Supplementary Material - Nonparametric Learning for Layered Segmentation of Natural Images

Soumya Ghosh and Erik B. Sudderth
Department of Computer Science, Brown University, Providence, RI, USA
sghosh@cs.brown.edu, sudderth@cs.brown.edu

1. Low rank Expectation Propagation

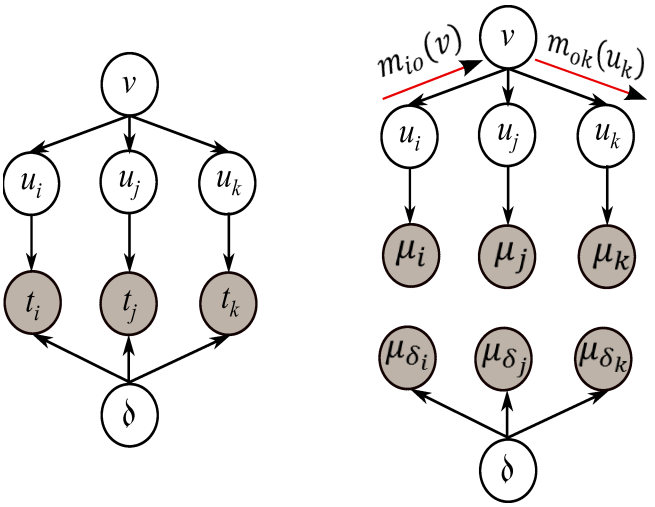

Figure 1. **True and Approximate distributions.** Graphical models representing the distribution of random variables in a layer (*We have left out the hyper-parameters on δ and v*). **Left**: True distribution. **Right**: Approximate distribution.

As previously noted, the random variables associated with each layer of our model can be treated independently of the others. Following the notation introduced in Section 3, we have

$$p(\boldsymbol{u}, \mathbf{v}, \delta \mid \boldsymbol{t}, \alpha) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I}) p(\delta|\alpha) \prod_{n=1}^{N} N(u_n|a_n^T\mathbf{v}, \psi_n)\mathbb{I}(t_n(\delta - u_n) > 0) \tag{1}$$

We approximate this distribution with a Gaussian distribution of the form:

$$q(\boldsymbol{u}, \mathbf{v}, \delta \mid \boldsymbol{t}, \alpha) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I})\mathcal{N}(\delta \mid \tilde{\mu}_p, \tilde{\sigma}_p^2) \prod_{n=1}^{N} \mathcal{N}(u_n \mid a_n^T\mathbf{v}, \psi_n)\mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2)\mathcal{N}(\delta \mid \tilde{\mu}_{\delta_n}, \tilde{\sigma}_{\delta_n}^2) \tag{2}$$

The graphical models corresponding to the true and approximate distributions are shown in Figure 1. EP proceeds by removing an approximate factor and substituting it with the corresponding true factor, giving rise to the augmented distribution. The moments of this augmented distribution are then computed and the parameters of the approximate factor is updated by matching the moments of the approximate and augmented distributions. Next, we demonstrate how these quantities are computed for our model.

Firstly, note that our approximation assumes independence between δ and $\{\mathbf{u}, \mathbf{v}\}$. From figure 1 and using standard

Gaussian BP results we have

$$q(\mathbf{v} \mid \boldsymbol{t}) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I}) \prod_{n=1}^{N} m_{no}(\mathbf{v}) \tag{3}$$

with

$$m_{no}(\mathbf{v}) \propto \mathcal{N}(\mathbf{v} \mid \boldsymbol{\tau}_{no}^{-1}\boldsymbol{\nu}_{no}, \boldsymbol{\tau}_{no}^{-1}),\ \boldsymbol{\tau}_{no} = \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n} a_n a_n^T \tag{4}$$

$$\boldsymbol{\nu}_{no} = \frac{\tilde{\nu}_n}{1+\psi_n\tilde{\tau}_n} a_n,\ \tilde{\nu}_n = \tilde{\tau}_n\tilde{\mu}_n,\ \tilde{\tau}_n = \tilde{\sigma}_n^{-2} \tag{5}$$

