Learning What and How of Contextual Models for Scene Labeling

Arpit Jain[1], Abhinav Gupta[2], and Larry S. Davis[1]

[1]University of Maryland College Park, MD, 20742
[2]Carnegie Mellon University, Pittsburgh, PA, 15213

Abstract. We present a data-driven approach to predict the importance of edges and construct a Markov network for image analysis based on statistical models of global and local image features. We also address the coupled problem of predicting the feature weights associated with each edge of a Markov network for evaluation of context. Experimental results indicate that this scene dependent structure construction model eliminates spurious edges and improves performance over fully-connected and neighborhood connected Markov network.

1 Introduction

Image understanding is one of the central problems in computer vision. Recently, there has been significant improvements in the accuracy of image understanding due to a shift from recognizing objects "in isolation" to context based recognition systems. Such systems improve recognition rates by augmenting appearance based models of individual objects with contextual information based on pairwise relationships between objects. These relationships can be co-occurrence relationships or fine-grained spatial relationships. However, most approaches have employed brute force approaches to apply context - all objects are first (probabilistically) detected and connected in a massive network to which probabilistic inference methods are applied. First, this approach is clearly not scalable; but more important, it suffers from the serious drawback that it treats all pair-wise relationships in an image as equally important for image analysis. For example, consider the image shown in Figure 1, where our goal is to identify the unknown label of the region outlined in red (which we will refer to as the target), given the labels of other regions in the image. The regions labeled as building tend to force the label of the target towards building (two building regions co-occur more often then building and car) and the region labeled car tends to force the label of the target to be road, since car above road has higher probability in the contextual model than car above building. In the case of fully-connected models, the edges from the building regions to the target region outnumber the edges from other regions to the target and therefore the target is incorrectly labeled as building. If we had only utilized the relationship from the region associated with the car and ignored the relationships from other objects to predict the label of the target, then we would have labeled the target correctly.

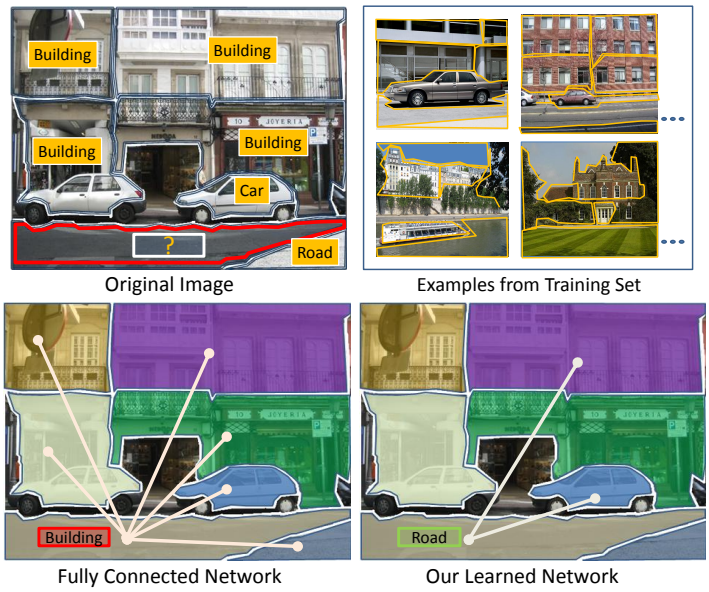

Fig. 1. An example from our dataset showing that all relations are not informative in fully a connected network and can lead to wrong labeling and how our proposed method learns "what" edges are important and removes dubious information

Other approaches for applying context to image understanding have considered fixed structure Markov networks where only nodes corresponding to neighboring segments are linked by an edge [2]. Such approaches are based on the reasonable assumption that neighboring segments carry significant contextual information and might be sufficient for recognition. However, such neighborhood based connectivity schemes have two shortcomings: (1) Images are two-dimensional projection of the three-dimensional world. Two objects which are far in the 3D world might appear very close in the image plane and therefore some noisy relationships are included in the network. (2) The assumption that neighboring segments provide sufficient contextual information is too strong and does not hold in many cases. For example, in a sunset scene, the relationship between the appearance of the sky (orange) and the appearance of large bodies of water is useful for recognition of such bodies of water, even though there can be intervening land regions between them.

