Gibbs Sampling via Neural Network Probability Estimation

Jenq-Neng Hwang, Alan Lippman, Eric Tsung-Yen Chen

Information Processing Laboratory
Department of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195

Abstract—This paper proposes a neural network approach, an efficient nonparametric approach, for Markov random field (MRF) modeling to provide a good estimate of Bayesian a posteriori probability. The proposed approach, neural network based MRF (NNMRF), overcomes the difficulties encountered in estimating the parameters of Gibbs distribution that characterize the MRFs and the underlying texture. Successful application of the NNMRF to textured image segmentation using the Gibbs sampling technique is also presented.

1. Introduction

Markov random field (MRF) based on Gibbs distribution has been widely used in textured image modeling [4, 7]. However, the model parameters that constitute the conditional probability in Gibbs distributions and necessary for the segmentation usually require tedious and laborious estimation. Different estimation techniques have been proposed, e.g., coding method [1], maximum pseudo likelihood [2], and linear least square methods using histogramming [4]. These methods, most of them cannot be used reliably, either require solutions to a set of nonlinear equations that are difficult and cumbersome to solve or least square solutions to a big set of linear equations. The method proposed here is basically a nonparametric (classifier) estimator for Gibbs distribution based on neural networks trained by neighborhood configurations of an underlying texture. This greatly simplifies estimation task in that the parameters associated with different cliques of an MRF model are inherently embedded in the neural network structure. In Section 2, we will discuss Gibbs sampling algorithm and its relationship with our proposed neural network based MRF (NNMRF) model. In Section 3, we will investigate the posterior probability estimation capabilities of the back-propagation neural network (BPNN) trained with Kullback-Leibler criterion. Textured image segmentation examples will be illustrated in Section 4 and followed by conclusion in Section 5.

2. Bayesian Image Segmentation via Gibbs Sampling

2.1 Markov Random Field (MRF) Model

We treat an image as a discrete 2-D random field defined over \( N_1 \times N_2 \) rectangular lattice of points.\[ S = \{(i,j) | 1 \leq i \leq N_1, 1 \leq j \leq N_2\} \]
A neighborhood system \( \eta \) defined on \( S \) is a collection of subsets of \( S \):
\[ \eta = \{(i,j) \in S, \eta \in \eta \} \]
which satisfies the following two conditions:
1. \((i,j) \in \eta \)
2. if \((k,l) \in \eta \), then \((i,j) \in \eta \) for any \((i,j) \in S \).

A neighborhood system can be extended to any order hierarchically, e.g., first order neighborhood system \( \eta^{(1)} \) contains the nearest four pixels surrounding \((i,j)\), and second order neighborhood system \( \eta^{(2)} \) contains the nearest eight pixels surrounding \((i,j)\), and so on (see Figure 1(a)). A clique, \( c \), associated with a neighborhood system is a subset of \( S \) such that
1. \( c \) contains a single pixel, or
2. for \((i,j) \neq (k,l)\), \((i,j) \in c \) and \((k,l) \in c \) implies that \((i,j) \in \eta \).

The collection of all cliques of \((S, \eta)\) is denoted by \( C \). Figures 1(b) and 1(c) show a second order neighborhood system \( \eta^{(2)} \) and its associated cliques.

A random field \( Y = \{Y_\eta\} \) defined over the lattice \( S \) is Gibbs distributed with respect to \( \eta \) if its joint distribution is of the form
\[ P(Y = y) = \frac{1}{Z} e^{-U(y)} \]  
where
\[ U(y) = \sum_{c \in C} V_c(y) \]  
is the "energy function", \( V_c(y) \) is the potential associated with a clique \( c \), and \( Z \) is the normalization factor.
Figure 1: (a) A hierarchically arranged neighborhood system in an MRF. (b) The second order neighborhood system. (c) The associated cliques for second order system.

