Multiscale Conditional Random Fields for Semi-supervised Labeling and Classification

David Duvenaud, Benjamin Marlin, Kevin Murphy
Laboratory for Computational Intelligence
Department of Computer Science
University of British Columbia, Canada
{*duvenaud,bmarlin,murphyk*}*@cs.ubc.ca*

***Abstract*—Motivated by the abundance of images labeled only by their captions, we construct tree-structured multi-scale conditional random fields capable of performing semi-supervised learning. We show that such caption-only data can in fact increase pixel-level accuracy at test time. In addition, we compare two kinds of tree: the standard one with pairwise potentials, and one based on noisy-or potentials, which better matches the semantics of the recursive partitioning used to create the tree.**

***Keywords*-conditional random fields, semi-supervised, multi-scale, segmentation, classification**

I. Introduction

A. Motivation

A central problem in training object localization models is that pixel-labeled images are rare and costly to produce. In contrast, data that have only weak labeling information, such as captions, are relatively abundant. We would like to construct a localization model which can incorporate such weakly labeled data into its training set. As we will show, this can be done by constructing a unified model of object presence at both the scale of the whole image as well as the scale of individual super-pixels.

B. Multi-scale Approaches

Recent work has shown that combining evidence from multiple scales is an effective strategy for image classification [1] [2] [3]. These approaches exploit the fact that label fields at different scales must agree with each other to some degree.

This observation motivates the following procedure for building a joint model across scales: First, segment an image at different scales. Next, estimate a class' presence or absence separately for each segment. Finally, combine these local estimates with a conditional random field (CRF).

C. Previous Approaches to Multiscale CRFs

The construction of multi-scale CRFs was previously demonstrated in [2] and [3]. In these experiments, each image was segmented independently at multiple scales. Potentials were trained to estimate the local evidence that each node contained the class of interest. Each node was then connected to a parent node from the next coarser level of segmentation, whose segment most overlapped the child segment. This procedure resulted in a tree-structured CRF which could combine local evidence from all nodes in a sensible way. Figure 1 shows an example of such a CRF.

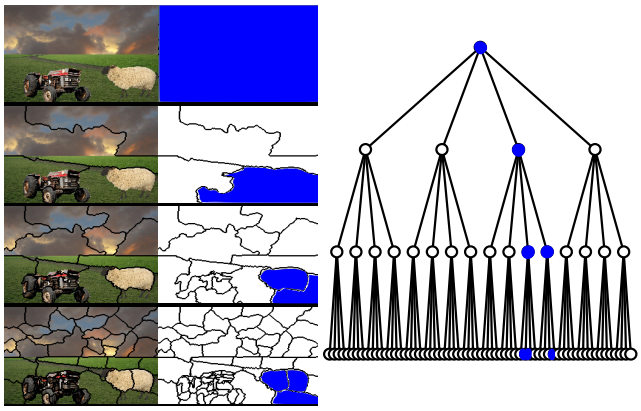

Figure 1. An example of a CRF defined on a recursively segmented image. Left: An example image, segmented at 4 levels of detail. Centre: The image segments, coloured to denote the presence of a sheep within them. Right: The tree-structured CRF corresponding to the image segments, where each segment is attached to all of its sub-segments.

Because this model defined a joint probability over labels at all scales, it could perform both image classification and object localization. In addition, it could effectively incorporate evidence from other image classifiers to perform better localization on unlabeled images. [2]

D. Contributions

In this work, we further develop the use of such tree-structured multiscale CRFs. However, in contrast to previous approaches, we construct the joint CRF before training, which allows evidence from partially labeled data to propagate through the tree, acting as a training signal at all scales. Thus, our model can perform semi-supervised learning, taking advantage of weakly labeled data. This is a significant advantage, since there is an abundance of weakly labeled image data on the web: specifically, images which only have

captions, (such as 'cat') but no pixel-level labels. We show that such caption-only data can in fact increase pixel-level accuracy at test time.

The other main contribution of this paper is the use of an exact, recursive image segmentation, in which each segment is composed entirely of smaller image segments. This induces an OR-structured correspondence between the labels of each segment and its subsegments. We examine the effect of introducing the noisy-or factor in section IV.

