Grayscale image segmentation using reversible jump MCMC. Intermediate report.

Pavel Senin

November 9, 2006

Contents

1 Introduction

I have chosen project aiming working software piece that employs Reversible Jump MCMC approach. The reason to chose Reversible Jump is because this approach in my opinion mimics ntural human ability to segment images that are noizy or have very little features to do segmentation exactly. Basically this method search within the multidimentional variants space trying to maximize classes separation. The approach itself seems to be havyweight and computationally costly, but second reason to choose this way is my own interest in understanding of unsupervised learning and ICA.

In my view, resulting software would accept any image as input data and will produce segmented image as output logging on fly all found specificity of the provided image itself. This report will narrow to detailed model description, the algorithm moves description and detailed description of each methods used. As the supplementary materials produced R code will be placed along with segmentation results and performance benchmarking.

R (or 'GNU S'), a freely available language and environment for statistical computing and graphics was chosen as the programming environment. So far I've started a little coding and able to load and save images, extract raster information as a numerical matrix with intensity value for each of the pixels. I did code for the image segmentation using hardcoded Gaussian parameters (Move 1 of following algorithm). The main issues for now are functions for building sampling distributions and sampling itself along with simulated annealing method implementation.

As I mentioned in project proposal, I found available for benchmarking images but in this case it would be hard to estimate performance due to working in R, most of benchmarking results are found for C/C++ compiled code.

Fo the last month of this semester I will try to finish my coding along with the rest of the model description. For my class presentation I would like to choose to present simulated annealing method because of it's crucial role for optimization in the segmentation algorithm.

Rest of this report is just very first draft for my workflow and it's background.

2 Model description

2.1 Model build

The model itself is built according to (NEED CITATION, Kato,99). Let's suppose that the observed image is: $F = \left\{ \vec{f}_s \,\middle|\, s \in S, \forall i : 0 < \vec{f}_s^i < 1 \right\}$, where vector $\vec{f}_s$ is vector that carried intensity of colour for pixel s. The segmentation itself is just labeling of each pixel $s \in S$ by label $\omega_s \in \Lambda = \{1, 2, ..., L\}$. $\omega \in \Omega$ denotes a labeling (or segmentation), Ω is a set of all possible labeling.

We regard our image as a sample drawn from unknown Gaussian mixture distribution. The goal of our analysis is inference about the number L components:

- the component parameter $\Theta = \{\forall\lambda : 1 \leq \lambda \leq L, \Theta_\lambda = (\vec{\mu_\lambda}, \Sigma_\lambda)\}$;
- the component weights p_λ $(1 \leq \lambda \leq L)$ summing to 1;
- the clique potential (or inter-pixel interaction strength), β;
- the segmentation ω.

The joint distribution of above variables $L, p, \beta, \omega, \Theta, F$ given by formula :

$$P(L, p, \beta, \omega, \Theta, F) = P(\omega, F \,|\Theta, \beta, p, L\,)\, P(\Theta, \beta, p, L) \tag{1}$$

In our context it is natural to impose the independence of labeling Θ, inter-pixel composition β, component weights p, and labels set power L as parameters that given randomly in every processed image. Therefore their joint probability reduces to

$$P(\Theta, \beta, p, L) = P(\Theta)\, P(\beta)\, P(p)\, P(L) \tag{2}$$

The posterior distribution of (F, ω) may be expressed as:

$$P(F, \omega \mid \Theta, \beta, p, L) = P(F \mid \omega, \Theta, \beta, p, L)\, P(\omega \mid \Theta, \beta, p, L) \tag{3}$$

As declared earlier pixel classes (segmentation itself) represented as a multivariate Gaussian distribution and the underlied MRF (Markov Random Field) process follows a Gibbs distribution defined over a first order neighborhood system (Figure 1). Previous equation could be factored out as:

$$P(F \mid \omega, \Theta, \beta, p, L) = P(F \mid \omega, \Theta) =$$

$$\prod_{s\in S} \left(\frac{1}{\sqrt{(2\pi)^3 \left|\sum_{\omega_s}\right|}} \exp\left(-\tfrac{1}{2} \left(\vec{f}_s - \vec{\mu}_{\omega_s}\right) \sum\nolimits_{\omega_s}^{-1} \left(\vec{f}_s - \vec{\mu}_{\omega_s}\right)^T \right) \right) \tag{4}$$

