Projected Gradient method for Inference in Markov Random Fields

You Lu
Department of Computer Science
Virginia Tech
you.lu@vt.edu

Abstract

Belief propagation(BP) has been widely used to compute marginal probabilities for Markov random Fields(MRFs). However, BP has no convergence guarantee even on the convex objective function. Gradient based optimization method has been studied for years. With properly set learning rate, it is guaranteed that will converge to the optimum. Thus, in this project, we try to use projected gradient method to replace the BP to compute the marginals for the MRFs. In our experiments, we found that even though projected gradient method is convergent, it is much slower than BP.

1 Introduction

Markov random fields(MRFs) Wainwright et al. (2008) are widely used in many applications, including image segmentation Domke (2013), part-of-speech tagging Chen et al. (2011), and social network analysis Zheleva et al. (2010).

A fundamental limitation to their practical use is the difficulty associated with computing various statistical quantities, e.g., marginals, data likelihoods etc. Belief Propagation(BP) is the most widely used method for computing the marginals. Convex BP Wainwright (2006) uses a convex variational relaxation to construct the objective function. However, even the objective is convex, the convex BP is not always convergent Kolmogorov (2006). Some methods have been proposed to solve this problem. The sequential tree-reweighted BP has convergence guarantee, but it requires to modify the tree appearance probabilities at each iteration. Roosta et al Roosta et al. (2008) show that parameter of the MRF and the edge weights satisfy some relationships, the convex BP will converge.

Gradient based optimization methods Boyd and Vandenberghe (2004) has been well-studied for many years. They are easy to implement, have low iteration complexity, and theoretical convergence guarantee. In this paper, we try to replace the BP with projected gradient method to compute the marginals, i.e., beliefs, for MRFs.

2 Backgrounds

In this section, we introduce MRFs and convex BP.

2.1 Markov Random Fields

A Markov random field (Wainwright et al., 2008; Wainwright, 2006) associates a collection of random variables, $X = \{X_1, ..., X_n\}$, with the vertices of an undirected graph. Consider an undirected graph $G = (V, E)$, where $V = 1, ..., n$ is the vertex set and $E \subset V \times V$ is the edge set. Each vertex s is associated with a multinomial random variable X_s, taking values in the set $\mathcal{X}_s = \{1, 2, ..., m\}$. We

Preprint. Work in progress.

use x_s to denote a particular state of X_s. In this paper, we focus on pairwise Markov random fields. The probability mass function of a pairwise Markov random field can be written as

$$p(x|\theta) = \exp\{\sum_{s\in V} \theta_s(x_s) + \sum_{(s,t)\in E} \theta_{st}(x_s, x_t) - A(\theta)\}, \tag{1}$$

where θ is the parameter of the Markov random field that we want to learn. We define θ as

$$\begin{aligned} \theta_s(x_s) &:= \sum_{j=1}^{m} \theta_{s;j}\mathbb{I}_j(x_s), \\ \theta_{st}(x_s, x_t) &:= \sum_{j=1}^{m}\sum_{k=1}^{m} \theta_{st;jk}\mathbb{I}_j(x_s)\mathbb{I}_k(x_t), \\ \theta &= \{\theta_s|s \in V\} \cup \{\theta_{st}|(s,t) \in E\}, \end{aligned}$$

where $\theta_s = \{\theta_{s;j}|j = 1, ..., m\}$ and $\theta_{st}(x_s, x_t) = \{\theta_{st;jk}|j, k = 1, ..., m\}$ are vectors of parameters, $\mathbb{I}$ is the indicator function

$$\mathbb{I}_j(x_s) = \begin{cases} 1 & \text{if } x_s = j \\ 0 & \text{otherwise,} \end{cases}$$

and the function

$$A(\theta) = \log \sum_X \exp\{\sum_{s\in V} \theta_s(x_s) + \sum_{(s,t)\in E} \theta_{st}(x_s, x_t)\} \tag{2}$$

is the logarithm of the normalizing constant.

2.2 Convex Belief Propagation

In most situations, when the network is not a tree, the normalizing function $A(\theta)$ is intractable. One popular method is to use *belief propagation* (BP) (Wainwright et al., 2008; Pearl, 2014) to approximate the $A(\theta)$ function. In this paper, we focus on a form of convex belief propagation that optimizes an alternate inference objective based on a distribution of spanning trees over the MRF graph (Wainwright, 2006; Wainwright et al., 2005). This method, known as *tree-reweighted belief propagation* (TRW-BP), approximates $A(\theta)$ with a convex function $B(\theta)$. The convex function $B(\theta)$ is defined as

$$B(\theta) = \max_{\tau\in\mathbb{L}(G)} \{\langle\theta, \tau\rangle - B^*(\tau)\}, \tag{3}$$