II. Semi-Supervised Learning in Pairwise CRFs

In this section, we give details of our segmentation algorithm, define the multiscale CRF with pairwise potentials, show how to compute the likelihood, and show how learning can be done via the Expectation-Maximization algorithm.

A. Exact Recursive Segmentation

In previous work [2] [3], segmentations at different levels of detail were constructed independently of one another. This approach is efficient, but has the disadvantage that the segments at one level of detail may not significantly overlap with the segments at the next coarsest level.

As opposed to segmentation images at each scale independently, we construct an exact recursive segmentation. We define a multi-scale segmentation to be recursive when each segment is contained in exactly one other segment at the next coarser level of detail.

To produce the exact recursive segmentation used by our model, we first segment the image at the finest spatial scale using a Quick-shift based super-pixel algorithm [4]. We then run a sparse affinity propagation clustering algorithm [5] to cluster adjacent image regions. The similarity between regions is simply the L_2 distance between their mean colors. We perform several rounds of clustering, corresponding to increasingly coarse levels of detail. We stop when there are 6 or fewer segments, which are merged in a final step.

In all experiments performed in this paper, we truncated the recursive segmentation at four levels of recursion, leaving approximately 30 segments per image at the finest level of detail. Figure 2 shows some examples of recursive segmentations using this algorithm.

B. Model Semantics

As in previous work, we construct the CRF by connecting nodes whose image regions have maximal overlap. However, in our case, each node is completely contained in exactly one other node by design. We denote the containing region to be a parent region, and the nodes contained within it to be the children of that region.

The image segments at all levels of detail can be denoted by S_r, where r denotes the segment number. The model contains one label node $Y_r^{(c)}$ for each class c and each element of the recursive image partition S_r. Setting $Y_r^{(c)} = 1$ is interpreted as meaning that the image region defined by segment S_r contains part of an object from class c, while setting $Y_r^{(c)} = 0$ is interpreted as meaning that the image region S_r does not contain part of an object from class c.

Figure 2. Example multi-scale segmentations from the VOC 2008 dataset. Rows one to four: Image segmentation at progressively finer levels of detail.

C. Local Evidence Potentials

The local evidence log-potential for each node $Y_r^{(c)}$ in this model depends linearly on the feature vectors x_r for the region S_r. We define the local evidence log-potential in Equation 2.1, where $\mathbf{w}_l^{(c)}$ are the feature-to-label weights for class c and segmentation level l.

$$\phi_f\left(y_r^{(c)}, x_r\right) = y_r^{(c)}(x_r^T \mathbf{w}_l^{(c)}) \tag{2.1}$$

Weights are shared across all nodes in a given level of detail l of the segmentation tree within each object class. In the experiments below, the weight vectors $\mathbf{w}_2^{(c)}$ and $\mathbf{w}_3^{(c)}$ were also constrained to be equal.

We make a separate copy of each CRF for each class. Each class's training and testing was performed separately, so from this point on, we drop the $^{(c)}$ superscript for notational simplicity.

D. Independent Model

As a baseline, we can consider an image model consisting solely of unconnected local evidence potentials. In this "independent" model, every region label is predicted separately,

and the model becomes equivalent to a per-region logistic regression on each region's image features. The parameters of the independent model only can be trained on fully-labeled data.

The likelihood of a node label assignment $\mathbf{y}$ in the independent model is as follows:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{N}\exp\left(\phi_f(y_i, x_i)\right) \tag{2.2}$$

where N is the number of nodes in the tree.

E. Pairwise Potentials

To construct a joint distribution over label nodes, we introduce pairwise potentials connecting neighbouring nodes. These pairwise potentials depend only on a 2x2 table of parameters θ, indexed by values taken by the nodes at each end of the potential.

$$\phi_{pair}\left(y_i, y_j\right) = \theta(y_i, y_j) \tag{2.3}$$

In our experiments, three sets of pairwise parameters were learned: One set for the potentials connecting global nodes to their children, another set for the potentials connecting the two middle layers, and a third set for the potentials connecting middle-layer nodes to bottom-layer nodes.

