

Crowd Segmentation Progress

January 15, 2017

1 Problem Formulation

The goal of quality evaluation is two-folds. Given N worker responses, find:

1. the quality of the bounding boxes (BB) drawn by worker
2. the best proposed region for a given object in an image

2 Key Assumptions and Intuitions

1. In literature, there are scoring functions that require “ground truth” and ones that are “unsupervised”.
2. If a worker’s response differs greatly from the ground truth, then work quality (Q_w) is low.
3. If most of the workers’ response differ greatly from ground truth for a particular image, then task difficulty (D_t) should be high. As a corollary, if the spread of the J_i distribution is large, then D_t should also be high.

To compute the latent quantities Q_w, D_t , we propose an iterative EM-like algorithm, where at every step, we assume that the ground truth bounding box (BB_G) is the current estimate of the maximum likelihood region. The maximum likelihood region is constructed by adding in sub-regions from a tile-graph.

3 Generative Process

The generative model is inspired by [3], the process is as follows:

- A task is defined by an object-image pair i :
 - z_i is hidden variable that completely describes the ground truth BB from the image (e.g. set of all points in BB_G).
 - ϕ_j is some descriptive image-related quantity extracted from BB_G . This can either be a 1-D scalar aggregate or a multidimensional quantity. (e.g. boundary complexity of the image boundary)
 - J_i is the set of all workers j that annotated the object-image.

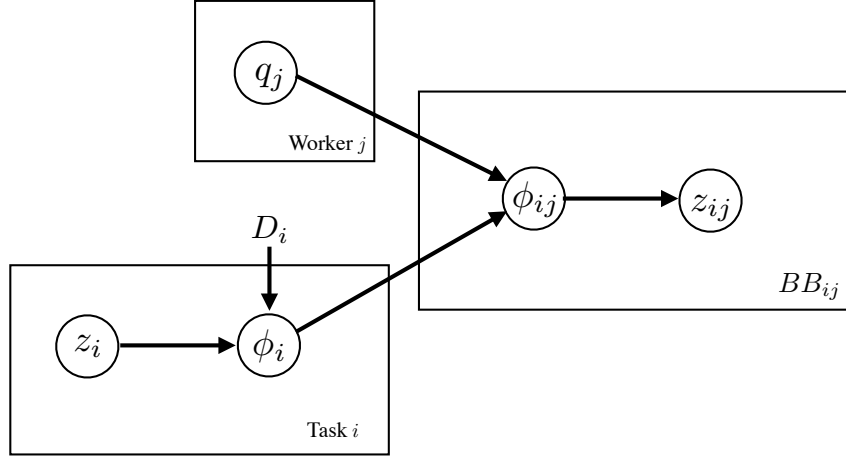


Figure 1: Proposed probabilistic graphical model for crowdsourcing image segmentation.

- D_i is the task difficulty of object-image i . The ϕ_j image summary is determined by both the objective description of $BB_G(z_i)$ and the difficulty of the task. The task difficulty is a measure of how far off are most of the worker's responses compared to BB_G .

$$D_i = \sum_{j \in J_i} dist(\phi_j, \phi_i) \quad (1)$$

Here, we can model the distribution of ϕ_i as :

$$p(\phi_i|z_i) = N(\phi_i; z_i, D_i^2)$$

This agrees with our intuition that in a distribution of image descriptions for all workers, the larger the spread means that the task is difficult. The distribution is centered around z_i (i.e. complete description of BB_G).

- By definition, we assume here that task difficulty is completely a result of the image itself and independent of any worker qualities. The worker j is described by the quality of work that he produces parameterized as q_i . We can also try to model user expertise, but this is less important in salient, common-object segmentation. If we chose to take the prior approach, we assume that q_i only parameterizes the vision-related quantities affecting worker j 's judgement on ϕ_{ij} .

$$q_j = dist(\phi_{ij}, \phi_i) \quad (2)$$

- Both the characteristics of the ground truth BB (ϕ_i) and the worker's ability to segment the image (q_i) would determine how the worker would perceive the image (ϕ_{ij}) and what worker's BB would look like (z_{ij}).

