DATA-DRIVEN SCENE UNDERSTANDING BY ADAPTIVE EXEMPLAR RETRIEVAL

Xionghao Liu1, Wei Yang1, Qing Wang1,*, Liang Lin1,2, Jian-Huang Lai1,2

1Sun Yat-Sen University, Guangzhou 510006, China
2Department of Computing, The Hong Kong Polytechnic University
lxiongh@126.com, platero.yang@gmail.com, ericwangqing@gmail.com, linliang@ieee.org, stsljh@mail.sysu.edu.cn

ABSTRACT
This article studies a data-driven approach for semantically scene understanding, without pixelwise annotation and classifier pre-training. Our framework parses a target image with two steps: (i) retrieving its exemplars (i.e. references) from an image database, where all images are unsegmented but annotated with tags; (ii) recovering its pixel labels by propagating semantics from the references. We present a novel framework making the two steps mutually conditional and bootstrapped under the probabilistic Expectation-maximization (EM) formulation. In the first step, the references are selected by jointly matching their appearances with the target as well as the semantics. We process the second step via a combinatorial graphical representation, in which the vertices are superpixels extracted from the target and its selected references. Then we derive the potentials of assigning labels to one vertex of the target, which depends upon the graph edges that connect the vertex to its spatial neighbors of the target and to its similar vertices of the references. Two steps can be both solved analytically, and the inference is conducted in a self-driven fashion. In the experiments, we validate our approach on two public databases, and demonstrate superior performances over the state-of-the-art methods.

Index Terms— scene understanding, semantic segmentation, semantic-aware sparse coding, multi-image graphical model

1. INTRODUCTION

Significant progresses have been identified in solving the task of semantic image understanding [15, 21, 4, 12, 7]. However, these methods usually build upon supervised learning with fully annotated data that are expensive and sometimes limited in large-scale scenarios. Several weakly supervised methods were proposed [16, 5, 19, 17, 18] to reduce the overload of data annotation, which can be trained with only image-level labels indicating the classes presented in the images. Recently, data-driven approaches [9, 8, 10, 11] receive increasing attentions, which tend to leverage knowledges from auxiliary data in weakly supervised fashions, and demonstrate very promising applications. Following this trend, one interesting but challenging problem arises for the scene understanding: How to parse the raw images in virtue of the strength of numerous unsegmented but tagged images, as the image-level present tags can be achieved more easily. The difficulties are however obvious due to the fact that one scene image probably contains several objects of diverse appearances. In this work, we investigate this problem by developing a unified framework, in which the two following steps perform iteratively, as Fig. 1 illustrates.

In Step 1, we search for similar images as the exemplars (i.e. references) matching to the target image from the auxiliary database (in Fig. 1 (b)), and these references are required...
to share similar semantic concepts with the target. To find the reliable references that are highly co-related with the target, we learn from the recent advance of sparse coding techniques [20]. We believe the target image can be represented by a sparse combination among the auxiliary data. Moreover, we enforce the representation to be semantically meaningful: The references that we will select should contain consistent tags. The tags of the target image can be also taken into account during the iteration, as they can be determined by the last label assignment (in Step 2). We solve this step using the proximal gradient method, so that we select the references according to their co-efficiencies in the solution. This process is partially motivated by the approach by Liu et al. [10] in Multimedia community, and they employed the sparse representation on patch-level for joint image labeling.

In Step 2, we assign labels to the pixels of the target by propagating semantics from the selected references. We create a graphical model, in which the vertices are the superpixels from the target image and its references. There are two types of edges over the graph inspired by [6]: (i) the inner-edges connecting the spatial adjacent vertices within the target; (ii) the outer-edges connecting the vertices of the target to those of its references. The potentials are then derived into an MRF form by aggregating the two types of edge connections, which can be fast solved by the Graph Cuts algorithm [4]. Extracting the graph over multi-images was also discussed in [18] for scene labeling. Their model differed from ours in two aspects. First, they made connections on all images in the training set, while our approach builds on only the selected references for the target image. To find the reliable references that are highly co-related with the target, we learn from the recent advance of sparse coding techniques [20]. We believe the target image can be represented by a sparse combination among the auxiliary data. Moreover, we enforce the representation to be semantically meaningful: The references that we will select should contain consistent tags. The tags of the target image can be also taken into account during the iteration, as they can be determined by the last label assignment (in Step 2). We solve this step using the proximal gradient method, so that we select the references according to their co-efficiencies in the solution. This process is partially motivated by the approach by Liu et al. [10] in Multimedia community, and they employed the sparse representation on patch-level for joint image labeling.

