

---

# Supplement to Infinite Mixture Prototypes for Few-Shot Learning

---

Kelsey R. Allen<sup>1</sup> Evan Shelhamer<sup>\*2</sup> Hanul Shin<sup>\*1</sup> Joshua B. Tenenbaum<sup>1</sup>

## A. Appendix

### A.1. Implementation Details

For all few-shot experiments, we use the same base architecture as prototypical networks for the embedding network. It is composed of four convolutional blocks consisting of a 64-filter  $3 \times 3$  convolution, a batch normalization layer, a ReLU nonlinearity, and a  $2 \times 2$  max-pooling layer per block. This results in a 64-dimensional embedding vector for omniglot, and a 1600 dimensional embedding vector for mini-imagenet. Our models were trained via SGD with RMSProp (Tieleman & Hinton, 2012) with an  $\alpha$  parameter of 0.9.

For Omniglot, the initial learning rate was set to 1e-3, and cut by a factor of two every 2,000 iterations, starting at 4,000 iterations. Optimization is stopped at 160,000 iterations. We use gradient accumulation and accumulate gradients over eight episodes before making an update when performing 5-way training. Both  $\sigma_l$  and  $\sigma_u$  are initialized to 5.0.  $\sigma_l$  is learned jointly during training while we found learning  $\sigma_u$  on Omniglot to be unstable and so it is therefore fixed.  $\alpha$  was set to 0.1.

For mini-ImageNet, the initial learning rate was set to 1e-3, then halved every 20,000 iterations, starting at 20,000 iterations. Optimization is stopped at 100,000 iterations. Both  $\sigma_u$  and  $\sigma_l$  are initialized to 15.0 and both are learned jointly. We found that on average,  $\sigma_l$  stabilized around 12, and  $\sigma_u$  stabilized around 25.  $\alpha$  was set to  $10^{-5}$ . Clusters were still regularly created even with such a small  $\alpha$ .

### A.2. Controlling for the Number of Gradients Taken During Optimization

Consider the gradient of the loss: it has the dimensions of shot  $\times$  way because every example has a derivative with respect to every class. In this manner, by default, the

---

<sup>\*</sup>Equal contribution <sup>1</sup>Department of Brain and Cognitive Sciences, Center for Brains, Minds, and Machines (CBMM), CSAIL, MIT, Cambridge, MA <sup>2</sup>Computer Science, UC Berkeley, Berkeley, CA. Correspondence to: Kelsey R. Allen <krallen@mit.edu>.

episode size determines the number of gradients in an update. Quantitatively, 20-way episodes accumulate 16 times as many gradients as 5-way episodes. By sampling 16 5-way episodes and accumulating the gradients to make an update, we achieve significantly better results, matching the results obtained with 20-way episodes within statistical significance in most settings. Note that agreement across conditions may not be perfectly exact because subtle adjustments to hyperparameters might be necessary. See Table 1 for the quantitative results of these control experiments.

### A.3. Alternative Infinite Mixture Model Algorithms

Here we discuss two alternatives to IMP for performing inference in infinite mixture models. We will first discuss an approximation to a Gibbs sampler for estimating the MAP of a Chinese restaurant process (CRP) (Aldous, 1985). We will then discuss an expectation maximization procedure which maintains soft assignments throughout inference.

The generative model of the CRP consists of sampling assignments  $z_1, \dots, z_J$  which could take on cluster values  $c = 1, \dots, C$  from the CRP prior with hyperparameter  $\alpha$ , which controls the concentration of clusters, and number of cluster members  $N_c$ . Cluster parameters  $\mu_c$  are sampled from a base distribution  $H(\theta_0; \mu_0, \sigma_0)$ , and instances  $x_j$  are then sampled from the associated Gaussian distribution  $N(\mu_{z_j}, \sigma_{z_j})$ .  $\theta$  consists of the means  $\mu$  and sigmas  $\sigma$ .

The CRP generative model is defined as

$$p(z_{J+1} = c | z_{1:J}, \alpha) = \frac{N_c}{N + \alpha} \text{ for } c \in \{1 \dots C\} \text{ and}$$
$$p(z_{J+1} = C + 1 | z_{1:J}, \alpha) = \frac{\alpha}{N + \alpha}$$

for assignments  $z$  of examples  $x$  to clusters  $c$ , cluster counts  $N_c$ , and parameter  $\alpha$  to control assignments to new clusters.  $N$  is the total number of examples observed so far.

One popular sampling procedure for parameter estimation is Gibbs sampling (Neal, 2000). In Gibbs sampling, we draw from a conditional distribution on the cluster assignments until convergence. The conditional draws are:

$$p(z_{J+1} = c | z_{1:J}, \alpha) \propto \begin{cases} N_{c,-j} \int P(x_j | \theta) dH_{-j,c}(\theta) & \text{for } c \leq C \\ \alpha \int P(x_j | \theta) dH_0(\theta) & \text{for } c = C + 1 \end{cases} \quad (1)$$

Table 1. Results on Omniglot for different gradient accumulations. Bolded results are not significantly different from each other, showing that equalizing the number of gradients can equalize the accuracy.