4 Metrics for Φ Functions

There are two sets of annotations that we used as ground truth comparison for computing these metrics: gold-standard annotations cross-matched with MSCOCO ([COCO]) and detailed annotation boundaries drawn by me with the same web interface ([Self]). Note that some of the COCO annotations lack exact cross matches. The metrics computed for objects that are not in the COCO database are flagged and not used for computing the evaluation metrics. However, inexact annotations of the same object (e.g. book-labeled object with only book cover annotation) is still incorporated in the computed metrics. The evaluation metrics can be grouped into three categories:

Area-based: These methods include precision, recall, area ratio or boundary complexity.

$$\text{Precision} = \frac{\text{area}(BB_i \cup T)}{\text{area}(T)} \quad (3)$$

$$\text{Recall} = \frac{\text{area}(BB_i \cup T)}{\text{area}(BB_i)} \quad (4)$$

$$\text{Jaccard} = \frac{\text{area}(BB_i \cap T)}{\text{area}(BB_i \cup T)} \quad (5)$$

Area ratio is a simple baseline proposed by [2] based on the intuition that larger objects should be easier to annotate than smaller objects, so larger annotations should be better than smaller ones.

$$\text{Area ratio} = \frac{\text{area}(BB_i)}{\text{Total image area}} \quad (6)$$

Boundary-based: While precision, recall, and majority-vote are simple metrics, since they are bounded by [0,1], metrics computed against BBG should always be 1. In addition these projection functions do not capture the full resolution of the bounding box.

[2] proposes a bipartite-matching measure based on the Euclidean distance between two BBs. First, they randomly sample $m=300$ points along the annotation boundary, then compute all pairwise Euclidean distance. Then, the Kuhn-Munkres algorithm is used to match together the orientation of the two annotations, and returns the assignments that yields the minimum Euclidean. Finally, the normalized score (NME) of an annotation i is computed as:

$$\text{score} = 1 - \frac{\text{dist}_i}{\max(\text{dist})} \quad (7)$$

where $\max(\text{dist})$ is the maximum Euclidean distance of all the annotations computed in our dataset. Our implementation differs slightly in that we conduct a B-spline parametric interpolation of use $m=50$ points along the boundary rather than random sampling, in order to speed up the computation in the Munkres algorithm. These implementation details should have little effect on the Euclidean scores computed.

Another simple baseline used by [2] is simple a unnormalized count of the number of control points in a user’s annotation (Num Points), based on the intuition that a more

carefully-annotated would result in a better annotation. Since some objects may have inherently simple geometries that could be well-annotated with a small number of control point, to account for the object’s boundary complexity, one possible derived measure could be to normalize by the max number of control point) of the particular object.¹

Contrast-based: These methods examine how close is BB to regions of contrasts detected by CV algorithms (saliency maps, Bayesian Matting) or edge detectors. A major problem when implementing these methods is group and match CV regions to BB annotations, since CV methods often yield over-segmented regions.

5 Preliminary Experiment

We ran a preliminary experiment where each HIT consisted of one annotation task for a specific pre-labelled object in the image, as shown in Fig.2. There is a total of 46 objects in 9 images from the MSCOCO dataset[1]. These objects and images are intentionally chosen so that they represent a variety of image difficulty (based on object clutter-ness) and potential logical error and level of ambiguity. The average number of objects annotations that each worker completed was 10.16. The average time to complete each HIT is 83.96 seconds and workers are compensated for 5 cents per HIT. For each object, we collected annotations from a total of 40 independent workers.



Figure 2: An example can be seen [here](#).

5.1 Data Observations

- **Basic statistical summary:** Most workers makes decent annotation that closely follows the ground-truth BB, since the mean is close to one and standard deviation

¹Since this is a constant for each object i , it would not affect the form of the J_i distribution.

is large for most metrics. The annotations with metric scores significantly below a threshold are likely mistakes due to task ambiguity and ground-truth mismatches, as we can see that applying work quality filter significantly improves the mean and SD.

All	Mean	SD	Filter>0.6	Mean	SD
Precision [COCO]	0.87	0.22	Precision [COCO]	0.93	0.069
Recall [COCO]	0.9	0.12	Recall [COCO]	0.92	0.072
Jaccard [COCO]	0.79	0.22	Jaccard [COCO]	0.86	0.084
NME [COCO]	0.94	0.12	NME [COCO]	0.96	0.055
Num Points	26	19	Num Points	26	19
Precision [Self]	0.86	0.21	Precision [Self]	0.92	0.076
Recall [Self]	0.9	0.14	Recall [Self]	0.93	0.074
Jaccard [Self]	0.78	0.22	Jaccard [Self]	0.86	0.086
NME [Self]	0.94	0.13	NME [Self]	0.96	0.053
Area Ratio	0.063	0.089	Area Ratio	0.063	0.089