where

$$\mathbb{L}(G) := \{\tau \in \mathbb{R}_+^d | \sum_{x_s} \tau_s(x_s) = 1, \sum_{x_t} \tau_{st}(x_s, x_t) = \tau_s(x_s)\},$$

and τ is an element of $\mathbb{L}(G)$. The vector τ is called the pseudo-marginal or belief vector. Specifically, τ_s is the unary belief of vertex s and τ_{st} is the pairwise belief of edge (s, t). For convenience, we define that

$$\tau = \{\tau_s|s \in V\} \cup \{\tau_{st}|(s,t) \in E\}.$$

In the analysis of convex belief propagation, we define a strongly convex function $B^*(\tau)$, which has the following form:

$$B^*(\tau) = -\sum_{s\in V} H(\tau_s) + \sum_{(s,t)\in E} \rho_{st} I_{st}(\tau_{st}),$$

where

$$\begin{aligned} H(\tau_s) &= \sum_{x_s} \tau_s(x_s) \log \tau_s(x_s) \\ I_{st}(\tau_{st}) &= \rho_{st} \sum_{x_s,x_t} \tau_{st}(x_s, x_t) \log \frac{\tau_{st}(x_s, x_t)}{\tau_{x_s}(x_s)\tau_{x_t}(x_t)} \end{aligned}$$

are the unary entropy and mutual information, respectively, and ρ_{st} is the edge appearance probability under the distribution of spanning trees.

Equation 3 can be solved via TRW-BP (Wainwright et al., 2005). Let λ_{st} be the message from vertex t to vertex s. The update rules of messages and beliefs are as follows:

$$\lambda_{st} \propto \sum_{s} \exp\{\frac{1}{\rho_{st}}\theta_{st}\} \times \frac{\tau_s}{\lambda_{ts}}, \tag{4}$$

where

$$\tau_s \propto \exp\{\theta_s\} \times \prod_{t \in N(s)} \lambda_{ts}^{\rho_{st}}, \tag{5}$$

and

$$\tau_{st} \quad \propto \quad \exp\{\theta_s + \theta_t + \frac{1}{\rho_{st}}\theta_{st}\} \times \frac{\prod_{v \in N(s)} \lambda_{vs}^{\rho_{vs}}}{\lambda_{ts}^{1-\rho_{st}}} \times \frac{\prod_{v \in N(s)} \lambda_{vt}^{\rho_{vt}}}{\lambda_{st}^{1-\rho_{ts}}}. \tag{6}$$

From Equation 4, Equation 5, and Equation 6 we can see that the update rules for the beliefs and messages are just closed form solutions of them. It is efficient but has no convergence guarantee.

We focus on the tree-reweighted form of convex BP in this project, but other forms of convex BP can also be used in our approach.

3 Projected Gradient method for Markov Random Fields

Notice that the computation of τ can be seen as a constraint optimization problem:

$$\begin{aligned} &\max_{\tau} L(\tau) = \langle \theta, \tau \rangle + \sum_{s \in V} H(\tau_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}) \\ &\text{s.t.} \quad \tau \in \mathbb{L}(G) \end{aligned} \tag{7}$$

To use projected gradient to optimize τ, we need to first use gradient based method to update τ, and then project the updated τ to the polytope $\mathbb{L}(G)$.

3.1 Gradient Ascent to Update Beliefs

The gradients of unary belief τ_s and the pair-wise belief τ_{st} are different. The gradient of τ_s is:

$$\nabla_{\tau_s} L(\tau) = \theta_s - 1 - \log \tau_s + \sum_{t \in N(s)} \rho_{st}.$$

The gradient of τ_{st} is

$$\nabla_{\tau_{st}} L(\tau) = \theta_{st} - \rho_{st}(1 + \log \tau_{st} - \log \tau_s - \log \tau_t)$$

Thus, we can have the update rule of τ that

$$\tau' = \tau_t + \alpha_t \nabla_\tau L(\tau) \tag{8}$$

where τ_t is the τ at iteration t, the α_t is the gradient, the τ' is the unprojected τ, and $\nabla_\tau L(\tau) = [\nabla_{\tau_{v_1}} L(\tau), ..., \nabla_{\tau_{v_V}} L(\tau), \nabla_{\tau_{e_1}} L(\tau), ..., \nabla_{\tau_{e_E}} L(\tau)]^T$ is the concatenation gradient vector of τ.

