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

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

Bert Huang
Department of Computer Science
Virginia Tech

Consistency-based Approximate Inference for Learning Graphical Model Parameters

Shuangfei Fan
Virginia Tech
Blacksburg, VA 24061
sophia23@vt.edu

Sirui Yao
Virginia Tech
Blacksburg, VA 24061
ysirui@vt.edu

Bert Huang
Virginia Tech
Blacksburg, VA 24061
bhuang@vt.edu

Abstract

Graphical model provides a way of representing probabilistic relationships between random variables, representing the data that we are interested in by a suitable graphical model can help us understand more about the relationship between difference variables. We provide a method to increase the consistency of the predictions with approximate inference by optimizing over a dual objective which extends the energy functional with penalty on disagreement. The experiments on the problem of image segmentation show that the dual objective performs better both in accuracy and consistency.

1 Introduction

Graphical model provides a way of representing probabilistic relationships between random variables, therefore represent the data that we are interested in by a suitable graphical model can help us understand more about the relationship between difference variables. For example, in the task of pixel level label prediction, though the feature representation plays an important role in classify individual pixels, information such as image edges, appearance consistency and spatial consistency are also important to be taken into consideration when assigning labels, so that we can more comprehensively account for all the factors and obtain accurate and precise results.

For this reason, Probabilistic graphical models have gained popularity in dealing with pixel-level labelling tasks to help enhance the prediction accuracy. They can provide smoothness constraints that encourage label agreement between similar pixels as well as spatial and appearance consistency of the labelling output. Therefore, it helps to refine weak and coarse pixel-level label predictions and produce sharp boundaries and fine-grained segmentations [10].

Various probabilistic graphical models have different properties. For pixel-level labelling tasks, since we need to consider both individual pixel feature representations as well as interactions between interaction among adjacent variables, Conditional Random Fields are the ideal choice. It combines the ability of graphical models to compactly model multivariate data with the ability of classification methods to perform prediction using large sets of input features [10].

One popular algorithm to run inference on Conditional Random Fields is Loopy belief propagation, whose efficiency is heavily dependent on the tree-width of the network. The computational and space complexity of this inference algorithm grow exponentially as the tree-width of the network increases. While in most real life applications such as image segmentation, the tree-width in the pixel grid is not small, therefore, exact inference in general Conditional Random Fields sometimes become infeasible. To deal with this, we adopted the strategy of truncated fitting. Unlike general inference algorithms, which are based on optimization, and iterates updates until some convergence threshold is reached, Truncated fitting is to fit the marginals produced after a fixed number of updates, with no assumption of convergence. Research shows that his leads to significant speedups [4].

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

The biggest contribution of our approach is taking into consideration the consistency between unary belief and pairwise belief, which is an important property that should be held in many problems that are solved by inference algorithm on the graphical model, we want to design a new learning objective function to include the requirement of consistency for approximate inference. The idea is to make all the pairwise beliefs on a variable they shared agree with each other on that variables belief. Therefore we can define a loss function to minimize the absolute value of the difference between them.

At last, we apply back propagation to compute the gradient of the PGM parameters for approximate inference to learn the model parameters. Since generally it is fast to find the gradient by automatic differentiation [8], Autodiff technique is also applied.

2 Literature Review

2.1 Learning Graphical Model Parameters with Approximate Marginal Inference[4]

While Maximum a Posteriori (MAP) estimate, which seeks to find the optimum joint prediction, is a popular inference problem in Conditional Random Fields prediction tasks, this paper points out the disadvantages of this strategy and turn the objective into selecting the most likely value for each component independently by adopting a different alternative utility function. The new inference problem is called Maximum Posterior Marginal (MPM) inference and mainly has two advantage over MAP. First, the actual maximizing joint probability $p(x|y)$ in a MAP estimate might be extremely small, so much so that the limited numbers of examples might not be necessary to exactly predict the true output. Second, MAP does not distinguish between a prediction that contains only a single error at some component x_j, and one that is entirely wrong. Note that the results of MAP and MPM inference will be similar if the distribution $p(x|y)$ is heavily peaked at a single configuration x.