F. Likelihood

The likelihood of observing a particular configuration of label nodes $\mathbf{y}$ given feature vector x is defined as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\prod_{i}^{N}\exp\left(\phi_f(y_i, x_i) + \phi_{pair}(y_i, y_{par(i)})\right) \tag{2.4}$$

Here $par(i)$ denotes the index of parent of node i. As a special case, the root has no parent node, and $\phi_{pair} = 0$.

G. Likelihood Gradients

We now show how to compute the gradient of the log-likelihood in the presence of missing (unlabeled) data.

First, let $\mathbf{y}^{mis}$ denote the missing labels while $\mathbf{y}^{obs}$ denotes the observed labels. The marginal probability of observing $\mathbf{y}^{obs}$ can be obtained by summing out over all joint configurations of the missing labels $\mathbf{y}^{mis}$:

$$P(\mathbf{y}^{obs}|\mathbf{x}) = \sum_{\mathbf{y}^{mis}} P(\mathbf{y}^{obs}, \mathbf{y}^{mis}|\mathbf{x}) \tag{2.5}$$

We can then define the posterior probability of the missing labels given the observed labels:

$$P(\mathbf{y}^{mis}|\mathbf{y}^{obs}, \mathbf{x}) = \frac{P(\mathbf{y}^{mis}, \mathbf{y}^{obs}|\mathbf{x})}{P(\mathbf{y}^{obs}|\mathbf{x})} \tag{2.6}$$

We compute both of these quantities using belief propagation [6]. Then, the gradient for the weights with respect to the expected complete log-likelihood $\mathcal{L}$ is given by:

$$\frac{\partial E[\mathcal{L}]}{\partial \mathbf{w}_l^c} = \sum_{i \in Layer(l)} \left[P(y_i|\mathbf{y}^{obs}, \mathbf{x}) - P(y_i|\mathbf{x})\right] x_i \tag{2.7}$$

This gradient sums across training examples. In addition, an L_2 regularization penalty was placed on the image feature weights $\mathbf{w}$.

H. Learning

Learning was broken into three stages as follows:

1) The image feature weights $\mathbf{w}$, initialized to zero, were trained in the independent model by supervised training on fully labeled training images.
2) Pairwise factors ϕ_{pair} were added to the CRF, and the feature weights $\mathbf{w}$ along with the pairwise parameters θ were learned jointly by supervised training on the fully labeled training examples.
3) Caption-only data was added to the dataset, and the model was trained in a semi-supervised way using the E-M algorithm shown above.[1]

III. Experiments

A. Performance Metric

Performance was measured by the accuracy a as defined in the VOC 2008 challenge as

$$a = \frac{\text{tp}}{\text{tp} + \text{fp} + \text{fn}} \tag{3.8}$$

where tp, fp, and fn mean true positive, false positive, and false negative, respectively [2]. True positive is the number of foreground pixels correctly predicted. False positive is the number of background pixels incorrectly predicted. False negative is the number of foreground pixels incorrectly predicted. Here, "foreground" refers to the parts of the image containing the class of interest.

B. Pascal VOC Dataset

The data used for these experiments were gathered from the training and validation sets provided in the PASCAL Visual Object Classes(VOC) Challenge 2008 segmentation dataset [7]. This dataset contains 1023 fully labeled images, each approximately 500x500 pixels. Each pixel is either assigned to one of 20 classes, assigned to the "background" class, or labeled as "don't care". The predicted labels of pixels labeled "don't care" do not count towards the accuracy score.

[1]The function minimizer used in the M step was *minFunc* by Mark Schmidt, which implements the L-BFGS algorithm. This software is available at *http://www.cs.ubc.ca/ schmidtm/Software/minFunc.html*

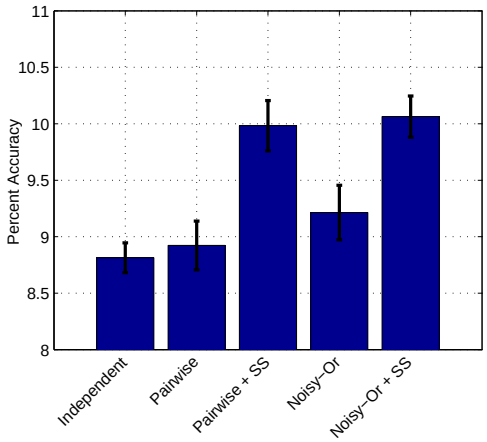

Figure 3. Pixel-level test accuracy across all models.