Proceeding further we have:

$$P(\omega \mid \Theta, \beta, p, L) = P(\omega \mid \beta, p, L) = \frac{1}{Z(\beta, p, L)} \exp\left(-U(\omega \mid \beta p, L)\right) \tag{5}$$

where $U(\omega \mid \beta, p, L)$ is energy function:

$$U(\omega \mid \beta, p, L) = \sum_{s\in S} -\log(p_{\omega_s}) + \beta \sum_{\{s,r\}\in C} \delta(\omega_s, \omega_r) \tag{6}$$

$\delta(\omega_s, \omega_r) = 1$ if ω_s and ω_r are different and -1 otherwise. $Z(\beta, p, L) = \sum_{\omega\in\Omega} \exp\left(-U(\omega \mid \beta, p, L)\right)$ is normalization constant or partition function. C denotes the set of cliques and $\{s, r\}$ is doubleton containing the neighboring pixel sites r and s.

Note that the whole posterior distribution could be derived from Gibbs distribution where the Gaussian distribution taken in account in the energy of the external field:

$$U(F \mid \omega, \Theta) = -\log\left(P(F \mid \omega, \Theta)\right) =$$

$$\sum_{s\in S} \left(\ln\left(\sqrt{(2\pi)^3}\right) + \tfrac{1}{2} \left(\vec{f}_s - \vec{\mu}_{\omega_s}\right) \sum\nolimits_{\omega_s}^{-1} \left(\vec{f}_s - \vec{\mu}_{\omega_s}\right)^T \right) \tag{7}$$

The partition function $Z(\beta, p, L)$ is not tractable according to (NEED CITATION), the comparison of the likelihood of two different MRF realizations from equation (5) is unfeasible. However Pseudo-Likelihood could be compared according to (NEED CITATION) as:

$$P(\omega \mid \beta, p, L) \approx \prod_{s \in S} \frac{p_{\omega_s} \exp\left(-\beta \sum_{\forall r:\{s,r\} \in C} \delta(\omega_s, \omega_r)\right)}{\sum_{\lambda \in \Lambda} p_\lambda \exp\left(-\beta \sum_{\forall r:\{s,r\} \in C} \delta(\lambda, \omega_r)\right)} \tag{8}$$

Now, we are using facts that $P(F)$ is constant for any particular image and Equation (1), Equation (2), Equation (4) and Equation (8) we are able to approximate the posterior density $P(L, p, \beta, \omega, \Theta \mid F) = P(L, p, \beta, \omega, \Theta, F) / P(F)$:

$$\begin{aligned} P(L, p, \beta, \omega, \Theta \mid F) &= P(F \mid \omega\Theta) P(\omega \mid \beta, p, L) P(\Theta) P(\beta) P(p) P(L) \\ &\approx \prod_{s \in S} \left(\frac{1}{\sqrt{(2\pi)^3 \left|\sum_{\omega_s}\right|}} \exp\left(-\frac{1}{2}\left(\vec{f}_s - \vec{\mu}_{\omega_s}\right) \sum_{\omega_s}^{-1} \left(\vec{f}_s - \vec{\mu}_{\omega_s}\right)^T\right)\right) \\ &\times \prod_{s \in S} \frac{p_{\omega_s} \exp\left(-\beta \sum_{\forall r:\{s,r\} \in C} \delta(\omega_s, \omega_r)\right)}{\sum_{\lambda \in \Lambda} p_\lambda \exp\left(-\beta \sum_{\forall r:\{s,r\} \in C} \delta(\lambda, \omega_r)\right)} \\ &\times P(\beta) P(L) \prod_{\lambda \in \Lambda} P(\vec{\mu_\lambda}) P(\Sigma_\lambda) P(p_\lambda) \end{aligned} \tag{9}$$

Concerning the priors, this model follows (NEED CITATION) and choose uniform reference priors for L, $\vec{\mu_\lambda}$, Σ_λ, p_λ where $\lambda \in \Lambda$.