They also introduce the strategy of truncated fitting for inference procedure considering that the computational and space complexity of exact inference algorithm grow exponentially as the tree-width of the network increases. Normally, inference algorithms are based on optimization, where one iterates updates until some convergence threshold is reached. In truncated fitting, marginals are generated only after a fixed number of updates. So instead of viewing the inference process as an optimization, it makes more sense to consider it as a large, nonlinear function, with no assumption of convergence for the inference procedure.

As for inference methods, they introduced two approaches, mean field and tree reweighted belief propagation. Both of them involve steps where the first step is to take a product of a set of terms, and then normalize.

2.2 Conditional Random Fields as Recurrent Neural Networks [10]

This paper introduces a new form of convolutional neural network that combines the strengths of Convolutional Neural Networks (CNNs) and Conditional Random Fields (CRFs)-based probabilistic graphical modeling. This algorithm is applied in the context of image segmentation. They build a conditional random filed by modeling pixel labels as random variables conditioned upon observations of individual image pixel features, then back-propagation is used to optimize model parameters. During back-propagation, error derivatives w.r.t. the filter inputs are calculated by sending the error derivatives w.r.t. the filter outputs through the same M Gaussian filters in reverse direction. CRF-RNN can work as a part of a traditional deep neural network. It is capable of passing on error differentials from its outputs to inputs during back-propagation based training of the deep network while learning CRF parameters.

They tested this system on the popular Pascal VOC segmentation benchmark, and achieves a new state-of-the-art. This proves that the uniting of the strengths of CNNs and CRFs in a single deep network can contribute to improvement on performance.

2.3 Discussion

All the prior work has built a solid foundation for us to apply our idea in the image segmentation task. They also provide advanced technical details to improve the performance of the Conditional Random Field framework. However, the consistency between unary belief and pairwise belief is not discussed

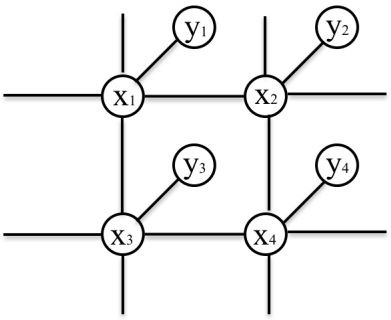


Figure 1: Conditional Random Field Grid

in those paper. Therefore, on top of their accomplishment, we want to do further research and study the concept of inconsistency so as to deepen our understanding on structure prediction.

3 Conditional random fields

Many tasks involve predicting a large number of variables that depend on each other as well as on other observed variables. Predicting Variables which are related, which is called Structured prediction. Structured prediction methods are essentially a combination of classification and graphical modeling[7][9].

CRF is a popular probabilistic method for structured prediction, which has wide application in many areas, including natural language processing, computer vision, and bioinformatics. Based on Markov Random Field (MRF), CRF is often interested in modeling the conditional probability of x, given observations y. For example, as shown in Figure 1, in the context of image segmentation, x represents the label of each pixel on the grid, while y stands for single pixel features. The conditional probability of x, given observations y involves the product of two terms is given as Eq.(1).The first term is the product over the set of cliques in the graph, while the second is over all individual variables[4].

$$p(x|y) = \frac{1}{Z(y)} \prod \phi(\mathbf{x_c}, y) \prod \phi(x_i, y) \tag{1}$$

4 Variational inference

The original belief propagation algorithm was proposed by Pearl in 1988 for finding exact marginals on trees, and turns out to be able to perform surprisingly well when applied to general graphs, those that can contain loops. Therefore, LBP is a message passing algorithm which is exact in trees and approximate in general graph.

