Image edge detection using the hidden Markov model based on normal distribution with optimum expectation

Hassan Sharghi gallejoog¹, Reza Hashemi², Mehdi Rajabi³

¹. Department of Statistics, Ardabil branch, Islamic Azad University, Ardabil, Iran
². Dept. of Statistics, Razi University, Kermanshah, Iran,
³. Department of Mathematics Ardabil branch, Islamic Azad University, Ardabil, Iran

Corresponding Author email: mhd_rajabi@yahoo.com

ABSTRACT: THE HIDDEN MARKOV METHOD IS A ROBUST MODEL WHICH IS EMPLOYED IN VARIOUS FIELDS OF SCIENCE. ALTHOUGH IT HAS FREQUENTLY BEEN USED IN EDGE DETECTION THE WAVELET TRANSFORMATION HAS ALWAYS BEEN INCLUDED IN THE PROCESS. IN THIS PAPER, WE MODELED THE HIDDEN MARKOV METHOD BASED ON THE NORMAL DISTRIBUTION FOR IMAGE EDGE DETECTION (WITHOUT USING THE WAVELET TRANSFORMATION). THE ESTIMATION OF THE INCLUDED PARAMETERS WILL BE PRESENTED BY USING THE EM ALGORITHM FOLLOWED BY THE HIDDEN STATES, WHICH ARE CONSIDERED AS EDGES DETERMINED BY THE VITERBI ALGORITHM. FINALLY, WE WILL COMPARE OUR PROPOSED ALGORITHM WITH THAT OF CANNY AND OTHER ALGORITHMS WHICH ARE BASED ON THE HIDDEN MARKOV MODEL AND THE WAVELET TRANSFORMATION.

Key words: hidden Markov; wavelet; Viterbi algorithm; edge detection; EM algorithm.

INTRODUCTION

Edge detection is the first step and one of the most important components of image processing, which could be viewed as preprocessing. The general approach of the edge detection is allocation to each pixel of the image a number between zero and 250, which determines the level of the energy of the pixel. A great difference in the level of the energy of neighboring pixels indicates an edge in the image. The edge detection algorithm is based upon detection of points with various amounts of energy, which is actually the edge of the image.

The Canny algorithm, which is quite famous, works with local maximum produced by signal processing and using the Hsian matrix of edge detection.

The hidden Markov model is a strong method to analyze correlated data with possible change points because of its strong mathematical background, leading to a great number of its applications. In this context, the hidden Markov model is used to distinguish the pixels with "high" and "low" energy amounts as image edges.

The first application of the hidden Markov model to image edge detection was introduced by Romberg (1999). Another edge detection model was introduced by Sun (2004) based on the hidden Markov model and the wavelet. Among recent works we can refer to Zhang (2009) which used a zero mean normal and a Laplace distribution as distribution generators of the hidden states of pixels. The present work can be considered as an improvement over the Zhang's model in that the mean will be estimated based upon the intensity of the points with a high level of energy in the image.

THE HIDDEN MARKOV MODEL FOR IMAGE EDGE DETECTION

In the Hidden Markov Model, we assume that observations $O_1, O_2, O_3, \ldots, O_T$ are produced by hidden states $q_1, q_2, q_3, \ldots, q_M$. We also assume that these hidden states are produced by a normal distribution. Given that the image data are based on their amount of energy, the edge of the images occur at the points with energy jumps. Therefore, these points can be regarded as transitions in hidden states.

In order to locate image edges, $M$ normal distributions with different means will be considered in the model, shown in Figure 1 for $M=2$.

Based upon these observations, the estimations of parameters will be conducted using the EM algorithm, in which the maximum likelihood estimation for incomplete data of the HMM model will be locally optimized.
THE PARAMETERS OF THE HIDDEN MARKOV MODEL

An HM model with the following parameters will be considered:

1) \( N \) Number of states in the model. These states is denoted by \( S = \{s_1, s_2, s_3, \ldots, s_N\} \). The states in the arbitrary time \( t \), is denoted by \( q_t \). \( Q = \{q_1, q_2, \ldots, q_T\} \) indicates the total states belonging to time \( T \).

