Scene Labeling Using Sparse Precision Matrix

Nasim Souly Mubarak Shah
Center for Research in Computer Vision (CRCV), University of Central Florida (UCF)
nsouly@eecs.ucf.edu, shah@crcv.ucf.edu

Abstract

Scene labeling task is to segment the image into meaningful regions and categorize them into classes of objects which comprised the image. Commonly used methods typically find the local features for each segment and label them using classifiers. Afterwards, labeling is smoothed in order to make sure that neighboring regions receive similar labels. However, these methods ignore expressive connections between labels and non-local dependencies among regions. In this paper, we propose to use a sparse estimation of precision matrix (also called concentration matrix), which is the inverse of covariance matrix of data obtained by graphical lasso to find interaction between labels and regions. To do this, we formulate the problem as an energy minimization over a graph, whose structure is captured by applying sparse constraint on the elements of the precision matrix. This graph encodes (or represents) only significant interactions and avoids a fully connected graph, which is typically used to reflect the long distance associations. We use local and global information to achieve better labeling. We assess our approach on three datasets and obtained promising results.

1. Introduction

Semantic image segmentation, assigning a label to each pixel of an image, is a classic and challenging task in computer vision, due to the efforts needed to simultaneously segment and recognize the image regions. One of the widely used approaches to address this problem is to exploit MAP (maximum a posteriori) inference in a multi-class conditional random field (CRF). This is the extension of the binary CRF, which has been widely used to find foreground-background in images. Common CRF models are defined over pixels, patches or super-pixels of the image. These models generally comprise of the unary or association potential, which measures how likely a pixel (or a super-pixel) can be assigned a particular label without taking into account the properties of other parts of the image, and the smoothing or interaction potential, which assesses (evaluates) how the labels of the other connected nodes (pixels or super-pixels) interact to maximize the assignment agreement.

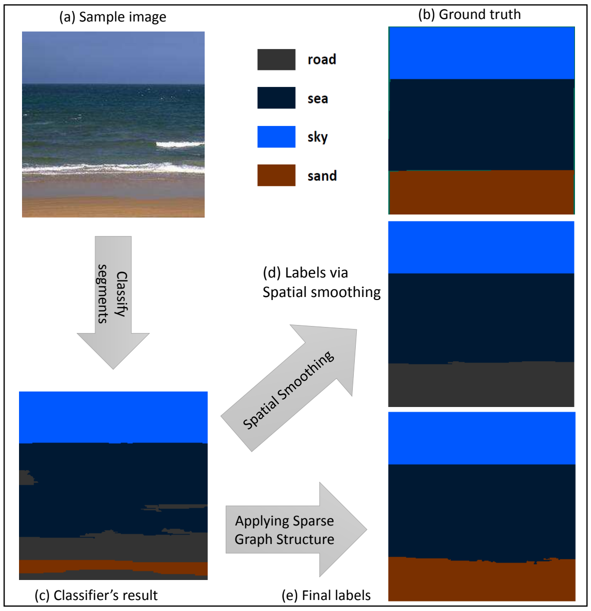

Figure 1. Given an image, we aim to improve the semantic labels of the regions, originally miss-labeled by classifiers. (a) shows a query image, (b) shows the human annotated image (ground truth), (c) shows labels obtained by classifiers, (d) shows labels via spatial smoothing and (e) shows our results.

Commonly, the structure of the CRF is specified manually; in images typically a 2D lattice is used to build an adjacency CRF using the neighboring pixels. However, this model has two important limitations. First, it is unable to incorporate long-range (long-distance) connections between different regions of the image. Second, it may not be able to model the contextual relationships among labels and may not be capable of capturing the complexity in the labels.
One of the approaches to overcome this problem is to use a *fully connected CRF*, in which pair-wise potentials are defined between all pairs of the nodes (pixels or super-pixels). However, the main limitation of this method is the complexity of the inference. The overwhelming number of edges in

the model makes the problem difficult to be solved in an acceptable time. Furthermore, since the optimization solution to the multi-label CRF is not exact, the complexity in the structure leads to reduced accuracy.

In this paper, we propose to learn the label graph (the correlation graph between labels in the dataset) and find the structure of the super-pixels within the image using the sparse *precision matrix* (also called concentration matrix) estimated using graphical lasso. We aim to infuse the relations between labels in the model, without expensive learning of parameters in training the CRF. By doing this, we find the compatibility among labels instead of using Potts model for all pairs of labels, thus the cost for different combination of labels would be dependent on their correlation and the way they influence each other. However, we consider only nodes (regions) in the image, which have interactions with other regions and which are not limited to spatial smoothness only.Also, our model facilitates using smaller elements (smaller super-pixels or pixels) of the image. Due to the fact that, we do not need to encode all interactions between these elements, we can find finer and more accurate boundaries using smaller super-pixels. In our approach, in addition to utilizing the scene semantics by employing the structure and dependency among labels and regions, we also exploit global context by refining local probabilities achieved by classifiers using a retrieval set, which is obtained based on k nearest neighbors of image employing GIST features.