Note that a Gibbs distribution is a kind of exponential distribution which states that the joint distribution of $Y$ can be decomposed into local potential functions controlling the stability of the lattice system. More specifically, the less the potential the higher the probability. The potential functions $\{V_i(y)\}$ defined over cliques of neighborhood system $\eta$ are usually in parametric form with each (weighting) parameter associated to each type of cliques [4, 10].

### 2.2 Parameter Estimation

One commonly used parameter estimation method for MRF model is the coding method [1], which is basically a maximum likelihood method under conditional independence assumption i.e., the Markov assumption. In this method, each pixel in image is "coded" based on the order of neighborhood system we assume, i.e., pixels that are not neighbors are assigned the same label. Although we may easily conduct goodness-of-fit test for this method, it requires solution to a set of nonlinear equations that are difficult and cumbersome to solve and can not be used reliably. A computationally more feasible method in a least square framework [4], which determines parameter values by solving linear equations using histogramming. Due to the requirement of counting local configuration based on single realization and solving the pseudo inverse of a large size matrix, the latter method also results in an unreliable estimate.

### 2.3 Neural Network Based Markov Random Field (NNMRF)

Equation (1) requires that each clique type associated with the assumed neighborhood system being well defined. Since the full conditional distributions uniquely determine the joint, we are interested in the distribution of center pixel $y_i$ conditioned on its neighbors $y_{\delta i}$, i.e., $P(y_i|y_{\delta i})$.

Notice that instead of using the complicated parametric form of cliques in specifying $P(y_i|y_{\delta i})$, we can formulate it easily in a neural network setting as a classification problem.

In a classification application, it is normally assumed that the input vector, $x \in \mathbb{R}^n$, belongs to one of $M$ classes, $c_i, 1 \leq i \leq M$. The main objective of a classification task is to decide to which of the $M$ classes the vector $x$ belongs. The decision can be made based on some forms of deterministic discriminate function, e.g., the Euclidean distance measure. A more general decision rule is based on the probabilistic decision, such as maximum a posteriori (MAP) approaches which guarantees the minimum classification error. In a MAP approach, for each of the classes, one requires to estimate the posterior probability, $P(c_i|\eta)$, which is usually computed via Bayes rule. Since $P(c_i)$ is easy to compute, most conventional pattern classification literature has been focusing on the research of estimating the likelihood $P(\eta|c_i)$. On the other hand, when a neural network (such as a BPNN) is used for this classification task, there is usually an input layer of $n$ neurons corresponding to the n-dimensional input vector $x$, one or two layers of "well-chosen" hidden neurons, and one output layer of $M$ neurons with each one representative of one of the $M$ different classes. It has been shown that for a 2-class recognition problem, the single-output activation value $y$ of a BPNN trained by the standard back-propagation learning, which minimized the mean squared error (MSE) between the actual output $y$ and the desired output $t$ (either 0 or 1), indeed provides a least square estimate of the posterior probability [11, 12]. Therefore neural network classifiers bypass the stage of estimating the likelihood and directly estimate the posterior probability.

In a neural network based Markov random field (NNMRF), the training is constructed by simply feeding each BPNN with neighborhood configuration $y_{\delta i}(=x)$ of each pixel $y_i$ and assigning center value as target $c = c_i$, in a classification formulation (i.e., each gray level of $y_i$ is assigned as one class). After training, the BPNN responds to the neighborhood configuration inputs by providing the posterior probabilities as outputs for the center cite being assigned to all gray levels, i.e., the BPNN represents a model of underlying field which is conventionally modeled by Gibbs distribution of parametric form in $P(y_i|y_{\delta i})$.