3.2 Project Beliefs to the Feasible Polytope

3.2.1 Rewrite constraints

To do the projection, we need to first rewrite the constraint, i.e., $\tau \in \mathbb{L}(G)$ to the equality and inequality constraints. Note that τ must be non-negative. Thus, the inequality constraints of τ will be:

$$\tau \geq 0$$

Then we rewrite the constraints $\sum_{x_s} \tau_s(x_s) = 1$, and $\sum_{x_t} \tau_{st}(x_s, x_t) = \tau_s(x_s)$ to the

$$P\tau = b$$

format. Each row of P is a constraint of τ, and since it is a matrix product, the column dimension of P is equal to the τ's dimension. Note that $\tau = [\tau_{v_1}, ..., \tau_{v_V}, \tau_{e_1}, ..., \tau_{e_E}]^T$. Let the number of states of the MRF be D, then the τ's dimension is $V \times D + E \times D^2$.

For each node s, we need to have $\sum_{x_s} \tau_s(x_s) = 1$, so we have V equality constraints for unary beliefs. For each edge (s, t) and each state i, we need to have that $\sum_{x_t} \tau_{st}(x_{si}, x_t) = \tau_s(x_{si})$, so we have $2 \times E \times D$ equality constraints for pair-wise beliefs. Thus, the P has $V + 2 \times E \times D$ rows. The detailed construction of P is as below.

Let P_1 be the sub-matrix of P that represents the constraints for unary beliefs, and P_2 be the sub-matrix that represents the constraints for pair-wise beliefs. We have that $P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$. Let P_{1i} be the constraint for the ith unary belief. Since we want to have that $\sum_{x_s} \tau_s(x_s) = 1$, we can construct the P_{1i} as below:

$$P_{1i} = [0, 0, ..., 0 \quad \underbrace{1, ..., 1}_{\text{coefficients of } \tau_i} \quad , 0, ..., 0] \tag{9}$$

The intuition of construct P_2 is that for any edge (s, t), we need to have that $\sum_{x_t} \tau_{st}(x_s, x_t) - \tau_s(x_s) = 0$, which implies that for any state d of x_s, we should have that $\sum_{x_t} \tau_{st}(x_{sd}, x_t) - \tau_s(x_{sd}) = 0$. Let (s, t) be the jth edge in the network. Note that for each edge, it has $2 \times D$ constraints. For the dth state of v_k, we can construct its corresponding constraint as below:

$$P_{2,2D(j-1)+d} = [0, ..., 0, \underbrace{-1}_{\text{coefficients of } \tau_i(x_{sd})}, 0, ..., 0, \underbrace{1, ..., 1}_{\text{coefficients of } \tau_{kl}(x_{kd}, x_t)}, 0, ..., 0] \tag{10}$$

Next, we turn to construct the b to make $P\tau = b$. Let $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$. Note that for each unary belief, its constraint is $\sum_{x_s} \tau_s(x_s) = 1$. Thus we have that $b_1 = [1, ..., 1]^T$. For each pair-wise belief, we have that $\sum_{x_t} \tau_{st}(x_s, x_t) - \tau_s(x_s) = 0$. Thus, we have that $b_2 = [0, ..., 0]^T$.

3.2.2 Treat the Projection as a Quadratic Programming Problem

With the written constraints, we can now rewrite the projection step as a quadratic programming problem. Let τ' be the updated τ obtained from Equation 8. Let $\Pi(z)$ be the projection function, we can treat the projection as the following problem:

$$\Pi(z) = \min_{\tau} ||\tau - z||_2^2 \tag{11}$$

$$\text{s.t.} \quad P\tau = b, \quad \tau \geq 0 \tag{12}$$

$$\tag{13}$$

3.3 Projected Gradient method for Computing the Marginals

With the gradient based update rule for τ, and the projection function, we can now build the projected gradient method for computing the beliefs. The algorithm details are in Alrorithm 1. Note that the complexity of computing the gradient is the same as using BP to update the beliefs. Since the projection step requires to solve a quadratic programming problem, it may make this algorithm slower than BP. However, since projected gradient method has convergence guarantee, and $L(\tau)$ is a strongly convex function, when the learning rate is properly set, this algorithm is guaranteed to converge to the optimum.

Algorithm 1 Projected gradient method for computing beliefs

1: Initialize $t = 1$ and τ_t.
2: While τ has not converged
3: $\quad \tau' = \tau_t + \alpha_t \nabla_\tau L(\tau)$
4: $\quad \tau_{t+1} = \Pi(\tau')$.
5: $\quad t = t + 1$
6: end

4 Empirical Study

In this section, we test our method's performance on image segmentation. We use convex BP learning framework to learn the parameters for the MRF. We then compute the beliefs for each image in the test set using the learned parameters.

Experiment settings. We use horse dataset Kolesnikov et al. (2014) in our experiments. We randomly select 30 images and compute the beliefs for them. We fix the image size as 10×10. We compare the gradient-based inference method with the convex BP. We test the number of iterations needed to converge, and the per-iteration running time for each method.