In this paper, we evaluate the importance of individual contextual-constraints and use a data-driven model for selection of **what** contextual constraints should be employed for solving a specific scene understanding problem, and for constructing a corresponding Markov-network. Unlike previous approaches that use fully connected or fixed structures based on neighborhood relationships, our approach predicts the structure of the Markov network (i.e., selects edges). Selection of edges is generally dependent on a combination of global and local factors

such as discriminativeness of regions. However, identifying the variables/factors associated with predicting the importance of a contextual edge a priori is difficult. Instead, we take a data driven approach to predict the importance of an edge, in which scenes similar to a "test" image are identified in the training dataset and utilized to predict which regions should be linked by an edge in the Markov network corresponding to the test image - referred to as edge prediction. Figure 1(a) shows an example of edge prediction from our test dataset. Our approach appropriately eliminates the edges from most of the building regions to the target and maintains the edge from the car. This leads to a correct labeling of the target.

To learn a data-driven(non-parametric) model of edge importance, we have to compute the importance of edges in the training data-set itself. This requires evaluating each edge in the training data-set with respect to other edges in the training data-set. Edges that represent consistent spatial-relationships between pairs-of-nouns are retained as informative edges and the rest are dropped. If a single 2D-spatial relationship was sufficient to represent constraints between a pair of nouns, then extracting consistent edges would be straight-forward. However, relationships between pairs of nouns are themselves scene-dependent (due to viewpoint, functional-context, etc.). For example, based on viewpoint, a road might be either below a car or around a car (see Figure 1(b)). Similarly, relationships are also based on function-context of an object. For example, a bottle can either be on the table or below the table based on its function (drinking vs. trash). Therefore, we cluster the relationships between pairs of nouns based on scene properties. For each cluster, we then learn feature-weights which reflect how much each feature of the vector of variables capturing spatial relationships is important for evaluating constraint/relationship satisfaction. For example, in a top-down view, road being "around" car is most important. Our approach not only learns the construction model for Markov networks, but also learns the feature weights which define how to evaluate the degree to which a relationship between a pair of nouns is satisfied. Again, instead of explicitly modeling the factors on which these feature weights depend, we utilize a data driven approach to produce pseudo-clusters of images and estimate **how** each contextual edge should be evaluated (See Figure 1(b)) in each cluster.

The contributions of our paper are: (1) A data driven approach for predicting **what** contextual constraints are important for labeling a specific image that uses only a subset of the relationships used in a fully-connected model. The resulting labeling are both more accurate and computed more efficiently compared to the fully-connected model. (2) A model for predicting **how** each contextual edge should be evaluated. Unlike previous approaches, which utilize a single spatial-relationship between a pair of objects (car above road), we learn a scene dependent model of context and can encode complex relationships (car above road from a standing person's viewpoint, but road around car from a top-down viewpoint).

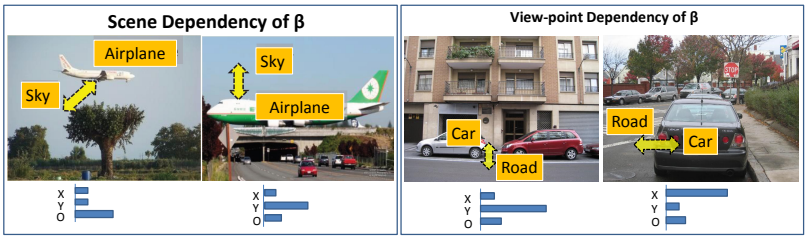

Fig. 2. The figure shows examples of how feature weights are a function of both local and global factors. Here we show how feature weights depend on function context and viewpoint. Pairwise features X,Y,O refer to differences in x-coordinates,difference in y-coordinates and overlap between two regions respectively

2 Related Work

Recent research has shown the importance of context in many image and video understanding tasks [3, 1, 4–6]. Some of these tasks include segmentation and recognition. For object recognition, researchers have investigated various sources of context, including context from the scene [7], objects [4] and actions [22]. Scene based context harnesses global scene classification such as urban, landscape, kitchen etc to constrain the objects that can occur in the scene (for example, a car cannot occur in a kitchen). On the other hand, object based contextual approaches model object-object co-occurrence and spatial relationships to constrain the recognition problem (for example, car above road). Recent research suggests that the object-object based contextual models outperform the scene based contextual models [8]. Our work builds upon this and tries to improve how object-object relationships should be utilized on selected pairs of regions instead of all region pairs.