Table 1: Left: Statistics for all workers; Right: for good workers only [metric \geq 0.6]

- Both the number of tasks each worker takes on and average time in a task follows a Pareto-like, long-tail distribution, which is typical for crowdsourcing applications.
- **Data Fitting Procedure:** We are interested in figuring out what functional form these Φ functions are distributed as. We fitted the histograms against 84 different probability distribution functions², using the maximum-likelihood estimators of these distributions. Then, a Kolmogorov-Smirnov test assessed the statistical significance of whether the fitted function and the data follow the same distribution. We quantify the best fits using minimal residual sum-of-square (RSS) and the p-value resulting from the KS-test. To preserve the tails of these distributions, no filtering for selecting good workers only was done in the fitting procedure.

5.2 Overall Distribution

Overall distribution contains each of the metrics computed from all tasks submitted by all workers (N=1947). The histogram distribution (fixed bin size =50) of these metrics resembles a long-tail, exponential-decaying distribution. As shown in the best-fitting functions in Table 3, there are many pdfs that fits one metric but not another.

One particular distribution that yields the best fits for many metrics is the Johnson unbounded (SU) distribution, which we summarized in Table 4. The Johnson SU distribution is a transformed Gaussian where the data $x \mapsto \gamma + \sigma \sinh^{-1}(\frac{x-\xi}{\lambda})$, which effectively maps the typical two-parameter Gaussian to a more flexible, four parameter pdf to better account for the skewness (heavily right-skewed) and kurtosis (long-tail) of the distribution.

²Most of the functions in `scipy.stats`: [alpha, anglit, arcsine, beta, betaprime, bradford, burr, cauchy, chi, chi2, cosine, dgamma, dweibull, expon, exponpow, exponweib, f, fatiguelife, fisk, foldcauchy, foldnorm, frechet_l, frechet_r, gamma, gausshyper, genexpon, genextreme, gengamma, genhalflogistic, genlogistic, genpareto, gilbrat, gompertz, gumbel_l, gumbel_r, halfcauchy, halflogistic, halfnorm, hypsecant, invgamma, invgauss, invweibull, johnsonsb, johnsonsu, ksone, kstwobign, laplace, levy, levy_l, loggamma, logistic, loglaplace, lognorm, lomax, maxwell, mielke, nakagami, ncf, nct, ncx2, norm, pareto, pearson3, powerlaw, powerlognorm, powernorm, rayleigh, rdist, recipinvgauss, reciprocal, rice, semicircular, t, triang, truncexpon, truncnorm, tukeylambda, uniform, vonmises, vonmises_line, wald, weibull_max, weibull_min, wrapcauchy]