2.2 Sampling from Posterior

Herein we will construct MCMC sampler from posterior distribution which is used in equation (9) for our segmentation model. Classical MCMC cannot be used, because of changing dimensionality of the parameter space. Reversible Jump MCMC however capable to hold this issue (CITATION NEED RJMCMC). Our set of unknowns is $\{L, p, \beta, \omega, \Theta\}$, lets denote it as X and let $\pi(X)$ be the target probability measure (the posterior distribution). The wildly used tool to sample from such distribution is Metropolis-Hasting method (CITATION NEED). When the current state is X the new state is drawn from an arbitrary joint distribution $q(X, X')$. The new state is accepted with probability

$$A(X, X') = \min\left(1, \frac{\pi(X')\, q(X, X')}{\pi(X)\, q(X', X)}\right) \tag{10}$$

We have multiple parameter subspaces of different dimensionality and it is nessesary to devise move types between subspaces (CITATION?). These moves will be combined in so called hybrid sampler (CITATION?) by random choice between available moves at each transition. We denote the moves as $m \in M = \{1, 2, ..., M\}$ and let $q(X, X')$ be the probability of proposing the move type m and state X' when the current state is X. Not all move types available from the

beginning, so for some certain $X, q_m(X, \cdot)$ might be 0 for some m. Furthermore $q_m(X, \cdot)$ is is a sub-probability measure on m at X' and $\sum_{m \in M} q_m(X, \cdot) \leq 1$ (CITATION ORIGINAL RJ PAPER). The proposed state is than accepted with probability

$$A_m(X, X') = \min\left(1, \frac{\pi(X')\, q_m(X, X')}{\pi(X)\, q_m(X', X)}\right) \tag{11}$$

NEED TO PUT MORE JUSTIFICATION AND EXPLANATION HERE.

2.3 Hybrid Sampler

For sketched in previous section model we need to work out sampler. It will consist of five moves:

1. Sampling the class labels ω, i.e. resegment the image;
2. Sampling Gaussian parameters $\Theta = \{(\mu_\lambda, \vec{\Sigma_\lambda}) \mid \lambda \in \Lambda\}$;
3. Sampling the mixture weights $p_\lambda\, (\lambda \in \Lambda)$;
4. Sampling the MRF hyperparameter β;
5. sampling the number of classes L, i.e. splitting one of mixture components into two or merge two of them into one.

Cases 1 - 4 are not random instead of case 5 where at each sweep we decide randomly about splitting or combining.

2.4 Move 1 - image segmentation

This move is just classical image segmentation with known parameters, i.e. we have fixed parameters - their estimates,

$$P(L, p, \beta, \omega, \Theta \mid F) \approx P\left(F \mid \omega, \widehat{\Theta}\right) P\left(\omega \mid \widehat{\beta}, \widehat{p}, \widehat{L}\right) P(\Theta)\, P(\beta)\, P(p)\, P(L)$$

$$\approx \prod_{s \in S} \left(\frac{1}{\sqrt{(2\pi)^3 \left|\widehat{\Sigma}_{\omega_s}\right|}} \exp\left(-\frac{1}{2} \left(\vec{f_s} - \vec{\widehat{\mu}}_{\omega_s}\right) \widehat{\Sigma}_{\omega_s}^{-1} \left(\vec{f_s} - \vec{\widehat{\mu}}_{\omega_s}\right)^T \right) \right)$$

$$\times \prod_{s \in S} \widehat{p}_{\omega_s} \exp\left(-\widehat{\beta} \sum_{\forall r : \{s,r\} \in C} \delta(\omega_s, \omega_r) \right) \tag{12}$$

We have eliminated $Z(\beta, p, L) = Z\left(\widehat{\beta}, \widehat{p}, \widehat{L}\right)$ because it is constant for the Ω. Furthermore sub-chain could be sampled using Gibbs sampler since ω takes discrete values over finite set Λ.

2.5 Move 2 - estimating Gaussian parameters

This move is aiming mean and covariance matrix of the pixel classes. By setting variables L, p, β, ω to their estimates $\widehat{L}, \widehat{p}, \widehat{\beta}, \widehat{\omega}$ Equation (9) reduces to the form:

$$P(L, p, \beta, \omega, \Theta \mid F) \approx P(F, \widehat{\omega} \mid \Theta)\, P(\Theta) =$$

$$\prod_{\lambda\in\Lambda}\prod_{s:\widehat{\omega}_s=\lambda} P\left(\vec{f}_s\middle|\,\vec{\mu}_\lambda,\Sigma_{lambda}\right)P\left(\vec{\mu}_\lambda\right)P\left(\Sigma_\lambda\right)=$$
$$\times\prod_{s\in S}\frac{1}{\left((2\pi)^3|\Sigma_\lambda|\right)^{|\Sigma_\lambda|/2}}\exp\left(-\tfrac{1}{2}\sum_{s:\widehat{\omega}_s=\lambda}\left(\vec{f}_s-\vec{\mu}_\lambda\right)\Sigma_\lambda^{-1}\left(\vec{f}_s-\vec{\mu}_\lambda\right)^T\right)$$
$$\times\prod_{\lambda\in\Lambda}P\left(\vec{\mu}_\lambda\right)P\left(\Sigma_\lambda\right)P\left(p_\lambda\right)\quad(13)$$

where $|\Sigma_\lambda|$ is number of sites labeled by λ.