In order to demonstrate the performance of our method, we report experimental results on three benchmark datasets including MSRC2 [20], Stanford Background [7] and SIFT-flow [16]. In summary, we make the following contributions:

- We find the structure of the graphical model between labels and regions using sparse precision matrix to exploit helpful long distance interactions without considering all connections.
- We improve the scores of super-pixels by combining local classifiers results and probabilities obtained by a retrieval set based on global information of a scene.
- We incorporate discovered significant interactions between labels (positive or negative correlations) in pairwise cost term of the energy minimization problem.

The rest of the paper organized as follow: section 2 reviews related work proposed for scene labeling, and in section 3 our proposed method is described in detail. The experiments and evaluations of our method are presented in section 4, and finally in section 5 we conclude our paper.

2. Related Work

Recently, semantic segmentation has been the subject of many research works. Proposed methods are different in terms of employed features and descriptors, primitive elements (pixels, patches or regions), classifier choices and incorporation of different techniques for context. A majority of methods employ Conditional Random Fields [14]. These methods use mainly appearance (local features) as unary potential and smoothness between neighboring elements as the pairwise term [20]. In order to integrate potentials of the features at different levels (pixels and super-pixels) higher order CRF have also been explored [19], [7].

In addition to local features, some methods benefit from object detectors and combine the results from detectors and context information [14], [26]. In some approaches, the image segments are labeled by transferring the labels from a dataset of known labels. To do so, for a given image, similar images are retrieved from a sample data using a nearest neighbor algorithm, then by using Markov Random Field model, pixels (or super-pixels) in the image are labeled [16], [25] and [27]. There are many extensions of this type of labeling, for instance, in [2] authors propose to learn the weights of descriptors in an off-line manner to reduce the impact of incorrectly retrieved super-pixels. Also, authors in [22] proposed to use a locally adaptive distance metric to find the relevance of features for small patches in the image and to transfer the labels from retrieved candidates to small patches of the image. In [8], instead of using a retrieval set to transfer the labels, a graph of dense overlapping patch correspondences is constructed; and the query image is labeled by using established patch correspondences.

In some other papers, authors incorporate context information in their modeling, using global features of the image or applying co-occurrence of the labels [29]. Deep learning techniques have also been used in scene labeling. For each pixel of the image, multi-scaled features are obtained and a neural network is trained to aggregate feature maps and to label the regions with highest scores. Note that, these models need a large data for training [3], [23]. In [11] authors proposed to represent an image as a collage of warped, layered objects which are sampled from reference images. For a given image, they retrieve a dictionary of object segment candidates that match the image, then represent the image by combining these matched segments. For this purpose, they need a dataset of label exemplars. Moreover, in [10] the authors use detectors to find the bounding boxes of the objects and label regions using information from detectors and surface occlusions, in addition, they use RGB-depth to understand the scene.

In contrast to these models, we automatically find the relations of classes, and incorporate the context in refining scores as well as pairwise costs to achieve better label assignments without highly expensive training or merely using common scene to model the relations among classes.

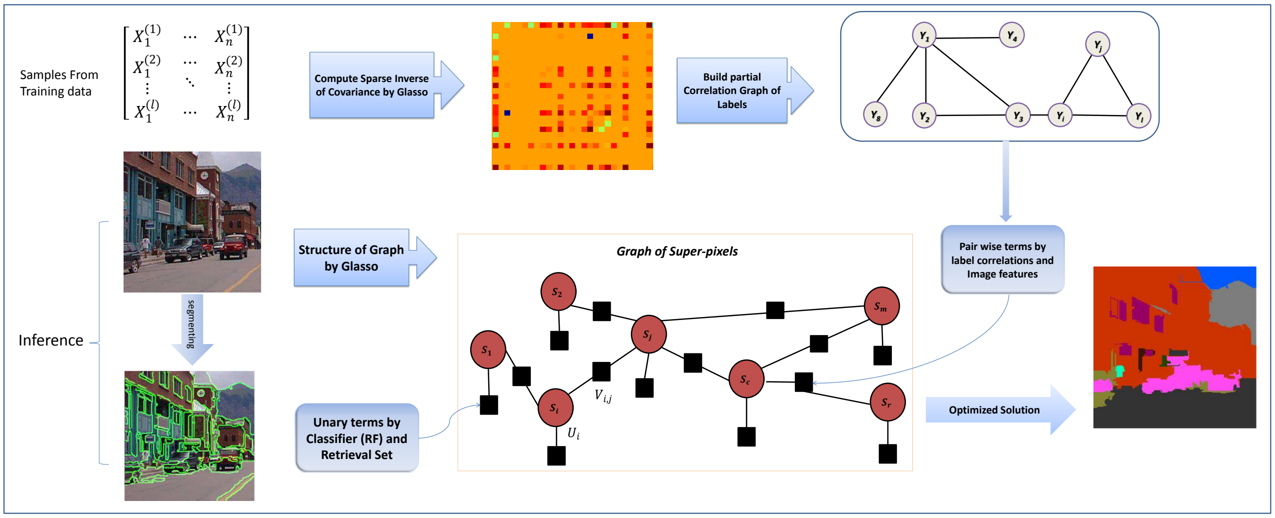
Samples From Training data
Compute Sparse Inverse of Covariance by Glasso
Build partial Correlation Graph of Labels
Inference
segmenting
Structure of Graph by Glasso
Graph of Super-pixels
Pair wise terms by label correlations and Image features
Unary terms by Classifier (RF) and Retrieval Set
Optimized Solution
$V_{i,j}$
U_i

Figure 2. The overview of our approach: We begin by extracting the feature matrix, and segmenting the image into super-pixels. Then classifiers (random forest) are trained. We detect the relations between labels using the sparse estimated partial correlation matrix of the training data. In the inference part, for a given image the label scores are obtained via the classifiers (random forest and nearest neighbors), then the energy function of a sparse graphical model on super-pixels is optimized to label each super-pixel.

