Project 3: Markov Random Field for Image Segmentation

Wenting Li

December 19, 2018

1 Introduction

Bayes Networks have a shown powerful capability by constructing a directed graph, but in some domains like the image processing and computer vision, the undirected graph mode brings us more convenience to represent the mutual dependence between nodes. Markov Random Filed (MRF) or Markov Network using an undirected graph $G = (V, E)$ is able to characterize the spatial or non-causual dependencies of random variables satisfying the Markov Conditions (MC). Here the Markov Condition means any node X_i of the graph is independent to other nodes if its neighbor is given, or

$$p(X_i|X_{-1}) = p(X_i|N_i) \tag{1}$$

where X_{-i} denotes the nodes not including the node X_i and N_i is the neighbor of X_i. Based on the MC property, the pairwise local independence [1] and global independence [2] Markov properties can be obtained.

Then the crucial point of MRF is the computation of the joint probability. As most MRF can be factorized according to the maximum clique [3], the introduction of the potential function of cliques significantly augments the efficiency of calculating joint probability. According to the Hammersley–Clifford theorem the joint probability distribution can be factorized as the product of the potential function of the maximum cliques , or

$$p(X) = \frac{1}{Z}\Pi_{c\in\mathbb{C}}\Psi(X_c) \tag{2}$$

where the potential function $\Psi(X_c)$ can be arbitrarily defined by any non-negative functions, and Z is the partition function. Although the freedom of defining the potential function brings great

[1] Pairwise Independence means any pair of nodes are independent if all other nodes are given.

[2] Global independence of MRF means that if all the paths are blocked between two nodes X_i and X_j, these two nodes are independent.

[3] Maximum Clique $\mathbb{C}$ is unique denoting the smallest cliques that cover all the nodes.

convenience to the application of MRF, the computational complexity of Z may become intractable if the network is large, thus properly selecting the potential function is paramount for a MRF model.

One popular way of defining the potential function is the log-linear potential function, i.e.,

$$\Psi(X_c) = \exp(-w_c E(X_c)) \tag{3}$$

where w_c is the parameter and $E(\cdot)$ is the energy function. The benefit of choosing log-linear potential function is that the joint probability can be written as the Gibbs distribution and the multiplication is converted to summation. There are various ways of defining potential functions and the MRF structures for the diverse applications.

The most common applications of MRF include the image reconstruction, image segmentation, 3D vision and object labeling. The basic idea of modeling image with MRF include the following considerations: 1) the the pixels of images can be treated as nodes of an undirected graph; 2) the hidden variables corresponding to the pixels can represent the meanings or properties of various pixels; 3) the joint probability of the pixels and the hidden variables can be utilized for various applications, such as segmentation of the images.

Therefore, the aim of this project is to segment the background and foreground of an image using the Label-Observation MRF model. The labels X_i are defined to represent the background and foreground respectively. The intensity Y_i is continuous observations. The Potts model is required to model the correlations between any two nodes X_i, X_j, and the intensity Y_i are assumed to follow the Gaussian distribution. The bias term in this project is ignored. The parameters of the MRF model are given, and the problem is to determine the labels X_i of each nodes.

The remaining parts of this report are following: Section 2 talks about the theoretical analysis of the MRF, and the Iterated Conditional Modes (ICM) to solve the issue of background and foreground segmentation, and then Section 3 validates the theory with the numerical experiment. The results with various initial conditions are analyzed, and then Section 4 concludes the results. The matlab codes are attached in the end.

2 Theory and Analysis

Due to the various structures and definition of potential functions, there are a variety of MRF models. Among them the pairwise MRF models have been extensively applied. For the discrete random variables, there are Potts model, Ising model and the label-observation models; as for the continuous random variables, the Gaussian graphical models can be applied. Besides the potential functions characterizing the correlations between nodes, the bias term for each node is also suggested to improve the performance by considering the prior distribution.