Thus, we have the following result

$$q(\mathbf{v} \mid \boldsymbol{t}) \propto \mathcal{N}(\mathbf{v} \mid, \boldsymbol{\tau}_{pos}^{-1}\boldsymbol{\nu}_{pos}, \boldsymbol{\tau}_{pos}^{-1}) \tag{6}$$

$$\boldsymbol{\tau}_{pos} = \boldsymbol{I} + \sum_{n=1}^{N} \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n} a_n a_n^T \tag{7}$$

$$\boldsymbol{\nu}_{pos} = \sum_{n=1}^{N} \frac{\tilde{\nu}_n}{1+\psi_n\tilde{\tau}_n} a_n \tag{8}$$

We can remove the effect of an approximate factor by dividing out the corresponding message.

$$q(\mathbf{v} \mid \boldsymbol{t}_{-n}) \propto \mathcal{N}(\mathbf{v} \mid, \boldsymbol{\tau}_{-n}^{-1}\boldsymbol{\nu}_{-n}, \boldsymbol{\tau}_{-n}^{-1}) \tag{9}$$

$$\boldsymbol{\tau}_{-n}^{-1} = (\boldsymbol{\tau}_{pos} - \boldsymbol{\tau}_{no})^{-1} \tag{10}$$

$$\boldsymbol{\nu}_{-n} = \boldsymbol{\nu}_{pos} - \boldsymbol{\nu}_{no} \tag{11}$$

Note that $\boldsymbol{\tau}_{-n}^{-1}$ can be efficiently computed using the following rank one update:

$$\boldsymbol{\tau}_{-n}^{-1} = \boldsymbol{\Sigma} - (-m)\frac{\boldsymbol{\Sigma} a_n a_n^T \boldsymbol{\Sigma}}{1 - m a_n^T \boldsymbol{\Sigma} a_n} \tag{12}$$

$$m = \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n}\ and\ \boldsymbol{\tau}_{-n}^{-1} = \boldsymbol{\Sigma} \tag{13}$$

Next observe that

$$q(u_n|\boldsymbol{t}) \propto \mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2) m_{on}(u_n) \tag{14}$$

$$q(u_n|\boldsymbol{t}_{-n}) \propto m_{on}(u_n) \tag{15}$$

$$m_{on}(u_n) \propto \mathcal{N}(u_n \mid \tau_{on}^{-1}\nu_{on}, \tau_{on}^{-1}) \tag{16}$$

A little algebra reveals that the parameters of m_{on} are given by

$$\tau_{on}^{-1} = \psi_n + a_n^T \boldsymbol{\tau}_{-n}^{-1} a_n\ and\ \tau_{on}^{-1}\nu_{on} = a_n^T \boldsymbol{\tau}_{-n}^{-1}\boldsymbol{\nu}_{-n} \tag{17}$$

Similarly, the parameters of the distribution $q(\delta \mid \boldsymbol{t}_{-n}) \propto \mathcal{N}(\delta \mid \tau_{-\delta_n}^{-1}\nu_{-\delta_n}, \tau_{-\delta_n}^{-1})$ can be computed. Finally, the moments of the following augmented distribution need to be computed:

$$q(u_n, \delta \mid \boldsymbol{t}_{-n})\mathbb{I}(t_n(\delta - u_n) > 0) = q(\delta \mid \boldsymbol{t}_{-n}) q(u_n \mid \boldsymbol{t}_{-n})\mathbb{I}(t_n(\delta - u_n) > 0) \tag{18}$$

A little bit of algebra leads to the following closed form formula for the relevant normalization constants.
Normalization constant of the augmented distribution (0^{th} order moment):

$$P = \Phi\left(\frac{t_n(\mu_{-\delta_n} - \mu_{-n})}{\sqrt{\sigma_{-n}^2 + \sigma_{-\delta_n}^2}}\right) = \Phi(h_n) \tag{19}$$