In most previous work, relationships are represented by graphical models such as belief networks [22] or CRFs [4], and the parameters of the graphical models are learned using graph cuts [23] or max-margin method [24]. One of the common problems with such approaches is determining "what" edges in the graphical model should be used for inference. While fully-connected networks provide the largest number of constraints, they are hard to evaluate and also include weak edges which can sometimes lead to higher belief entropy. Fixed structure approaches, such as neighborhood based MRF's[2], are computationally less demanding but ignore vital long range constraints. Other approaches such as [9] perform approximate inference by selecting fewer edges based on object co-occurrences and discriminability. There has been some work on learning the structure of a graphical model from the training dataset itself [10]. Here, the edges are learned/inserted based on the consistency of relationships throughout the dataset. However, most of the contextual relationships are scene based and might not hold true for all scenarios. In such situations, structure-learning approaches tend to drop the informative edges, since they are not consistent throughout. Instead, we predict the relevant contextual relationships based on

the scene being analyzed. In our approach, instead of learning a fixed structure from the training dataset, we learn the space of allowable structures and then predict a structure for a test image based on its global scene features and local features.

Our work is similar in spirit to "cautious" collective inference [11, 12]. Here, instead of using all relationships, the relationships which connect discriminative regions are used for initial iterations and the number of relationships used are increased with each iteration. However, the confidence in the classification of a region is itself a subtle problem and might be scene-dependent. Instead, we learn a decision model for dropping the edges/relationships based on global scene parameters and local parameters. Our work is also related to the feature/kernel weighting problem [13]. However, instead of learning weights of features/kernel for recognition problems, we select features for a constraint satisfaction problem. Therefore, the feature weights are on pairwise features and indicate "how" the edge in a Markov network should be evaluated. This is similar to [1] in which the prior on possible relationships between pairs of nouns is learned, where each relationship is based on one pair-wise feature. However, this approach keeps the priors/weights fixed for a given pair of nouns whereas in our case we learn a scene-dependent weight function.

3 Overview

Given a set of training images with ground truth labeling of segments, our goal is to learn a model which predicts the importance of an edge in a Markov network given the global features of the image and local features of the regions connected by that edge. We also want to learn a model of image and class-specific pairwise feature weights to evaluate contextual edges. Instead of modeling the latent factors and using a parametric approach for computing edge importance and feature-weights, we use a data-driven non-parametric approach to model these. Learning a non-parametric model of edge-importance would require computing edge importance in the ensemble of Markov networks of the set of training images. Edge importance, however, itself-depends upon feature weights; feature weights determine if contextual constraints are satisfied or not. On the other hand, the feature weights, themselves, depend on the structure of the Markov networks in the training dataset, since only the images for which nouns are (finally) linked by an edge should be evaluated to compute the feature weights. We propose an iterative approach to these linked problems. We fix the feature weights to estimate the current edge-importance function, followed by fixing the edge-importance function to re-evaluate feature weights.

Learning. Figure 3 shows an overview of our iterative learning algorithm. Assume that at some iteration, we have some contextual edges in the training data-set and feature weights associated with each contextual edge. For example, in figure 3, out of the six occurrences of road and car, we have contextual edges in five cases with their corresponding weights. Based on the current feature weights, we first estimate how likely each edge satisfies the contextual relationship and its

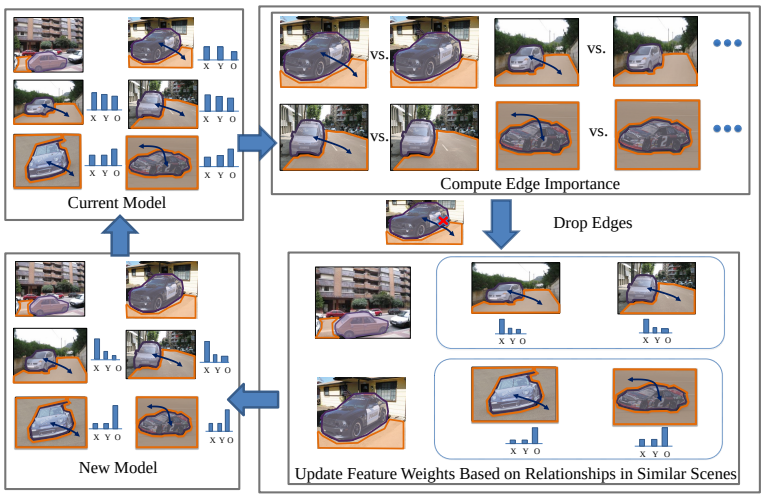

Fig. 3. Overview of our approach: we propose an iterative approach of **what** constitutes the space of important edges and **how** these edges can be evaluated. Our approach simultaneously learn the construction model $\mathcal{F}_e()$ and differential feature weights β

importance in identifying the labels of the regions. To compute the importance, we compare labeling performance with and without the edge in the Markov network. For example, in the first case the relative locations of the car and road are not coherent with other similar examples in the training dataset (the road is neither around/overlapping the car nor is it to the right of the car as in the other cases). Therefore, in this case the edge linking the car and road is not informative and the Markov network without the edge outperforms the Markov network with the edge.