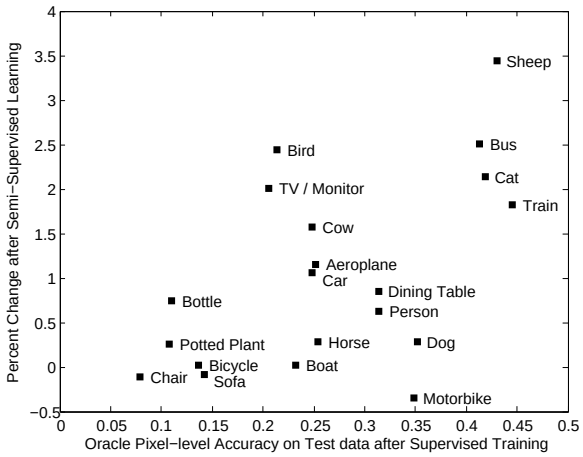

Figure 4. A plot of the change in test error versus the pixel-level accuracy on the test set after supervised training, when the global node was set to the true value.

C. Image Features

The image features used were: colour histograms (100 dimensions), histogram of oriented gradients (200 dimensions) [8], textons (500 dimensions) [9], and the 5x5 discretized location of the segment (25 dimensions). With a bias term added, each feature vector had 826 dimensions. However, the model is somewhat agnostic to the image features computed, and allows the use of different feature vectors at different levels in the segmentation hierarchy. For instance, one may want to use GIST [10] features at the highest spatial scale, as in [11]. However, exploratory experiments did not show a significant difference in performance when the top level features were replaced with a GIST vector.

D. Dataset Balancing

In these experiments, we balanced the dataset for each class separately by removing approximately 80% of images that did not contain the class of interest from the training sets.

E. Cross-validation

Error bars depicting one standard error were produced by conducting experiments on five training/test splits of the data. Within each split, the L_2 regularization parameter λ was chosen by nested cross-validation: Each training set was split into five inner training/validation splits. For both the supervised case and the semi-supervised case, the setting of λ that had the best average accuracy on the validation set was chosen to train the model on the whole training set for that fold.

Each outer fold had 400 fully-labeled training examples, 400 caption-only training examples, and 200 test examples.

F. Results

Figure 3 shows mean performance across all 21 classes in the VOC dataset, averaged over 5 folds. Error bars represent one standard error. The mean improvement in accuracy after semi-supervised training is statistically significant.

To better understand where the improvement from semi-supervised learning came from, we produced figure 4. This figure shows the change in pixel-level accuracy on each class after semi-supervised training, versus the pixel-level accuracy after supervised training when the top-level node was clamped to the true value. Only those classes which were already well-localized after supervised training were able to take advantage of the caption-only data.

G. Smoothing

In Figure 5, we compare the pixel level probabilities amongst the three models for an image drawn from the test set. In the image shown, the independent model finds several disparate patches which match the class of interest. In the tree models, evidence from the whole tree is combined, resulting in a smoother labeling at the pixel level.

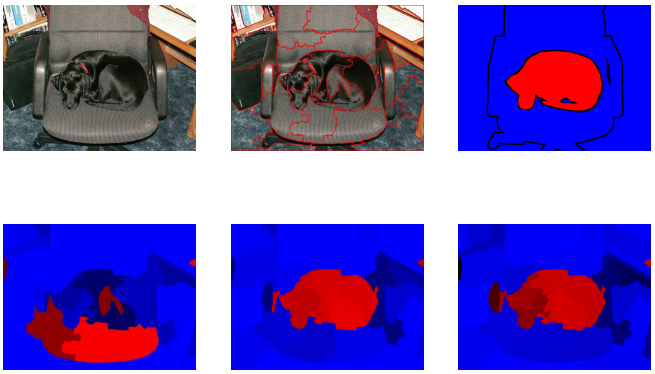

Figure 5. Detecting a dog. Top left: Original image. Top center: Segmentation at bottom level. Top right: True pixel labels. Bottom left: Pixel probabilities for independent model. Bottom center: Pixel probabilities for pairwise model. Bottom right: Pixel probabilities for noisy-or model.

