Image Segmentation Based on Bethe Approximation for Gaussian Mixture Model

Fan CHEN, Kazuyuki TANAKA and Tsuyoshi HORIGUCHI*

Graduate School of Information Sciences, Department of Computer and Mathematical Sciences, Tohoku University, Aramaki-aza-Aoba 04, Aoba-ku, Sendai 980-8579, Japan

Received August 20, 2004; final version accepted October 22, 2004

We propose an image segmentation algorithm under an expectation-maximum scheme using a Bethe approximation. In the stochastic image processing, the image data is usually modeled in terms of Markov random fields, which can be characterized by a Gibbs distribution. The Bethe approximation, which takes account of nearest-neighbor correlations, provides us with a better approximation to the Gibbs free energy than the commonly used mean-field approximation. We apply the Bethe approximation to the image segmentation problem and investigate its efficiency by numerical experiments. As a result, our approach shows better robustness and faster converging speed than those using the mean-field approximation.

KEYWORDS: image segmentation, Bethe approximation, EM algorithm, Gaussian mixture model

1. Introduction

As a common preprocess of higher-level image processing such as image understanding and pattern recognition, an image segmentation attracts much interest of many researchers. Various approaches for the image segmentation have been proposed; see [1–3] as reviews. According to different features used in those approaches so far, they can be classified roughly as a histogram-based approach [4], an edge-based approach [5], a region-based approach [6] and other types of approaches [7]. Although the histogram-based approach is easy to implement, it fails to get continuous regions in some cases due to the lack of consideration on spatial information, and does not work well when there are no obvious peaks that represent different classes in the histogram. The edge-based approach is similar to the way in which human perceives objects. However, the performance of edge-based approach is rather sensitive to the noise in the image and has a difficulty when we deal with natural images which always present textured regions. The region-based approach is more robust to the noise, and provides a segmentation with continuous regions in terms of a well-defined criterion for regional homogeneity in the cost of more expensive computational time and memory than other approaches.

In the region-based approach, we regard the given image as consisting of homogenous regions, which should be constant in intensity but actually have fluctuations for some reasons. This approach is divided into several methods, such as the region growing method [8], the region splitting and merging method [9], and the clustering methods and so on. In fact, the clustering method based solely on intensity does not give satisfiable results in general. Stochastic methods provide us with a soft way to embed the spatial information into the segmentation process [10]. In the stochastic image segmentation, correct decision of the class number, proper selection of the inference criterion and accurate description of the spatial information are some of key factors that affect the performance. Although some unsupervised model-selection approaches [11, 12] have been proposed to determine the number of classes, but most of them are computationally heavy.

The Maximum a Posteriori (MAP) and the Maximum Posterior Marginal (MPM) are two commonly used criteria in the Bayesian image processing. The former focuses on the maximization of the whole posterior probability while the later tries to achieve the maximum posterior probability on each pixel. In this sense, the MPM works better in reducing the number of misclassified pixels than the MAP [18] and is thought to be suitable for the image segmentation in the present paper.

A method by using the Markov Random Field (MRF) is a natural way to incorporate spatial correlations, in which the local characteristics of an image are specified in a way that the probability for each pixel depends only on its neighbors [13]. Especially, the blob-like homogenous regions are described using an isotropic MRF [14]. According to the Hammersley–Clifford theorem [15], the joint distribution of MRF is equivalent to a Gibbs distribution. Since the MRF is usually intractable for random systems, a Gibbs sampler combined with a simulated annealing scheme [13] was firstly introduced to solve the image processing problem, which is actually rather computationally burdensome. Later, an Expectation-Maximization (EM) algorithm [19] was applied to the image segmentation. Here the EM algorithm aims to find a local optimum from a data set with incomplete data and consists of two steps: in the E-step, expectation...

* E-mail: tsuyoshi@statp.is.tohoku.ac.jp
values of incomplete data are calculated while in the M-step hyperparameters are updated by maximizing a specified criterion. Comer and Delp [20] proposed an EM/MPM algorithm, in which MPM and EM algorithm are two separated processes, and a Gibbs sampler was used in the MPM to generate new states.

In order to release the computational burden, a mean-field approximation of statistical physics is used and an EM/MPM algorithm using the mean-field approximation was proposed by Zhang [16]. Since then, a lot of works based on the mean-field approximation have been proposed. For example, using the idea of the mean-field approximation, Celeux et al. [17] proposed a series of EM/MPA algorithms, which includes a mean EM/MPA algorithm, a mode EM/MPA algorithm, and a simulated EM/MPA algorithm. In a latter section, we use the simulated EM/MPA, which has the best performance among those three, for a comparison with results obtained in the present paper.

A Bethe approximation [21, 22] is generally a better approximation than the mean-field one since it takes nearest-neighbor correlations between effective fields and bonds into account. In the review of Tanaka [24], the EM/MPM approach for the binary image restoration under the Bethe approximation was explained in detail. Here we consider a Hamiltonian for the image segmentation in a more general case and try to derive the iterative equations. With the update rules by the Bethe approximation, we show that a better performance is achieved for the proposed image segmentation problem.

In Section 2, we formulate the image segmentation problem by using a Gaussian Mixture Model (GMM) and give the EM updating scheme. In Section 3, the Bethe approximation is applied to the image segmentation problem and updating rules for optimal hyper-parameters are derived. We give some comments on the extension to color images in Section 4. Numerical experiments and discussions are made in Section 5. We give a conclusion of our work in Section 6.