SHOT	BATCH-WAY	EPISODE-WAY	5-WAY		20-WAY	
			1-SHOT	5-SHOT	1-SHOT	5-SHOT
1	20	20	<b>98.5</b>	<b>99.6</b>	<b>95.0</b>	<b>98.8</b>
1	20	5	<b>98.3</b>	<b>99.5</b>	<b>94.8</b>	<b>98.6</b>
1	5	5	97.7	<b>99.4</b>	92.1	98.0
5	20	20	<b>97.8</b>	<b>99.6</b>	<b>93.2</b>	<b>98.6</b>
5	20	5	<b>97.9</b>	<b>99.6</b>	<b>92.9</b>	<b>98.5</b>
5	5	5	96.8	<b>99.4</b>	89.8	97.7

For the case of a spherical Gaussian likelihood, let us define  $\mathcal{N}_c = \mathcal{N}(x_i; \mu_c, \sigma)$  as the likelihood of assigning  $x_i$  to cluster  $c$  and  $\mathcal{N}_0 = \mathcal{N}(x_i; \mu_0, \sigma + \sigma_0)$  as the likelihood of assigning  $x_i$  to a new cluster drawn from the base distribution (Gaussian with mean  $\mu_0$  and  $\sigma_0$ ). We can then write:

$$\begin{aligned}
 p(z_i = c | \mu) &= \frac{N_{k,-n} \mathcal{N}_c}{\alpha \mathcal{N}_0 + \sum_{j=1}^C N_{j,-n} \mathcal{N}_j} \\
 p(z_i = C + 1 | \mu) &= \frac{\alpha \mathcal{N}_0}{\alpha \mathcal{N}_0 + \sum_{j=1}^C N_{j,-n} \mathcal{N}_j} \\
 p(\sigma_c | z) &= \frac{\sigma \sigma_0}{\sigma + \sigma_0 N_c} \\
 p(\mu_c | z) &= \mathcal{N} \left( \mu_c; \frac{\sigma \mu_0 + \sigma_0 \sum_{i, z_i = c} x_i}{\sigma + \sigma_0 N_c}, \sigma_c \right)
 \end{aligned}$$

Unfortunately, because inference must be performed during every episode of our learning procedure, and there are many episodes, Gibbs sampling until convergence is impractical. We therefore use the approach from (Raykov et al., 2016) to approximate the procedure with a single pass over all data in the episode. This approximates the MAP by considering only the most probable cluster assignment for each data point, and updating cluster parameters based on these assignments. A full discussion is given in Raykov et al. (2016), and we include their method here for reference (Algorithm 1). While their method is fully-unsupervised, we employ a cross-entropy loss on the query points given the updated means and counts for the *labeled* clusters, for end-to-end optimization of classification, and initialize clusters with the class-wise means as in IMP.

Results for 5-way 1-shot Omniglot and mini-ImageNet are in Table 2. Unlabeled points are often incorrectly assigned to the labeled clusters, which both reduces the variance of that cluster, and increases its likelihood via the prior. The hard assignments lead to unstable clustering, making learning substantially more challenging.

We additionally implemented a simple expectation maximization approach (Algorithm 2). Here we maintain soft assignments  $z$  throughout, and use the updates to the cluster

means  $\mu_c$  as in (Kimura et al., 2013). Our three main differences are to: 1. include labeled points for initialization; 2. instead of having a fixed truncation parameter  $T$  for the maximum number of available clusters, we instantiate new clusters when the probability of a new cluster exceeds a certain threshold  $\epsilon$ ; 3. we do not estimate variances, as this led to very unstable results. Instead of estimating variances based on assignments, we use the same variance learning technique as IMP, which provides significant improvement. The best value of  $\alpha$  was one for which no new clusters were created in both Omniglot and mini-ImageNet.

Table 2. Ablation experiments comparing different inference schemes for infinite mixture prototypes. Accuracies are for semi-supervised 5-way 1-shot episodes, with 5 unlabeled examples per class, and 5 distractors.

METHOD	OMNIGLOT	MINI-IMAGENET
MAP-DP ( $\mu, \sigma$ )	70.0 ± 0.1	UNSTABLE
EM	95.9 ± 0.2	41.0 ± 0.6
HARD DP-MEANS	98.0 ± 0.2	45.2 ± 1.0
IMP	<b>99.0 ± 0.1</b>	<b>49.6 ± 0.6</b>

We additionally tested the hypothesis that the CRP prior was leading to worse performance by ablating it. With the prior ablated, the EM approach improves to 48.6% accuracy on mini-ImageNet, and 98.0% accuracy on Omniglot. While this is still below IMP’s performance, this gives some explanation for why the EM inference procedure fails.