3. Proposed Approach

Our approach consists of two main steps. The first step consists of off-the-shelf parts including feature extraction and classifier training based on local features of the sample training images. Also, in this phase using the training data, we capture the structure of semantic label interactions graph to be later employed in the pair-wise cost computation. In the second step, which is the inference, for a given query image, using scores computed by the classifiers for each possible label, and the pair-wise costs obtained by label correlations and appearance features of the image, the MAP inference in CRF framework is applied and each super-pixel is assigned a label. An overview of our proposed model is shown in figure 2.

In training, first we segment images using efficient graph based segmentation [4]. Next, for each super-pixel, local features, including SIFT, color histogram, mean and standard deviation of color, area and texture, are extracted. Given these local features, classifiers (random forest) are trained to label super-pixels using their local features. Also, in training phase we build the sparse precision matrix based on the sample data to highlight the important relations (positive or negative correlations) between labels. In testing, for a query image we find the unary terms, for its segments, using scores from local classifiers refined with the probabilities obtained from a retrieval set based on global features. Then, we use a fast implementation of graphical lasso to find the structure of the dependency graph between super-pixels and assign weights to edges based on correlation values. Finally, we use α expansion to optimize the energy function and assign a label to each super-pixel.

3.1. Graphical Lasso and Sparse Precision Matrix

In order to find the structure of the graph of our model, we employ the precision matrix (the inverse of covariance matrix) to capture the dependency between variables. The partial correlation between two variables X and Y, given other variable Z, measures the association between X and Y, after regressing X and Y on Z. If the partial correlation between two variables given all other variables is zero, there will be no edge between them in the corresponding partial correlation graph. The matrix of partial correlations between variables can be defined using the inverse of covariance matrix Ω. Therefore, zeros in the inverse covariance indicate that there is no edge in the graph. Even though empirical covariance of the data is a decent approximation of the true covariance, this is not valid for the precision matrix. Furthermore, when the dimension of the data increases, the covariance matrix may not be invertible. We assume the data follows Gaussian distribution and use graphical lasso.

Let $X = (X^{(1)}, ..., X^{(p)})$ be a p-dimensional random vector. Assume, we have a set of n random samples $X_1, ..., X_n$, we are interested in identifying conditional independence between the pair of variables (features) $X^{(i)}$ and $X^{(j)}$, given other variables. In doing so, X can be represented by a graph $G = (V, E)$, where vertices correspond to p variables and the edges represent the correlations between variables. In the Gaussian (Normal) distribution, the correlation and dependency graph are equivalent. Even though the data may not have a normal distribution, since conditional independence graphs are hard to estimate, employing partial correlation is a reasonable alternative to find the structure of the interactions between the variables. Let the matrix $C = \{\rho_{i,j}\} \in \mathbb{R}^{p \times p}$ be a partial correlation matrix, where $\rho_{i,j}$ captures the partial correlations between variables $X^{(i)}$ and $X^{(j)}$, and

$$\rho_{i,j} = -\Omega_{i,j} / \sqrt{\Omega_{i,i}, \Omega_{j,j}}, \tag{1}$$

where $\Omega = \Sigma^{-1}$ is the inverse of the covariance matrix of the data with covariance Σ. Using sample covariance matrix to estimate the matrix C is not proper for high dimensional data, due to the limited number of samples, the covariance matrix may not be invertible. Also, more importantly, the inverse of empirical covariance matrix may not be sparse and consequently not resulting in a sparse graph. In order to find the structure of the graph and obtain a certain number of influential edges, it is desirable to have zeros in the precision matrix, since zeros determine the independent (uncorrelated) variables. Therefore, imposing sparsity constraint on the elements of precision matrix enforces that insignificant and noisy relations are discarded and meaningful dependencies are persevered. To achieve sparsity, [30] proposed to use a lasso (Least Absolute Shrinkage and Selection Operator) model [24] to estimate each variables using others as predictor and by applying L_1 regularization on coefficients to enforce sparsity. Therefore, the edges in the graph are removed for the variables for which corresponding coefficients are zero. In [5], an algorithm, named graphical lasso (glasso), was proposed to maximize the Gaussian log-likelihood of the data with L_1 penalty on precision matrix elements to impose sparsity. This approach uses block coordinate gradient to solve the optimization problem, which is fast and suitable for our application. Let S be the empirical covariance matrix of the data, then Ω can be obtained by,

$$\arg\max_{\Omega} \text{ logdet } \Omega - \text{tr}(S\Omega) - \lambda \left\|\Omega\right\|_1, \tag{2}$$

where tr is the trace of the matrix and $\left\|\right\|_1$ is the L_1norm (sum of the absolute values) of the matrix. In brief, one can model the dependency between variables using their partial correlation graph. The partial correlation graph has an edge between j and k when $\rho_{j,k} \neq 0$. Furthermore, as mentioned above, partial correlation has a direct relation with inverse of covariance of the data (equation 1). Therefore, by estimating a sparse precision matrix (inverse of covariance), one could obtain the structure of the dependency graph between variables, where zeros in the precision matrix mean there is no edge between corresponding variables. In following sections we explain each part of the approach in detail.