2. EM/MPM Algorithm for Stochastic Image Segmentation

2.1 Formulation for EM/MPM estimation

Suppose an image composed of $K$ classes on a lattice, denoted by $\Lambda$,

$$\Lambda = \{i \mid 1 \leq i \leq N\},$$

(1)

where $N$ is the number of all pixels in the given image and $K$ is the number of classes. The configuration of intensity space is expressed by $Y = \{y_i \mid i \in \Lambda, y_i \in \{0, 1, \ldots, 255\}\}$. A label space for the classes, $Z = \{z_i \mid i \in \Lambda, z_i \in \{1, 2, \ldots, K\}\}$, which denotes to which class the current pixel belongs, is assumed to be determined from the image. Hence the segmentation of the given image becomes a problem to recover this label space out of the corresponding intensity space.

In order to take account of the homogeneity inside a region, we need to introduce proper spatial correlations. To reduce the computational complexity, it is natural and reasonable to assume that the state of a pixel depends only on those at its neighbors. Thus we also assume that the label space is a Markov random field. According to the Hammersley–Clifford theorem, $P(Z \mid \beta)$ is described using a Gibbs distribution, i.e.,

$$P(Z \mid \beta) = \frac{\exp(-\beta H^\text{prior}_Z)}{\sum_{\{z\}} \exp(-\beta H^\text{prior}_Z)},$$

(2)

where in our case we put

$$H^\text{Prior}_Z = - \sum_{(ij)} f_{ij}(z_i, z_j).$$

(3)

Here $\beta$ is the inverse temperature of the prior Gibbs distribution, and $f_{ij}(a, b)$ is a function that describes the interaction between classes at pixel $i$ and pixel $j$. In Eq. (3), $(ij)$ denotes that the summation is to be taken over nearest neighbor pairs of pixels $i$ and $j$. Especially, we assume the symmetry for $f_{ij}(a, b)$ with respect to $a$ and $b$: $f_{ij}(a, b) = f_{ij}(b, a)$.

If we assume that $y_i$ is conditionally distributed on $z_i$, we have

$$P(Y \mid Z, \theta) = \prod_i P(y_i \mid z_i, \theta) = \exp\left[\sum_i g(z_i, y_i, \theta)\right],$$

(4)

where $\theta$ is a vector of hyper-parameters and $g(z_i, y_i, \theta)$ is a function that describes the effect $z_i$ receives from $y_i$ under the hyper-parameter $\theta$.

We derive the $a$ posteriori probability $P(X \mid Y, \theta, \beta)$ by using the Bayesian theorem:

$$P(Z \mid Y, \theta, \beta) \propto P(Y \mid Z, \theta)P(Z \mid \beta).$$

(5)

Using Eqs. (2) and (4), the $a$ posteriori probability takes the following form:
\begin{align}
P(Z \mid Y, \theta, \beta) &= \exp(-\beta \mathcal{H}(Z \mid Y, \theta, \beta)) \frac{1}{\sum_{\{z\}} \exp(-\beta \mathcal{H}(Z \mid Y, \theta, \beta))}, \tag{6} \\
\mathcal{H}(Z \mid Y, \theta, \beta) &= - \sum_{\{y\}} f_0(\bar{z}, \bar{z}_i) - \frac{1}{\beta} \sum_{i} g(\bar{z}_i, y_i, \theta) \tag{7} \\
&= - \sum_{\{y\}} \sum_{k=1}^{K} \sum_{k'=1}^{K} f_0(k, k') \delta_{\bar{z}_i, \bar{z}_{i,k}} - \frac{1}{\beta} \sum_{i} \sum_{k=1}^{K} g(k, y_i, \theta) \delta_{\bar{z}_{i,k}}. \tag{8}
\end{align}

The Hamiltonian \( \mathcal{H}(Z \mid Y, \theta, \beta) \) given above represents a rather general setting for the image segmentation. In our following numerical experiments, the prior of \( Z \) is considered to be a Potts model, in which two nearest pixels tend to take the same label value. As we have mentioned in Section 1, the given image is thought to consist of such regions that have fluctuations around different mean intensity values; the image can be well modeled by a mixture model with a finite number of mean intensities. Here, the intensity space \( Y \) is assumed to be described by Gaussian Mixture Model (GMM), in which the intensity of a pixel is only conditionally Gaussian distributed on its corresponding label value and the intensities at any two pixels are independent from each other. Namely, we have

\begin{align}
f_0(k, k') &= \delta_{k,k'}, \tag{9} \\
g(k, y_i, \theta) &= - \log \left( \frac{\sigma_k}{\sigma_i} \alpha_k \right) - \frac{1}{\sigma_k^2} (y_i - \mu_k)^2, \tag{10}
\end{align}

where \( \delta_{a,b} \) is the kronecker delta and \( \theta = (\mu_1, \sigma_1, \mu_2, \sigma_2, \ldots, \mu_K, \sigma_K) \); \( \mu_k \) and \( \sigma_k \) are the mean and variance of class \( k \), respectively.

The estimation of an optimal label space \( Z \) is usually done by maximizing the posterior probability of \( Z \) over observed \( Y \) and hyper-parameters \( (\theta, \beta) \), i.e., \( P(Z \mid Y, \theta, \beta) \), which is expressed by a Gibbs canonical distribution with the prior assumed as the MRF. We call this criterion the Maximum a Posteriori (MAP), i.e., \( Z^* = \arg \max_{Z} P(Z \mid Y, \theta, \beta) \). In statistical physics, the configuration of largest posterior probability is determined by minimizing the following Gibbs free energy:

\begin{align}
\mathcal{F}(P) &= \sum_{\{z\}} \mathcal{H}(Z \mid Y, \theta, \beta)P(Z \mid Y, \theta, \beta) - \frac{1}{\beta} S(P) \\
&= - \sum_{\{y\}} \sum_{k=1}^{K} \sum_{k'=1}^{K} f_0(k, k') \delta_{\bar{z}_i, \bar{z}_{i,k}} - \frac{1}{\beta} \sum_{i} \sum_{k=1}^{K} g(k, y_i, \theta) \delta_{\bar{z}_{i,k}} - \frac{1}{\beta} S(P), \tag{11}
\end{align}

where \( S(P) \) is the corresponding entropy given by

\begin{align}
S(P) &= - \sum_{\{z\}} P(Z \mid Y, \theta, \beta) \log P(Z \mid Y, \theta, \beta), \tag{12}
\end{align}

and we put \( \langle Q \mid Y, \theta, \beta \rangle \equiv \sum_{Z} P(Z \mid Y, \theta, \beta)Q \) for a quantity \( Q \). See [23] for the Gibbs free energy.