Belief Propagation involves message passing. The first step is to multiply the factors assigned to each clique, forming the initial potentials. A node passes a message to an adjacent node only when it has received all incoming messages, excluding the message from the destination node to itself. For example, the clique c_i multiplies all incoming messages from its other neighbors with its initial clique potential, forming a factor whose scope is the clique. It then sums out all variables except those in the sepset between c_i and c_j, and sends the resulting factor as a message to c_j[5].

When a clique has received all messages, it multiplies them with its own initial potential. The result is a factor called the beliefs. Messages are calculated as Eq.(2);Belief are calculated as Eq.(3).

$$m_{s\to t}(x_t) := \sum_{x_s} (\phi_{st}(\mathbf{x_s}, \mathbf{x_t}) \prod_{\mathbf{u}\in\mathbf{Neighbor(s)}\setminus\mathbf{t}} \mathbf{m_{u\to s}}(\mathbf{x_s})) \tag{2}$$

$$b_t(x_t) \propto \prod_{s\in Neighbor(t)} m_{s\to t}(x_t) \tag{3}$$

5 Truncated fitting

Inference algorithms are based on optimization where one iterates updates until some convergence threshold is reached. However it is time-consuming in real world applications, so we derived algorithms for truncated fitting to fit the marginals produced after a fixed number of updates, with no assumption of convergence and this is proved to lead significant speedup[3]. In our experiment, we define the learning objective of loopy belief propagation in terms of the approximate marginals obtained after a fixed number of iterations, so in this way BP doesn't run until converge. Since this optimization problem is not convex, we need to define and differentiate a loss defined on the current predicted marginals. For the loss function, we can use the negative likelihood, however we need to reformulate the differentiate now, this procedure can be done either manually or automatically.

5.1 Autograd with warm start

For truncated fitting, instead of deriving the gradient manually we applied autograd[6] to calculate the gradient automatically. However one problem of using autograd to calculate the gradient is that it is going to be much more slower than using the gradient computed with the maximum likelihood formula. So one solution is to combine these two methods, we first optimize the problem with the gradient computed from the maximum likelihood for a fixed iteration as a fast warm start to get a better initialization than random value, then we continue with the autograd to fine tune it. The experiment showed that it is faster with a warm start than using autogard only.

5.2 Dual problem

Consistency is one of the important properties for a true distribution, we want it also to be true for our predicted distribution. So the most principled way to do this is to replace the energy functional with the dual objective with a constraint on agreement of the pairwise beliefs and the unary beliefs. Therefore by optimizing the dual problem we introduced penalties on the inconsistency and this also provide an upper bound on the negative log-likelihood[2] [1].

The classic loss function is the negative likelihood which is defined as:

$$-L(\boldsymbol{\theta}, \boldsymbol{x}) = -\log p(\boldsymbol{x}; \boldsymbol{\theta}) = -\boldsymbol{\theta} \cdot f(\boldsymbol{x}) + \boldsymbol{A}(\boldsymbol{\theta}) \tag{4}$$

where

$$\boldsymbol{A}(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in M} \boldsymbol{\theta} \cdot \boldsymbol{\mu} + H(\boldsymbol{\mu}) \tag{5}$$

is the log-partition function. Applying Danskin's theorem we have

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \frac{d\boldsymbol{A}}{d\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\mu} \in M} \boldsymbol{\theta} \cdot \boldsymbol{\mu} + H(\boldsymbol{\mu}) \tag{6}$$

However in general graphs, this optimization problem is intractable in both the polytope M and calculating the entropy. So for Loopy belief propagation, we define a polyhedral which is the outer bound of the marginal polytope and we also applied Bethe approximation to derive an approximation of energy functional $(\boldsymbol{A}(\boldsymbol{\theta}))$, therefore the negative likelihood becomes:

