Pointly-supervised Scene Parsing with Uncertainty Mixture

Hao Zhao[1], Ming Lu[2], Anbang Yao[2], Yiwen Guo[2], Yurong Chen[2], Li Zhang[1]
[1]Department of Electronic Engineering, Tsinghua University
[2]Cognitive Computing Laboratory, Intel Labs China
{zhao-h13@mails,chinazhangli@mail}.tsighua.edu.cn
{ming1.lu, anbang.yao, yiwen.guo, yurong.chen}@intel.com

Abstract

Pointly-supervised learning is an important topic for scene parsing, as dense annotation is extremely expensive and hard to scale. The state-of-the-art method harvests pseudo labels by applying thresholds upon softmax outputs (logits). Our method, by contrast, builds upon uncertainty measures instead of logits and is free of threshold tuning. We motivate the method with a large-scale analysis of the distribution of uncertainty measures, using strong models and challenging databases. This analysis leads to the discovery of a statistical phenomenon called uncertainty mixture. Inspired by this discovery, we propose to decompose the distribution of uncertainty measures with a Gamma mixture model, leading to a principled method to harvest reliable pseudo labels. Beyond that, we assume the uncertainty measures for labeled points are always drawn from the certain component. This amounts to a regularized Gamma mixture model. We provide a thorough theoretical analysis of this model, showing that it can be solved with an EM-style algorithm with convergence guarantee. Our method is also empirically successful. On PascalContext and ADE20k, we achieve clear margins over the baseline.

1. Introduction

Dense annotation for scene parsing is very expensive. Thus annotating scenes with point clicks and semantic class assignment is an appealing alternative, and training with this kind of labels is called ponitly-supervised scene parsing in this paper. Specifically speaking, full supervision corresponds to the right-bottom parts of each panel in Fig 1, and point supervision is enlarged and overlapped onto the input images as the left-top parts demonstrate. This work focuses on harvesting pseudo labels for the pointly-supervised training set. Specifically speaking, one can train a model using only the point supervision, which is referred to as the first round model. This model would produce a semantic label prediction for every pixel in the training set, as demonstrated in the right-top parts of each panel in Fig 1. They are referred to as pseudo labels throughout this paper. They are erroneous, but containing right ones. The central problem considered here is how to harvest as-many-as-possible good pseudo labels for training. The state-of-the-art method [3] expands point supervision into regions and regards them as trustworthy pseudo labels. It is proven effective on public benchmarks yet there exist two critical issues:

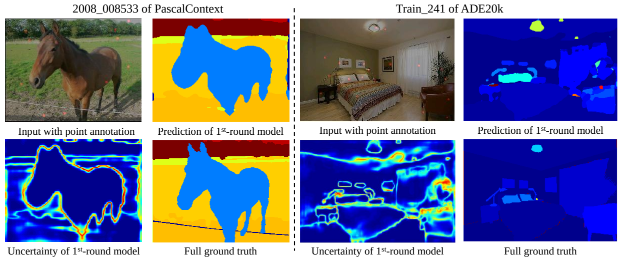
2008_008533 of PascalContext
Train_241 of ADE20k
Input with point annotation
Prediction of 1st-round model
Input with point annotation
Prediction of 1st-round model
Uncertainty of 1st-round model
Full ground truth
Uncertainty of 1st-round model
Full ground truth

Figure 1. Here we show input images with point supervision, pseudo label maps, uncertainty maps and full ground truth.

Firstly, it relies upon softmax outputs, or say logits. It is known that logits can be over-confident upon wrong prediction. The reason behind is that softmax result is only a single point estimate of the predictive distribution. Instead, we compute the uncertainty measures for these predictions, depicted in the left-bottom parts of each panel in Fig 1. They faithfully reflect the confidence of the network outputs. It is obvious that harvesting pseudo labels with low uncertainty (low color temperature in Fig 1) is a promising solution. However, how do we properly define 'low uncertainty'?

Secondly, harvesting pseudo labels using logits would introduce thresholds. It is very time-consuming to tune thresholds for modern deep networks. Meanwhile, using the natural thresholds generated by *argmax* would lead to a trivial usage of the pseudo labels. One may argue that using uncertainty measures still involves defining a *low uncertainty threshold*, as just mentioned in the last bullet point. We show that this problem can be resolved by exploiting a newly discovered statistical phenomenon called uncertainty mixture. It allows us to decide on the optimal threshold of uncertainty measures in an automatic way. Specifically speaking, we decompose the uncertainty measures for unlabeled points with a Gamma mixture and harvest pseudo labels belonging to the certain component.