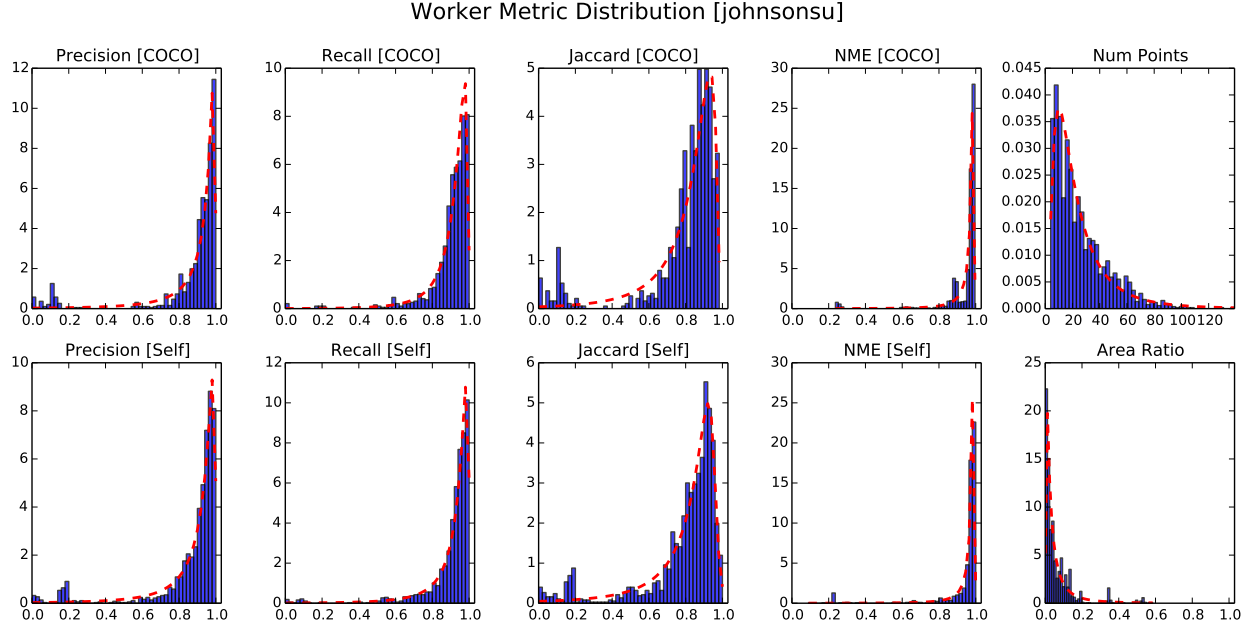


Figure 3: Normalized histogram of metric values, fitted with a Johnson SU distribution.

5.3 Worker Distributions

Recall that J_i is the set of all workers j that annotated the object-image i , we are interested in finding out how these workers are distributed in order to deduce worker quality.

In the data fitting procedure, the bin size is an important hyperparameter. When the bin size is small, the histogram is very smoothed, so many different functional forms can be fitted. Since our data is $N=40$, we pick a bin size of 30. Due to the large number of J_i distributions, we conducted the fitting procedure on a smaller candidate set of more interpretable functional forms.³ Table 5 summarizes the best functional fit for each metric.

While the Gaussian does not yield the best fit, it yields a more interpretable and may be easier for inference. Some of the non-Gaussianity may also be due to the small sample size in our preliminary experiment ($N \approx 40$ for each J_i distribution). We are interested in assessing whether it can be a “good-enough” fit based on RSS. When looking at the

5.3.1 Testing Assumption 3: Influence of task difficulty on spread

	Precision[C]	Recall[C]	Jaccard[C]	Precision[S]	Recall[S]	Jaccard[S]
Average	0.19	-0.17	0.11	0.20	0.25	0.04
Ground Truth	0.03	-0.18	-0.03	-0.07	0.02	-0.07

Table 2: Pearson’s linear correlation coefficient when compared to **Average** (average number of points in BB drawn by all worker) and **Ground Truth** (number of points in BB drawn by me). [C],[S] short for [COCO] and [Self].

³Gaussian, Johnson SU, Power law, Pareto, and t-distributions

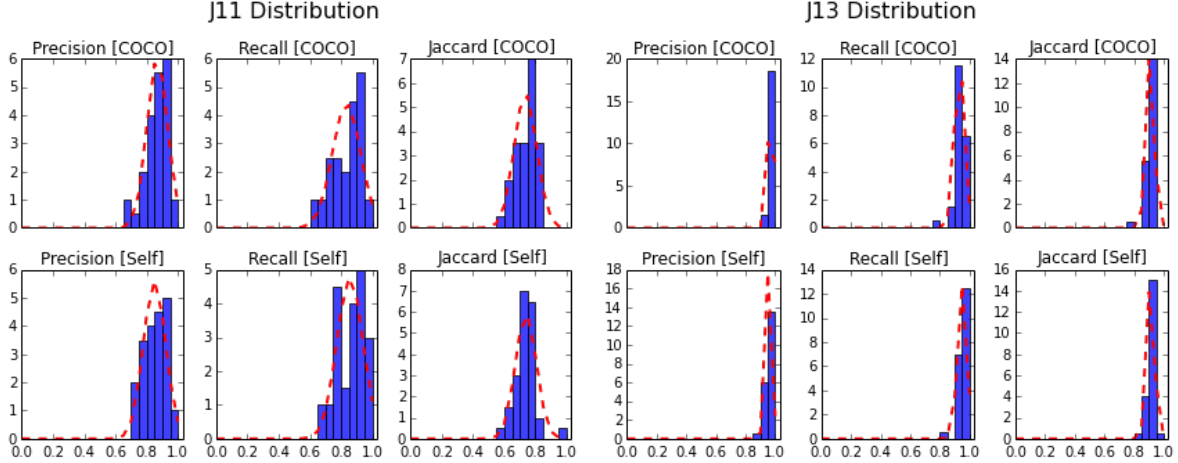


Figure 4: Example J_i distributions for a small, round-edged object (Right) and a medium-size, square-edged object (Left), qualitatively shows how task difficulty relates to width of Gaussians.