3.2. Local Classifiers

In this section, we explain the first step of the model. In training, we start with segmenting each sample image into super-pixels using efficient graph-based segmentation method [4], followed by computing a feature vector(including, SIFT, color mean) for each super-pixel in the image. Since the ground truth for each image is pixel based, each super-pixel is assigned a label which correspond to the majority of its pixels. We use the same features as used in [25]. In order to rescale the classifier scores and give chance to other classes to compete during optimization phase, we use a sigmoid function. By doing so, if the classifier mislabels a super-pixel, theres is more chance that the label would be changed during the inference phase. We adapt the parameters of the sigmoid function using the validation data. We use random forest classifiers [15] to classify each super-pixel in a test image. Due to the fact that super-pixels may break the structure of the data, since training data inevitably is noisy, the bagging using subset of training examples and subsets of features is used to reduce the effects of the noisy data. Unlike some of the other methods, which train object detectors in addition to the region classifiers, we only use region features and small scale classifiers to obtain the initial label scores for each super-pixel. In our experiments, random forest achieved better results in terms of average accuracy among all the classes, even though we randomly discard some of the samples during the training, due to large number of super-pixels.

3.3. Global Retrieval

Since the local classifiers treat each super-pixel individually, the context information may be missed, therefore we propose to refine the scores obtained from the classifiers by leveraging the global feature extracted from the data. By doing this, we enforce that global information of the scene and geometrical features play a role in labeling the data. We use GIST features to retrieve a subset of the nearest neighbors of the query image from the training data. We use the method proposed in [17] to speedup the retrieval process and make it scalable for large databases. Next, we compute the probability of assigning each label, l, at a specific location by counting the number of super-pixels with the label l in the retrieval set, and normalize it with respect to the total number of labels. Thus, we have a probability as $p_g(label = l_i|location = (x, y))$. Finally, for each label we modify the obtained scores from the classifiers with these probabilities (corresponding to the super-pixels) using the following late fusion formulation:

$$\text{w}_{new}(i, j) = \text{w}(i, j)^{\gamma} \times p_g(i, j)^{1-\gamma}, \tag{3}$$

where γ is the combination coefficient and $\text{w}_{new}(i, j)$ is the new probability of i_{th} label for the super-pixel j.

3.4. Scene Graph Structure

In order to capture the structure of the label graph, we start by building a matrix comprising of the sample data.

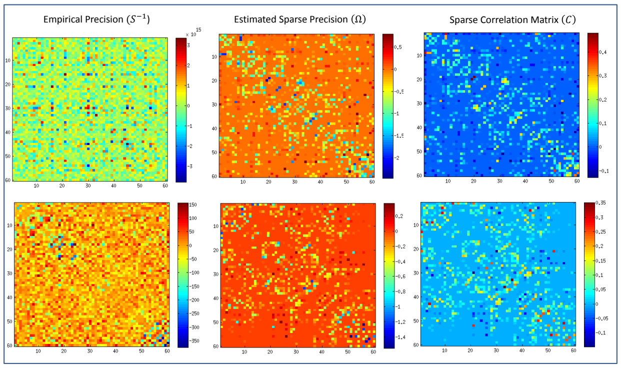

Figure 3. The first row is obtained by using the output (scores) of each classifier and treating it as a random sample. The second row is obtained using the features of the super-pixels to find the correlation between them. The first column corresponds to empirical inverse of covariance matrix of the data; as shown the entries are very noisy and finding true interactions among the super-pixels is difficult. However, the estimated sparse precision matrix provide fewer and more meaningful interactions.

Each image in the training set is represented by a vector of size equal to the number of classes; we want to discover the influences of labels, therefore our random variables (features) are labels in the dataset. For example, in the SIFT-flow data set we have 33 labels, therefore each vector has dimension of 33. The value of a particular variable (specific label) in this representation is the probability of seeing that label in the image. These probabilities are obtained by counting the pixels belonging to the class and normalizing them by the image size. Then, using equation 2 the precision (concentration) matrix is estimated. The degree of the sparsity is handled by parameter λ. Partial correlations of labels are used to find the interaction between labels, which will be used for pairwise-cost (interaction potential) in the CRF formulation instead of using Potts model. Therefore, if two connected nodes do not have the same label, assigning different pair of labels contribute differently in finding the conditional probability of the assignment.