Beyond the direct application of Gamma mixture, a new regularized model tailored for our problem is proposed, analyzed, implemented and evaluated. We assume the uncertainty measures for labeled points are drawn from the certain component. Intuitively, this helps the mixture model to better capture the shape of the certain component. Mathematically, this amounts to an added regularization term in the objective function of an EM procedure. Since the model has not been visited before, we present a systematic exposition of its analytical properties, from convergence guarantee to solver details. Empirically, this regularized Gamma mixture harvests pseudo labels of higher quality than the baseline and leads to better scene parsing performance, in all experimental settings we inspected.

Last but not least, our method is extensively benchmarked on challenging public datasets, namely PascalContext and ADE20k. It turns out that our method works robustly in various settings. This robustness is attributed to the nature of the method: our mixture modeling decides on the optimal threshold of uncertainty measures automatically. On an absolute scale, our method collaborates well with strong network architectures and training techniques, resulting in new state-of-the-art performance on both datasets. We believe our solution to be a useful one due to its robustness and good performance.

2. Uncertainty Mixture

A fully-supervised scene parsing dataset is denoted as $\{I^i, L^i\}$, in which I^i is the image and L^i is the label map. A pointly-supervised version of this dataset is depicted as $\{I^i, P^i\}$ and P^i is the degenerated version of L^i. First, we train a model $M(*;\Theta^1)$ using only P^i and we call it the first-round model. Formally, we solve this problem:

$$\arg\min_{\Theta^1} \sum_i CE(M(I^i;\Theta^1), P^i)$$

in which $CE(*,*)$ is the cross-entropy loss function. Only labeled points in P^i provide supervision signals for the model. Note that $M(I^i;\Theta^1)$ gives a softmax result (logit) for every pixel in I^i. By taking the index of the maximum value in $M(I^i;\Theta^i)$, we can get the pseudo label map F^i. If we use F^i as it is, this can be considered as a product of applying natural *argmax* thresholds upon $M(I^i;\Theta^1)$. It is also possible to exploit thresholds on $M(I^i;\Theta^1)$ to harvest a subset of F^i as pseudo labels.

For image I^i, we generate an uncertainty map U^i. Our goal is to look for rules to tell certain pixels from uncertain ones. We take Fig 2 as an example, in which two pixels are highlighted. In the pseudo label map generated by the first round model, pixel A is wrongly predicted as *blanket* so that we should not use it for training. Meanwhile, pixel B is correctly predicted as *bed* and considered as a good pseudo label. It can be clearly seen that in the uncertainty map, pixel A has higher color temperature than pixel B.

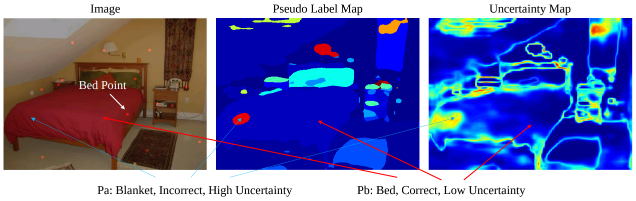
Image
Pseudo Label Map
Uncertainty Map
Bed Point
Pa: Blanket, Incorrect, High Uncertainty
Pb: Bed, Correct, Low Uncertainty

Figure 2. Pixel A: a wrong pseudo label with high uncertainty. Pixel B: a correct pseudo label with low uncertainty.

We conduct an analysis as such:

We assume there are C categories and F^i_j to be the binary mask of category j in the corresponding pseudo label map. Here $j = 1, ..., C$. Applying this mask on the uncertainty map U^i gives us U^i_j which is the uncertainty measures for a specific category. By stacking vectorized U^i_j and excluding zero entries, we collect all the uncertainty measures for points labeled as j, and this array is called U_j.

The uncertainty mixture phenomenon emerges in this analysis: The histogram of U_j is shaped as a two-peak mixture, as Fig 3 demonstrates.

3. Pointly-supervised Scene Parsing

In the mixture of uncertainty measures, we name the low uncertainty component as LUC and the high uncertainty component as HUC. We regard LUC and HUC as Gamma distributions parameterized by $\theta_1 = \{\alpha_1, \beta_1\}$ and $\theta_2 = \{\alpha_2, \beta_2\}$:

$$p_1(x;\alpha_1,\beta_1) = \frac{x^{\alpha_1 - 1} e^{-\frac{x}{\beta_1}}}{\beta_1^{\alpha_1}\Gamma(\alpha_1)}$$

$$p_2(x;\alpha_2,\beta_2) = \frac{x^{\alpha_2 - 1} e^{-\frac{x}{\beta_2}}}{\beta_2^{\alpha_2}\Gamma(\alpha_2)}$$

Here $\Gamma(*)$ is the Gamma function.

And $p_1 + p_2$ characterizes the underlying distribution of every histogram of U_j as visualized in Fig 3. In order to determine the parameters θ_1, θ_2, we can use a standard EM algorithm described in [4] or our regularized EM algorithm. As such, we can harvest pseudo labels belonging to LUC and train the model with them.