In this section, we phrase the problem in a probabilistic formulation, and then discuss the Expectation Maximization inference framework for optimization.

2. PROBLEM FORMULATION

In this section, we phrase the problem in a probabilistic formulation, and then discuss the Expectation Maximization inference framework for optimization.

2.1. Probability Model

Let $\Delta = \{I_k, L_k\}_{k=1}^N$ denote a set of images $\{I_k\}$ with image-level labels $\{L_k\}$. Each image $I_k$ is represented as a set of superpixels $\{x_i^k\}_{i=1}^{n_k}$, where $n_k$ is the number of superpixels in $I_k$.

![Fig. 2. Illustration of the semantic-aware sparse coding. Top: The target image is denoted by the pentagon) and each auxiliary image denoted by an triangle. The darked triangles represent the images selected as the references. bottom: The grey squares represent semantic labels that are introduced as constraints during the optimization. And we select a subset of auxiliary images as references for the target image.](image-url)
M-step minimizes the energy $L(Q,Y_t)$ with respect to $Y_t$ with $Q(\alpha)$ fixed.

(i) The E-step: Approximating $Q(\alpha)$:

The posterior of the latent variable $Q(\alpha)$ is defined as,

$$Q(\alpha) = P(\alpha|I_t,Y_t,\Delta) = \frac{1}{Z} \exp\{-E_\alpha(\alpha, I_t, Y_t, \Delta)\}, \quad (5)$$

where $Z$ is the normalization constant of the probability. The energy $E_\alpha$ evaluates the appearance and semantics consistency, which is specified as,

$$E_\alpha(\alpha, I_t, Y_t, \Delta) = E_{Sc}(\alpha, I_t, \Delta) + \lambda E_{Sa}(\alpha, Y_t, \Delta). \quad (6)$$

The first term $E_{Sc}$ measures the appearance similarity between $I_t$ and images in $\Delta$, defined as,

$$E_{Sc} = \frac{1}{2} \| F(I_t) - B\alpha \|_2^2 + \beta \| \alpha \|_2^1 \quad (7)$$

where $\beta$ is the tradeoff parameter used to balance the sparsity and the reconstruction error. $F(\cdot)$ is an $m$-dimensional global feature of an image, and $B \in \mathbb{R}^{m \times N}$ is a matrix consisting of all the features of images in $\Delta$.

The second term $E_{Sa}$ in Eq. (6) measures semantic consistency, defined as,

$$E_{Sa} = \frac{1}{2} \sum_{i,j \in N} \| \alpha_i - \alpha_j \|_2 W_{ij} + \gamma \alpha^T D \alpha, \quad (8)$$

where $W_{ij} = \frac{|L_i \cap L_j|}{|L_i| |L_j|}$ measures the semantic similarity between $(I_i, I_j) \in \Delta$, and $D$ is a diagonal matrix whose entry $D_{kk} = 1 - \frac{|L_k \cap L_k|}{|L_k|^2}$ measures the semantic dissimilarity between $I_k \in \Delta$ and the target image $I_t$. $L_i$ is the latent labels of the target image, which is unknown at the beginning\(^1\), and it can be determined from $Y_t$ during the later iterations.

![Illustration of the combinatorial graphical model. The dark circles represent the superpixels; the fours over the square region are extracted from the target image while the others from references that are denoted by dashed regions. Two types of edges are introduced: inner-edges connecting the spatial neighbors within the target image (denoted by red wavy lines), and the outer-edges connecting the superpixels of the target to those of its references (denoted by straight green lines).](image)

Intuitively, the first term of Eq.(8) encourages the images sharing similar labels to be selected as the references. And the second term of Eq. (8) penalizes the images which are semantically dissimilar with $I_t$ to reconstruct the target image.

**Algorithm 1 Adaptive Reference Retrieval**

**Input:** Target image feature $F(I_t)$, codebook $B$, semantic constraints $\Lambda$, and the threshold $\sigma$ for stop.