First and Second order moments for δ:

$$E[\delta] = \mu_{-\delta_n} + t_n \frac{\sigma^2_{-\delta_n} N(h_n)}{\Phi(h_n)\sqrt{\sigma^2_{-n} + \sigma^2_{-\delta_n}}} \tag{20}$$

$$E[\delta^2] = 2\mu_{-\delta_n} E[\delta] - \mu^2_{-\delta_n} + \sigma^2_{-\delta_n} - \frac{\sigma^4_{-\delta_n} h_n N(h_n)}{\Phi(h_n)(\sigma^2_{-n} + \sigma^2_{-\delta_n})} \tag{21}$$

First and Second order moments for u_n:

$$E[u_n] = \mu_{-n} - t_n \frac{\sigma^2_{-n} N(h_n)}{\Phi(h_n)\sqrt{\sigma^2_{-n} + \sigma^2_{-\delta_n}}} \tag{22}$$

$$E[u_n^2] = 2\mu_{-n} E[u_n] - \mu^2_{-n} + \sigma^2_{-n} - \frac{\sigma^4_{-n} h_n N(h_n)}{\Phi(h_n)(\sigma^2_{-n} + \sigma^2_{-\delta_n})} \tag{23}$$

where $\mu_{-n} = \tau^{-1}_{on} \nu_{on}$, $\mu_{-\delta_n} = \tau^{-1}_{-\delta_n} \nu_{-\delta_n}$, $\sigma^2_{-\delta_n} = \tau^{-1}_{-\delta_n}$, $\sigma^2_{-n} = \tau^{-1}_{on}$.

The parameters of the approximate factor corresponding to u_n can now be computed and the posterior on $\mathbf{v}$ updated using a rank one update, analogous to standard Gaussian process classification [1]. A final issue worth noting is that we have a non standard prior on δ which is difficult to deal with. We approximate the prior on δ with another Gaussian factor. The moments required for computing the parameters of this Gaussian are estimated numerically. Since, δ is an unidimensional quantity, numerical moment computation is easy and efficient. Furthermore, these moments are required only once per EP sweep, where a sweep is defined as circling through all the super-pixels. Thus the added computational cost of numerical moment computation is negligible.

1.1. Computational Complexity

Observe that we only explicitly maintain a Gaussian posterior distribution on $\mathbf{v}$ which is a D dimensional quantity. Thus, the complexity of one EP sweep is $O(ND^2)$ as opposed to standard Gaussian process classification which has a complexity of $O(N^3)$ where N is the number of super-pixels. Observe that for any candidate partition, the prior for all layers can be evaluated in parallel. Thus, the cost of running T search iterations, each iteration running t sweeps of EP is $O(tTND^2)$.

2. Likelihood Evaluation

The likelihood computation involves evaluating the independent color and texture integrals

$$\int_\Theta p(\mathbf{x}|\boldsymbol{z}, \Theta) p(\Theta|\rho) d\Theta = \int_{\theta^c} p(\mathbf{x^c}|\boldsymbol{z}, \theta^c) p(\theta^c|\rho^c) d\theta^c \int_{\theta^t} p(\mathbf{x^t}|\boldsymbol{z}, \theta^t) p(\theta^t|\rho^t) d\theta^t \tag{24}$$

which is a standard multinomial-Dirichlet integral. We provide the solution to the color integral here for the sake of completeness (*To simplify notation we denote θ^c, $\mathbf{x}^c$ by just θ and $\mathbf{x}$*).