We implement the methods in Python. To do the projection, we use the implemented function in CVXPY to solve the quadratic programming problem, i.e., Equation 13.

Convergence analysis. In our experiments, we find that the gradient-based inference is slower than the convex BP, i.e., Table 1. Since it needs to do the projection, i.e., Step 4 in Algorithm 1, every iteration, it is slower than the convex BP. Besides, it needs more iterations to converge. The Figure 1 plots the convergence curves for 3 different images.

Table 1: Comparisons of convex BP and Gradient-based inference on number of iterations needed to converge and per-iterations running time.

Method	Averaged per-iteration running time(s)	Number of iterations needed
Gradient-based Inference	0.214799	153.06
Convex BP	0.000415	84.23

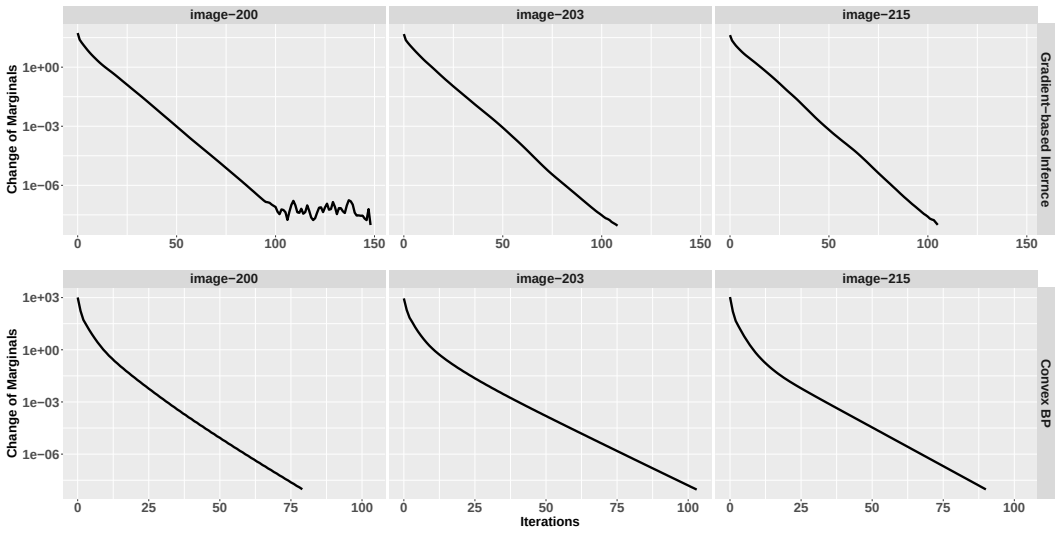

Figure 1: The convergence curves of three examples.

5 Conclusion

In this project, we try to use projected gradient method to replace BP to computing the marginals for MRFs. The update of τ can be divided to two steps. First, we use gradient ascent to update τ, and then we project the updated τ to the feasible polytope. We treat the projection step as a quadratic programming problem. We conduct experiments on small networks, i.e., 10×10 grid network. We found that the projected gradient method indeed will converge to the optimal. We compare it with the BP and found that BP is much faster than the projected gradient method. We conclude that this is because to project the τ' to the feasible polytope, we need to solve a quadratic programming problem, which needs a lot of time. In the future, we will try to accelerate the projection using other methods.

References

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Chen, D., Dyer, C., Cohen, S. B., and Smith, N. A. (2011). Unsupervised bilingual pos tagging with markov random fields. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 64–71. Association for Computational Linguistics.

Domke, J. (2013). Learning graphical model parameters with approximate marginal inference. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2454–2467.

Kolesnikov, A., Guillaumin, M., Ferrari, V., and Lampert, C. H. (2014). Closed-form approximate crf training for scalable image segmentation. In *European Conference on Computer Vision*, pages 550–565. Springer.

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1568–1583.

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Roosta, T. G., Wainwright, M. J., and Sastry, S. S. (2008). Convergence analysis of reweighted sum-product algorithms. *IEEE Transactions on Signal Processing*, 56(9):4293–4305.

Wainwright, M. J. (2006). Estimating the"wrong"graphical model: Benefits in the computation-limited setting. *Journal of Machine Learning Research*, 7(Sep):1829–1859.

Wainwright, M. J., Jaakkola, T. S., and Willsky, A. S. (2005). A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Zheleva, E., Getoor, L., and Sarawagi, S. (2010). Higher-order graphical models for classification in social and affiliation networks. In *NIPS Workshop on Networks Across Disciplines: Theory and Applications*, volume 2. Citeseer.