**Output:** Semantic sparse coding coefficient $\alpha^*$.\(^\dagger\)

1. **while** $\| \alpha_{k+1}^* - \alpha_k^* \|_2 > \sigma$ **do**
2. **compute** the gradient of $g(\alpha)$ at $\alpha^k$, $\nabla g(\alpha^k) = B^T (B \alpha^k - F(I_t)) + \lambda \Lambda \alpha^k$.
3. $z_k^* = \arg \min \{ z - \alpha^k \}^T \nabla g(\alpha^k) + \beta \| z \|_1 + \frac{\gamma}{2} \| z - \alpha^k \|_2^2$ where $L > 0$ is a parameter.
4. **iteratively increasing** $L$ by a constant factor until the condition $g(z_k^*) \leq M_k^0 (\alpha^k, z_k^*) := g(\alpha^k) + \nabla g(\alpha^k)^T (z_k^* - \alpha^k) + \frac{\gamma}{2} \| z_k^* - \alpha^k \|_2^2$ is met, else return to step 3.
5. Update $\alpha_{k+1}^* := \alpha_k^* + \nu_k (z_k^* - \alpha_k^*)$, where $\nu_k \in (0,1]$
6. $k := k+1$
7. **end while**
8. $\alpha^* = \alpha_k^*$

(ii) The M-step: estimating $Y_t$:

The M-step performs to minimize the following energy function with respect to $Y_t$:

$$E_M(Y_t) = -\sum_\alpha Q(\alpha) \ln P(I_t, Y_t, \alpha|\Delta). \quad (9)$$

However, summing out $\alpha$ for all possibility demands very expensive computation cost, particularly to process a large number $N$ of data. Instead, we seek a lower-bound of $E_M(Y_t)$. Assume that we can infer $\alpha^*$ with the maximized probability $Q(\alpha^*)$ by the E-step. Then we can define the joint distribution of $(I_t,Y_t)$ conditional on $Q(\alpha^*)$, and we have

$$\sum_\alpha P(I_t, Y_t|\Delta; \alpha^*) > \sum_\alpha P(I_t, Y_t, \alpha|\Delta). \quad (10)$$

It is straightforward in the context of our task, as the cumulative density of assigning labels from good references (i.e. given $\alpha^*$) is higher than that with general cases. Thus, we set the lower-bound as,

$$E_M(Y_t) > -\sum_\alpha Q(\alpha) \ln P(I_t, Y_t, |\Delta; \alpha^*), \quad (11)$$

where $Q(\alpha)$ is fixed by the last E-step. The energy to be minimized can be further simplified as,

$$E_M(Y_t) = -\ln P(I_t, Y_t|\Delta, \alpha^*), \quad (12)$$

where we will specify $P(I_t, Y_t |\Delta, \alpha^*)$ with a combinatorial graph model in Sec. 3.2.

\(^1\) We initialize $L_t$ as the whole label set of the database.
3. INFERENCE AND IMPLEMENTATION

Within the EM formulation, the inference algorithm iterates with two steps: i) computing \( \alpha^* \) in the E-step for reference retrieval and ii) solving the labeling \( Y_t^* \) with the selected references in the M-step.

3.1. Adaptive Reference Retrieval

Maximizing \( Q(\alpha) \) is equivalent to minimizing the energy defined in Eq. (6) w.r.t \( \alpha^* = \arg \min_{\alpha} E(\alpha, I_t, Y_t, \Delta). \) Notice that \( E(\alpha, I_t, Y_t, \Delta) \) can be regarded as an semantic-aware sparse representation, where we jointly model the appearance reconstruction with semantic consistency. Fig. 2 intuitively illustrates this model, and it can be rewritten as,

\[
E(\alpha) = \frac{1}{2} \| F(I_t) - B \alpha \|_2 + \beta \| \alpha \|_1 + \frac{1}{2} \lambda^T A \alpha, \tag{13}
\]

where \( A = 2(L + \gamma D), \) and \( L \) is the Laplacian matrix of \( W \) (e.g. \( L = \text{diag}(\sum_{j=1}^N W_{ij}) - W \)). The semantic associated terms in Eq. (13) can be phrased in convex forms, thus we can use the proximal gradient method to solve this problem easily, and the optimization is shown in Algorithm 1.

Given the optimized \( \alpha^* \), we can simply select the references according to coding co-efficiencies, e.g. select by thresholding. And we set \( \alpha_k = 0 \) if image \( I_k \) is not selected.

3.2. Aggregated Label Assignment

Given the references determined by \( \alpha^* \), we propagate their semantic labels to \( I_t \) by constructing a combinatorial graph. We extract superpixels from both \( I_t \) and the references as graph vertices, and connect them with probabilistic edges incorporating their affinities, as Fig. 3 illustrates.

Two types of edges are considered over the graph, and each superpixel of the target connects with the \( p \) most similar superpixels of each reference.

