Learning Deep Inference Machines

J. Andrew Bagnell, Alex Grubb, Daniel Munoz, Stephane Ross
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
{ dbagnell, agrubb, dmunoz, sross } @cs.cmu.edu

Introduction. The traditional approach to structured prediction problems is to craft a graphical model structure, learn parameters for the model, and perform inference using an efficient– and usually approximate– inference approach, including, *e.g.*, graph cut methods, belief propagation, and variational methods. Unfortunately, while remarkably powerful methods for inference have been developed and substantial theoretical insight has been achieved especially for simple potentials, the combination of learning and approximate inference for graphical models is still poorly understood and limited in practice. computer vision, for instance, there is a common belief that more sophisticated representations and energy functions are necessary to achieve high performance which are difficult for theoretically sound inference/learning procedures.

An alternate view is to consider *approximate inference as procedure*: we can view an iterative procedure like belief propagation on a random field as a network of computational modules taking observations, other local computations on a graph (messages), and providing intermediate output messages and final output classifications over nodes in the random field. As a concrete example, belief propagation computes marginal distributions over variables by iteratively visiting all nodes in the graph structure and passing messages to neighbors which consist of "cavity marginals", *i.e.*, a sequence of marginals with the effect of each neighbor removed. To train such an algorithm, we consider training a general classifier using standard supervised learning techniques such that the output of the classifier corresponds to these marginals given the input variables and messages (*e.g.* by minimizing the logistic loss). In this sense, the classifier is trained to approximate the computations that occur during belief propagation. Note however that in our case, there is no graphical model of the data, i.e. there need not be any probabilistic model that corresponds to the computations performed by the classifier. The inference procedure is instead thought of as a black box function that is trained to yield correct predictions. We note that our approach builds directly on recent work, most notably [1, 2, 3], reducing structured classification to a series of simpler supervised learning problems. Most work here trains relatively short sequences of classifiers making simple, non-iterative decisions. Our approach apes the structure of approximate inference algorithms to benefit from the proven success of iterative methods for decoding the best structured output, but trains the inference network directly to maximize performance.

Training Inference Machines. If we consider a variational inference or belief propagation method and instantiate one computational module for each computation of node beliefs, we end up with a tremendously deep network: in our results below, for instance, we have a network with depth $O(10^5)$. Such networks are (generally) difficult to train using only gradient descent methods. Recent results have demonstrated the power of combining training source *local* to a module with a *global* objective function to coordinate behavior.

Building Inference. In many iterative inference procedures, there is a natural source of local information which is simply attempting to predict at each node the ideal classification/regression that we hope would result at the end of an inference procedure. For instance, in belief propagation where the computations are over node beliefs, we can simply attempt to target the ideal, single node supervised classification given

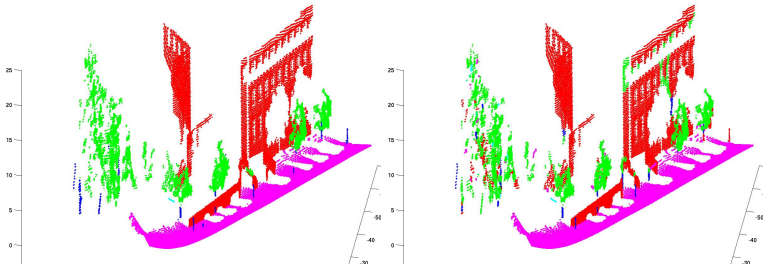

Figure 1: Ground truth and classification of a test scene given by building a deep loopy belief propagation machine.

previous computations. This target is generally unachievable early in the chain, or inference would be unnecessary, but it provides a valuable initialization.

For modest depth networks, we propose to use *stacking* [2], where modules are trained sequentially, each learning given the output of previous modules in the network. For larger depth networks, we use a new approach, Stochastic Mixing Iterative Learner (SMILe) [4], which can be seen as combing the stochastic mixing of SEARN [1] with the simple supervised training approach of stacking. In either case, our target training is always the local module-level ideal output.

Fine-tuning Inference. In many instances, the performance of an initial training can be significantly improved by fine-tuning the entire network to optimize performance on the final structured prediction at the end of the sequence of modules. A few methods are appropriate for this end-to-end optimization, notably Policy Search by Dynamic Programming [5], SEARN [1] and [6] back-propgation. We present results using "Boosted Back-propagation" [7] that optimizes the network over a space of functions instead of a space of parameters. The combination of functional gradient descent with the error-propagation mechanics of back-propagation [6] enables us to use arbitrary learning machines and training procedures though-out the network.

Our work can be seen to build on the recent work by [3] on training filters, a linear-chain inference algorithm, and we note that although they use different techniques, they adopt the same strategy of incremental building and refinement for training inference.

Results. Preliminary results on LADAR-point cloud classification 1 demonstrate results that compete with the state-of-the-art and often with dramatically lower computational effort. We additionally present results on image de-noising using a deep network that mimics a simple variational inference strategy.

References

[1] H. Daume, J. Langford, and D. Marcu. Search-based structured prediction. Machine Learning Journal, 2009.

[2] William W. Cohen and Vitor R. Carvalho. Stacked sequential learning. In *IJCAI'05*, 2005.

[3] John Langford, Ruslan Salakhutdinov, and Tong Zhang. Learning nonlinear dynamic models. In *ICML*, 2009.

[4] S. Ross and J. A. Bagnell. Reductions for imitation learning. Technical report, CMU, Robotics Institute, `http://robotwhisperer.org/preprints/`, 2010.

[5] J.A. Bagnell, S. Kakade, A. Ng, and J. Schneider. Policy search by dynamic programming. In *NIPS*, 2003.

[6] L. Bottou and P. Gallinari. A framework for the cooperation of learning algorithms. In *NIPS*, 1991.

[7] A. Grubb and J. A. Bagnell. Boosted backpropagation. Technical report, CMU, Robotics Institute, `http://robotwhisperer.org/preprints/`, 2010.

Topic: graphical models, Preference: Oral, Presenting Author: J. A. Bagnell