$$\begin{aligned} -L(\boldsymbol{\theta}, \boldsymbol{x}) = -\log p(\boldsymbol{x}; \boldsymbol{\theta}) &= -\boldsymbol{\theta} \cdot f(\boldsymbol{x}) + \boldsymbol{A}(\boldsymbol{\theta}) \\ &\approx -\boldsymbol{\theta} \cdot f(\boldsymbol{x}) + \widetilde{\boldsymbol{A}}(\boldsymbol{\theta}) \\ &= -\boldsymbol{\theta} \cdot f(\boldsymbol{x}) + \max_{\boldsymbol{\mu} \in F} \boldsymbol{\theta} \cdot \boldsymbol{\mu} + H_{\mathrm{B}}(\boldsymbol{\mu}) \end{aligned} \tag{7}$$

where

$$\widetilde{\boldsymbol{\mu}}(\boldsymbol{\theta}) = \frac{d\widetilde{\boldsymbol{A}}}{d\boldsymbol{\theta}} \tag{8}$$

In many applications we also want the pairwise beliefs and unary beliefs agree with each other, so we add penalty on the disagreement and derive the dual objective function:

$$-L_{\mathtt{dual}}(\boldsymbol{\theta}, \boldsymbol{x}) = -\boldsymbol{\theta} \cdot f(\boldsymbol{x}) + \max_{\boldsymbol{\mu} \in F} [\boldsymbol{\theta} \cdot \boldsymbol{\mu} + H_{\mathrm{B}}(\boldsymbol{\mu}) + \boldsymbol{\lambda}^{\top} c(\boldsymbol{\mu})] \tag{9}$$

6 Experiments

In this section, we describe experiments that test the performance of the proposed dual objective and whether it can help reduce the inconsistency between pairwise beliefs and unary beliefs.

6.1 Setup

The experiment is image segmentation problem which using a pairwise, 4-connected grid-graph and learning uses the L- BFGS optimization algorithm. Generally the threshold for training stage should be tight since a loose threshold may lead to a bad estimated risk gradient, and learning terminating with a bad search direction. However there is not much influence of a loose threshold during test time. So in the experiments, we use different convergence thresholds for learning stage an test time where for training we use 10^{-5} and for test we use 10^{-4}.

6.2 Data

We experimented with the Weizman horse dataset. It consists of 328 side-view color images of horses and the corresponding segmentation mask (the pixel is either 0 which represent background or 1 which represent horse). For our experiments, considering the computation cost of autograd we only use 32 images for training and 10 images for testing.

For unary features, we start with calculating the RGB values of each pixel. Since a linear classification in RGB space is not very likely to categorize the classes, we then expend with quadratic features and a bias feature, therefore the feature size is expended to 10.

6.3 Evaluation

We trained model with truncated fitting with 10 iterations of inference and test with truncated fitting with 3 iterations of inference. We experiment on two objective functions : energy functional and dual objective, the results are showing in Table 1, we also plot ROC curve in Fig. 3. The result shows that not only the inconsistency reduced by optimizing over dual objective, the error rate also decreased. We also plot the result of the segmentation on horse dataset in Fig. 3 The output shows that it can separate the horse from the background using not only unary features but also relational information based on graphical model.

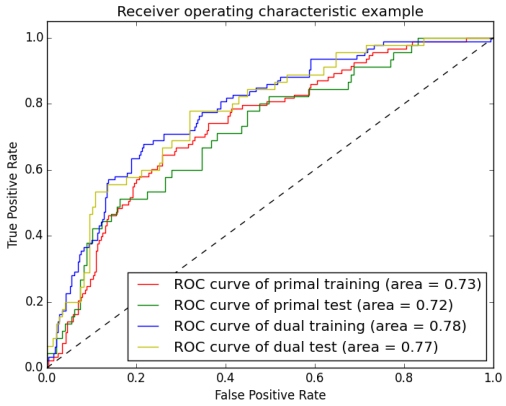
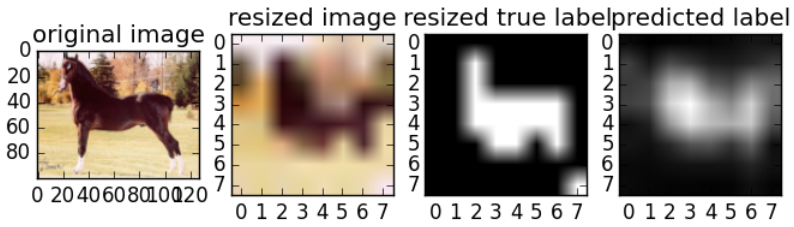