Here we use the Maximum Posterior Marginal (MPM) as the criterion for the estimation of label space. Let the marginal posterior probability of pixel \( z_i \) over \( Y \) and hyper-parameters be \( P(z_i \mid Y, \theta, \beta) \) defined as follows:

\begin{align}
P(z_i \mid Y, \theta, \beta) &= \prod_{k=1}^{K} P(z_i = k \mid Y, \theta, \beta)^{\delta_{z_{i,k}}}, \tag{13}
\end{align}

where

\begin{align}
P(z_i = k \mid Y, \theta, \beta) &= \sum_{\{z\}} P(Z \mid Y, \theta, \beta) \delta_{z_{i,k}} = \langle \delta_{z_{i,k}} \mid Y, \theta, \beta \rangle. \tag{14}
\end{align}

The MPM criterion is given as

\begin{align}
z_i^* &= \arg \max_k \sum_{\{z\}} P(Z \mid Y, \theta, \beta) \delta_{z_{i,k}} = \arg \max_k \langle \delta_{z_{i,k}} \mid Y, \theta, \beta \rangle. \tag{15}
\end{align}

The MAP estimation is usually implemented by some global stochastic maximization techniques such as the simulated annealing, while the MPM estimation tries to get the optimal value for each pixel and thus allows different decoupling ways to approximate the segmentation process, such as the mean-field approximation, the Bethe approximation and so on. On the other hand, since a better image segmentation should minimize the number of misclassified pixels, the MPM suggests better performance by reducing the misclassified probability at each pixel. In the MPM estimation described here, the label value of each pixel is done by minimizing the above free energy according to its marginal posterior probability.

The optimal solutions for \( \theta \) and \( \beta \) are estimated by maximizing \( \log P(Y \mid \theta, \beta) \):
\[ (\theta^*, \beta^*) = \arg \max_{(\theta, \beta)} \log P(Y | \theta, \beta), \]  

(16)

where \( P(Y | \theta, \beta) \equiv \sum_{(\omega, \rho)} P(Y, Z | \theta, \beta) \) is the marginal likelihood of \((\theta, \beta)\) on a currently given intensity space \( Y \). In the marginal likelihood approach, the observed intensity space \( Y \) is regarded as a suitably generated sample by the label space \( Z \) under certain hyper-parameters \((\theta, \beta)\). By maximizing this likelihood, we achieve an optimal hyper-parameter estimation on a limited set of samples, \( Y \).

Due to the existence of incomplete data, we use EM algorithm to find an optimal label configuration. The whole process is iterated until a convergence condition is met. We summarize the algorithm and update rules as follows; a detailed review of the EM/MPM scheme can be found in [24].

1. **E-Step**: We calculate the following expectation values \( \langle \delta_{z,k} | Y, \theta, \beta \rangle \) and \( \langle \delta_{z,k} \delta_{l,j} | Y, \theta, \beta \rangle \).

2. **M-Step**: All parameters are estimated and label space is updated by the following rules:

\[
\begin{align*}
(\theta^*, \beta^*) &= \arg \max_{(\theta, \beta)} \log P(Y | \theta, \beta), \\
\langle \delta_{z,k} | Y, \theta, \beta \rangle^* &= \arg \min_{\langle \delta_{z,k} | Y, \theta, \beta \rangle} \mathcal{F}(P), \\
\langle \delta_{z,k} \delta_{l,j} | Y, \theta, \beta \rangle^* &= \arg \min_{\langle \delta_{z,k} \delta_{l,j} | Y, \theta, \beta \rangle} \mathcal{F}(P).
\end{align*}
\]

2.2 EM/MPM algorithm by mean-field approximation

Before entering into Section 3, we make a short review on a mean-field approximation. As the first order approximation of cluster variation method, the mean-field approximation takes a simple form and gives us a good approximation with low computational complexity in certain cases. The basic idea is to treat two-body interactions approximately by neglecting the fluctuations from the mean value. As a result, the label space behaves as being composed of independent variables on each pixel. Accordingly, the posterior probability of \( Z \) over \( Y \) under the mean-field approximation reads \( P_{\text{MF}}(Z | Y, \theta, \beta) = \prod_{i} P_{\text{MF}}(z_i | Y, \theta, \beta) \). To search for the optimal value of \( \langle \delta_{z,k} | Y, \theta, \beta \rangle_{\text{MF}} \) that minimizes the free energy, we differentiate the free energy with respect to \( \langle \delta_{z,k} | Y, \theta, \beta \rangle_{\text{MF}} \) as follows:

\[
\frac{\partial}{\partial \langle \delta_{z,k} | Y, \theta, \beta \rangle_{\text{MF}}} \mathcal{F}_{\text{MF}}(P) = 0,
\]

(17)

where \( \mathcal{F}_{\text{MF}}(P) \) is the mean-field energy defined by

\[
\mathcal{F}_{\text{MF}}(P) = -\sum_{ij}^{K} \sum_{k=1}^{K} f_{ij}(k, k') \langle \delta_{z,k} \delta_{z,k'} | Y, \theta, \beta \rangle_{\text{MF}}
- \frac{1}{\beta} \sum_{i}^{K} g_i(k, y_i, \theta) \langle \delta_{z,k} | Y, \theta, \beta \rangle_{\text{MF}}
+ \frac{1}{\beta} \sum_{i}^{K} \sum_{l}^{K} \langle \delta_{z,k} \delta_{l,j} | Y, \theta, \beta \rangle_{\text{MF}} \log \{ \langle \delta_{z,k} \delta_{l,j} | Y, \theta, \beta \rangle_{\text{MF}} \}.
\]