We define \( P(I_t, Y_t | \Delta, \alpha^*) \) in Eq. (12) on the graphical model, and is denoted as,

\[
P(I_t, Y_t | \Delta, \alpha^*) = \frac{\sum_{i=1}^n \psi(y_t^i | \alpha^*, \Delta) + \sum_{(x^i_t, x^j_t) \in \omega} \phi(y_t^i, y_t^j, x^i_t, x^j_t)}{\sum_{(x^i_t, x^j_t) \in \omega}}, \tag{14}
\]

where \( \omega \) is the inner edges. The optimization of Eq. (12) becomes an tractable graphical model optimization problem.

To derive the potentials of assigning labels to one vertex of the target, i.e., \( \psi(y_t^i | \alpha^*, \Delta) \) in Eq. (14), we propose a semantic-based superpixel density prior, which is defined as,

\[
\psi(y_t^i | \alpha^*, \Delta) = \sum_{k=1}^N \alpha^*_k \rho(x^i_t, I_k) \delta(y_t^i \in L^k), \tag{15}
\]

where \( \rho(x^i_t, I_k) \) denotes the density of superpixel \( x^i_t \) in image \( I_k \), which is defined as,

\[
\rho(x^i_t, I_k) = \frac{1}{N^k} \sum_{(x^i_t, x^j_t) \in \xi} \| f(x^i_t) - f(x^j_t) \|_2, \tag{16}
\]

where \( \xi \) are outer-edges, \( N^k \) is the number of outer-edges, and \( f(\cdot) \) is the feature vector of a superpixel. This density measures the similarity between the superpixel \( x^i_t \) in the target and its neighboring superpixels connected by outer-edges in the reference image \( I_k \), thus it implicitly exhibits the probability that \( x^i_t \) sharing the same labels with its reference \( I_k \).

Algorithm 2 Overall procedure of our framework

**Input:** Target \( I_t = \{x^i_t\}_{i=1}^n \), auxiliary \( \Delta = \{I_k, L_k\}_{k=1}^N \).

**Output:** Label of each superpixel \( Y_t = \{y^i_t\}_{i=1}^n \).

1: **Initial:** \( L_t^0 \) contains all labels, and \( n = 1 \).
2: **while** \( L_t^{n+1} \neq L_t^n \) **do**
3: Minimize the energy \( E(\alpha) \) defined in Eq. (6) using Alg. 1.
4: **for all** \( x^i_t \) in \( I_t \) **do**
5: **for all** image \( I_k \) in \( B \) **do**
6: Select the \( p \)-most similar superpixels \( O_t^k = \{x^j_t\}_{j=1}^p \).
7: Construct \( O_t^k = \cup_k O_t^k \).
8: **end for**
9: Add \( (x^i_t, x^j_t) \) to \( \omega \) for all \( x^j_t \in O_t^k \).
10: Add \( (x^i_t, x^j_t) \) to \( \xi \) for all neighbors & \( x^j_t \) of \( x^i_t \), \( i \neq j \).
11: **end for**
12: Minimize Eq. (14). Optimize the latent label \( Y_t^* \) using alpha-beta swap algorithms of graph cuts.
13: Update \( L_t^{n+1} \) as the unique set of \( Y_t^* \).
14: \( n \leftarrow n + 1 \)
15: **end while**

The pairwise potentials, i.e. \( \phi(y^i_t, y^j_t, x^i_t, x^j_t) \) in Eq. (14), encourages the smoothness between neighboring superpixels within the target, as,

\[
\phi(y^i_t, y^j_t, x^i_t, x^j_t) = \| f(x^i_t) - f(x^j_t) \|_2 \delta(y^i_t \neq y^j_t), \tag{17}
\]

where \( \delta(\cdot) \) is the indicator function.

Eq. (15) is known as the data term. It ensures that the current labeling \( y^k_t \) is coherent with the observed data. It penalizes a label \( y^k_t \) to \( x^i_t \) if it is different with the observed data. Eq. (17) is known as the smooth term. It ensures that the overall labeling is smooth. Notice that the smooth term satisfies the constrains: \( \phi(y^i_t, y^j_t, x^i_t, x^j_t) = 0 \) if \( y^i_t = y^j_t \), and \( \phi(y^i_t, y^j_t, x^i_t, x^j_t) \geq 0 \), thus approximate solutions Eq. (14) can be found using alpha-beta swap algorithms of graph cuts.

The sketch of our framework is shown in Algorithm 2.

4. EXPERIMENT

We evaluate our framework on two challenging datasets MSRC21 [15] and VOC 2007 [2] by comparing with state-of-the-art. We also conduct an empirical study on the effectiveness of the proposed EM iterations.