H. Improving Performance

The performance of the models in these experiments is unimpressive relative to the state of the art. However, there are several reasons to expect that significantly better performance can be achieved, at the cost of slower training times[2].

To improve test accuracy, any of the following steps could be taken:

- The recursive segmentation can be made finer. In the experiments performed above, the recursive segmentation was only four levels deep, leaving relatively large segments at the finest scale. Since images can only be labeled per-segment, a coarse segmentation puts an upper bound on the pixel-level accuracy.
- The training set can be left unbalanced. In the experiments above, the training datasets were balanced by removing approximately 80% of images that did not contain the class of interest.
- The number of unlabeled examples can be increased relatively easily. To gather caption-only training data for the "dog" image model, for example, it suffices to merely find images that somewhere contain a dog, with no further labeling required. Note that these models can safely incorporate datasets having some incorrect labels, by giving images probabilistic labels.

In the following section, we investigate the possibility of improving performance by introducing a different factor joining the layers of the tree.

IV. Noisy-Or Tree Models

A. Motivation

In a multiscale CRF, how should we specify the joint probability of a group of child nodes and their common parent? When we segment an image recursively, then we will only observe a parent node to be if at least one child node is on. Thus, a factor joining parents and children should only put probability mass on states where either the parent is off, or where the parent and one or more children are also on.

Such factors have the same semantics as a logical OR-gate, whose probabilistic analogue is the noisy-or factor. As shown in [6], belief propgation messages for a noisy-or factor can be computed in time linear in the number of children in the factor. Thus noisy-or factor is appealing because it closely matches the semantics of the multiscale tree, while having the same time complexity as the pairwise model.

[2]In the experiments above, the tree models take approximately 2 hours to train per class on a 2GHz CPU, for a given setting of the hyperparameters. The main bottleneck in training the model is in performing inference at each step on each of the the partially-labeled examples. However, this inference step can be computed in parallel over all examples.

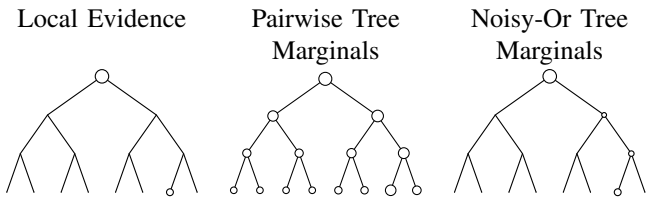

Figure 6. Left: Local evidence before belief propagation. Middle: Marginals after belief propagation in a pairwise tree. Right: Marginals after belief propagation in a noisy-or tree. Node size is proportional to marginal probability.

B. Definition

The noisy-or factor has the following semantics: The parent[3] node y_p turns on with probability θ independently for each child y_i that is turned on. Here i ranges from 1 to c, the number of children of y_p. Thus the noisy-or log-potential can be defined as:

$$\phi_{no}\left(y_p, y_{1\ldots c}\right) = \log\left(1 - \prod_{i=1}^{c}(1-\theta)^{y_i}\right)^{y_p} + \log\left(\prod_{i=1}^{c}(1-\theta)^{y_i}\right)^{(1-y_p)} \tag{4.9}$$

In a form that is easier to read, we can replace the success rate θ with the failure rate $q = 1 - \theta$:

$$\exp(\phi_{no}\left(y_p, y_{1\ldots c}\right)) = \left[1 - \prod_{i=1}^{c} q^{y_i}\right]^{y_p} \left[\prod_{i=1}^{c} q^{y_i}\right]^{(1-y_p)} \tag{4.10}$$

C. Multiple-Instance Learning

The noisy-or model is typically used in the multiple instance learning (MIL) setting [12]. In the MIL setting, the training set is constructed of subsets of training examples, where each subset is labeled as a positive example if any members of the subset are examples of the class of interest. The MIL setting is very similar to the semi-supervised object-detection task we are given here.

D. Evidence Flow in Trees

To illustrate the difference between pairwise trees and noisy-or trees, we show a synthetic example. Figure 6 contrasts evidence flows between the two models. Given strong evidence that a class is present somewhere in the image, and weak evidence that it is present at one location, the pairwise tree adjusts its probability strongly everywhere. The noisy-or tree only adjusts its probability significantly in the regions containing weak evidence.