For K segments and N super-pixels we have,

$$\int_\theta p(\mathbf{x}|\boldsymbol{z}, \theta^c) p(\theta|\rho^c) d\theta = \prod_{k=1}^{K} \int_{\theta_k} p(\theta_k|\rho^c) \prod_{n=1}^{N} p(\boldsymbol{x}_n|z_n, \theta_k)^{\mathbb{I}(z_n=k)} d\theta_k \tag{25}$$

$$= \prod_{k=1}^{K} \int_{\theta_k} \Delta(\rho^c) \prod_{w=1}^{W_c} \theta_{kw}^{\rho^c_w - 1} \prod_{n=1}^{N} \prod_{w=1}^{W_c} (\theta_{kw}^{x_{nw}})^{\mathbb{I}(z_n=k)} d\theta_k \tag{26}$$

$$= \prod_{k=1}^{K} \Delta(\rho^c) \int_{\theta_k} \prod_{w=1}^{W_c} \theta_{kw}^{\rho^c_w - 1} \prod_{w=1}^{W_c} (\theta_{kw})^{\sum_n x_{nw} \times \mathbb{I}(z_n=k)} d\theta_k \tag{27}$$

$$= \prod_{k=1}^{K} \Delta(\rho^c) \int_{\theta_k} \prod_{w=1}^{W_c} (\theta_{kw})^{x^k_w + \rho_w - 1} d\theta_k \tag{28}$$

$$= \prod_{k=1}^{K} \frac{\Delta(\rho^c)}{\Delta(\rho^c + x^k)} \tag{29}$$

In the above derivation $\Delta(\rho^c) = \frac{\Gamma(\sum_w \rho_w^c)}{\prod_w \Gamma(\rho_w^c)}$ and $x_w^k =$ number of times word w occurs with segment k. Putting it all together we have

$$\int_\Theta p(\mathbf{x}|\boldsymbol{z}, \Theta) p(\Theta|\rho) d\Theta = \prod_{k=1}^{K} \frac{\Delta(\rho^c)}{\Delta(\rho^c + x_k^{(c)})} \frac{\Delta(\rho^t)}{\Delta(\rho^t + x_k^{(t)})} \tag{30}$$

3. Search Details

In this section we provide details of our search algorithm.

3.0.1 Search Pseudo-code

Get the initial partition $\mathbf{z}^0$ using k-means.
Set maxIter = 200, $i = 1$, bestMode $= \mathbf{z}^0$
while $i \leq$ maxIter **do**
  while $p(\mathbf{z}^i \mid \mathbf{x}, \eta) \geq p(\mathbf{z}^{i-1} \mid \mathbf{x}, \eta)$ **do**
    Apply shift move to $\mathbf{z}^{i-1}$ to get $\mathbf{z}^i$
    bestMode $= \mathbf{z}^i$
    $i = i + 1$
  end while
  if $i \leq$ maxIter **then**
    Select a move from the set { Merge, Swap, Split }
    Apply the selected move to $\mathbf{z}^{i-1}$ to get $\mathbf{z}^i$
    if $p(\mathbf{z}^i \mid \mathbf{x}, \eta) \geq p(\mathbf{z}^{i-1} \mid \mathbf{x}, \eta)$ **then**
      bestMode $= \mathbf{z}^i$
    end if
    $i = i + 1$
  end if
end while
return bestMode

3.1. Shift move details

***Notation note**: z_n is a categorical random variable assuming one of K values, where K is the number of components in the partition $\mathbf{z}$. t_n on the other hand is a binary random variable indicating whether super-pixel n is assigned to layer k or not. A is a N-by-D matrix, with rows $a_1^T \ldots a_N^T$*

We are interested in optimizing $p(\mathbf{z} \mid \mathbf{x}, \eta)$ with respect to $\mathbf{z} = \{z_1, z_2 ... z_n\}$. In the shift move we assign each $z_n = \hat{k}$ such that $\hat{k} = \underset{k}{argmax}\ p(z_n = k \mid z_{-n}, \alpha, A, \Psi) p(\mathbf{x} \mid \mathbf{z}, \rho)$. Note that this implies we are optimizing $p(\mathbf{z} \mid \mathbf{x}, \eta)$ one z_n at a time.

