Holistic Scene Understanding

Jian Yao

August 6, 2015

1 License

If you use this code in your research, please cite:

@inProceedings{ HolisticSceneUnderstanding
author = "J. Yao and S. Fidler and R. Urtasun",
title = "Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation",
booktitle = "CVPR",
year = {2012}
}

We have project website at

http://ttic.uchicago.edu/ yaojian/HolisticSceneUnderstanding.html

Note: To make the code run, you also need to download learning and inference code as instructed in the following. Then don't forget to cite [2],[3] and follow their instructions of usage.

For questions concerning the code, please contact Jian Yao <yzyaojian AT gmail DOT com> and Sanja Fidler <sanja.fidler AT gmail DOT com>.

2 Overview

This package of matlab script is mainly designed for Holistic Scene Understanding problems with related papers[1]. To make it run, we use the following resources:

1. Automatic Labelling Environment(ALE)(Lubor Ladicky and etc.)
   http://www.robots.ox.ac.uk/ lubor/

2. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials(Philipp and etc.)
   http://graphics.stanford.edu/projects/densecrf/

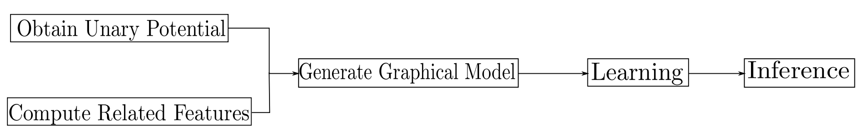
3. Object Detection with Discriminatively Trained Part Based Models(P. Felzenszwalb and etc.)
   http://www.cs.berkeley.edu/ rbg/latent/index.html

4. Distributed Message Passing for Large Scale Graphical Models(Alex G. Schwing and etc.)
   http://www.alexander-schwing.de/projects.php

We take this opportunity to say thanks.

There are basically 3 steps in our code:

- Obtain unary potential for all the nodes in the graphical model
- Connect those nodes and compute the related features
- Train the model on training dataset and do inference on test dataset



2.1 Obtain Unary Potential

For MSRC dataset, we use the ALE software to get the unary potential of superpixels. You need to change the ALE code so that it can output the potentials. For Pascal VOC 2010, we use fully connected CRF to get the unary potential of superpixels. We use the Deformable Part Based model[5] to get bounding boxes and compute the unary potentials as detailed in the paper[1]. As for the unaries for the scenes, we use the method of a standard bag-of-words spatial pyramid and train a linear one-vs-all SVM classifier.

None of them are embeded in our code, so you have to run separately and save it in the correct path. We suggest you save it in

$$data/DATASET\_NAME/Features$$

as indicated in the file: *dataset_globals.m*.

In this package, for your convinience, we precompute the unary potentials.

2.2 Construct the Garph

In this step, we connect the nodes in the graph and compute the features as we presented in the paper. But you could also add more features as you want only if the nodes in this graph will not be changed. For more details about the graph structure, please go to Section 3 in the paper[1].
Once this step is finished, the graph will be also generated since we encode the graph struture in feature computing.

2.3 Learning and Inference

We use the learning method in paper [3] and the inference code in paper [2]. Please go to http://www.alexander-schwing.de/projects.php [2] to download the inference code and to see how to use it.

3 Usage

System Requirement: Linux 64bit with 4G+ memory.

There are four folders in the root: computeFeature, data, external, utility. *computeFeature* consists of all the codes that compute features. We put all the related data in *data* while *external* has resources from other people. *utility* is the folder that consists of the necessary functions in the model.

There are also two files in the root. In *dataset_globals.m*, you will have access to set the parameters and the pathes. *segmentation.m* is the main function.

For your convenience, we precomputed the following data for **original MSRC dataset**:

1. Superpixel, Supersegment Unary Potential;
2. Object Detection Data;
3. Scene Classification Data;
4. Superpixels in two scales.