(18)

The mean-field approximation was first applied to image segmentation by Zhang and a detailed derivation can be found in [16]. We give some results by using the mean-field approximation here.

\[
\langle \delta_{z,k} | Y, \theta, \beta \rangle_{\text{MF}} = \exp \left[ \beta \sum_{j \in N_i} \sum_{k=1}^{K} f_{ij}(k, k') \langle \delta_{z,k} \delta_{z,k'} | Y, \theta, \beta \rangle_{\text{MF}} + g_i(k, y_i, \theta) \right]
\sum_{l=1}^{K} \exp \left[ \beta \sum_{j \in N_l} \sum_{k=1}^{K} f_{ij}(l, k') \langle \delta_{z,k} \delta_{l,j} | Y, \theta, \beta \rangle_{\text{MF}} + g_i(l, y_i, \theta) \right],
\]

(19)

and

\[
\langle \delta_{z,k} \delta_{l,j} | Y, \theta, \beta \rangle_{\text{MF}} = \langle \delta_{z,k} | Y, \theta, \beta \rangle_{\text{MF}} \langle \delta_{l,j} | Y, \theta, \beta \rangle_{\text{MF}},
\]

(20)

where we use \( N_i \) to denote the set of nearest neighbors to pixel \( i \).

3. EM/MPM Algorithm by Bethe Approximation

By taking nearest-neighbor correlations between effective fields and bonds into account, the Bethe approximation usually brings us with better results than the mean-field approximation. Here we extend the Bethe approximation to a general case and apply it to our segmentation problem.

Under the Bethe approximation, the entropy \( S(P) \) is defined as:
\[ S(P) = \sum_i S_i + \sum_{(ij)} (S_{ij} - S_i - S_j), \]  
\( (21) \)
in which the second term compensates for the entropy of pairs and thus the fluctuations of pairs are included. For the square lattice, it is equal to

\[ S(P) = -3 \sum_i S_i + \sum_{(ij)} S_{ij}. \]  
\( (22) \)

Thus the free energy under the Bethe approximation is given by

\[ \mathcal{F}(P) = -\frac{1}{\beta} \sum_i \sum_{k=1}^{K} g_i(k, y_i, \theta) (\delta_{z_i,k} | Y, \theta, \beta) - \sum_{(ij)} \sum_{k=1}^{K} \sum_{k'=1}^{K} f_{ij}(k,k') (\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta) \]
\[ - \frac{3}{\beta} \sum_i \sum_{k=1}^{K} (\delta_{z_i,k} | Y, \theta, \beta) \log((\delta_{z_i,k} | Y, \theta, \beta)) \]
\[ + \frac{1}{\beta} \sum_{(ij)} \sum_{k=1}^{K} \sum_{k'=1}^{K} (\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta) \log((\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta)). \]  
\( (23) \)

In order to minimize the free energy under the following constraints:

\[ \sum_{k=1}^{K} (\delta_{z_i,k} | Y, \theta, \beta) = 1, \]  
\( (24) \)
\[ \sum_{k=1}^{K} \sum_{k'=1}^{K} (\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta) = 1, \]  
\( (25) \)
\[ \sum_{k=1}^{K} (\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta) = (\delta_{z_i,k} | Y, \theta, \beta), \]  
\( (26) \)

we define the following functional with Lagrange terms:

\[ \mathcal{L} = \beta \mathcal{F}(P) + \sum_i \eta_i \left[ \sum_{k=1}^{K} (\delta_{z_i,k} | Y, \theta, \beta) - 1 \right] + \sum_{(ij)} v_{ij} \left[ \sum_{k=1}^{K} \sum_{k'=1}^{K} (\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta) - 1 \right] \]
\[ + \sum_{(ij)} \sum_{k=1}^{K} \sum_{k'=1}^{K} l_{ij}^{k'} \left[ (\delta_{z_i,k} | Y, \theta, \beta) - \sum_{k=1}^{K} (\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta) \right]. \]  
\( (27) \)

where \( \{\eta_i\}, \{v_{ij}\}, \{l_{ij}^{k'}\} \) are Lagrange multipliers. By taking variation of \( \mathcal{L} \) with respect to \( \langle \delta_{z_i,k} | Y, \theta, \beta \rangle \) and \( \langle \delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta \rangle \), respectively, we find the following equations:

\[ -g_i(k, y_i, \theta) - 3[\log(\delta_{z_i,k} | Y, \theta, \beta) + 1] + \sum_{j \in N_i} \lambda_{ij}^{k'} + \eta_i = 0, \]  
\( (28) \)
\[ -\beta f_{ij}(k,k') + \log(\delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta) + 1 + v_{ij} - \lambda_{ij}^{k} - \lambda_{ij}^{k'} = 0. \]  
\( (29) \)

By solving these equations, we get

\[ \langle \delta_{z_i,k} | Y, \theta, \beta \rangle = \frac{\exp \left[ \frac{1}{3} \left( -g_i(k, y_i, \theta) + \sum_{j \in N_i} \lambda_{ij}^{k'} \right) \right]}{\sum_{l=1}^{K} \exp \left[ \frac{1}{3} \left( -g_l(l, y_i, \theta) + \sum_{j \in N_i} \lambda_{ij}^{k'} \right) \right]}, \]  
\( (30) \)
\[ \langle \delta_{z_i,k}\delta_{z_j,k'} | Y, \theta, \beta \rangle = \frac{\exp[\beta f_{ij}(k,k') + \lambda_{ij}^{k} + \lambda_{ij}^{k'}]}{\sum_{l=1}^{K} \sum_{l'=1}^{K} \exp[\beta f_{ij}(l,l') + \lambda_{ij}^{l} + \lambda_{ij}^{l'}]}. \]  
\( (31) \)