After computing the importance of each edge, a few non-informative edges are eliminated. At this stage, we fix our edge importance function and utilize it to estimate the new pair-wise feature weights. For computing the new feature weights, we retrieve similar examples from the training dataset and analyze which pair-wise features are consistent throughout the set of retrieved samples. The weights of the consistent features are increased accordingly. In the example, we can see that for the images with a top-down viewpoint, the overlap feature becomes important since in the retrieved samples the road region was generally overlapping the car region. Once the feature weights are updated, we obtain a new non-parametric model of both edge-importance and feature weights. This new model is then used to evaluate the edge importance and drop further edges and recompute feature weights.

Inference. The inference procedure is illustrated in Figure 4. An image is first segmented into regions. For segmentation, we use the SWA algorithm [14] and stability analysis for estimating the stable segmentation level [15]. We then predict the importance of each edge based on global features and local features

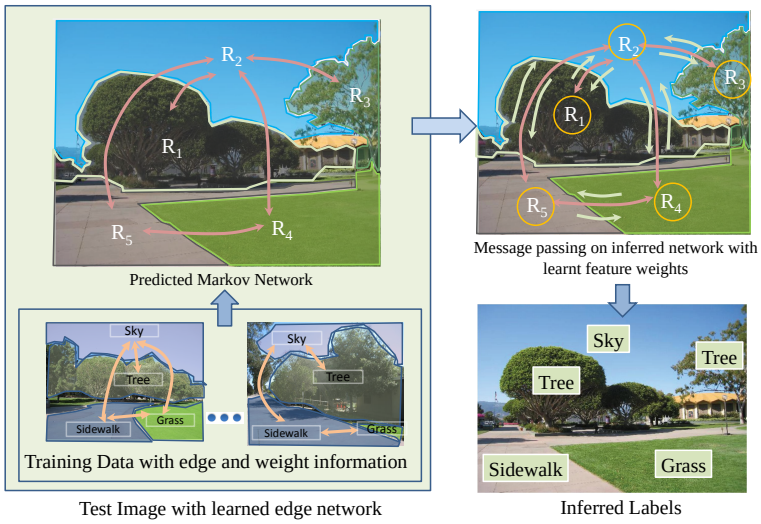

Fig. 4. Inference algorithm for our approach: Using the global and local features computed from the segmentation, we first predict the structure of the Markov network by matching possible edges to edges in the training set (locally weighted regression). We also estimate the feature weights β for each edge in the Markov network. Finally we use message passing to predict the labels

of the regions connected by the edge. Based on the importance of edges, we construct a Markov network for inference. For each edge, we also compute feature weights that should be utilized to evaluate context on that edge. The labels are then predicted using the message passing algorithm over the constructed Markov network with the estimated feature weights.

4 Mathematical Formulation

We now more formally describe our approach to learn "what" edges constitute the space of efficient networks and "how" to evaluate these edges in those networks. Our motivation is that not all edges in the complete Markov network are informative. So, we want to include in our Markov network only those edges which are generally informative, given the image, and also predict the corresponding feature weights which describe how to evaluate the constraints specified by the selected edges. Formally, our goal is to learn two functions from training data: $\mathcal{F}_e(G^t, R_i^t, R_j^t)$ and $\beta(G^t, n_i^t, n_j^t)$; where $\mathcal{F}_e()$ evaluates whether there should be an edge between regions i and j of image t and $\beta()$ represents the vector of pair-wise feature weights. The function $\mathcal{F}_e()$ depends on the global scene features G^t and the local features of region i and j represented by R_i^t and R_j^t. On the other hand, the feature weights depend on the global scene features and the pair of noun classes for which the pair-wise features are being evaluated.

The functions are learned based on a cost function, which minimizes the cost of labeling in the training dataset. The task is to predict all the nouns in an image, and therefore our cost function can be formulated as follows: Suppose our vocabulary consists of m object-classes, and let y_i^t be an m-dimensional vector which represents the ground-truth annotation for region i in training image t. Function $f_A(R_i)$ evaluates the appearance of the region by comparing it with the appearance models of the m labels and returns an m-dimensional vector with appearance distance scores. The cost function is then formulated as:

$$\mathcal{C} = \sum_t (\sum_i y_i^t f_A(R_i) + \sum_{(j,k)\in\Lambda^t} y_{jk}^t F_C(G^t, R_{jk})) \tag{1}$$

In this cost function, y_{jk}^t is a m^2 dimensional vector which represents pair-wise ground truth annotations and F_C is a m^2 dimensional-vector representing how well the pair-wise features R_{jk} match the contextual parameters for all m^2 pairs of labels. Λ^t represents the set of chosen edges for an image t based on function $\mathcal{F}_e$ and, therefore, we define Λ^t as: $\{(j,k) : \mathcal{F}_e(G^t, R_i^t, R_j^t) > \alpha\}$.

