–964, Dec. 2010.
[46] B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, "Max-margin parsing," in *Proc. Empirical Methods Natural Lang. Process.*, 2004, pp. 1–8.
[47] D. Tao, X. Lin, L. Jin, and X. Li, "Principal component 2-dimensional long short-term memory for font recognition on single Chinese characters," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2015.2414920.
[48] G. Tsechpenakis and D. Metaxas, "CoCRF deformable model: A geometric model driven by collaborative conditional random fields," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2316–2329, Oct. 2009.
[49] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy, "Accelerated training of conditional random fields with stochastic gradient methods," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 969–976.
[50] C. Wang, N. Komodakis, and N. Paragios, "Markov random field modeling, inference & learning in computer vision & image understanding: A survey," *Comput. Vis. Image Understand.*, vol. 117, no. 11, pp. 1610–1627, Nov. 2013.
[51] Y. Weiss, "Comparing the mean field method and belief propagation for approximate inference in MRFs," in *Advanced Mean Field Methods*, D. Saad and M. Opper, Eds. Cambridge, MA, USA: MIT Press, 2001.
[52] M. Wytock and Z. Kolter, "Sparse Gaussian conditional random fields: Algorithms, theory, and application to energy forecasting," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1265–1273.
[53] C. Xu, D. Tao, and C. Xu, "Large-margin multi-view information bottleneck," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1559–1572, Aug. 2014.
[54] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2015.2417578.
[55] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring Artificial Intelligence in the New Millennium*, D. Saad and M. Opper, Eds. 2003, ch. 8, pp. 239–269.
[56] R. Yan, L. Shao, and Y. Liu, "Nonlocal hierarchical dictionary learning using wavelets for image denoising," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4689–4698, Dec. 2013.
[57] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma, "2D conditional random fields for Web information extraction," in *Proc. 22nd Int. Conf. Mach. Learn.*, Bonn, Germany, 2005, pp. 1044–1051.
[58] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3241–3253, Aug. 2014.
[59] T. Zhou, D. Tao, and X. Wu, "NESVM: A fast gradient method for support vector machines," in *Proc. IEEE 10th Int. Conf. Data Mining*, Dec. 2010, pp. 679–688.

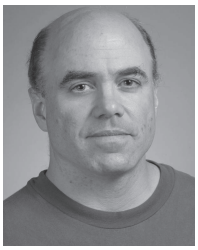
Dongjin Song received the B.Eng. degree from the University of Science and Technology of China, in 2007, and the M.Phil. degree from Hong Kong Polytechnic University, in 2010. He is currently pursuing the Ph.D. degree with the Department of Electrical Computer Engineering, University of California at San Diego. He is mainly interested in machine learning, optimization, applied statistics, and their applications for recommender system, social networks analysis, and computer vision. His research results have been published at the IEEE TRANSACTIONS ON IMAGE PROCESSING, ACM SIGKDD, AAAI, DCC, and ASONAM. He received the best scientific project prize in the Data for Development Challenge in 2013, and top 10 finalist in the Italian Telecom Big Data Challenge in 2014.

Wei Liu is currently a Research Scientist of IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. He received M.Phil. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 2012. He is the recipient of 2011-2012 Facebook Fellowship and the 2013 Jury Award for best thesis of Department of Electrical Engineering, Columbia University. He holds adjunct faculty positions in Rensselaer Polytechnic Institute and Stevens Institute of Technology. His research interests include machine learning, data mining, information retrieval, computer vision, pattern recognition, and image processing. He has published more than 80 peer-reviewed journal and conference papers, including Proceedings of IEEE, IEEE TMI, IEEE TIP, IEEE TCSVT, ACM TIST, NIPS, ICML, KDD,CVPR, ICCV, ECCV, IJCAI, AAAI, MICCAI, SIGIR, SIGCHI, ACL, etc. His current research work is geared to large-scale machine learning, big data analytics, large-scale multimedia content search engine, mobile computing, and parallel computing.



Tianyi Zhou received the B.Eng. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2008. He is now working on machine learning at the University of Washington, Seattle. He was a Research Assistant at Nanyang Technological University, Singapore, and the University of Technology Sydney at Australia, and was a Research Intern at Microsoft research, Redmond. He has published 20+papers on NIPS, ICML, AISTATS, KDD, ICDM, IJCAI, ISIT, MachineLearning (Springer), and the IEEE TRANSACTIONS ON IMAGE PROCESSING, which have been cited by 400+ times. He won ICDM best student paper award at 2013. His current research mainly focuses on developing streaming, distributed, and subset selection based large-scale machine learning techniques.

Dacheng Tao (F'15) is currently a Professor of Computer Science with the Centre for Quantum Computation and Intelligent Systems, and the Faculty of Engineering and Information Technology with the University of Technology, Sydney. He mainly applies statistics and mathematics to data analytics and his research interests spread across computer vision, data science, image processing, machine learning, neural networks, and video surveillance. His research results have expounded in one monograph and 100+ publications at prestigious journals and prominent conferences, such as the IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM, and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM'07, the best student paper award in IEEE ICDM'13, and the 2014 ICDM 10 Year Highest-Impact Paper Award.



David A. Meyer is currently a Professor with the Mathematics Department and a Research Scholar at the Institute on Global Conflict and Cooperation at the University of California, San Diego (UCSD). He received the B.A./M.A. degree from The Johns Hopkins University, and the Ph.D. degree from MIT, all in mathematics. He held Visiting and post-doctoral positions in the Mathematics and Physics (Relativity Group) Department at Syracuse University, and the Physics Department at Duke University, before coming to UCSD as a Post-Doctoral Researcher in the Physics Department. His research interests range from mathematical physics, specifically quantum computing, to data analysis, including time series and image processing.