Introducing \( \phi_{ij}^k \) by using \( \lambda_{ij}^{k'} = \sum_{l \in N_i \setminus \{j\}} \phi_{ij}^l + g_i(k, y_i, \theta) \), we obtain the following expressions from Eqs. (30) and (31):

\[ \langle \delta_{z_i,k} | Y, \theta, \beta \rangle = \frac{\exp \left[ g_i(k, y_i, \theta) + \sum_{j \in N_i} \phi_{ij}^l \right]}{\sum_{l=1}^{K} \exp \left[ g_l(l, y_i, \theta) + \sum_{j \in N_i} \phi_{ij}^l \right]}, \]  
\( (32) \)
\begin{equation}
(\delta_{z,k} \mid Y, \theta, \beta) = \exp \left[ \beta_{ij}(k, k') + g_i(k', y, \theta) + \sum_{r \in N_{(j)}} \phi_{jr} + \sum_{r \in N_{(j)}} \phi_{jr}^l \right] \cdot \frac{\sum_{l=1}^{K} \sum_{r \in N_{(j)}} \exp \left[ \beta_{ij}(l, l') + g_j(l, y, \theta) + g_{ij}(l', y, \theta) + \sum_{r \in N_{(j)}} \phi_{jr} + \sum_{r \in N_{(j)}} \phi_{jr}^l \right]}{\sum_{l=1}^{K} \sum_{r \in N_{(j)}} \exp \left[ \beta_{ij}(l, l') + g_j(l, y, \theta) + g_{ij}(l', y, \theta) + \sum_{r \in N_{(j)}} \phi_{jr} + \sum_{r \in N_{(j)}} \phi_{jr}^l \right]} .
\end{equation}

By using the constraint (26), we obtain the following relation:
\begin{equation}
\exp(\phi_{ij}) = \frac{\sum_{l=1}^{K} \exp \left[ g_j(l, y, \theta) + \sum_{r \in N_{(j)}} \phi_{jr} \right] \sum_{l=1}^{K} \exp \left[ \beta_{ij}(l, l') + g_j(l', y, \theta) + \sum_{r \in N_{(j)}} \phi_{jr} + \sum_{r \in N_{(j)}} \phi_{jr}^l \right]}{\sum_{l=1}^{K} \sum_{r \in N_{(j)}} \exp \left[ \beta_{ij}(l, l') + g_j(l, y, \theta) + g_{ij}(l', y, \theta) + \sum_{r \in N_{(j)}} \phi_{jr} + \sum_{r \in N_{(j)}} \phi_{jr}^l \right]} .
\end{equation}

Now by putting \( \rho_{ij}^k = \exp(\phi_{ij}^k) / \sum_{l=1}^{K} \exp(\phi_{ij}^l) \), we have from Eq. (34)
\begin{equation}
\rho_{ij}^k = \frac{\sum_{l=1}^{K} \exp \left[ g_j(l, y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l \sum_{l=1}^{K} \exp \left[ \beta_{ij}(l, l') + g_j(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l \prod_{r \in N_{(j)}} \rho_{jr}^l}{\sum_{l=1}^{K} \sum_{r \in N_{(j)}} \exp \left[ g_j(l, y, \theta) + \beta_{ij}(l, l') + g_{ij}(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l \prod_{r \in N_{(j)}} \rho_{jr}^l} .
\end{equation}

Because \( \sum_{l=1}^{K} \exp \left[ \beta_{ij}(l, l') + g_{ij}(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l \) depends on \( k \), we rewrite Eq. (35) as follows:
\begin{equation}
\rho_{ij}^k = \frac{1}{W} \sum_{l=1}^{K} \frac{\sum_{l=1}^{K} \exp \left[ g_j(l, y, \theta) + \beta_{ij}(l, l') + g_{ij}(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l}{\prod_{r \in N_{(j)}} \rho_{jr}^l} .
\end{equation}

\begin{equation}
W = \left\{ \sum_{l=1}^{K} \sum_{r \in N_{(j)}} \exp \left[ g_j(l, y, \theta) + \beta_{ij}(l, l') + g_{ij}(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l \prod_{r \in N_{(j)}} \rho_{jr}^l \right\} .
\end{equation}

\begin{equation}
V = \left\{ \sum_{l=1}^{K} \exp \left[ g_j(l, y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l \right\} ^{-1} .
\end{equation}

Since \( \rho_{ij}^k \) is the normalized form of \( \exp(\phi_{ij}^k) \), we finally reach the iterative function for \( \rho_{ij}^k \):
\begin{equation}
\rho_{ij}^k = \frac{\sum_{l=1}^{K} \exp \left[ \beta_{ij}(l, l') + g_{ij}(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l}{\sum_{l=1}^{K} \sum_{r \in N_{(j)}} \exp \left[ \beta_{ij}(l, l') + g_j(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l} .
\end{equation}

Then \( (\delta_{z,k} \mid Y, \theta, \beta) \) and \( (\delta_{z,k} \delta_{z,k'} \mid Y, \theta, \beta) \) are given in terms of \( \rho_{ij}^k \) as follows:
\begin{equation}
(\delta_{z,k} \mid Y, \theta, \beta) = \exp \left[ g_j(k, y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^k .
\end{equation}

\begin{equation}
(\delta_{z,k} \delta_{z,k'} \mid Y, \theta, \beta) = \frac{\sum_{l=1}^{K} \exp \left[ g_j(l, y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l}{\sum_{l=1}^{K} \sum_{r \in N_{(j)}} \exp \left[ g_j(l, y, \theta) + \beta_{ij}(l, l') + g_{ij}(l', y, \theta) \right] \prod_{r \in N_{(j)}} \rho_{jr}^l} .
\end{equation}

where
\begin{equation}
\pi_{ij}^{kl} = \exp \left[ g_j(k, y, \theta) + \beta_{ij}(k, k') + g_{ij}(k', y, \theta) \right] .
\end{equation}