Here we formally summarize the whole pipeline of training pointly-supervised scene parsing models with uncertainty mixture. We divide the method into five steps:

(1) Train the first round model $M(*;\Theta^1)$ on the original pointly-supervised dataset $\{I^i, P^i\}$;

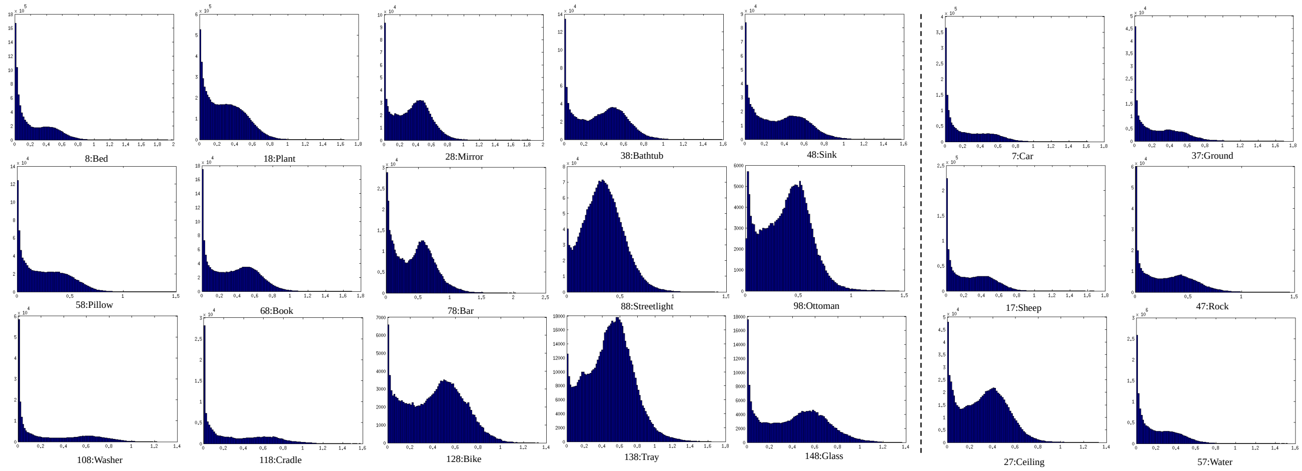

Figure 3. This figure illustrates the ubiquitous existence of the uncertainty mixture phenomenon. Separated by the dash line, left panels correspond to ADE20k and right panels correspond to PASCALContext. Under each panel denotes the class index and name.

(2) Generate the uncertainty measure array U_j for each category j on the whole training set;

(3) Estimate the Gamma mixture parameters θ_{1j}, θ_{2j} by applying the EM algorithm on U_j;

(4) For each unlabeled pixel with pseudo label j and uncertainty measure x, compare $p_1(x; \alpha_{1j}, \beta_{1j})$ and $p_2(x; \alpha_{2j}, \beta_{2j})$. If the former is smaller, harvest this pixel and its pseudo label. Combining all harvested pseudo labels and the original dataset dataset, we get the new dataset with good pseudo labels $\{I^i, G^i\}$;

(5) Finetune the first round model on the augmented dataset $\{I^i, G^i\}$, to get the final model $M(*; \Theta^2)$.

4. Regularized Gamma Mixture Model

In short, our regularized model incorporates the uncertainty measures for labeled points into the formulation. Similar to the generation of the category-wise vector of U_j, we collect the uncertainty measures for labeled points (of category j) as $\overline{U_j}$. The task remains the same: automatically estimating the parameters $\{\theta_{1j}, \theta_{2j}\}$. The core assumption is that $\overline{U_j}$ is drawn from LUC. Empirically $\overline{U_j}$ samples have an uncertainty measure marginally larger than zero. There are other two possible assumptions: (1) $\overline{U_j}$ is drawn from HUC; (2) $\overline{U_j}$ is drawn from a third *very certain* component. Yet apparently they are not reasonable.

Adding this data term into the Gamma mixture model [4] leads to a maximum likelihood estimation problem that has not been studied before. So in this section we provide a thorough analysis of it, giving answers to several important questions: (1) Can this problem be solved with an EM-algorithm? (2) Is it guaranteed to converge like the standard EM-algorithm? (3) What are the detailed steps to solve it?

In this short version, mathematical details are omitted. Interested readers can refer to the full version.