### 2.4 Hierarchical Gibbs Sampling

Recently, Gibbs sampler [7] provides a Bayesian inference mechanism for applications like image analysis and other areas of spatial statistics that suffered from computational prohibition previously [3, 6, 13]. An important application is the Bayesian segmentation of textured images where we focus on estimation of global maximum a posteriori.
(MAP) configuration through iterative updating local pixel label. Given $N$ textures in an image, the goal is to assign each pixel, $y_i$, a label $l_y$ (where $l_y = 1, ..., N$) so that $P(l|y)$ is maximized. According to the Bayes rule,

$$P(l|y) = \frac{P(y|l)P(l)}{P(y)} \propto P(y|l)P(l)$$  \hspace{1cm} (3)

where the prior probability $P(l)$ is the high level spatial interaction term specifying region formation that can be also modeled by Gibbs distribution [4], and $P(y|l)$ specifies the low level texture characteristics of type $l$. Note $P(y|l)$ can be written as the joint distribution, $P_i(y_{1:i}|y_s)$, the convergence towards maximum(s) of the corresponding joint distribution $P_i(y)$ is guaranteed under rather general conditions where Brooks expansion [1] is valid. The Markovian updating scheme proceeds as follows. Given an arbitrary starting set of values $X^0=(x_1^0, ..., x_k^0)$, we update a single component from $x_i^0$ to $x_i$ according to the distribution $P(x_i|x_1, ..., x_{i-1})$, then from $x_i^0$ to $x_i$ according to the distribution $P(x_i|x_1^0, ..., x_i^0)$, and so on up to $x_i$ from $P(x_i|x_1^0, ..., x_{i-1})$ to complete one iteration. Under mild conditions [7], $X^t$ can be regarded as simulated observation from (near) maximum(s) of $P(X)$ as $t \to \infty$. According to Equation (3), we have to apply the Gibbs sampler based on NNMRF in a hierarchical manner. For each pixel, we first calculate the prior probability, $P(l)$, to which texture type it belongs based on a low-order Gibbs distribution. Then we send the pixel values of its neighbors to the trained NNRMFs to get the likelihood values of $P(y|l)$. Based on the products of these two terms, we can update its label $l_y$ that stands for the most likely texture type. Then we keep updating each pixel's label to complete one iteration. Several iterations may be needed to guarantee convergence.

3. Performance of Neural Network Based Density Estimator

From Section 2, we know that NNMRF model can provide conditional probabilities which is conventionally characterized by Gibbs distribution in parametric form. However, the standard MSE criterion used in training BPNN may have some drawbacks.

3.1 Kullback-Leibler Training Criterion

More specifically, the MSE criterion used in BPNN training tends to give better estimates at large values than at small values [11], which will severely degrade the MRF modeling capability when all the posterior probabilities generated by neural network are required to make a meaningful Gibbs sampling. One criterion called Kullback-Leibler (KL) criterion[9,11] which overcomes this deficiency and provides a better probability estimate can be adopted. The KL criterion, derived from information theory, suggests the relative entropy, $E_{KL}$, to be a better measure of the difference between the desired probability $\{t_i = P(c_i|x)\}$ and the actual estimated probability $\{y_i\}$ determined by the activation values of output neuron:

$$E_{KL} = \sum \int_x P(c_i|x) \log \frac{P(c_i|x)}{P(x)P(y_i|x,w)} P(x) dx \hspace{1cm} (4)$$

where $w$ denote all the weight parameters associated with the neural network. Note $E_{KL}$ is always nonnegative and is equal to zero if and only if $y_i(x,w) = P(c_i|x)$ for all $x$.

Given $K$ finite training vectors $\{x^{(i)}, k = 1,..., K\}$, Equation (4) can be approximated by:

$$E_{KL} = \sum_{i=1}^K \sum_{k=1}^K \left[ \log \left( \frac{1+t^{(i)}_k}{1+y^{(i)}_k} \right) + \left( 1-t^{(i)}_k \right) \log \left( \frac{1-y^{(i)}_k}{1+y^{(i)}_k} \right) \right]$$ \hspace{1cm} (5)

where we confine the targets $\{t^{(i)}_k\}$ and actual outputs $\{y^{(i)}_k\}$ to be within the interval $[-1,1]$. In this setting, the desired posterior probability is denoted as $\frac{1}{2}(1 + t^{(i)}_k)$, and similarly the actual estimated posterior probability is denoted as $\frac{1}{2}(1 + y^{(i)}_k)$.