One specific application of MRF is to segment the background and foreground of images. The pixels X_i of images are treated as the random variables that have binary variables, representing the foreground and backgrounds, and each random variable is connected with a label Y_i that describes the intensity of the corresponding pixel. To segment the background and foreground, the special structure of "Label-Observation MRF" is constructed, as shown in Fig. 1. There are two types of random variables, the labels $X_i \in \{0, 255\}$ and the observations variables $Y_i \in [0, 255]$ are continuous. All the X_i are fully connected but there are no connections between Y_i.

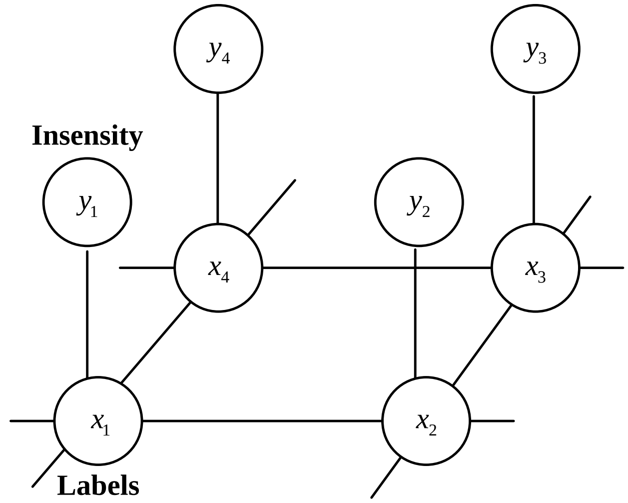


Fig. 1: *The Label-Observation MRF Model*

With this model established, the joint probability can be constructed based on the potential function of the X_i's and Y_i, X_i. The following will talk about the definition of these potential functions and one iterative algorithm that can implement this method efficiently–Iterative Conditional Mode (ICM) algorithm.

2.1 MRF model for image segmentation

The label of X_i determines whether Y_i belongs to the background or forewordground in this project, or the label can indicate the class that the pixel i is. All the X are connected while there is no connection among observation Y. This model is called the label-observation model.

The joint probability $p(X,Y)$ of this model can be computed by the product of the local potential functions of the maximum cliques, which can be any proper non-negative function, and unitary potential functions,

$$p(X,Y) = \frac{1}{Z}\Pi_{i,j}\Psi(X_i,X_j)\Pi_i\Phi(X_i,Y_i) \tag{4}$$

where Z is the partition function to normalize,$\Psi(X_i,X_j)$ is the pairwise potential function and $\Phi(X_i,Y_i)$ is to the unitary potential function.

In this project, the pairwise energy function is defined based on the Potts model, and the delta impulse function is employed as follows,

$$E(X_i,X_j) = 1-\delta(X_i - X_j) \tag{5}$$

where δ is the delta impulse function and it equals to 1 when it's argument is 0.

As the observations Y_i are assumed to follow the Gaussian distribution, the unary energy function is defined as

$$E(X_i, Y_i) = -\ln N(\mu_x, \sigma_x^2) \tag{6}$$

$$= -\ln[\frac{1}{\sigma_x\sqrt{2\pi}}\exp(-\frac{1}{2}(\frac{Y_i - \mu_x}{\sigma_x})^2)] \tag{7}$$

where μ_x, δ_x are the mean and standard deviation of the Gaussian distribution of Y_i given $x = 255$ and $x = 0$ respectively.

By ignoring the bias terms, the joint probability distribution for all pixels $X = \{X_i\}$and their intensities $Y = \{Y_i\}$ can be defined as

$$p(X, Y) = \frac{\exp[-\Sigma_{x_i \in V}\alpha E(X_i, Y_i) - \Sigma_{x_i \in V}\Sigma_{y_j \in N_{X_i}}\beta E(X_i, X_j)]}{Z} \tag{8}$$

where α and β are the unary and pairwise potential parameters and N_{X_i} is the 4-neighbor of X_i, and Z is the partition function.

Based on the log-linear potential function through (8), the joint probability can be obtained, but the unitary potential function can be various. The aim of the unitary potential function is to capture the relationship between labels X and observations Y. Moreover, adding some bias terms can also benefit this model but in this project the bias items are ignored.