In addition, in order to capture the structure of the graph for the image elements (here super-pixels) in the inference step, we use the graphical lasso to obtain the relationships between the super-pixels. By doing this, if two super-pixels are related but assigned irrelevant labels, the cost of the assignment is increased. To do so, each super-pixel is treated as a random variable, and by using the classifiers which are trained for class labels, we generate samples for these variables. Thus, the length of each vector is equal to the number of super-pixels and we will have L vectors, where L is number of the classes. Then, we again use graphical lasso and estimate a sparse precision matrix (inverse of covariance), and subsequently obtain the partial correlation graph, where the zero indicates no edge between super-pixels. Note that since the number of super-pixels can be large, the covariance matrix may be singular and not invertible, due to the fact that the number of available samples (e.g. scores from classifiers) is limited. Therefore, in such cases using the sparse estimation can be beneficial. Not only do we find the structure and the connections between regions, we also use these values to incorporate relevancy of super-pixels in pairwise potentials. As it is shown in figure 3, the inverse of the sample covariance matrix is very noisy due to the fact that the covariance matrix can be singular (or close to singular). By using the graphical lasso we can capture the structure of the graph, (as shown in the figure) and also preserve the correlation between spatial neighbors of super-pixels. The alternative for finding the relations between super-pixels is to use their features and try to find dependency between super-pixels, by sparse representation of each super-pixels using other super-pixels as predictors. The example is shown in the bottom row of the figure 3. We use features of super-pixels after reducing the dimension by PCA. In our experiments, we use the scores from classifiers as sample data since they are more efficient.

3.5. Energy Function Optimization

As we obtain graph structure for the query image, we build a CRF over the super-pixels given the features of the image, and formulate an energy function E as follows:

$$E(y,x) = \sum_{s_i} U(y_i,x) + \tau \sum_{i,j \in rel\_set(i,j)} V(y_i,y_j,x), \quad (4)$$

where the goal is to assign a label $y_i \in \mathfrak{L} = 1,2,...,l$ to each super-pixel i, while leveraging correlations between labels to refine the individual labeling. Also, we aim to incorporate local smoothness between relevant super-pixels as well. $rel\_set(i,j)$ represents the set of the edges, which correspond to non-zero entires in the precision matrix. And, τ is a weight to control the balance of smoothness. The unary term, U, here is defined as the cost of assigning a label c to a super-pixel s_i, which we obtain by using scores provided by the classifier w_i for a particular super-pixel:

$$U(y_i = c|x_{s_i}) = 1 - \frac{1}{1+e^{-\mathrm{w}_{i,c}}}. \quad (5)$$

The pairwise term considers both appearance similarity between super-pixel i and j as well as correlation between labels, as follows:

$$\begin{aligned} V(y_i = l, y_j = k|x_{s_i}, x_{s_j}) &= \delta(l,k) \times F(s_i,s_j), &(6)\\ \delta(l,k) &= -\log(\sigma(\rho_{l,k})), &(7)\end{aligned}$$

where $\rho_{l,k}$ is the correlation between labels which is found in the training step, σ is a sigmoid function, and F is the measure of similarity between super-pixels based on color and position features. This term adds cost to the energy