[3]Here we are using "parent" and "child" to denote relative position in the image segmentation, not in the sense of a Directed Acyclic Graph.

E. Likelihood

The likelihood of the noisy-or tree model is similar to that of the pairwise tree model. Essentially, each set of pairwise potentials between a parent and all of its children is replaced by one noisy-or factor:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\prod_{i}^{N}\exp\left(\phi_f(y_i, x_i) + \phi_{no}(y_i, y_{children(i)})\right) \tag{4.11}$$

Where, as a special case, nodes at the bottom layer of the tree have no children, and $\phi_{no} = 0$.

F. Computing Expected Complete Likelihood

To compute the expected complete likelihood of the noisy-or factors conditioned on local evidence at each of the child nodes, we could simply use the junction-tree algorithm. However, this method would require a sum over all possible states of each group of children. This sum is exponential in the number of children and may be prohibitively slow. Fortunately, the expected likelihood can be calculated in linear time.

To see that this is the case, consider computing the expected likelihood of a family of nodes $y_p, y_1 ... y_c$, each with local evidence $P(y_i|e_i)$ representing the contribution from the local potentials $\phi_f(y_i, x_i)$. Note that when the parent node is off ($y_p = 0$), the sum over all child nodes has a factorized form:

$$\sum_{y_{1\ldots c}} P(y_p = 0|y_{1\ldots c})P(y_{1\ldots c}|e) = \sum_{y_{1\ldots c}}\prod_{i=1}^{c} q^{y_i}P(y_i|e_i) \tag{4.12}$$

Bringing sums inside of products, we obtain the efficient form:

$$\sum_{y_{1\ldots c}} P(y_p = 0|y_{1\ldots c})P(y_{1\ldots c}|e) = \prod_{i=1}^{c}\sum_{y_i} q^{y_i}P(y_i|e_i) \tag{4.13}$$

Which skips the exponential sum. Thus we can compute $P(y_p = 0|y_{1\ldots c})$ efficiently. We can also trivially compute $P(y_p = 1|y_{1\ldots c}) = 1 - P(y_p = 0|y_{1\ldots c})$. The normalization constant $P(e)$ can be computed in the same manner. Thus we can compute every quantity needed efficiently.

G. Training

The gradients for the image feature weights W are identical to those of the pairwise tree model once the node marginals have been computed, and can be estimated with the same E-M algorithm as the pairwise trees.

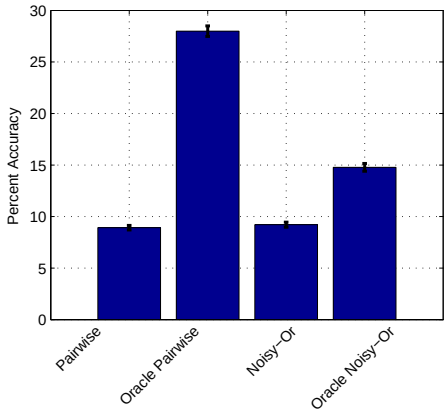

Figure 7. Pixel-level test accuracy in the pairwise and noisy-or models, compared to the case where the global-level nodes were clamped to their true value.

H. Learning the Noisy-Or Failure Rate Parameter

In a fully observed, recursively segmented image, the maximum likelihood estimate for the failure rate parameter q will always be zero, since a parent node will be observed to be on if and only if a child is on. However, on partially observed data, this is not necessarily the case.

In initial experiments, the parameter q was learned in parallel with the feature weights W, but as the model converged, the learned q parameter again tended towards zero. For the experiments below, this parameter was fixed to 0.01.

I. Performance

In figure 3, we can see that while the noisy-or model offers a slight advantage over the pairwise model in the fully supervised setting, it has the same performance as the pairwise model in the semi-supervised setting.

To further shed light on the results in section III, we conducted a number of experiments investigating the flow of evidence within CRFs for the two models.

J. Introducing an Oracle

Following [2], we examined the effect of introducing a perfect oracle at the root node of the CRF. This let us examine how well the models localize given the correct object classification, and allowed us to compute an upper bound on the possible performance boost in pixel-level accuracy attainable by incorporating evidence from a better global-level classifier. Figure 7 shows the results. We see that both models obtain a large increase in pixel-level accuracy when combined with an oracle. This result is consistent with results in [2].