1. for each super-pixel n
   (a) for each layer k
      i. If super-pixel n is defined for layer k; Compute the approximate posterior cavity distribution on $\mathbf{v}$; $q(\mathbf{v}|\boldsymbol{t}_{-n}) \propto \mathcal{N}(\mathbf{v}|\boldsymbol{\mu}_{-n}, \Sigma_{-n})$ and the approximate posterior cavity distribution for the layer's threshold δ_k; $q(\delta_k|\boldsymbol{t}_{-n}) = N(\delta_k|\mu_{-\delta_n}, \sigma^2_{-\delta_n})$
      ii. If super-pixel n is not defined for layer k (ie it has already been assigned to a previous layer) the posterior distributions on $\mathbf{v}$ and δ_k are themselves the cavity distributions.

iii. Next, compute the parameters of the conditional distribution $q(u_n|\mathbf{v}, \boldsymbol{t}_{-n}) = q(u_n|\mu_*, {\sigma^2}_*)$, given by

$$\mu_* = a_n^T \boldsymbol{\mu}_{-n} \tag{31}$$
$$\sigma_*^2 = \Psi_n + a_n^T \Sigma_{-n} a_n \tag{32}$$

iv. Finally, compute $\pi_{nk} = p(t_n = 1|t_{-n})$ as follows

$$\pi_{nk} = E_q[\mathbb{I}(u_n < \delta_k)] \tag{33}$$
$$= \int\int \mathbb{I}(u_n < \delta_k) N(u_n|\mu_*, \sigma_*^2) N(\delta_k|\mu_{-\delta_n}, \sigma^2_{-\delta_n}) du_n d\delta_k \tag{34}$$
$$= \Phi\left(\frac{\mu_{-\delta_n} - \mu_*}{\sqrt{\sigma_*^2 + \sigma^2_{-\delta_n}}}\right) \tag{35}$$

v. The probability of super-pixel n getting assigned to layer k is given by

$$p(z_n = k \mid z_{-n}) = p(u_n < \delta_k \mid u_n > \delta_l) = \pi_{nk} \prod_l (1 - \pi_{nl});\ with\ l = 1...k-1 \tag{36}$$

vi. Compute the posterior probability of the super-pixel assignment

$$p(\mathbf{z} \mid \mathbf{x}, \rho, \alpha) \propto p(z_n = k \mid z_{-n}) \int p(\mathbf{x} \mid \mathbf{z}, \theta) p(\theta \mid \rho) d\theta \tag{37}$$

(b) Finally, assign n to layer $\hat{k}$ which maximizes posterior probability

$$\hat{k} = \underset{k}{argmax}\ p(z_n = k \mid z_{-n}) \int p(\mathbf{x} \mid \mathbf{z}, \theta) p(\theta \mid \rho) d\theta \tag{38}$$

(c) For all layers affected by the shift of super-pixel n, update the corresponding posterior distribution on $\mathbf{v}$ by a EP projection for the relevant super-pixel. Care is taken such that when a previously invalid super-pixel gets shifted into a layer, the old posterior is treated as the new cavity distribution. Likewise when a super-pixel is shifted out of a layer, the old cavity distribution is treated as the new posterior.

4. Probability to Correlation mapping details

Covariance Calibration. We are interested in estimating a mapping between the correlation (c) of a pair of Gaussian random variables (u_i and u_j), and the conditionally learned probability q_{ij} of the corresponding super-pixels i and j being assigned to the same layer. According to our generative model, two super-pixels i and j can be assigned to the same layer k iff both u_i and u_j are less than the threshold δ_k. Hence, the probability of two super-pixels being assigned to layer k is

$$p_-|\delta_k = \int_{-\infty}^{\delta_k}\int_{-\infty}^{\delta_k} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}\right) du_i du_j \tag{39}$$

Furthermore, we can marginalize out the unknown thresholds δ_k

$$q_-^k(\alpha, \rho) = \int_{-\infty}^{\infty}\int_{-\infty}^{\delta_k}\int_{-\infty}^{\delta_k} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}\right) p(\delta_k|\alpha) du_i du_j d\delta_k \tag{40}$$

Let us further define

$$q_+^k(\alpha, \rho) = \int_{-\infty}^{\infty}\int_{\delta_k}^{\infty}\int_{\delta_k}^{\infty} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}\right) p(\delta_k|\alpha) du_i du_j d\delta_k \tag{41}$$

which is the probability that both u_i and u_j are greater than the δ_k. Note that neither q_- nor q_+ can be computed in closed form and are both numerically approximated.