2) \( M \) Number of symbols \( \{1, 2, 3, \ldots, M\} \), used to explain the observations.

3) A set of probability transitions \( A = \{a_{ij}\} \), whose components will be defined as follows:

\[
1 \leq j \leq N, 1 \leq i \leq N, t \leq T
\]

\[
a_{ij} = p(q_{t+1} = s_j \mid q_t = s_i) = \begin{cases} 1 & \text{if } i = j \text{ and } t = 1 \ldots T \text{, } i, j = 1, \ldots, N \\ \sum_{j=1}^{N} a_{ij} & \text{if } t = 1, \ldots, T \\
\end{cases}
\]

4) \( B = \{b_j(o_t)\} \) a set of probability of observations in each state, where

\[
b_j(v_t) = p(o_t = v_j \mid q_t = i) = \begin{cases} 1 & \text{if } o_t = v_j, t = 1 \ldots T, j = 1, \ldots, M \\ \sum_{j=1}^{M} b_j(o_t) & \text{if } t = 1, \ldots, T \end{cases}
\]

5) \( \pi = \{\pi_i\} \) The vector of initial distribution.

\[
\pi_i = p(q_1 = s_i) = \begin{cases} 1 & \text{if } i = 1 \ldots N \\ \sum_{i=1}^{N} \pi_i & \text{if } \end{cases}
\]

Using these notations, the hidden Markov model will be denoted by \( \lambda = (A, B, \pi) \)

THE FORWARD-BACKWARD ALGORITHM

In order to estimate the parameters of the model, we will need to use forward/backward algorithms. The forward variable can be defined as \( \alpha(i) \)

\[
\alpha_i(i) = p(o_1, \ldots, o_t, s_i = s_j \mid \lambda) \quad i = 1, \ldots, N, t = 1, \ldots, T
\]

which is the probability of observation of partial sequence \( O = o_1, o_2, \ldots, o_t \) (up to \( t \)), and in time \( t \) will be at state \( s_i \), given the following \( \lambda \) model.

The following procedure will be employed to determine the forward variable:

6) Initial valuation : \( \alpha_1(i) = \pi_i b_i(o_1) \quad i = 1, \ldots, N \)

7) Induction:

\[
\alpha_i(j) = \sum_{i=1}^{N} \alpha_i(i) p_j(o_i) \quad j = 1, \ldots, N, t = 1, \ldots, T - 1
\]

\[
P(O \mid M) = \sum_{i=1}^{N} \alpha_i(i)
\]

8) Final step:

The backward variable can be defined as:

\[
\beta_i(i) = p(o_{t+1}, o_{t+2}, \ldots, o_T \mid q_t = s_i, \lambda)
\]

This variable is equal to the probability of observations \( O_{t+1}, O_{t+2}, \ldots, O_T \) Given that it is in time \( t \) and at state \( s_i \). This variable can be estimated in the backward procedure.

1) Initial valuation: \( \beta_1(i) = 1 \quad i = 1, \ldots, N \)

2) Induction

\[
\beta_i(i) = \sum_{j=1}^{N} \beta_i(j) p_i(o_i) \quad i = 1, \ldots, N, t = 1, \ldots, T - 1
\]

3) Final value:

\[
P(O \mid \lambda) = \sum_{i=1}^{N} \beta_i(i) b_i(o_i) \pi_i
\]

The probability of transition from the state \( s_i \) at time \( t \) to state \( s_j \) at time \( t+1 \) will be denoted by \( \xi(i, j) \)
And defined as follows:

\[ \xi(i, j) = P(q_i = s_j, q_{i+1} = s_j | o_1, o_2, ..., o_t, \lambda) \quad j, i \in \{1, ..., N\} \quad t = 1, ..., T \]