Many of the best-fitting functional form of Φ in Table 5 may make the inference problem challenging. From Fig. —. we see that the number points measure —, follows a Gaussian, since this is a measure — based only the detexterity of the worker rather than the logical errors of the BB drawn We are interested in testing our hypothesis that if work quality exhibits a large spread, that means that the object i is probably very hard to annotate. If we exclude difficult task due to task ambiguity, we hypothesize that the number of points in the image object(as a measure of boundary complexity) should depend on the standard deviation of the fitted Gaussian. While this assumption qualitatively holds true, the Pearson’s correlation coefficient and linear fits shows that there is **no linear relationship between the number of points in an object and the standard deviation of the fitted Gaussian**.

References

- [1] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5):740–755, 2014.
- [2] Sirion Vittayakorn and James Hays. Quality Assessment for Crowdsourced Object Annotations. *Proceedings of the British Machine Vision Conference 2011*, pages 109.1–109.11, 2011.
- [3] Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The Multidimensional Wisdom of Crowds. *NIPS (Conference on Neural Information Processing Systems)*, 6:1–9, 2010.

A Tables

metric	Function Name	RSS	D-value	p-value
Precision [COCO]	beta	9.11	0.48	1.02e-05
Recall [COCO]	loggamma	4.56	0.46	2.76e-05
Jaccard [COCO]	gompertz	13.3	0.46	2.76e-05
NME [COCO]	cauchy	51.7	0.84	1.25e-16
Num Points	johnsonsb	0.000239	1	2.16e-23
Precision [Self]	johnsonsu	6.83	0.34	0.00443
Recall [Self]	johnsonsu	5.49	0.42	0.000178
Jaccard [Self]	johnsonsu	4.57	0.28	0.0317
NME [Self]	cauchy	35.6	0.8	4.01e-15
Area Ratio	gengamma	27.1	0.34	0.00443

Table 3: Shapewise, the RSS is a better measure of functional fit than p-value, so we use this for evaluating the best-fitting pdf for each measure. This table summarizes the best-fitting function for each metric.

metric	RSS	D-value	p-value	ξ	λ	Shift	Scale
Precision [COCO]	10	0.36	0.0021	5.2	0.75	1	0.00011
Recall [COCO]	7	0.44	7.2e-05	5.9	1.1	1	0.00062
Jaccard [COCO]	14	0.3	0.017	5.6	1.1	0.99	0.0017
NME [COCO]	2.2e+02	0.7	1.1e-11	1.3	0.61	0.99	0.0032
Num Points	0.00044	1	2.2e-23	-6.2	1.2	0.8	0.21
Precision [Self]	6.8	0.34	0.0044	5.6	0.84	1	0.00018
Recall [Self]	5.5	0.42	0.00018	5.5	0.91	1	0.00029
Jaccard [Self]	4.6	0.28	0.032	1.6	0.95	0.96	0.039
NME [Self]	1.1e+02	0.64	7.8e-10	1.2	0.61	0.99	0.0037
Area Ratio	30	0.32	0.0089	-4.9	0.78	-0.0002	0.00012

Table 4: Johnson SU fitting coefficients.

Code is available on [Github](#). Web-app is hosted on [here](#)

Function	metric	Mean	SD	RSS	D-value	p-value
t	Precision [COCO]	1.52	0.962	14.9	0.7	2.62e-07
exponpow	Jaccard [COCO]	8.24	0.397	10.2	0.633	4.64e-06
exponpow	Precision [Self]	2.7	0.715	9.12	0.733	5.59e-08
johnsonsu	Recall [Self]	0.951	0.601	11.9	0.5	0.000616
johnsonsu	NME [Self]	0.637	0.946	142	0.5	0.000616
t	Num Points	41.3	56.9	0.00092	1	1.8e-14
johnsonsu	Area Ratio	0.888	0.459	111	0.433	0.00461
t	Recall [COCO]	1.01	0.986	6.21	0.8	2.05e-09
exponpow	Jaccard [Self]	3.85	0.886	10.3	0.9	8.38e-12
johnsonsu	NME [COCO]	0.775	0.369	117	0.733	5.59e-08

Table 5: Best functional fit (as determined by RSS) for all objects J_i distribution.