2.6 Move 3 - sampling Mixture Weights

¡not filled yet¿

2.7 Move 4 - sampling Hyperparameter β

¡not filled yet¿

3 Move 5 - Estimating number of classes

¡not filled yet¿

3.1 Sampling distribution

¡not filled yet¿

3.2 Splitting classes (generating new parameters and reallocating labels)

¡not filled yet¿

3.3 Merging classes (generating new parameters and reallocating labels)

¡not filled yet¿

3.4 Move acceptance probability

4 Optimization according to the MAP criteria, Simulated annealing

This section contains description of MAP estimator that provides us with an image segmentation$\widehat{\omega}$ and model parameters $\widehat{L},\widehat{p},\widehat{\beta},\widehat{\Theta}$. The estimation is done via stochastic relaxation algorithm. The MAP estimator is given by:

$$\left(\widehat{\omega},\widehat{L},\widehat{p},\widehat{\beta},\widehat{\Theta}\right)^{(MAP)}=\arg\max_{L,p,\beta,\omega,\Theta}P\left(L,p,\beta,\omega,\Theta\mid F\right)\qquad(14)$$

with the following constaraints:

$$\omega \in \Omega, \tag{15}$$

$$L_{min} \leq L \leq L_{max}, \tag{16}$$

$$\sum_{\lambda \in \Lambda} p_{lambda} = 1, \tag{17}$$

$$\forall \in \Lambda : 0 \leq \mu_{,i} \leq 1, \tag{18}$$

$$\forall \in \Lambda : 0 \leq \Sigma_{,i,i} \leq 1, -1 \leq \Sigma_{,i,j} \leq 1 \tag{19}$$

First equation of this section itself is a combinatorial optimization problem which requires special algorithms such as simulated annealing (CITATION NEEDED). In our case this process could be formulated as follows:

5 Algorithm, RJMCMC segmentation

1. Set $k = 0$ and initialize $\widehat{L}^0, \widehat{p}^0, \widehat{\beta}^0, \widehat{\Theta}^0$, and set temperature τ^0.

2. A sample $\left(\widehat{\omega}^k, \widehat{L}^k, \widehat{p}^k, \widehat{\beta}^k, \widehat{\Theta}^k\right)$ is drawn from the modified posterior distribution using the hybrid sampler defined in section (NUMBER NEEDED) before. The modification is due to simulated annealing constaraint - temperature τ_k:

$$\prod_{s \in S} \frac{1}{\left((2\pi)^3 \left|\Sigma_{\omega_s}\right|\right)^{1/2\tau_k}} \exp\left(-\frac{1}{2\tau_k}\left(\vec{f}_s - \vec{\mu}_{\omega_s}\right)\Sigma_{\omega_s}^{-1}\left(\vec{f}_s - \vec{\mu}_{\omega_s}\right)^T\right)$$

$$\times \prod_{s \in S} \frac{\exp\left(\frac{\log\left(p_{\omega_s}\right)}{\tau_k} - \frac{\beta}{\tau_k} \sum\limits_{\forall r:\{s,r\} \in C} \delta(\omega_s, \omega_r)\right)}{\sum\limits_{\lambda \in \Lambda} \exp\left(\frac{\log\left(p_{\lambda}\right)}{\tau_k} - \frac{\beta}{\tau_k} \sum\limits_{\forall r:\{s,r\} \in C} \delta(\lambda, \omega_r)\right)} \tag{20}$$

2.1. $\widehat{\omega}^k$ is drawn from distribution in Equation (NUMBER NEEDED).

2.2. $\widehat{\Theta}^k$ is drawn from distribution in Equation (NUMBER NEEDED).

2.3. $\widehat{p}^k$ is drawn from distribution in Equation (NUMBER NEEDED).

2.4. Sampling MRF hyperparameter $\widehat{\beta}^k$ from Equation (NUMBER NEEDED).

2.5. $\widehat{L}^k$ is estimated using the reversible jump techniques from the section (SECTION NUMBER NEEDED)

3. GOTO to step 2 with $k = k + 1$ and T_{k+1} until $k \leq K$.