To make it run, the things you need to do are

step 1 : Download the inference code from `http://www.alexander-schwing.de/dcbp_1.1.zip`, and put it in 'external/cBP-MPIGeneral' as indicated in the 'dataset_globals.m'. Also download the learning code from `http://www.alexander-schwing.de/dSP_3.zip` or from `http://www.alexander-schwing.de/projectsGeneralStructuredPredictionLatentVariables.php` **Source Code v3** in Downloads, put it in 'external/libHCRF'. Configure your operating system to get it work on your computer;

step 2 : Download the data and put it in the CVPR2012_code root directory or you may modify the dataset_globals.m file if you want to put the data somewhere else;

step 3 : Write a parameter file like files in the directory of 'configs', otherwise you will use the default parameter setting. Then run the script 'holistic.m', e.g. holistic('msrc', name_of_config_file). Please look at the script exp_msrc for an example. This script will also run the full Table we had in the paper. The config file "msrc_full" corresponds to the config of our full model;

step 4 : Check the results which should be the same as that in the paper.

3.1 Explanation of Feature Encoding

In case you want to add your own features and try on different dataset, we provide the details of encoding the features which follows the data format in [2].

We write the CRF in a more general way where $\epsilon = 1$ as the following:

$$\sum_{(x,y)\in\mathcal{D}} \epsilon \ln \sum_{\hat{y}} \frac{\theta^T \phi(x,y) + L(\hat{y},y)}{\epsilon} - d^T\theta + \frac{C}{p}||\theta||^p$$

Where:

- θ is the k dimensional vector that give different weights for each feature, here k is the number of features we use in the experiment.

- $\phi(x,y)$ is the feature which also has k dimension. And each component

$$\phi_i(x,y) = \sum_{v\in\mathcal{V}_{i,x}} \phi_{v,i}(y) + \sum_{\alpha\in E_{i,x}} \phi_{\alpha,i}(y) \quad i = 1,\cdots,k$$

- d is empirical means which has the form:

$$d = \sum_{(x,y)\in\mathcal{D}} \phi(x,y)$$

- We use p norm, and C is the trade-off coefficient for the regularization.

- ϵ refers to the temperature factor which usually equals '1' in our code to make the general formulation to be CRF model.

We followed the feature format in the inference code[2],

1. Unary potential:
   (a) feature{r}.sample{i}.local{s}.NumStates
   (b) feature{r}.sample{i}.local{s}.pot
   (c) feature{r}.sample{i}.local{s}.connTo
2. Pairwise Potential:
   (a) feature{r}.sample{i}.factor{s}.size
   (b) feature{r}.sample{i}.factor{s}.pot

where r ranges from the number of features, i ranges from the number of training(if in train) or testing samples(if in inference), s is the number of nodes in unary potential and is the number of edges in pairwise potential. And for whatever features, the index of the node and the edge should be consistent.

Note:

- If there's only unary potential in some feature, then $1(c)$ and $2(a)$can be ignored, and $2(b)$ should be $cell\{0,0\}$ in matlab.
- If there's only pairwise potential in the feature, then $1(b)$ needs to be $zeros(NumStates, 1)$ and $1(c)$ needs to be the index of the edge of which the node is one end. Also $2(a)$ is a vector consists of the corresponding size of each dimension in $2(b)$. Here the order in $2(a)$ has to be the order of matlab dimension of $2(b)$.
- The GroundTruth name for VOC 2010 should be like 2007_000032_GT.bmp to be consistent with MSRC, what you only need to do is to rename the original file.

4 Reference

[1]*Jian Yao, Sanja Fidler, Raquel Urtasun; Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation CVPR 2012*

[2]*A.G. Schwing, T. Hazan, M. Pollefeys and R. Urtasun; Distributed Message Passing for Large Scale Graphical Models,CVPR 2011*

[3]*T. Hazan and R. Urtasun; A Primal-Dual Message-Passing Algorithm for Approximated Large Scale Structured Prediction, NIPS 2010*

[4]*P. Arbelaez, M. Maire, C.Fowlkes and J. Malik; Contour Detection and Hierarchical Image Segmentation, PAMI 2011*

[5]*P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan; Object Detection with Discriminatively Trained Part Based Models, PAMI 2010*