The experiments in this section examine the semi-supervised 5-way 1-shot setting, with 5 unlabeled examples of each character and 5 distractor classes (see Section 4.2 of the paper for more experimental detail). In this setting, there is no effect of multi-modality in the labeled examples, and so any improvements by IMP are attributed to the way it clusters *unlabeled* data relative to these inference methods.

## References

- Aldous, D. J. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII1983*, pp. 1–198. Springer, 1985.
- Kimura, T., Tokuda, T., Nakada, Y., Nokajima, T., Matsumoto, T., and Doucet, A. Expectation-maximization algorithms for inference in dirichlet processes mixture. *Pattern Analysis and Applications*, 16(1):55–67, 2013.
- Neal, R. M. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.
- Raykov, Y. P., Boukouvalas, A., Little, M. A., et al. Simple approximate map inference for dirichlet processes mixtures. *Electronic Journal of Statistics*, 10(2):3548–3578, 2016.
- Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

**Algorithm 1** MAP-DP approach for inference.  $n_s$  is the number of labeled classes (way).  $q(i, c)$  is  $\log p(i, c)$ , the joint probability of cluster  $C$  and assignment  $i$ .  $\mathcal{N}(x; \mu, \sigma)$  is the Gaussian density.  $\alpha$  is the concentration hyperparameter of the CRP.

---

```

initialize  $\{\mu_1, \dots, \mu_{n_s}\}$  ▷ Initialize a cluster for each labeled class by taking class-wise means
initialize  $\{\sigma_1, \dots, \sigma_{n_s}\}$  ▷ Initialize cluster variances based on equation 4.
initialize  $\{z_1, \dots, z_I\}$  ▷ Initialize cluster assignments for labeled data points. All unlabeled cluster assignments start at 0.
 $C = n_s$  ▷ Initialize number of clusters  $C$ 
▷ Begin clustering pass
for each example  $i$  do
  for each cluster  $c \in \{1, \dots, C\}$  do
     $N_c \leftarrow \sum_i z_{i,c}$ 
     $\sigma_c \leftarrow \frac{\sigma\sigma_0}{\sigma + \sigma_0 N_c}$ 
     $\mu_c \leftarrow \frac{\sigma\mu_0 + \sigma_0 \sum_i z_{i,c} h_\phi(x_i)}{\sigma + \sigma_0 N_c}$ 
    estimate  $q_{i,c} \propto \log(N_{c,-i}) + \log(\mathcal{N}(x_i; \mu_c, \sigma_c))$ 
  end for
  estimate  $q_{i,C+1} \propto \log(\alpha) + \log(\mathcal{N}_0(x_i; \mu_0, \sigma_0))$ 
   $z_i \leftarrow \operatorname{argmin}(q_{i,1}, \dots, q_{i,C+1})$ 
  if  $z_i = C + 1$  then
     $C \leftarrow C + 1$ 
  end if
end for

```

---

**Algorithm 2** EM approach for inference.  $n_s$  is the number of labeled classes (way).  $q(i, c)$  is  $\log p(i, c)$ , the joint probability of cluster  $C$  and assignment  $i$ .  $\mathcal{N}(x; \mu, \sigma)$  is the Gaussian density.  $\alpha$  is the concentration hyperparameter of the CRP.  $\epsilon$  threshold for generating new cluster.

---

```

initialize  $\{\mu_1, \dots, \mu_{n_s}\}$  ▷ Initialize a cluster for each labeled class by taking class-wise means
initialize  $\{\sigma_1, \dots, \sigma_{n_s}\}$  ▷ Initialize cluster variances based on equation 4.
initialize  $\{z_1, \dots, z_I\}$  ▷ Initialize cluster assignments for labeled data points. All unlabeled cluster assignments start at 0.
 $C = n_s$  ▷ Initialize number of clusters  $C$ 
▷ Begin clustering pass
for each example  $i$  do
  for each cluster  $c \in \{1, \dots, C\}$  do
    estimate  $q_{i,c} \propto \log(N_{c,-i}) + \log(\mathcal{N}(x_i; \mu_c, \sigma_c))$ 
  end for
  estimate  $q_{i,C+1} \propto \log(\alpha) + \log(\mathcal{N}_0(x_i; \mu_0, \sigma_0))$ 
   $z_{i,c} \leftarrow \operatorname{softmax}(q_{i,1}, \dots, q_{i,C+1})$ 
  if  $z_{i,C+1} > \epsilon$  then
     $C \leftarrow C + 1$ 
     $\mu_C \sim \mathcal{N}(x_i, \mu_0, \sigma_0)$  ▷ Sample from the base distribution conditioned on the single observation  $x_i$ 
  end if
end for

```

---