Now observe that two super-pixels i and j can be assigned to the same layer, if they are both assigned to the first layer or if neither is assigned to the first layer but both are assigned to the second layer or if neither is assigned to the first two layers but both are assigned to the third layer and so on. We can thus express p_{ij} as

$$q_{ij} = q_-^1(\alpha,\rho) + q_-^2(\alpha,\rho)q_+^1(\alpha,\rho) + q_-^3(\alpha,\rho)q_+^1(\alpha,\rho)q_+^2(\alpha,\rho) + \dots \quad (42)$$

$$\approx \sum_{k=1}^{K} q_-^k(\alpha,\rho) \prod_{l=1}^{K-1} q_+^l(\alpha,\rho) \quad (43)$$

where we have explicitly truncated our model to have K (some large number) layers. The above equation defines the sought relationship and allows us to map conditionally learned q_{ij} to pairwise correlations of Gaussian random variables. The mapping is visualized in figure 2.

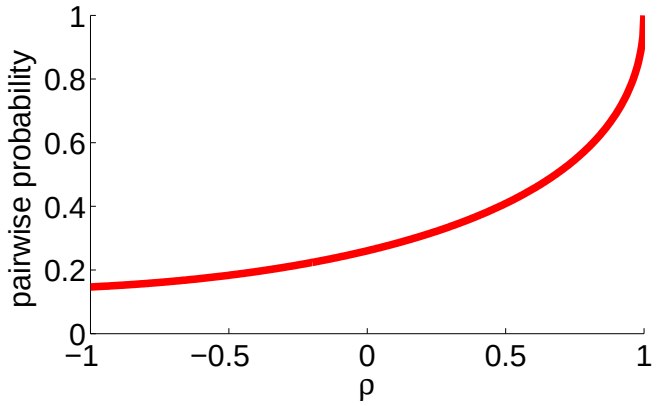

Figure 2. Mapping between correlation coefficients and pairwise probabilities

5. Experimental Details

This section provides further details about our experimental setup. Recall that we tune parameters of competing methods by performing a grid search over parameter values.

Mean shift has three tunable parameters, region band width (rbw), spatial bandwidth (spbw) and minimum region size (min), FH also has three parameters, a Gaussian kernel bandwidth σ, a scale parameter K and a minimum region size parameter(min). Normalized cuts and gPb have just one tunable parameter each – the number of segments (N) and a scale parameter (S) respectively. The parameters for all methods were tuned on the BSDS300 training set by performing a grid search and selecting values which jointly optimize rand index and segCover. The grid search was conducted over the following parameter values – a) MS - rbw= $\{1, 3, 6, 10, 15, 25\}$, spbw= $\{1, 3, 7, 10, 15, 25\}$, min= $\{500, 1000, 2000, 3000, 4000, 8000\}$ b) FH - $\sigma = \{0.2, 0.5, 0.8, 1.0\}$, K and $min = \{50, 100, 5001000\}$ c) Ncuts - $N = \{3, 5, 10, 15, 20, 25\}$ d) gPb - $S = \{0.05, 0.1, 0.13, 0.15, 0.2, 0.25, 0.3, 0.4\}$.

5.1. Additional Results

This section provides additional qualitative results. Figure 3 displays diverse modes found by the search procedure while figure 4 provides various examples of MAP partitions preferred by our model.