Contextual evaluation also requires feature-weighting, since all features are not equally important for contextual relationship evaluation. For example, while a difference in y-coordinate is important in evaluation of the contextual relationship between sky and water, the differences in x-coordinate is irrelevant. As discussed previously, these feature weights depend not only on the pair of nouns but also the global features of the scene. Therefore, if the function $f_{n_j,n_k}(G^t, R_{jk})$ represents the $(n_j, n_k)^t h$ element of F_C, we can write it as:

$$f_{n_j,n_k}(G^t, R_{jk}) = \sum_{l=1}^{L} \beta_{n_j,n_k}^l(G^t) C_{n_j,n_k}^l(G^t, R_{jk}^l) \tag{2}$$

where β^l represents the weight of the l^{th} pair-wise feature and is dependent on global scene features and the pair of nouns, and C^l is the context model which measures how well the l^{th} dimension of a pairwise feature R_{jk} satisfies the constraint learned for that dimension for the given pair of nouns.

Intuitively, equation 1 states that the cost can be minimized if: (1) We sum over the contextual constraints that have low cost, that is, Λ^t should only include informative edges. (2) the learned feature weights should be such that the dimensions which represent consistent relationships should have higher weight as compared to the other dimensions. Our goal is to minimize equation (1) with respect to all possible graphs in all training images and all possible weights. At that minima, we have a subset of edges for all the images in the training data-set and feature-weights at each edge. We then learn a non-parametric representation of $\mathcal{F}_e$ and β based on the importance and weights estimated for the edges in the training dataset. As we can see, the estimation of β in training images depends on edges that are important in the training images and the evaluation of the importance of edges depends on β. Therefore, we employ an iterative approach where we fix β and learn the function $\mathcal{F}_e$ and in the next step, based on the importance of edges in the training dataset, we re-estimate β.

4.1 Iterative Approach

Learning $\mathcal{F}_e$: Given feature-weights β, we predict whether an edge is informative or not. The information carried by an edge (representing potential contextual constraints on the pair-wise assignment of nouns to the nodes at the two ends of the edge) is a measure of how important that generic edge type is for inferring the labels associated with the nodes connected by the edge. The information carried in an edge depends on both global and local factors such as viewpoint and discriminability. Instead, of discovering all the factors and then learning a parametric-function; we use a non-parametric representation of the importance function. However, we still need to compute the importance of each edge in the training data-set.

To compute the importance of an edge, we use the message-passing algorithm. The importance of an edge is defined as how much the message passing through the edge helps in bringing the belief of nodes connected by the edge towards their goal belief (ground-truth). Suppose that the goal beliefs at node i and j are y_i and y_j respectively. The importance of the edge between i to j is defined as:

$$I(i \leftrightarrow j) = \frac{1}{iter} \sum_{k=1}^{iter} (y_i.b^k_{\mathcal{N}_i} - y_i.b^k_{\mathcal{N}_{i-(i,j)}}) + (y_j.b^k_{\mathcal{N}_j} - y_j.b^k_{\mathcal{N}_{j-(i,j)}}) \quad (3)$$

where $b^k_{\mathcal{N}_i}$ is the belief at node i at iteration k computed using messages from all the nodes (fully-connected setting); $b^k_{\mathcal{N}_{i-(i,j)}}$ is the belief at node i at iteration k computed using messages from all the nodes except $i \leftrightarrow j$ (edge-dropped setting). $iter$ is the total number of iterations of message passing algorithm.

Using this approach, the importance of each edge is computed based on the local message passing algorithm. It does not take into account the behavior of other similar edges (similar global scene features and connecting similar local regions) in the training dataset. For example, in a particular image from the set of beach scenes, the edge between sky and water might not be important; however if it is important in most other beach images, we want to increase the importance of that particular edge so that it is not dropped. We therefore update the importance of an edge by using the importance of the edges which have similar global and local features. This is followed by an edge dropping step, where the edges with low importance are dropped randomly to compute an efficient and accurate networks for the training images.