We also observe that the pairwise model receives a much greater boost in accuracy from the oracle than the noisy-or model. This result, combined with the example in Figure 6, suggests that the noisy-or model might be inappropriately

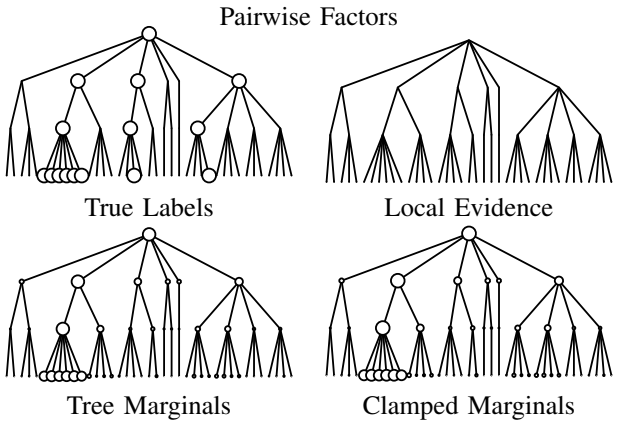

Figure 8. An example of belief propagation and evidence flow in the pairwise model, trained on real data. Node size is proportional to probability.

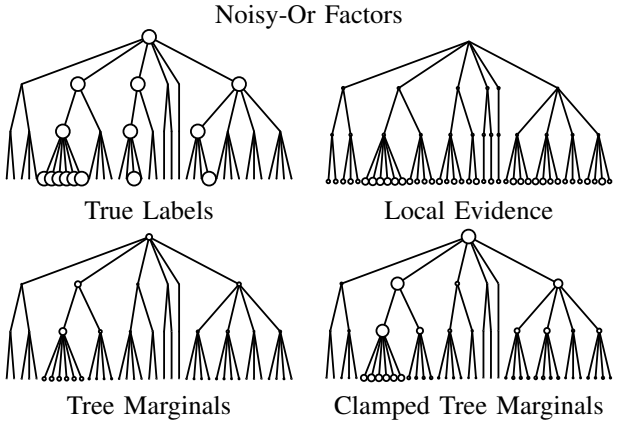

Figure 9. An example of belief propagation and evidence flow in a noisy-or tree, trained on real data. Node size is proportional to probability.

"explaining away" the oracle's being on by increasing the marginal probability of only a small number of bottom-level nodes.

K. Real Examples

To examine this phenomenon in more detail, we plot the evidence flow on a CRF defined on an image drawn from the test set. Figures 8 and 9 show the behavior of the two models on a real example.

The most illuminating feature of these figures is the difference in the marginals before and after the global node has been clamped to the true value. When the global node is clamped to the true value, we can see this evidence flowing down to the leaves in both models. As in Figure 6, we observe that in the pairwise model, evidence tends to flow down all branches to some degree. In the noisy-or model, we observe that evidence tends to flow down only one branch of the tree, and to a smaller degree than in the pairwise model.

V. Conclusion and Future Work

A. Large-Scale Experiments

Of the 400 caption-only images used here for semi-supervised learning, on most classes only 20-50 images in that set actually contained the class of interest. Given the improvement in performance observed after adding only this small number of examples to the test set, it seems worth noting that a large weakly-labeled dataset could easily be constructed for a small number of classes, to evaluate the effectiveness of yet adding more caption-only data.

B. Bounding Box Data

In the recursive tree models, we can effectively incorporate bounding box information by setting a node containing the entire bounding box to be 'on'. One next logical step would be to incorporate the large amounts of bounding-box labeled data available into the training set of multiscale CRFs.

C. Concluding Remarks

Our central motivation for using multi-scale CRFs was their ability to learn from weakly labeled data. This ability was clearly demonstrated: during semi-supervised training, we were able to observe evidence flowing from the labeled global-level nodes to the unlabeled pixel-level nodes, and at test time, we observed an increase in pixel-level accuracy. As large, caption-only datasets such as ImageNet [13] continue to grow, this ability will only become more useful.

