Supplementary Material for "Learning Ensembles of Potential Functions for Structured Prediction with Latent Variables"

Hossein Hajimirsadeghi and Greg Mori
School of Computing Science, Simon Fraser University, Canada
hosseinh@sfu.ca, mori@cs.sfu.ca

1. Computational Complexity of the Proposed Method

The computational complexity of each iteration of gradient boosting is divided into two parts: (1) computing point-wise pseudo-residuals and (2) training the base models. As discussed in Section 3.3, the former is obtained by inferring the CRF model for each data point and finding the marginal probabilities. We indicate the computational time of inferring the marginals of a CRF by T_{infer}. For example, for the tree/chain-structured CRF models of Section 4.1 and 4.2, $T_{infer} = O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ using belief propagation. For the cardinality model used in Section 4.3 $T_{infer} = O(m \log(m))$, where m is the number of instances in a bag. Finally, the total computational time of this part is obtained by summing over the whole data as $\sum_n T^n_{infer}$.

Next, the base models should be fitted to the point-wise pseudo-residuals. We assume a regression model can be trained to fit a set of training examples of size $|S|$ in $O(|S|\, d)$ time, where d is the size of the input feature vector. Given this assumption, each function approximation in the equations (17) and (18) takes $O(N\, d)$ and $O(\sum_n |\mathcal{E}^n|\, d)$ time, respectively. Finally, the computational time of fitting all functions would be $|\mathcal{Y}|\, O(N\, d) + |\mathcal{H}|\, O(\sum_n |\mathcal{E}^n|\, d)$.

We performed our experiments on an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz. As a numerical example, in the collective activity dataset, used in the experiments in Section 4.1.1 (which consists of 1908 training examples with average 5 local observations per example and the feature vectors are 240 dimensional), the training time was around 10 seconds per iteration. For the nursing home dataset of Section 4.1.2, which has 1910 training examples with average 2 local obsevations per example and 5-D feature vectors, the training time was around 2 seconds per iteration.

2. Guidelines for initialization of the potential functions

In the proposed HCRF-Boost algorithm, the potential functions may be initialized to zero at the first iteration. However, because of the nonconvexity of the likelihood optimization problem, a more smart initialization can improve the results (Note that the stochastic gradient ascent algorithm already helps to avoid some local optima). In our empirical studies we found that initializing the potentials with a model poorly trained by the standard HCRF algorithm [10] or with a global model trained by SVM can yield decent results in a few iterations (even 10 iterations). In fact, since each iteration of gradient boosting adds an entire model, a big step can be taken at each iteration [3]. In all experiments of Section 4.1 we used 50 iterations with $\beta = 0.1$.

3. Experiments on Popular Mulit-Instance Learning Datasets

The standard MIL benchmark datasets for computer vision are the *Elephant*, *Fox*, *Tiger* image categorization datasets [1]. Though dated, these are the standard benchmark on which MIL algorithms are evaluated. In the image data sets, each bag is an image, and the instances inside the bag represent 230-D feature vectors of different segmented blobs of the image. These data sets contain 100 positive and 100 negative bags. In all the experiments, we have preprocessed datasets by scaling the features of the original datasets to the range $[0, 1]$. We evaluate our proposed HCRF-Boost method with the cardinality model of Section 4.3 on these datasets. The results are reported based on 10-fold cross-validation classification accuracy and compared with the well-known MIL methods in Table 1.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568. MIT Press, 2002. 1, 2

Table 1. Comparison between HCRF-Boost and well-known MIL methods.

Method	Elephant	Fox	Tiger
HCRF-Boost	87	**66**	**86**
MIRealBoost [6]	83	63	73
ClassSetMaxRBM^{XOR} [9]	**88**	60	83
SVR-SVM [8]	85	63	80
MIForest [7]	84	64	82
MIGraph [13]	85	61	82
miGraph [13]	87	62	86
PPMM Kernel [11]	82	60	80
AW-SVM [5]	82	64	83
AL-SVM [5]	79	63	78
MILES [2]	81	62	80
MI-Kernel [4]	84	60	84
mi-SVM [1]	82	58	79
MI-SVM [1]	81	59	84
EM-DD [12]	78	56	72

[2] Y. Chen, J. Bi, and J. Wang. Miles: Multiple-instance learning via embedded instance selection. *T-PAMI*, 28(12):1931–1947, 2006. 2

[3] T. G. Dietterich, G. Hao, and A. Ashenfelter. Gradient tree boosting for training conditional random fields. *Journal of Machine Learning Research*, 9(10), 2008. 1

[4] T. Gärtner, P. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In *International Conference on Machine Learning (ICML)*, pages 179–186, 2002. 2

[5] P. Gehler and O. Chapelle. Deterministic annealing for multiple-instance learning. In *AISTATS*, 2007. 2

[6] H. Hajimirsadeghi and G. Mori. Multiple instance real boosting with aggregation functions. In *International Conference on Pattern Recognition (ICPR)*, 2012. 2

[7] C. Leistner, A. Saffari, and H. Bischof. Miforests: Multiple-instance learning with randomized trees. In *Computer Vision–ECCV 2010*, pages 29–42. Springer, 2010. 2

[8] F. Li and C. Sminchisescu. Convex multiple-instance learning by estimating likelihood ratio. *Advances in Neural Information Processing Systems (NIPS)*, pages 1360–1368, 2010. 2

[9] J. Louradour and H. Larochelle. Classification of sets using restricted boltzmann machines. In *Uncertainty in Artificial Intelligence (UAI-11)*, 2011. 2

[10] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence (T-PAMI)*, 29(10):1848–1852, 2007. 1

[11] H.-Y. Wang, Q. Yang, and H. Zha. Adaptive p-posterior mixture-model kernels for multiple instance learning. In *International Conference on Machine learning*, 2008. 2

[12] Q. Zhang and S. Goldman. Em-dd: An improved multiple-instance learning technique. *Advances in Neural Information Processing Systems (NIPS)*, 14:1073–1080, 2002. 2

[13] Z. Zhou, Y. Sun, and Y. Li. Multi-instance learning by treating instances as non-iid samples. In *International Conference on Machine Learning (ICML)*, 2009. 2