Learning β: Given the importance function of the edges $\mathcal{F}_e()$, we estimate β. As stated above, we use locally weighted regression for estimating β, therefore we need to estimate individual feature weights for all edges. Given the cost function in equation 1, a gradient descent approach is employed to estimate the feature weights of edges. We obtain the gradient as:

$$\frac{\partial \mathcal{C}}{\partial \beta^l_{n_j,n_k}} = C^l(G^t, R_{jk}) \quad (4)$$

where $\beta^l_{n_j,n_k}$ is the weight of l^{th} feature for edge (j,k). The above equation states that for a given pair of nouns, if the l^{th} dimension of pairwise feature is consistent with the l^{th} dimension of pairwise features from similar images, then the value of $\beta^l_{n_j,n_k}$ should be increased. Therefore, the value of β is updated at each step using the gradient above and then normalized ($\sum_l \beta^l = 1$). Intuitively, this equation evaluates which contextual relationship is satisfied on average for a pair of nouns and increases it weight. For example, between sky and water the above relationship is always satisfied where as left/right has high variance (In images sky is sometimes on left and sometimes on right of water). Therefore, this equation increases the weight of dY (measuring above) and decreases the weight of dX (measuring left).

4.2 Inference

Given a segmentation of an image, we construct a Markov network for the image using the function $\mathcal{F}_e$. For this construction, we first compute the global features, G, of the image and the local features of every region in the segmentation. A potential edge in the network is then predicted using simple locally weighted regression:

$$\mathcal{F}_e(G, R_j, R_k) = \sum_{t,j_t,k_t} W(G, G^t, R_j, R_{j_t}, R_k, R_{k_t}) M(j_t \leftrightarrow k_t) \tag{5}$$

where $W()$ is the weight function based on distances between local and global features of training and test data and $M()$ is an indicator function which predicts whether the edge was retained in the training data or not. The feature weights are also computed using locally weighted regression. The labels are then predicted using the message passing algorithm over the constructed Markov network with the estimated feature weights.

5 Experimental Results

We describe the experiments conducted on a subset of the LabelMe [19] dataset. We randomly selected 350 images from LabelMe and divided the set into 250 training and 100 test images. Our training data-set consists of images with segmentations and labels provided [1]. We used GIST features [18] as global features for scene matching. For appearance modeling, we use Hoeim's features [17] together with class specific metric learning used by [20]. The pairwise relation feature vocabulary consists of 5 contextual relationships [2]. In all experiments, we compare the performance of our approach to a fully-connected Markov network and a neighborhood based Markov network. We measure the performance of our annotation result as the number of pixels correctly labeled divided by total number of pixels in the image, averaged over all images.

[1] grass, tree, field, building, rock, water, road, sky, person, car, sign, mountain, ground, sand, bison, snow, boat, airplane, sidewalk

[2] Contextual relations - above/below, left/right, greener, bluer, brighter

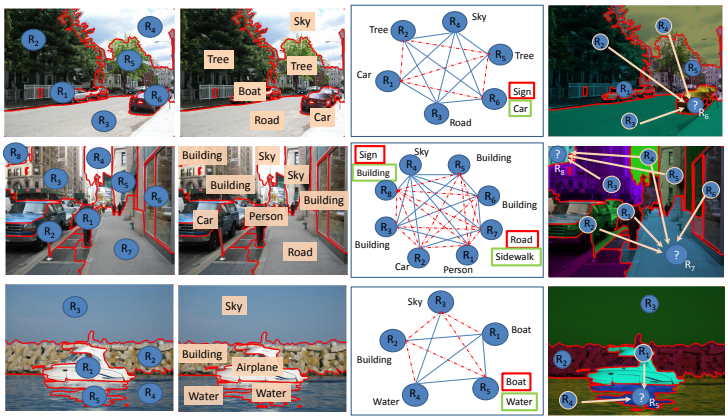

Fig. 5. A few qualitative examples from the LabelMe dataset of how constructing the network structure using our approach leads to an efficient Markov structure which also improves labeling performance

In the training phase, we run inference on each training image and utilize the ground truth to evaluate the importance of each edge. At each iteration, a few unimportant edges are dropped and feature weights are re-estimated. Figure 6 (a) show examples of how our approach captures the scene dependency of $\mathcal{F}(e)$ and β respectively. Fig 6 (b) shows the percentage improvement over a fully-connected network and a neighborhood based network with each iteration. The figure clearly shows that dropping edges not only provides computational efficiency, but also improves the matching scores in training due to the removal of spurious constraints or constraints which link regions which are not discriminative.