Implementation details: (1) we use method proposed in [1] to obtain about 400 superpixels for each image, for each
of which we compute a 225-dimensional feature vector as described in [3], and it covers various image information including color, texture, shape, and location. (2) To compute the global image feature, we first extract the dense SIFT descriptors of each image (intervals of 5 pixels). Then 200,000 descriptors are randomly sampled from images to generate 512 virtual words by sparse coding method introduced in [20]. After that, we compute the global feature of each image using bag-of-word model on a spatial pyramid, i.e., $1 + 2 \times 2 + 4 \times 4 = 21$ blocks of SIFT visual words from the 512-word dictionary. The feature vector length of each image is 10,752.

4.1. Results and Comparisons

**MSRC21 Dataset.** It is a 21-class dataset contains 591 images of 320 $\times$ 213 pixels with ground-truth annotations. For fair comparison, we use the standard training set and testing set split as proposed in [15]. This dataset fits our framework for its high co-occurrence between classes. We use the standard average per-class measure (average accuracy), to evaluate the performance, which is computed by first computing the percentage of correctly classified pixels for each class, and then average over all the classes. For each testing image, we use the rest in the testing set as the auxiliary data for our framework.

Given this insight, we compare the proposed method with the following state-of-the-art algorithms: K. Zh[22], MIM[18], PLSA-MRF[16] and MTL-RF[17]. In our experiment, we set parameters $p$ and $q$ in Algorithm 2 as 10 and 20, respectively. The sparsity parameter $\beta$ in Eq. (7) and semantic parameter $\gamma$ in Eq. (8) are set to 0.1 and 0.2 respectively, and we set parameter $\lambda$ in Eq. (6) as 1.

Table 1 shows the average accuracies for our approach in comparison with other competitive algorithms, from which we clearly see that our algorithm outperforms the others. Benefit from the semantic constraints incorporated in our approach, we achieve significant improvements for certain difficult classes, e.g., chair and cat. Serveral visualized results with the corresponding ground-truths are presented in Fig. 4, and more semantic segmentation results are in supplementary material as to the limited space of article. In addition, some failure examples on the MSRC-21 dataset is shown in Fig. 5. Note that we perform unsatisfactory for some categories, e.g. “boat”. This is mainly due to “boat” is always in small size, which leads to a lack of adequate information for feature extraction.

By using an unoptimized Matlab implementation, the segmentation task takes around 8 seconds (extract features: 1s; sparse coding with semantic constraints: 5s; graph cut: 2s) on a 64-bit system with Core-4 3.6 GHz CPU, 4GB Memory.

**VOC 2007 Segmentation Dataset.** This dataset contains 5011 training images with pixelwise annotations, and 4952 testing images with only the bounding boxes of the objects presented in the image are marked. There are total 20 classes for the classification, detection, and segmentation task. We conduct our experiments on its segmentation set using the official settings, which splits the data into 422 training/validation images and 210 testing images. The parameters setting of our framework is as the same as for MSRC21.

Few performance on VOC 2007 dataset is reported, due to the 20 extremely challenging categories it contains. However, the basic implementation of STF was kindly provided by authors of [14], and its parameters, e.g., number and depth of trees, are as same as provided. Here we compare with the weakly supervised STF by running the code provide by the author. We also compare our method with [22]. Results are reported in Table 2, and our methods outperforms [22] by 3%, which shows the superior performance of our method on extremely challenging scenarios.

Moreover, we validate the effectiveness of the proposed EM iterations from two aspects. First, we plot the energy $E_\alpha$ in each iteration, which is the energy of semantic-aware spare coding defined in Eq. (6), as shown in Fig. 6. Notice that, the energy $E_\alpha$ in Eq. 6 decreases with iterations increasing. The lower energy indicates that the references selected is better.

In addition, we choose some semantic segmentation results in each step of iteration, as Fig. 7 shows that the performance comes better with the increasing of iterations.
using the proximal gradient method. Our method is evaluated by selecting references that jointly match their appearance with the target as well as the semantic, and it can be solved using the proximal gradient method. Our method is evaluated on two datasets and outperforms the state-of-arts.

5. CONCLUSIONS

In this paper, a new framework is proposed for semantic segmentation where only image-level labels are available. The reference superpixels are selected by jointly matching their appearance with the target as well as the semantic, and it can be solved using the proximal gradient method. Our method is evaluated on two datasets and outperforms the state-of-arts.

References