2.1.1 MAP Inference

Given the observations Y of the images, the segmentation labels X can be computed by maximum a posterior probability $p(X|Y)$,

$$X^* = \arg\max_X p(X|Y) \tag{9}$$

$$= \arg\max_X p(X, Y) \tag{10}$$

$$= \arg\min_X \Sigma_{x_i \in V}\alpha E(X_i, Y_i) + \Sigma_{x_i \in V}\Sigma_{y_j \in N_{X_i}}\beta E(X_i, X_j) \tag{11}$$

There are multiple methods of computing the inference, including the approximation methods and the exact inference methods. The following section will apply the popular approximation inference method: Iterated conditional modes (ICM).

2.2 ICM

This algorithm is to approximate the global optimal inference by iteratively computing the local optimal inference. This algorithm is efficient but is influenced by different initial conditions.

a. As the conditional probability $p(X|Y)$ can be approximated by the product of local conditional probability, and according to the Markov condition, the local conditional probability can be calculated within the variables in the neighbor N_{X_i} of X_i,

$$p(X|Y) \approx \Pi_i p(X_i|X_{-i}, Y) \tag{12}$$

$$= \Pi_i p(X_i|X_{N_{X_i}}, Y_i) \tag{13}$$

$$\tag{14}$$

b. Thus each pixel i can be updated by maximizing the local conditional probability,

$$X_i^* = \arg\max_{X_i} p(X_i | X_{N_{X_i}}, Y_i) \quad (15)$$

$$= \arg\max_{X_i} p(X_i, X_{N_{X_i}}, Y_i) \quad (16)$$

$$= \arg\min_{X_i} \alpha E(X_i, Y_i) + \Sigma_{y_j \in N_{X_i}} \beta E(X_i, X_j) \quad (17)$$

where X_{-i} denotes the labels not including X_i.

The main drawback of this method is that initialization. The ICM can be easilty got stuck at some local optimal. Thus a proper initalization is necessary.

Algorithm 1 The Iterated Conditional Modes(ICM) Method

1: Input: MRF parameters $\mu_0, \sigma_0, \mu_{255}, \sigma_{255}, \alpha, \beta$
2: Initialization the matrix X by a zeros matrix.
3: **while** not converge **do**
4:

$$X_i = \arg\min_{X_i} \alpha E(X_i, Y_i) + \Sigma_{y_j \in N_{X_i}} \beta E(X_i, X_j)$$

5: where $i \in V$
6: Output: X

There are different ways of initializing the X, and the results can be influenced accordingly.

3 Experimental Results

3.1 Datasets

Fig. 2: *The image before segment*

Note the image is in PNG format. You need an image reader to read its values. Also, the MRF model should be over the entire image lattice of dimensions 640x360. 4-neighborhood should be used to identify the neighbor for each pixel.

(a) Initialization with all zeros *(b) Labels with all zeros initial conditions*

Fig. 3: Main figure caption

3.2 Tasks

Given the Y (pixel intensities) in attached image (640 by 360) as well as the MRF model parameters below,

$$\mu_0 = 106.2943, \sigma_0 = 5.3705, \mu_{255} = 141.6888, \sigma_{255} = 92.0635 \quad (18)$$

$$\alpha = 0.3975, \beta = 2.3472 \quad (19)$$

Perform the following tasks:

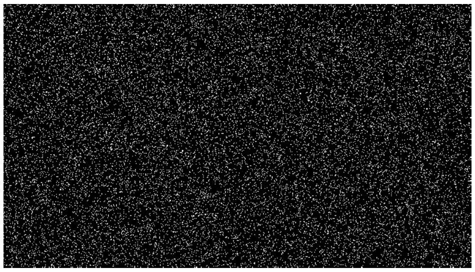
- Implement the Iterated Conditional Modes (ICM) method to perform MAP inference to find X, i.e., the label for each pixel (255 or 0) that maximizes $p(X|Y)$ to produce the segmented binary images, where the foreground pixel has an intensity of 255 and the background pixel intensity of 0.