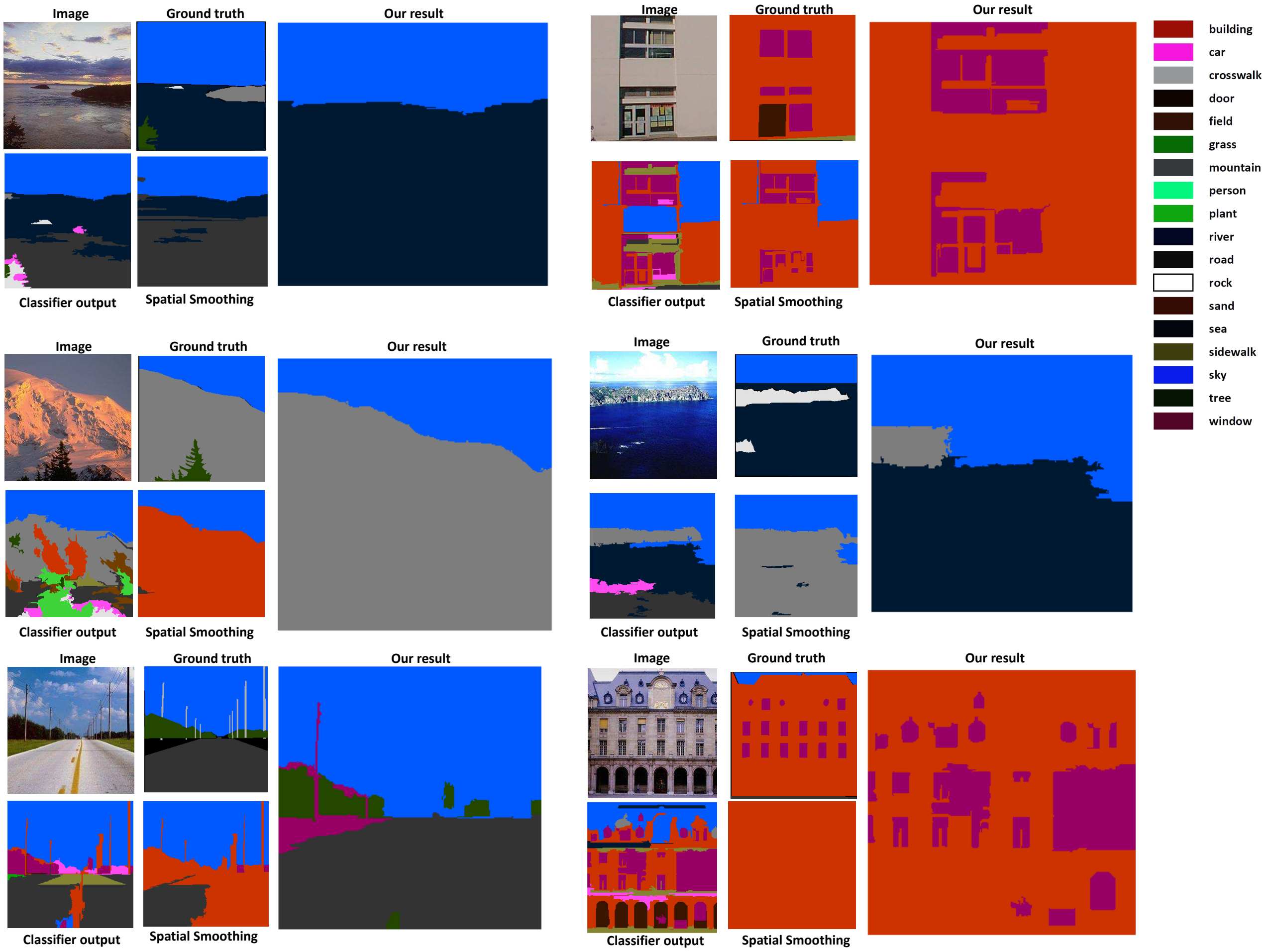

Figure 4. Some samples from SIFTflow data set: We show the image, the labels based on classifier scores, results after smoothing using spatial neighborhood and Potts model, and results using our method employing super-pixel correlation graphs.

in cases where related or similar super-pixels are given irrelevant labels. However, it also applies different costs to different combinations of labels. It should be noted that here that the edges are not limited to spatial neighbors of the super-pixels only, we also include significant (relevant) long interactions. However, despite fully connected configurations, we do not consider all interactions, thus only significant relations are taken into account. In this structure irrelevant and noisy interactions are avoided. Moreover, we incorporate the partial correlations between super-pixels in the function F given below. This provides the notion of dependency between super-pixels apart from only the appearance similarity.

$$F(s_i, s_j) = (w_1\, e^{-\|I_i - I_j\|} + w_2\, e^{-\|p_i - p_j\|}) R(s_i, s_j), \quad (8)$$

where I_i is the feature for super-pixel i, namely color mean , p_i is the center position of the super-pixel i, and R measures the relevancy of two super-pixels. This can be computed as $exp(\sigma(\rho_{s_i,s_j}))$, where σ is a sigmoid function.

4. Experiments and Results

We evaluate our method on three benchmark datasets. The first dataset is Stanford-background, which has 8 classes and 715 images, and following [21] data is randomly split into 80% for training and the rest for testing with 5-fold cross validation. As shown in table 1 we compare our results with state-of-the-art methods, and we achieve better results.

Table 1. Accuracy on StandfordBG dataset

Method	Avg Accuracy
Farabet natural [3]	81.4
Gould [9]	77.1
Shauai [21]	80.1
Local Classifier	72.8
Local Classifier + Global	78.9
Local + Global + Spatial smoothing	82.2
Ours Final (sparse structure)	**84.6**