3.2 Posterior Probability Approximation Capabilities

To verify the posterior probability approximation capabilities of a 2-layer (one hidden layer) BPNN trained with KL criterion, we conducted a simple 2-D binary classification task with data generated from two sets of Gaussian mixtures (class 1 has two Gaussian-mixtures and class 2 has three):
where \( N_i(m, \sigma) \) denotes a Gaussian (normal) distribution in \( i \)-th dimension with mean \( m \) and standard deviation \( \sigma \). The simulated data are shown in Figure 2. There are 6,000 random samples generated for each class for BPNN training. Another 10,000 uniform grid samples are generated for evaluating classification accuracy. Table 1 shows the classification accuracy of the 10,000 testing data and the percentage of variance explained (PVE) \([5]\) between the true Bayesian density and the estimated density:

\[
PVE = 100 \left(1 - \frac{E}{\Var}\right)\%
\]

where \( E = \frac{1}{N} \sum_s (\hat{P}_s - P_s)^2 \) denotes the mean squared error between the estimated density \( \hat{P}_s \) and the true density \( P_s \) over testing data, and \( \Var = \frac{1}{N} \sum_s (P_s - \Bar{P}_s)^2 \) denotes the sample variance.

Table 1: Classification accuracy of the 10,000 testing data and the PVE measures between the true Bayesian density and the estimated density.

<table>
<thead>
<tr>
<th>No. of Hidden Neurons</th>
<th>Classification Accuracy</th>
<th>PVE Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>86.12%</td>
<td>75.39%</td>
</tr>
<tr>
<td>20</td>
<td>85.81%</td>
<td>68.25%</td>
</tr>
<tr>
<td>30</td>
<td>85.83%</td>
<td>74.82%</td>
</tr>
</tbody>
</table>

Figure 2: Subset of training data from two different classes generated by Gaussian mixtures.

4. Experimental Simulation Results

Two different kinds of textured images with four gray levels are experimented: one has segmentation mask modeled by Gibbs distribution of a first order neighborhood system (see Figure 3(a)) and the other mask is artificially created by hand drawing (see Figure 3(b)). The corresponding textured images are shown in Figure 3(c) and (d). The textures are also generated by a second order Gibbs sampling procedures.

Figure 3: (a) The segmentation mask modeled by Gibbs distribution of second order. (b) The segmentation mask with shapes artificially created by hand drawing. (c) The corresponding textured image for (a). (d) The corresponding textured image for (b).

We compared NNMRF with that of [4] which determines parameter values by solving linear equations using histogramming. The two textures which constitute Figure 3(c) and 3(d) are generated from Gibbs distributions with parameters listed in Table 2 and these images are shown in Figure 4(a) and 4(b). From our simulations, the parameters estimated are not accurate enough for segmentation task (see Table 2 for comparison) and the segmented images after first and 150 iterations are shown in Figure 5(a) to 5(d).

For the NNMRF, the models of two different textures are built by training two BPNNs of 8 inputs (according to second order neighborhood system assumption) and 2 outputs (two gray levels) with KL criterion described above. The segmentation process based on hierarchical Gibbs sampling is described in section 2.4. Figure 6(a) and 6(b) show the segmentation results after 150 iterations of Gibbs sampling. The segmentation results can be further improved by smoothing with a simple \(3 \times 3\) median filtering over the segmentation label images. (see Figure 6(c) and 6(d)).

Figure 4: (a), (b) Textures generated from Gibbs distribution with parameters specified in Table 2.
5. Concluding Remarks

We proposed a neural network approach for modeling Markov random fields. Our designed simulation shows that a BPNN trained with Kullback-Leibler criterion can provide a fairly good estimate of Bayesian a posteriori probability. While conventional MRF modeling relies on difficult parameter estimation techniques, our method gives a simpler and more reliable solution which provides conditional probabilities required by Gibbs sampler. The proposed technique is verified by successful application to textured image segmentation tasks.

References