The labels X are initialized by various ways, and the corresponding labels output by the ICM are displayed. When using all zeros to initialize the X, the labels are shown in (b) Fig. 3. The ICM algorithm with this initialization converge after 26 iterations. The results clearly segment the plane from the background.

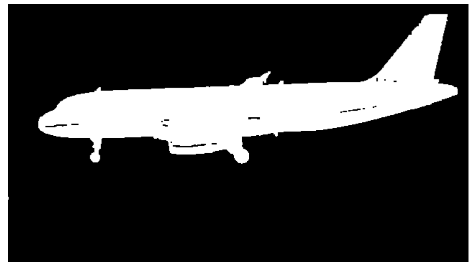
3.3 discusses and displays the image segmentation results under different initializations and iterations

All zeros are proper initialization and can converge pretty fast, as shown in Fig. 3, and if initialized by the random combination of 0 and 255 with different ratios as shown from the Fig. 4 to 6, the segmentation performance can be various. These is due to the limitation of ICM that only local optimal is ensured but not global optimal.

The ratio of the number of 255 and the total number of labels is variant from 10% to 70%, and the corresponding labels calculated by ICM are compared. The (a) of Fig. 4 to Fig. 6 show the different initialization with various ratios of 255 values.

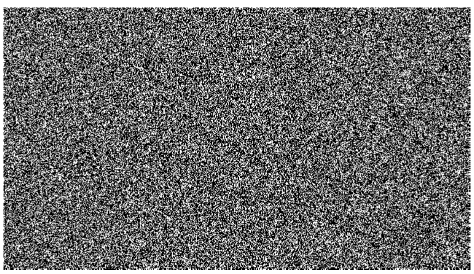


*(**a**) Initialization with 90% zeros and 10% 255 values*

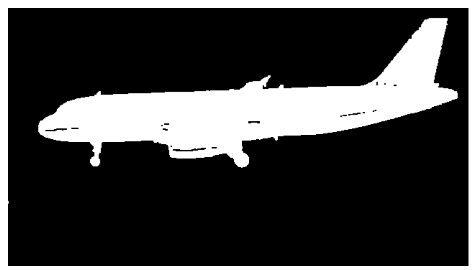


*(**b**) Lables output after 25 iterations with initialization of (a)*

Fig. 4: *Segmentation with the initialization of (a)*

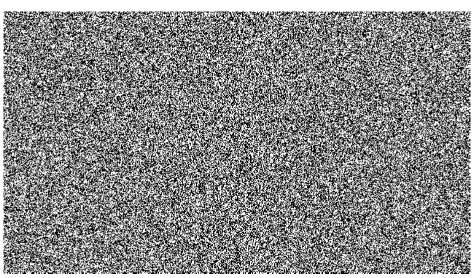


*(**a**) Initialization with 50% zeros and 50% 255 values*

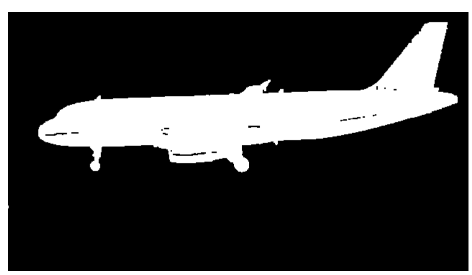


*(**b**) Lables output after 12 iterations with initialization of (a)*

Fig. 5: *Segmentation with the initialization of (a)*



*(**a**) Initialization with 30% zeros and 70% 255 values*



*(**b**) Lables output after 22 iterations with initialization of (a)*

Fig. 6: *Segmentation with the initialization of (a)*

1. When ratio of 255 is 10%, it takes about 25 iterations to converge and segmentation result is similar to that using all zeros initialization.
2. When ratio of 255 is 50%, it takes about 11 iterations to converge.
3. When ratio of 255 is 70%, it takes about 15 iterations to converge.