On test images, we first predict the Markov network using the learned $\mathcal{F}(e)$ and then utilize β to perform inference. Figure 7 show the performance of our approach compared to a fully-connected and a neighborhood connected Markov network on the LabelMe dataset at different thresholds of $\mathcal{F}(e)$. A higher threshold corresponds to dropping more edges. The values in parenthesis on the threshold axis shows the average percentage of edges dropped at that particular threshold. We also compared the performance of our approach to publicly available version of texton-boost (without CRF) on our LabelMe dataset and it yields approximately 50% as compared to 59% by our approach. It should be noted that this is similar to the performance of the local-appearance based model used in our approach. Therefore, our approach should also provide considerable improvement over the CRF version of the texton-boost as well. Above the performance chart, we show "what" edges are dropped at a given threshold. We also compare our approach to the exemplar based approach similar to [21] where the labels are transferred based on the edge matches in the training dataset.

Figure 5 shows representative examples where our approach performed better than the fully-connected network. The second column of the figure shows the

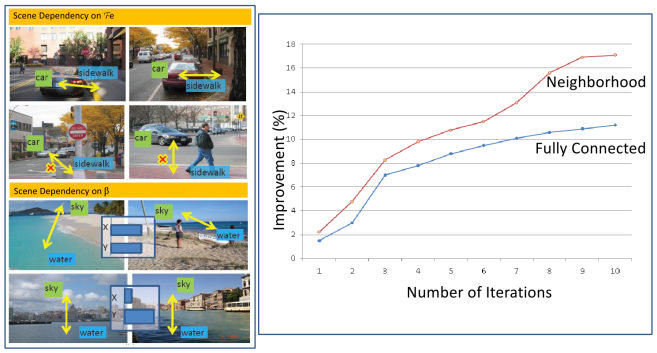

Fig. 6. (a) Scene dependency of $\mathcal{F}(e)$ and β. In the first case the edge between sidewalk and road is informative only when the car is parked or is nearby it. In the second case, in scenes like beaches, both x and y are important features of context; however when the viewpoint is such that water occupies most of the lower space, the importance of x decreases. (b) The graphs show the % improvement over the fully-connected and neighborhood based Markov network as the training continues

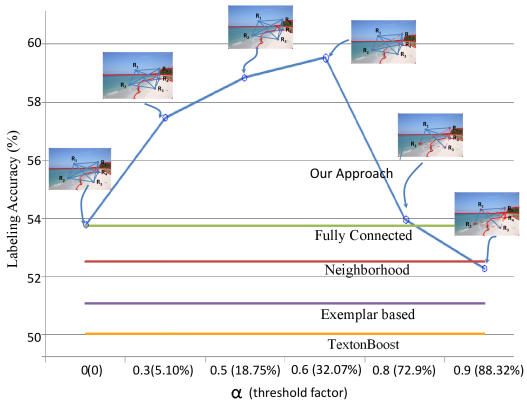

Fig. 7. The graph shows the improvement of our algorithm over the fully-connected network and neighborhood based network on the LabelMe dataset with an example of graph structures at different thresholds of $\mathcal{F}(e)$. The values in the parentheses shows the percentage of edges dropped at a given threshold of $\mathcal{F}(e)$

result obtained using just the appearance model (likelihood term). The third column shows our network compared to a fully-connected Markov network. The label marked in red is the result obtained using the fully-connected network while the label in green is the result obtained using our approach. In the last column, we show the regions of interest (where our approach dropped spurious edges which led to improvement in performance). In the first example, if a fully-connected network is utilized, the label of the red car on the right side of road is forced to signboard (by other car). This is because the car-signboard relationship

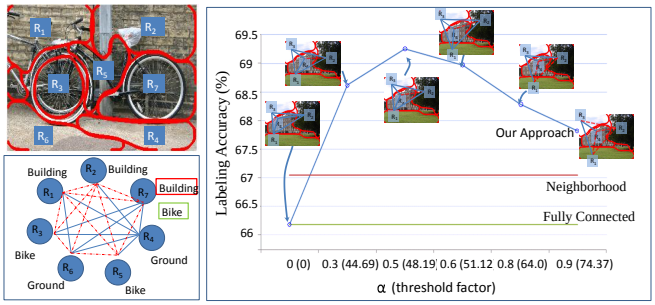

Fig. 8. (a) An example where our approach removed spurious edges and improved labeling performance. (b) Labeling accuracy of our algorithm compared to fully-connected and neighborhood connected Markov networks on the MSRC dataset with examples of graph structures at different thresholds of $\mathcal{F}(e)$

is stronger than car-car relation in the current spatial configuration and the bad appearance model predicts signboard as a more likely label. On the other hand, when the spurious edge is dropped, the labeling improves and the region is correctly labeled as a car. Similarly in the second example, we observe that the sidewalk is labeled as road in the fully-connected network (due to strong appearance likelihood and presence of buildings). On the other hand, the region labeled as person boosts the presence of sidewalk (people walk on sidewalks) and when spurious edges from buildings are dropped by our approach the labeling improves and the region is correctly labeled as sidewalk.