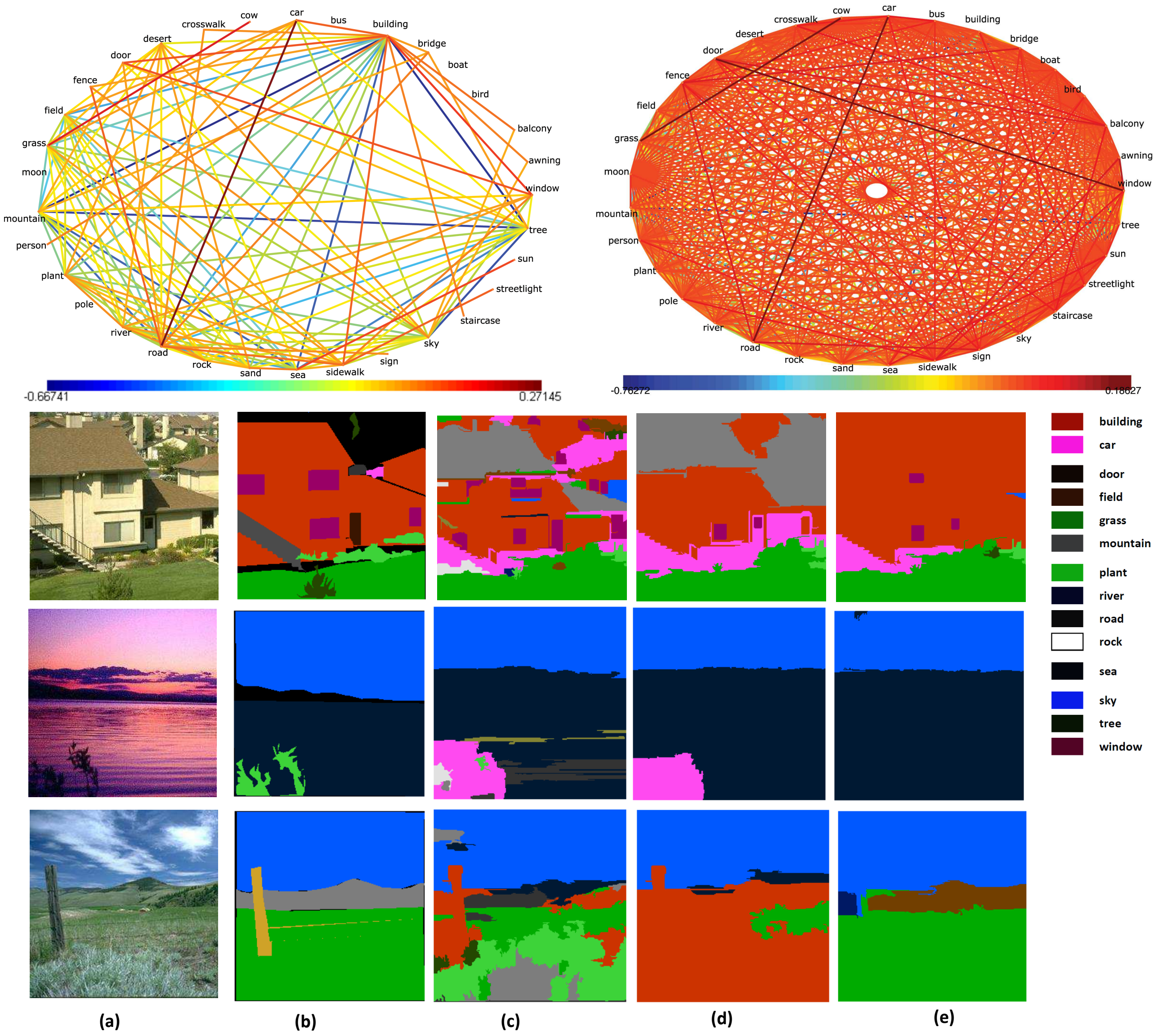

Figure 5. On the top row, we show two graphs: LEFT: obtained by the sparse partial correlation matrix , and the RIGHT: obtained using an empirical inverse of covariance matrix. As it is clear, more relevant relations are maintained and irrelevant edges are removed. Below the graphs, we show some sample images which have been properly labeled using the positive or negative correlation between labels. (a) sample image, (b) ground truth, (c) classifier result, (d) spatial neighborhood smoothing with Potts model, (e) results obtained by our approach.

The second dataset that we assess our approach with is SIFTflow dataset [16], which consists of 2,488 training images and 200 testing images from 33 classes collected from LabelMe [18]. The quantitative results of our approach are reported in table 2 and qualitative results are shown in figure 4. As it is shown, our method is able to achieve promising results without using computationally expensive features or object detectors. Note that the main aim of our method is to improve the local labeling via capturing the proper interactions among labels and super-pixels in addition to leverage from context information. Thus, improving the initial labeling leads to better final results.

We also applied our method on third dataset, MSRCV2 [20], which has 591 images of 23 classes. We use the provided split, 276 images in training and 255 images. Here again our method improves the classifiers results and achieves comparable results to the other methods which use different features. For instance, authors in [12] extract features for each pixel, and build a fully connected graph on pixel levels, where the unary classifier gives 84% accuracy.

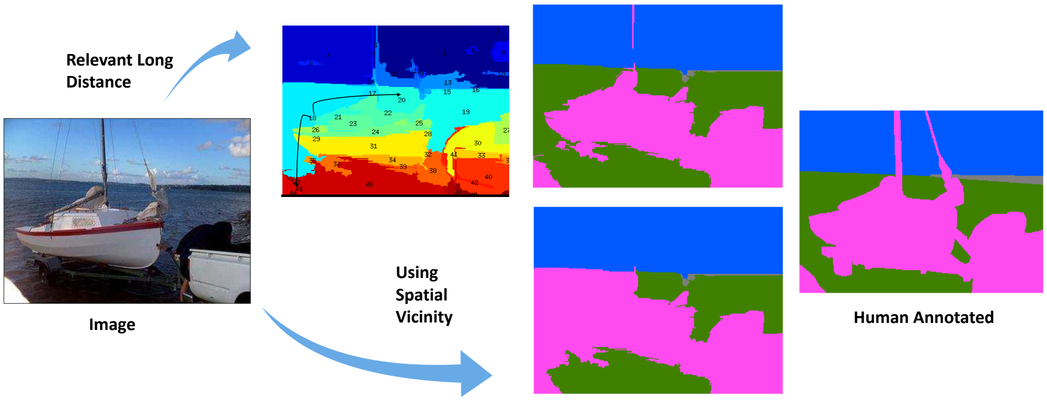

Figure 6. This example demonstrate that adding long distance edges can prevent over-smoothing and also refine the labels.