The segmentation results are almost the same.

In addition, here the 4-neighbor is applied but other ways of defining neighbors may obtain different results.

4 Conclusions

The ICM algorithm is validated on the image dataset, and the background is segmented from the foreground of the image. Due to the different initialization, the segmentation results and the number of iterations of ICM for the same dataset can be various. For this project, the results are not sensitive to the initialization, but ICM can only obtain the local optimal rather than the global optimal. Thus the issues of how to select a better initialization need the trial and error. In addition, the way of selecting neighboring is not studied. Here only the 4-neighbor is required but whether other ways of defining neighbors may or may not produce excellent results with different initialization has not been fully considered in this project.

What I have learned is that each algorithm has its advantages and limitations, so having a clear understanding of the theoretical assumptions before applying the algorithm is significant, otherwise it is not easy to interpret the bad results due to the algorithm itself. In addition, this project is only focusing on the implementation of the ICM, but more importantly, why the image segmentation is defined by such model and the why the parameters are given in this way have not deeply studied here, but these issues can be future work when really dealing with the realistic problems.

5 Source code

Add the path of the dataset 'Proj3 fall18.png', and then run code of 'ICM'. The resulting labels will be displayed in the image.

ICM

```
clc; clear all; close all;
%%%%%%%%%%%%%%%%% Iterated Conditional Modes (ICM) %%%%%%%%%%%%%%%%%%%%%%%%
%% load data
[Y1 ] = imread('Proj3_fall18.png','png' );
Y = double(Y1');
%% parameters
iter_num =100;
```

```
mu0 = 106.2943; sigma0= 5.3705; mu255 = 141.6888; sigma255= 92.0635;
alpha = 0.3975; beta= 2.3472;
%% initialize with different ratio of 255 values
X =  zeros(size(Y));
[row, col] = size(X);
seg = [0,255];  % X only has two possible values
%% different initialization by setting different ratio of 255 to 0
% ratio = 0.5;
% total = 640*360;
% index = randi(total,total*ratio,1);
% X(index) = 255;
% imshow(X');
err = zeros(1, iter_num);
for iter = 1:iter_num
for i =  1:row
    for j = 1:col
         E0 = potential(X, i,j, 0, Y );
         E255 = potential(X, i,j, 255, Y );
         [minE, classE] = min([E0, E255]);
         err(iter) = err(iter) + abs(X(i,j) - seg(classE));
         X(i,j) = seg(classE);
    end
end
fprintf('The error is %f after %d iterations \n', err(iter), iter);
    if err(iter) < 1e-3
        break
    end
end
imshow(X')
```

```
% this function is to compute the energy functions
function E = potential(X, i,j, value, Y )
        mu0 = 106.2943; sigma0= 5.3705; mu255 = 141.6888; sigma255= 92.0635;
        alpha = 0.3975; beta= 2.3472;
        neighbor =  find_4nei(X,i,j);
        if value == 0 %% if Xij = 0
            Ex0y = -log(1/(sqrt(2*pi)*sigma0)* exp((Y(i,j) - mu0)^2/(-2*
               sigma0^2)));
            Ex0x =  sum(neighbor~= 0);
            E = alpha * Ex0y + beta* Ex0x;
        elseif value == 255 %% if Xij = 255
            Ex255y = -log(1/(sqrt(2*pi)*sigma255)* exp((Y(i,j) - mu255)
               ^2/(-2*sigma255^2)));
```

```
            Ex255x =  sum(neighbor~= 255);
            E  = alpha * Ex255y + beta* Ex255x;
        end
end
```

```
% this function is to find the 4 neighbor of Xij
function nei  = find_4nei(X,i,j)
    [row, col] = size(X);
    nei = [];
    if 2<= i
       nei = [nei X(i-1,j)];
    end
    if i <= (row - 1)
       nei = [nei X(i+1, j)];
    end
    if j >= 2
       nei = [nei X(i, j-1)];
    end
    if j <= (col - 1)
       nei = [nei X(i, j+1)];
    end
end
```