Using the forward/backward variable, the probability can be determined by:

\[ \xi(i, j) = \frac{\alpha_i(i) a_{ij} b_j(o_{i+1}) \beta_{i+1}(j)}{\sum_{i} \sum_{j} \alpha_i(i) a_{ij} b_j(o_{i+1}) \beta_{i+1}(j)} \]

The probability of being at state \( s_i \) in time \( t \) based on the sequence of observations and the given model will be denoted by \( \gamma_t(i) \) and defined as:

\[ \gamma_t(i) = P(q_t = s_i | o_1, o_2, ..., o_t, \lambda) \quad i \in \{1, ..., N\} \quad t = 1, ..., T \]

and can be determined using the forward/backward variables as follows:

\[ \gamma_t(i) = \frac{\alpha_i(i) \beta_t(i)}{\sum_i \alpha_i(i) \beta_t(i)} \]

THE EM ALGORITHM

The EM algorithm is an iterative method for estimation, and it was first presented by Dempster et al (1997). This was employed by Bilms (1998) for the hidden Markov model. The EM algorithm can be used in two general cases.

In cases with missing values

In cases in which estimation of complete data is complicated.

In image processing, we can use the EM algorithm because of complications in parameter estimation due to the huge number of pixels in the image. For example, an ordinary picture has 512×512=262144 pixels.

This algorithm has two steps. Step E, i.e., the expectation step, used for incomplete data, and Step M, which maximizes this conditional expectation in respect to \( \lambda \). Then this estimated value will be regarded as the initial value for the next occurrence of the equation. This is repeated so that convergence is achieved.

The E step

The likelihood function for the complete data is as follows:

\[ L(\lambda | O, Q) = \prod_{t=1}^{T} P(q_t | q_{t-1}, \lambda) \]

Where \( \lambda \) is of the form \( \lambda = (B, A, \pi) \)

And the likelihood function for incomplete data could be achieved by

\[ L_{\gamma} (\lambda | O) = P(O | \lambda) = \prod_{t=1}^{T} P(q_t | q_{t-1}, \lambda) \]

In order to employ the EM algorithm, assuming that the parameters are known, using the conditional expectations of the logarithm of the likelihood function of the complete data, the q function can be written as:

\[ \mathbb{E}(\lambda | \lambda^{(k)}) = E(\log(P(O, Q | \lambda)) | O, \lambda^{(k)}) \]

The M step

In the M step, using the derivative of the Q function with respect to the parameters, we can achieve the maximum value of this function. Hence, we have

\[ \hat{\lambda} = \gamma(i) \]

\[ \tilde{a}_i = \frac{a_i(i)}{\sum \gamma(i)} \quad i = 1, 2, ..., T-1, i, j \in N \]

In order to estimate the expectations of the normal distributions via \( b_i(o_i) \) and \( O_i N(\mu, \sigma^2) \) \( i = 1, ..., M \), we have

\[ b_i(o_i) = \phi(o_i | \mu, \sigma^2) \]

\[ = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(o_i - \mu)^2}{2\sigma^2} \right) \]
and

\[ \sum_{t=1}^{T} \left( \sum_{i=1}^{N} \log(b_i(o_t))P(s_t = i | O, \lambda(i))^2 \right) \]

\[ = \sum_{t=1}^{T} \left( \sum_{i=1}^{N} \log\left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(o_t - \mu_i)^2}{2\sigma^2} \right\} \right) P(s_t = i | O, \lambda(i))^2 \right) \]

Using derivations with equal to zero, then we have the estimation

\[ \hat{\mu} = \frac{\sum_{t=1}^{T} (o_t)P(s_t = i | O, \lambda(i))}{\sum_{t=1}^{T} P(s_t = i | O, \lambda(i))} \]

The algorithm starts with an initial value and the algorithms of the IV section will be re-estimated, and through repetition of this procedure, we can achieve a local maximum of \( \lambda \).