Figure 2: ROC curve

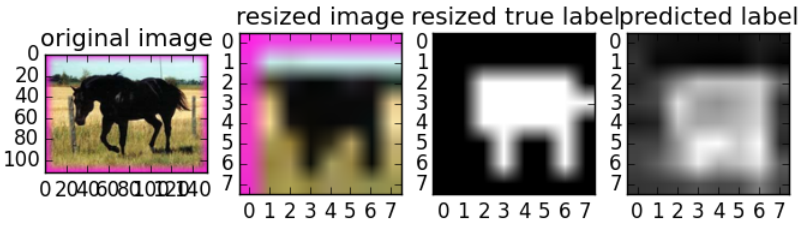
7 Future work

We will further modify this model mainly by following two approaches:

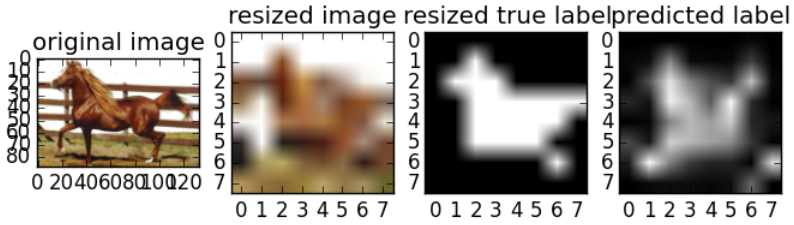
1. What we have done so far with the observed features are still basic and naive, we will further increase the features to higher dimension space. In Domke's paper [4], they expanding the



(a)



(b)



(c)

Figure 3: Predicted marginals for test images in horse dataset

Table 1: Evaluation of the horse dataset

	Primal		Dual	
	Train	Test	Train	Test
Error	0.248	0.229	0.239	0.229
Inconsistency	3.959	3.413	3.053	3.063

simple RGB feature to 64 features and append a 36-component Histogram of Gradients, resulting in 100 features. Edge features are also increased into a set of 42 features.

2. We will implement the Tree-Reweighted (TRW) inference in place of loopy belief propagation. TRW is a message passing algorithm that can use convex counting numbers and is related to the linear program relaxation of the MAP optimization problem. Then it is possible to guarantee that the algorithm continuously increases the dual objective, and hence it is convergent [5].

8 Conclusion

Training parameters of graphical model from real data involves many challenges. In this paper we talk about three of them. First of all, for the difficulty of doing exact inference, we discussed about approximate inference. Then for time complexity, we further apply truncated fitting into approximate inference. At last, as the major contribution of this paper, we optimize over a dual

objective which introduce penalty on disagreement between pairwise beliefs and unary beliefs to increase the consistency of the predictions. We implement our method on the problem of image segmentation and the results show that it performs better in both accuracy and consistency.

References

[1] J. Domke. Implicit differentiation by perturbation. In *Advances in Neural Information Processing Systems*, pages 523–531, 2010.

[2] J. Domke. Dual decomposition for marginal inference. In *AAAI*. Citeseer, 2011.

[3] J. Domke. Parameter learning with truncated message-passing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2937–2943. IEEE, 2011.

[4] J. Domke. Learning graphical model parameters with approximate marginal inference. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(10):2454–2467, 2013.

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

[6] D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: Effortless gradients in numpy.

[7] V. Stoyanov and J. Eisner. Minimum-risk training of approximate crf-based nlp systems. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–130. Association for Computational Linguistics, 2012.

[8] V. Stoyanov, A. Ropson, and J. Eisner. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *AISTATS*, pages 725–733, 2011.

[9] C. Sutton and A. McCallum. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011.

[10] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.