As given in the EM/MPM scheme, we estimate the optimal value of \( \beta \) by maximizing the log likelihood, \( \log P(Y \mid \theta, \beta) \), with respect to \( \beta \).
\[
\frac{\partial}{\partial \beta} \log P(Y \mid \theta, \beta) = \frac{\partial}{\partial \beta} \log \sum_{z_i} P(Y \mid Z, \theta)P(Z \mid \beta) \\
= \frac{\partial}{\partial \beta} \sum_{z_i} P(Y \mid Z, \theta) \exp(-\beta H_{2}^{\text{prior}}) \\
= \frac{\partial}{\partial \beta} \log \frac{\sum_{z_i} \exp(-\beta H_{2}^{\text{prior}})}{C_{17}} = 0, 
\]
from which we obtain the following equation:
\[
(H_{2}^{\text{prior}} \mid Y, \theta, \beta) = (H_{2}^{\text{prior}} \mid \beta). 
\]
Here \(Q \mid \beta \equiv \sum_{z} P(Z \mid \beta)Q\) for some quantity \(Q\). Recalling that the Hamiltonian of the prior takes the form of Eq. (3), we further derive the following equation.
\[
\sum_{(y)} \sum_{k=1}^{K} \sum_{k=1}^{K} f_{ij}(k,k') (\delta_{z_j,k} \delta_{z_j,k'} \mid Y, \theta, \beta) = \sum_{(y)} \sum_{k=1}^{K} \sum_{k=1}^{K} f_{ij}(k,k') (\delta_{z_j,k} \delta_{z_j,k'} \mid \beta). 
\]

In order to derive the update rules for \(\mu_k\) and \(\sigma_k\), we differentiate the log likelihood, \(\log P(Y \mid \theta, \beta)\), with respect to \(\theta\), i.e.,
\[
\frac{\partial}{\partial \theta} \log P(Y \mid \theta, \beta) = \frac{\partial}{\partial \theta} \log \sum_{z_i} P(Y \mid Z, \theta)P(Z \mid \beta) = 0. 
\]
We substitute the formula of \(P(Y \mid Z, \theta)\) into the above equation, we have
\[
\sum_{z_i} \frac{\partial P(Y \mid Z, \theta)}{\partial \theta} P(Z \mid \beta) = \sum_{z_i} \sum_{i} \frac{\partial g(z_i,y_i, \theta)}{\partial \theta} P(Y \mid Z, \theta)P(Z \mid \beta) \\
= \sum_{i} \sum_{k=1}^{K} \frac{\partial g(z_i,y_i, \theta)}{\partial \theta} \delta_{z_i,k} P(Y \mid Z, \theta)P(Z \mid \beta) \\
= \sum_{i} \sum_{k=1}^{K} \frac{\partial g(z_i,y_i, \theta)}{\partial \theta} (\delta_{z_i,k} \mid Y, \theta, \beta) \\
= 0. 
\]
The results are given as follows:
\[
\mu_k^* = \sum_i y_i \langle \delta_{z_i,k} \mid Y, \theta, \beta \rangle, \\
\sigma_k^* = \sqrt{\sum_i (y_i - \mu_k^*)^2 \langle \delta_{z_i,k} \mid Y, \theta, \beta \rangle} / \sum_i \langle \delta_{z_i,k} \mid Y, \theta, \beta \rangle. 
\]
In order to evaluate the classification result quantitatively, we introduce the mis-classification rate (MCR) criterion, \(r_{MC}\), which is defined by:
\[
r_{MC} = \frac{1}{N} \sum_i (1 - \delta_{z_i, z_i^{\text{true}}} ),
\]
where \(z_i^{\text{true}}\) is the true label value for pixel \(i\).

4. Extension to Color Image Segmentation

It is easy to extend the above algorithm to a segmentation for color images. Different from the gray-scale image for which each value in the intensity space is scalar, each value of the intensity space for a color image is represented by a vector in a color space. Several color spaces have been proposed for different contexts of image processing, e.g., RGB, HSI, YIQ and CIE spaces, etc. [7, 25–28] Although lots of discussions have been made so far, the selection of the best color space is still a difficult problem for the color image segmentation. The RGB space is quite commonly used because of its simplicity in implementation, where the color at each pixel of the image is generated by modulating the intensity of three primary colors: red, green, and blue [25]. A better color space than the RGB space in representing the colors of human perception is the HSI space, in which the color information is represented by hue and saturation values.
while brightness is represented by an intensity value [27]. The YIQ space is obtained by a linear transformation on the RGB space, where the Y component is a measurement of the luminance and is argued to be a likely candidate for edge detection while the I and Q components jointly describe the hue and the saturation of the image [28]. The CIE spaces provide an approximately uniform chromaticity scale, which allows the use of Euclidean distance in expressing the color difference of human perception, and thus is especially efficient in the measurement of small color difference [7].