We additionally tested our algorithm on the MSRC dataset. The training and testing data of the MSRC dataset is the same as in [16]. The dataset has 21 object classes. It should be noted that MSRC is not an ideal dataset since the number of regions per image is very low and therefore there are not many spurious edges that can be dropped. However, this experiment is performed in order to compare the performance of our baseline to other state-of-the-art approaches. Figure 8(a) shows the performance of our algorithm compared to fully-connected and neighborhood connected networks on the MSRC dataset. Our results are comparable to the state of the art approaches based on image segmentation such as [16]. Figure 8(b) shows an example of one case where dubious information is passed along edges in the fully-connected network leading to wrong labeling. Region 7, in the fully connected network, was labeled as building. The is because the building-building and bike-building contextual relationship is stronger than bike-bike relationship. But when the link of the bike region with regions labeled as building was removed through our edge prediction, it was correctly labeled as bike.

Acknowledgement: This research is supported by ONR grant N000141010766. This material is based upon work supported by, or in part by, the U. S. Army Research Laboratory and the U. S. Army Research Office under grant number W911NF-09-1-0383. The authors would also like to thank Alexei Efros, Martial Hebert and Tomasz Malisiewicz for useful discussions on the paper.

References

1. A. Gupta and Larry Davis, Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers, In ECCV (2008) 16-29.
2. P. Carbonetto, and N. Freitas, and K. Barnard, A statistical model for general contextual object recognition, In ECCV 2004.
3. S. Divvala, D. Hoiem, J. Hays, A. A. Efros, M. Hebert, An Empirical Study of Context in Object Detection, In CVPR 2009.
4. C. Galleguillos, A. Rabinovich and S. Belongie, Object Categorization using Co-Occurrence, Location and Appearance, In CVPR 2008.
5. J. Li and L. Fei-Fei.What, where and who? Classifying event by scene and object recognition In ICCV 2007.
6. X. He and R. Zemel, Latent topic random fields: Learning using a taxonomy of labels, In CVPR 08.
7. K. Murphy, A. Torralba, W. Freeman, Using the Forest to See the Trees:A Graphical Model Relating Features, Objects and Scenes, NIPS 2003.
8. A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie, Objects in Context,In ICCV 2007.
9. A. Torralba, K. P. Murphy and W. T. Freeman, Contextual Models for Object Detection using Boosted Random Fields, In Adv. in Neural Information Processing Systems (NIPS), pp. 1401-1408, 2005.
10. N. Friedman, The Bayesian structural EM algorithm, UAI 1998.
11. L.K McDowell, K. Gupta and David. Aha, Cautious Inference in Collective Classification, AAAI 2007.
12. J. Neville and D. Jensen, Iterative Classification in Relational Data, AAAI 2000 Workshop on Learning Statistical Models from Relational Data.
13. A. Rakotomamonjy, F. Bach,S. Canu and Y. Grandvalet, More Efficiency in Multiple Kernel Learning, ICML 2007.
14. M. Galun, E. Sharon, R. Basri and A. Brandt, Texture segmentation by multiscale aggregation of filter responses and shape elements, ICCV, 2003.
15. A. Rabinovich and T. Lange and J. Buhmann and S. Belongie, Model Order Selection and Cue Combination for Image Segmentation, In CVPR 2006
16. J. Shotton, M. Johnson, R. Cipolla, Semantic Texton Forests for Image Categorization and Segmentation, In CVPR 08.
17. D. Hoiem, A.A. Efros, and M. Hebert, Geometric Context from a Single Image, ICCV 2005
18. A. Oliva, and A. Torralba, Building the Gist of a Scene: The Role of Global Image Features in Recognition, Visual Perception 2006.
19. B. C. Russell, A. Torralba, K. P. Murphy and W. T. Freeman,LabelMe: a database and web-based tool for image annotation, IJCV 2008.
20. P. Jain and A. Kapoor, Probabilistic Nearest Neighbor Classifier with Active Learning, http://www.cs.utexas.edu/users/pjain/pknn/.
21. T. Malisiewicz and A. Efros, Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships. In NIPS, 2009
22. A. Gupta and L. S. Davis, Objects in Action:An Approach for Combining Action Understanding and Object Perception, In CVPR 2007
23. M. Szummer, P. Kohli and D. Hoiem, Learning CRFs using Graph Cuts. In: ECCV 2008.
24. I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun and Y. Singer, Large margin methods for structured and interdependent output variables. In JMLR, 6(Sep):1453-1484, 2005.