Table 2. Accuracy on SIFTflow dataset

Method	Avg Accuracy
Farabet [3]	78.5
Tighe [26]	78.6
Collage Parsing [28]	77.1
Shauai [21]	80.1
Gerorge without Fisher Vectors[6]	77.5
Gerorge Full [6]	81.7
Local Classifiers	71.2
Local Classifiers + Global	75.3
Local +Global + Spatial smoothing	77.7
Ours Final (sparse structure)	**80.6**

They improve the results by 2%, while our improvement is about 8% which is significant. If the classifiers are improved, our results can be improved even more.

Table 3. Accuracy on MSRC2 dataset

Method	Avg Accuracy
Harmony Potentials [1]	83
Fully Connected CRF [12]	86
Segment CRF with Co-Occurrence [13]	80
Local Classifier	76.6
Local Classifier + Global	77.1
Local +Global + Spatial smoothing	81.7
Ours Final (sparse structure)	**84.1**

In table 4 the average accuracy results per class are reported. As it is shown, our method does not compromise the per class accuracy for smoothing.

Table 4. Avg Accuracy Per Class

Method	StanfordBG	SIFTflow	MSRC2
Local Classifier	53.8	37.6	71.3
Our Result	77.3	45.8	76.8

4.1. Discussion

Our method improves results obtained from the classifiers in two folds. First, by imposing some constraints on label graph, more meaningful pairwise costs are applied for scene labeling. For example, in the label graph as shown in figure 5, building and mountain have negative partial correlation, on the other hand, building and windows have high positive correlation. Therefore, as shown in the top row of examples in figure 5, the mountain segments are refined. Also, since windows-building have less pairwise-cost, the windows super-pixels are not smoothed out as it was the case in column (d).

In addition, expanding the connectivities beyond immediate vicinities boosts the strength of the model. Selective edges based on partial correlation between segments prevent the model from over-smoothing and enforce the correlated segments to be assigned relevant labels. For instance, in the image shown in figure 6 super-pixel 18 and 15 are not immediately adjacent; however, in the sparse correlation matrix they are positively correlated. Thus, there is an edge between them and consequently, since their similarity and correlation is high, they are labeled correctly.

5. Conclusion

In this paper, we proposed to incorporate context information in both label space and observation space (superpixels) to boost local classifier results in order to better semantically label segments in an image. We used graphical lasso to estimate the sparse precision matrix of data to find relevant long distance interactions in addition to spatial smoothness. We have shown that, this model can refine label assignment using the correlation between labels as well as segments. Also, our model does not smooth out foreground labels as can be seen in spatial labeling. We reported improved experimental results on the SIFTflow, Standford background and MSRC2 benchmark datasets.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[2] D. Eigen and R. Fergus. Nonparametric image parsing using adaptive neighbor sets. In *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, pages 2799–2806. IEEE, 2012.

[3] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. *arXiv preprint arXiv:1202.2160*, 2012.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[6] M. George. Image parsing with a wide range of classes and scene-level context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3622–3630, 2015.

[7] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2009.

[8] S. Gould and Y. Zhang. Patchmatchgraph: Building a graph of dense patch correspondences for label transfer. In *Computer Vision–ECCV 2012*, pages 439–452. Springer, 2012.

[9] S. Gould, J. Zhao, X. He, and Y. Zhang. Superpixel graph label transfer with learned distance metric. In *Computer Vision–ECCV 2014*, pages 632–647. Springer, 2014.

[10] R. Guo and D. Hoiem. Labeling complete surfaces in scene understanding. *International Journal of Computer Vision*, pages 1–16, 2014.

[11] P. Isola and C. Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3048–3055. IEEE, 2013.

[12] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Adv. Neural Inf. Process. Syst*, 2011.

[13] L. Ladickỳ, C. Russell, P. Kohli, and P. H. Torr. Inference methods for crfs with co-occurrence statistics. *International journal of computer vision*, 103(2):213–225, 2013.

[14] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *Computer Vision–ECCV 2010*, pages 424–437. Springer, 2010.

[15] A. Liaw and M. Wiener. Classification and regression by randomforest.

[16] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011.

[17] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.

[18] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

[19] C. Russell, P. H. Torr, and P. Kohli. Associative hierarchical crfs for object class image segmentation. In *in Proc. ICCV*. Citeseer, 2009.

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[21] B. Shuai, G. Wang, Z. Zuo, B. Wang, and L. Zhao. Integrating parametric and non-parametric models for scene labeling.

[22] G. Singh and J. Kosecka. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3151–3157. IEEE, 2013.

[23] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011.

[24] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[25] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Computer Vision–ECCV 2010*, pages 352–365. Springer, 2010.

[26] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3001–3008. IEEE, 2013.

[27] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3748–3755. IEEE, 2014.

[28] F. Tung and J. J. Little. Collageparsing: Nonparametric scene parsing by adaptive overlapping windows. In *Computer Vision–ECCV 2014*, pages 511–525. Springer, 2014.

[29] T. L. Vu, S.-W. Choi, and C. H. Lee. Improving accuracy for image parsing using spatial context and mutual information. In *Neural Information Processing*, pages 176–183. Springer, 2013.

[30] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.