5. Results

5.1. Protocols

All our evaluations are done with PSPNets [7] with DRN backbones [6]. We consider three backbones of different capacities: 22-layer DRN, 54-layer DRN and 105-layer DRN. We report results on two representative benchmarks PASCALContext [2] and ADE20k [8], using the standard splits. For PASCALContext, 4998 samples are used for training and 5105 samples are used for testing. For ADE20k, 20210 images are used for training and 2000 images are used for testing. For a fair comparison, we use the same point annotation sets as [3] do. They have released the point annotation for ADE20k [5]. For the PASCALContext dataset, we generate the point annotation by selecting the mid-point of [1]'s scribbles, as [3] told us in mail. We use the category-wise and average intersection over union (IoU) as the metric. For both the 1st-round model and the fifth step in our pipeline, we train with 100 epochs, a polynomial learning rate annealing strategy of 0.9 alpha value, 0.5x to 2x random scaling, 10 degree random rotation and a crop size of 320×320 pixels. The testing is done in a single resolution.

5.2. Major Results

Here we present the major evaluation results of this study: our framework (section 4) improves the first-round model's performance on all data ponits we inspected and our regularized model is better than the original Gamma mixture formulation. Quantitative results are summarized in Table 1 for PASCALContext and Table 2 for ADE20k. Results for different backbone capacities are presented in different rows. The first column corresponds to the results at the first step. The second and third colums correspond to the performance of the fifth step. Gamma stands for the original Gamma mixture modelling and rGamma represents

	1st-round	Gamma	rGamma
22-layer	31.52	32.48 (+0.96)	**34.17** **(+2.65)**
54-layer	32.63	33.52 (+0.89)	**35.44** **(+2.81)**
105-layer	33.54	34.39 (+0.85)	**36.07** **(+2.53)**

Table 1. Quantitative results on PASCALContext. All numbers are measured in the metric of mean intersection over union (mIoU, %).

	1st-round	Gamma	rGamma
22-layer	24.53	25.45 (+0.92)	**27.00** **(+2.47)**
54-layer	25.20	26.29 (+1.09)	**27.19** **(+1.99)**
105-layer	26.33	27.44 (+1.11)	**28.79** **(+2.46)**

Table 2. Quantitative results on ADE20k. All numbers are measured in the metric of mean intersection over union (mIoU, %).

		ADE20k		PASCALContext	
	Arch	mR	mP	mR	mP
GMM	22	26.43	63.02	34.06	77.38
	54	26.87	63.76	37.19	79.91
	105	**27.64**	64.98	**39.03**	80.85
rGMM	22	25.45	65.72	33.28	81.28
	54	25.98	66.66	36.44	83.34
	105	26.89	**67.58**	38.26	**84.18**

Table 3. Transductive inference performance for our method. Short names mP/mR are categorical mean values for precision/recall, which are measured in percentage (%). Arch stands for the depth of backbones.

the proposed regularized model. Margins over the 1st-round model are also denoted.

From these results we can clearly draw the conclusion that training with pseudo labels harvested by our methods is much better than solely the point annotation. Notably, our algorithms (both *Gamma* and *rGamma*) are completely free of threshold tuning. This is credited to the discovery of the uncertainty mixture phenomenon, whose existence is retrospectively supported by these quantitative results. The regularized model out-performs the vanilla Gamma mixture solution with clear margins. It supports the necessity of exploiting the uncertainty measures for labeled points.

5.3. Transductive Inference Evalaution

In order to further evaluate the pseudo label quality, we report transductive inference results in Table 3. While weakly-supervised learning performance is demonstrated on the testing set, transductive inference performance is reflected with the unlabeled samples in the training set. We use two metrics: categorical mean precision and recall. For each one from the 150/60 classes in ADE20k and PASCAL-Context, we first calculate the precision and recall for harvested pseudo labels then average them. It can be seen that our regularization term leads to clearly higher precision and a little bit lower recall. This fact illustrates the working mechanism of rGMM: to harvest more accurate pseudo labels by selecting more strict thresholds.

6. Conclusions

We study the problem of pointly-supervised scene paring in this paper and make four contributions to the community. Firstly, we conduct a large-scale statistical analysis of the category-wise uncertainty measure for this setting. The major outcome of this analysis is the discovery of a phenomenon called uncertainty mixture. We believe it is of interest to many researchers as it reveals an intriguing property of deep models. Secondly, inspired by the phenomenon, we propose a pipeline that addresses the problem of harvesting pseudo labels in a principled manner. There is no need to tune thresholds on logits. Thirdly, a new regularized Gamma mixture model is presented and thoroughly analyzed, which is proved to be more effective than the vanilla model. Lastly but not least, we set new state-of-the-art results on two public benchmarks.

References

[1] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR 2016*.

[2] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR 2014*.

[3] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *AAAI 2019*.

[4] Andrew R Webb. Gamma mixture models for target recognition. *Pattern Recognition 2000*.

[5] Yunchao Wei, Shuai Zheng, Ming-Ming Cheng, Hang Zhao, and etc. Lid 2020: The learning from imperfect data challenge results. 2020.

[6] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR 2017*.

[7] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR 2017*.

[8] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR 2017*.