References

[1] K. Murphy, A. Torralba, and W. Freeman, "Using the forest to see the trees: a graphical model relating features, objects and scenes," *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[2] N. Plath, M. Toussaint, and S. Nakajima, "Multi-class image segmentation using conditional random fields and global classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 817–824.

[3] J. Reynolds and K. Murphy, "Figure-ground segmentation using a hierarchical conditional random field," in *Fourth Canadian Conference on Computer and Robot Vision, 2007. CRV'07*, 2007, pp. 175–182.

[4] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[5] B. Frey and D. Dueck, "Mixture modeling by affinity propagation," *Advances in neural information processing systems*, vol. 18, p. 379, 2006.

[6] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results," http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[8] N. Dalai, B. Triggs, R. I. Alps, and F. Montbonnot, "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005.

[9] P. Wu, B. Manjunanth, S. Newsam, and H. Shin, "A texture descriptor for image retrieval and browsing," in *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1999, pp. 3–7.

[10] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[11] K. Murphy, A. Torralba, D. Eaton, and W. Freeman, "Object detection and localization using local and global features," *Toward Category-Level Object Recognition*, pp. 382–400, 2006.

[12] T. G. Dietterich, R. H. Lathrop, T. Lozano-Perez, and A. Pharmaceutical, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[14] H. Harzallah, F. Jurie, and C. Schmid, "Combining efficient object localization and image classification," 2009.

[15] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," *Computer Vision–ECCV 2008*, pp. 705–718, 2008.

[16] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *ECML '07: Proceedings of the 18th European conference on Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 286–297.

[17] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, 2009.

[18] J. Lafferty, A. Mccallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*, 2001, pp. 282–289.

[19] J. Verbeek and B. Triggs, "Scene segmentation with crfs learned from partially labeled images," in *Advances in Neural Information Processing Systems*, 2007.

[20] M. Schmidt, K. Murphy, G. Fung, and R. Rosales, "Structure learning in random fields for heart motion abnormality detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[21] J. Zhu, E. P. Xing, and B. Zhang, "Partially observed maximum entropy discrimination markov networks," in *Advances in Neural Information Processing Systems*, 2008.

[22] X. He and R. S. Zemel, "Learning hybrid models for image annotation with partially labeled data," in *Advances in Neural Information Processing Systems*, 2008.

[23] T. S. Jaakkola and M. I. Jordan, "Variational probabilistic inference and the qmr-dt network," *Journal of Artificial Intelligence Research*, vol. 10, pp. 291–322, 1999.

[24] L. Du, L. Ren, D. Dunson, and L. Carin, "A bayesian model for simultaneous image clustering, annotation and object segmentation," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 486–494.

[25] P. Schnitzspan, M. Fritz, S. Roth, and B. Schiele, "Discriminative structure learning of hierarchical representations for object detection," in *CVPR*, 2009, pp. 2238–2245.

[26] J. Shi and J. Malik, *Normalized Cuts and Image Segmentation*, 2000.

[27] M. Szummer, P. Kohli, and D. Hoiem, "Learning crfs using graph cuts," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 5303. Springer, 2008, pp. 582–595.

[28] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 1124–1131.

[29] L. Liao, D. Fox, and H. A. Kautz, "Hierarchical conditional random fields for gps-based activity recognition," in *ISRR*, 2005, pp. 487–506.

[30] X. H. Richard, R. S. Zemel, and M. A. C. perpi nán, "Multiscale conditional random fields for image labeling," in *CVPR*, 2004, pp. 695–702.

[31] X. Feng, C. K. I. Williams, and S. N. Felderhof, "Combining belief networks and neural networks for scene segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 467–483, 2002.

[32] A. J. Storkey and C. K. I. Williams, "Image modelling with position-encoding dynamic trees," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 25, pp. 859–871, 2003.

[33] D. G. Lowe, "Object recognition from local scale-invariant features," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150–1157.

[34] A. Vezhnevets and J. M. Buhmann, "Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning," in *CVPR*, 2010, pp. 3249–3256.

[35] M. B. Blaschko, , A. Vedaldi, and A. Zisserman, "Simultaneous object detection and ranking with weak supervision," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2010.