References

[1] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. 3

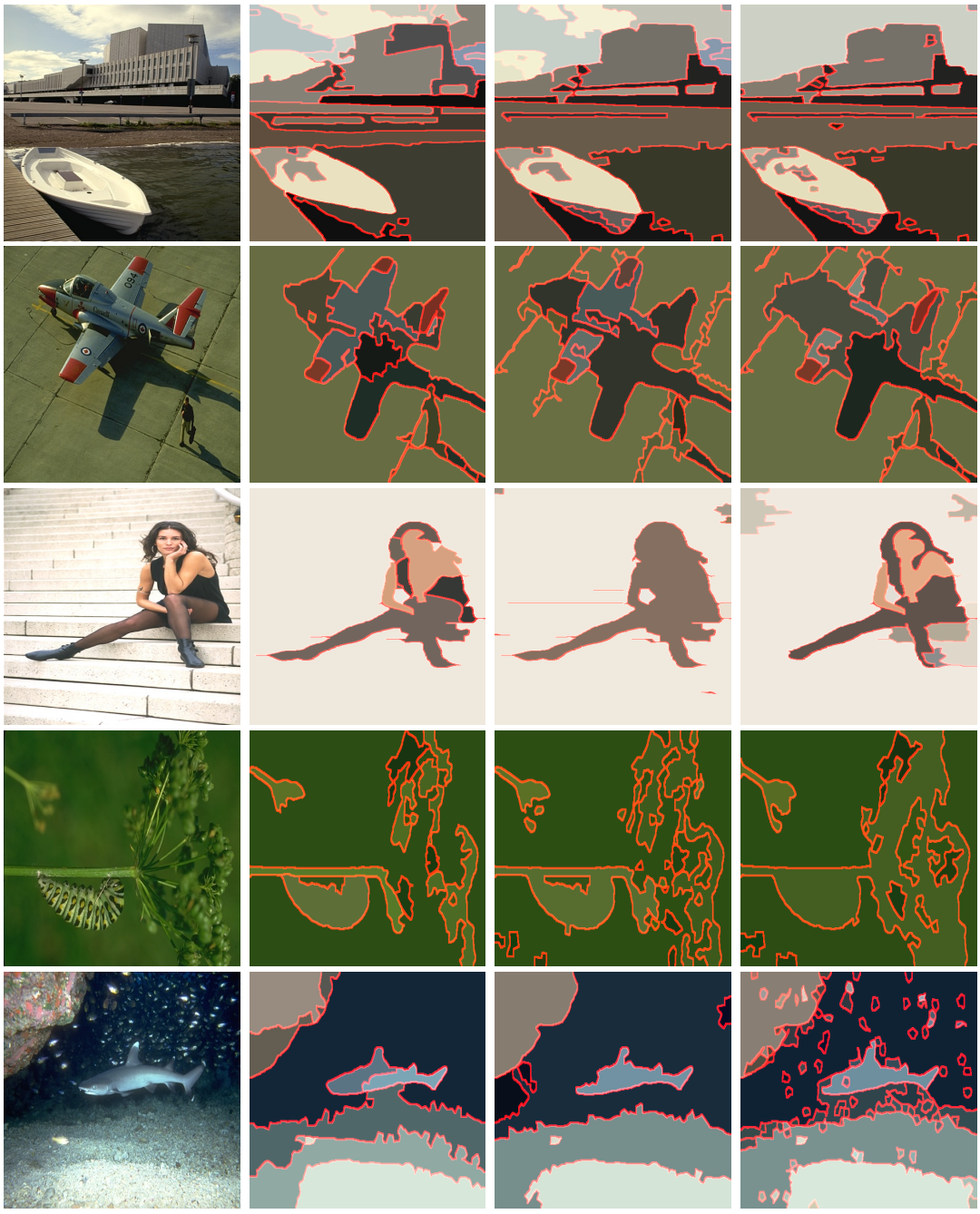

Figure 3. **Diverse Segmentations.** Diverse Segmentations discovered by our proposed algorithm. Each row depicts multiple partitions for a given image. Partitions in the second column are the MAP estimates. Other partitions with significant probability masses are shown in the third and fourth columns.

Figure 4. **Example Segmentations.** MAP partitions pf a random subset of LabelMe and BSDS images.