Despite the difference among those color spaces, the configuration of a color space is expressed in general by an expression,

$$Y_i = \frac{1}{C_3} \sum_{j=1}^{d} y_i^j,$$

where $d$ is the dimension of the selected color space. Accordingly, a mixture of multivariate Gaussian distribution is required to model the image data; Eq. (10) is replaced by the following equation for the color image,

$$g_i(k, y_i, \theta) = -\frac{1}{2} \log((2\pi)^d | \Sigma_k|) - \frac{1}{2} (y_i - \mu_k)^T \Sigma_k^{-1} (y_i - \mu_k), \quad (51)$$

where $\theta = (\mu_1, \Sigma_1, \mu_2, \Sigma_2, \ldots, \mu_K, \Sigma_K)$. Here $\mu_k$ and $\Sigma_k$ are a mean vector and a covariance matrix of class $k$, respectively. Since no exact form of the function $g_i(k, y_i, \theta)$ is required in our derivation, most of the equations are of the same form for the color image segmentation except for the substitution of $g_i(k, y_i, \theta)$ for $g_i(k, y_i, \theta)$. The update equations of mean vectors and covariance matrices for the color image segmentation are given as follows:

$$\mu_k^* = \frac{\sum_{i} y_i (\delta_{i,k} | Y, \theta, \beta)}{\sum_{i} (\delta_{i,k} | Y, \theta, \beta)}, \quad (52)$$

$$\Sigma_k^* = \frac{\sum_{i} (y_i - \mu_k^*)(y_i - \mu_k^T) (\delta_{i,k} | Y, \theta, \beta)}{\sum_{i} (\delta_{i,k} | Y, \theta, \beta)}. \quad (53)$$

5. Experimental Results

Our numerical experiments mainly focus on verifying the performance gain by the Bethe approximation over the result by the mean-field approximation. In our investigation, the robustness of the proposed method to the overlap between components of the GMM, which is usually caused by large variances of components, is a key factor to improve the performance of a segmentation algorithm. The improvement of approximation to the Gibbs free energy increases the robustness to the overlap level by making more accurate estimation of the overlap at each updating step, and thus results in better segmentation performance. For this reason, we use a set of generated images, which are designed to be well Gaussian distributed with different variance, for testing the proposed method.

Firstly, we use an image with equal variance $\sigma^2 = 300$ for all classes, as given in Fig. 1. Four components, A, B, C and D, whose layout is given in Fig. 2, are designed to have different mean values at 50, 100, 150, 200, respectively,
with the same fluctuation $\sigma^2 = 300$. In this case, this test image meets the assumption of GMM and thus is suitable to verify the efficiency of Potts model prior with the MF/MPM algorithm and the Bethe/MPM approximation. Here and hereafter, we refer to the EM/MPM algorithm using the Bethe approximation as the Bethe/MPM algorithm while referring to the EM/MPM algorithm using the mean-field approximation as the MF/MPM algorithm. For such a well GMM distributed image, both of the Bethe/MPM and the MF/MPM algorithm can give us satisfiable results, and thus we give only the result of the Bethe/EM algorithm. The estimated GMM after the segmentation is shown in Fig. 3 while the segmented image is given in Fig. 4. As we see clearly from Fig. 3, the histogram is well fitted by the four components of the estimated GMM. We further study the performance of EM/MPM algorithms under different values of $\beta$, and then the $r_{MC}$ is plotted as a function of $\beta$ in Fig. 5. The results by using some other algorithms, such as the Markov chain Monte-Carlo (MCMC), the deterministic EM/MAP algorithm and the simulated EM, are also given for comparison in Fig. 5. As depicted in Fig. 5, we note that in a wide range of $\beta$, the MCMC (dash line) gives us acceptable results, which verifies the efficiency of Potts Model assumption as a prior. Although the variance is small, almost all algorithms except the deterministic EM/MAP algorithm provide us with quite good results in a wide range of $\beta$, we have found that the Bethe/MPM algorithm outperforms the MP/MPM as for the robustness of $\beta$ when $\beta$ becomes larger. The estimated value of $\beta^*$ is $1.057171$, which is given in Fig. 6 with $\langle \mathcal{H}_Z^{\text{prior}} | Y, \theta, \beta \rangle$ and $\langle \mathcal{H}_Z^{\text{prior}} | \beta \rangle$ plotted as a function of $\beta$. In fact, $\beta$ is well estimated as a solution to Eq. (44), which is corresponding to a high value of $r_{MC}$ in Fig. 5. Since the image is generated to be well Gaussian distributed, we think that the Potts model assumption causes the bias between the maximization of log-likelihood and the minimization of $r_{MC}$. Because the purpose of our experiments is to investigate the robustness and the estimated $\beta$ does not give us meaningful information on this
investigation, we do not use the estimated value of $C_1$ in latter experiments, but we find a value of $C_1$ from Fig. 5.

In order to view the difference of performance for each algorithm quantitatively, minimum values of $r_{MC}$ from all tested algorithms are listed in Table 1. Since another factor we have to consider is the complexity of computation, we give also the corresponding iteration times and CPU time in Table 1. Although the Bethe/MPM algorithm runs slower at each step, it converges much faster than its mean-field counterparts. As for the reason, it is thought to be the better accuracy of the Bethe/MPM algorithm in generating next states that speeds up the convergence. In Table 1, we also note that the minimum value of $r_{MC}$ for the MCMC is higher than the one for the Bethe/MPM algorithm. Since $r_{MC}$ is rather low (≈5%), the difference between these two algorithms locates only in several pixels, which, as we can see from the images, are really doubtful that we cannot make sure to which class it originally belongs to.

An image with class variance $C_2 = 900$ is used to enlarge the difference between those algorithms as shown in Fig. 7. Mean values of all components for regions A, B, C and D are also 50, 100, 150 and 200, respectively. Thus the GMM assumption is violated by the large overlaps existing between two neighboring components.

Table 1. $r_{MC}$ and computation time for the test image given in Fig. 1. All the iteration times and CPU time given are corresponding to the minimum value of $r_{MC}$. Note that less iterations are required by the Bethe/MPM algorithm for the improvement of approximation accuracy. As a result, the Bethe/MPM algorithm computes faster than all others here.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Minimum value of $r_{MC}$ (%)</th>
<th>Iteration times</th>
<th>CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic EM</td>
<td>0.782015</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>MF/MPM</td>
<td>0.592041</td>
<td>31</td>
<td>4</td>
</tr>
<tr>
<td>Bethe/MPM</td>
<td>0.531006</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>Simulated EM</td>
<td>0.592041</td>
<td>300</td>
<td>66</td>
</tr>
<tr>
<td>MCMC</td>
<td>0.585982</td>
<td>1000</td>
<td>104</td>
</tr>
</tbody>
</table>

Fig. 7. A test image with $\sigma^2 = 900$. The mean values of each component are still 50, 100, 150 and 200, respectively. Thus the GMM assumption is violated by the large overlaps existing between two neighboring components.

Best result by the MF/MPM algorithm for $\sigma^2 = 900$. The robustness of the Bethe/MPM algorithm becomes much clearer in Fig. 11. Although the Bethe/MPM cannot escape from local optima, it seems that the better accuracy of the Bethe/MPM helps to decrease the probability to get trapped. In Table 2, the Bethe/MPM algorithm achieves a low value of $r_{MC}$ comparable to the value by the MCMC with a rather fast computation speed. As seen in Fig. 11, the Bethe/MPM algorithm does not show any obviously descending trend in performance as $\beta$ keeps increasing. In fact, the performance of the Bethe/MPM algorithm deteriorates rather quickly with the ascending of $\sigma^2$ and finally into a complete failure of segmentation. Actually, in this case, the performance is limited by the single-scale MRF and without using other structures, e.g., multi-scale techniques, even the robustness of MCMC is not much better than that of the Bethe/MPM. Thus within the regime of single-scale MRF, we come to the conclusion that the Bethe/MPM shows a rather good robustness.
Fig. 9. Best result by the Bethe/MPM algorithm for the test image in Fig. 7. The difference between the MF/MPM algorithm and the Bethe/MPM algorithm becomes much clear.

Fig. 10. Histogram of the estimated GMM by the Bethe/MPM algorithm for the test image in Fig. 7. Note that two sharp peaks exist at intensity 0 and 255, which violate the GMM assumption.

Fig. 11. $r_{\text{MC}}$ as a function of $\beta$ for algorithms. By generating more accurate new state at each step, the Bethe/MPM algorithm has better robustness to $\sigma^2$ than the MF/MPM algorithm.

Table 2. $r_{\text{MC}}$ and computation time for the test image given in Fig. 6. Note that although the iteration times of the Bethe/MPM algorithm is larger than the deterministic EM algorithm, the latter has failed to converge to a satisfiable local optimum.

<table>
<thead>
<tr>
<th></th>
<th>Minimum value of $r_{\text{MC}}$ (%)</th>
<th>Iteration times</th>
<th>CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic EM</td>
<td>20.59937</td>
<td>27</td>
<td>3</td>
</tr>
<tr>
<td>MF/MPM</td>
<td>4.144287</td>
<td>79</td>
<td>10</td>
</tr>
<tr>
<td>Bethe/MPM</td>
<td>2.246094</td>
<td>36</td>
<td>7</td>
</tr>
<tr>
<td>Simulated EM</td>
<td>3.088332</td>
<td>300</td>
<td>51</td>
</tr>
<tr>
<td>MCMC</td>
<td>2.197266</td>
<td>1000</td>
<td>106</td>
</tr>
</tbody>
</table>

Finally, we give some results for real images. There are two difficulties in evaluating the real image segmentation. One is the decision for the number of classes, and the other is that there is no true segmentation result for those real images. We choose three types of samples for comparison: a medical MRI image in Fig. 12, a texture image in Fig. 13, and a natural color image in Fig. 14. Here for simplicity the RGB color space is adopted for the sample color image in Fig. 14. For each sample, the same $\beta$ value is applied to both the Bethe/MPM and the MF/MPM; $\beta$ is set to 4, 10 and 2 for the MRI image, the texture image and the natural image, respectively. From Figs. 5 and 11, we find that the Bethe/MPM algorithm is kept better than the MF/MPM algorithm for the same value of $\beta$ within a wide range, although we have the different performance gain. Original images are given on the left, and results by the Bethe/MPM and the MF/
MPM are shown on the center and right, respectively. Note that a much better segmentation result is produced by the Bethe/MPM for the texture image. For the benefit of the robustness for the Bethe/MPM algorithm to a large value of $\sigma$, a large value of $\beta$ is used to deal with this texture image and achieve a better texture segmentation, while a large value of $\beta$ leads to over-segmentation results for the MF/MPM algorithm.

6. Conclusions

In the present paper, we have extended the Bethe approximation to a general case and proposed an EM/MPM
approach of image segmentation by applying the Bethe approximation to it; we denote it as the Bethe/MPM algorithm. Numerical experiments on both generated images and real images have been done to verify the performance gain of our approach over that using the MF/MPM algorithm. Comparing with other algorithms, the Bethe/MPM algorithm shows better performance, especially when the overlap between components of GMM becomes stronger; the Bethe/MPM algorithm outperforms others in robustness except the MCMC. Because of the limitation of single scale MRF, only weak texture images, which have a small variance in each class, are well dealt with. For those images with large variances, a multi-scale segmentation scheme has to be used, which will be included in our future work. The Bethe/MPM has a rather fast computational speed comparable to those by the MF/MPM algorithm, benefiting from its fast speed of convergence, and runs much faster than the simulated EM and MCMC.

We have checked the dependence of segmentation result on $\beta$, but a further analysis is still required on the selection of parameters, e.g., the number of classes, $K$, which is left as one of our future works. Although the simple Potts model is efficient for current image, more complicated prior must be introduced to avoid over-segmentation for natural images. We will also try to combine the segmentation result with perceptual feature extraction and perceptual grouping process for better image understanding.

REFERENCES