**THE VITERBI ALGORITHM**

In order to find the best sequence of hidden states \( S_1, S_2, \ldots, S_T \) given the observations \( o_1, o_2, \ldots, o_T \), the \( \delta(i) \) can be defined as:

\[ \delta(i) = \max_{q_{i-1}} P(q_i, q_{i-1}, \ldots, q_1 = s_1, o_1, o_2, \ldots, o_T | \lambda) \]

Regarding to the first observations \( o_1, o_2, \ldots, o_T \) produced a path which end to \( s_i \). It can be showed that \( \delta(j) \) could be obtained from the equality:

\[ \delta(j) = \max_{i \leq j < N} \left[ \delta(i) a_{ij} b_j(o_j) \right] \quad j = 1, \ldots, N \quad t = 2, \ldots, T \]

Figure 2. Method of obtaining the optimized path.

In order to find the sequence of optimum states the \( \delta(j) \) should be maximized for each \( t \) and \( j \), using the vector \( \Psi_t(j) \).

The procedure of optimization of the sequence of states is as follows:

1) Initial values

\[ \delta(i) = \max_{i} P(q_i = i, o_i) = \pi_i b_i(o_i) \quad i = 1, \ldots, N \]

\[ \psi_t(i) = 0 \]

2) Recursive equations

\[ \delta(t) = \max_{i \leq j < N} \left[ \delta(t) a_{ij} b_j(o_j) \right] \]

\[ \psi_t(j) = \arg\max_{i \leq j < N} \left[ \delta(t) a_{ij} \right] \quad t = 2, \ldots, T, \quad j = 1, \ldots, N \]

3) Final values and optimized path

\[ p^* = \max_{i \leq j < N} \left[ \delta(t) \right] \]

\[ q^*_t = \arg\max_{i \leq j < N} \left[ \delta(t) \right] \quad t = 2, \ldots, T \]

Using this procedure, hidden states for imagepixels could be obtained.
**IMAGE EDGE DETECTION**

In this step using the obtained hidden states in the section VI as a sequence, locations in which hidden states change will be regarded as the image edge, because change in hidden states is equivalent to change in the mean of the generator distribution and equivalently the source of observations. The method of detection edges of an image is shown in the figure 3.

![Diagram](image.png)

Figure 3. first row one row of image ,second row data of first row, and third row; the hidden state of second row and edge detection for it.

![Diagram](image.png)

Figure 4. Diagram of general form of presented model
INITIAL VALUE

For beginning convergence in the EM algorithm, an initial parameter is needed. We used the following parameters as default parameters for all the images.

Transition probability: \( p_{ij} = \begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix} \), probability of beginning state: \( \pi_i = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \), mean vector: \( \mu = \begin{pmatrix} 10 \\ 140 \end{pmatrix} \).

RESULTS

Wavelet transitions algorithms have been frequently used in edge detection. The main purpose of using these algorithms can be set as calculating derivative, declaring the maximum/minimum and places of energy jump (for more details read see). As mentioned in the Hidden Markov explanation, this model will calculate the best parameters to maximize the likelihood function by the EM algorithm. This action will maximize the difference between the hidden states and the result of the Viterbi algorithm will be the best state which most probably produced the observation. This action will compute the maximum and energy jump points more accurately and precisely than the wavelet transform. Since in comparison to its adjacent pixel, a pixel is considered an edge in addition to energy and energy direction, there should be a consideration of the whole energy of all pixels in the image and the probability of being an edge (by using the transition probability matrix the hidden states can be determined). As a result, edge detection can be done intelligently. This jump will be evaluated in proportion with the total image pixel’s jumps, which means just the amount of jump is not sufficient condition to be in the list of image edge points. This is the future that causes noise tolerance noise of the detected edges with our model.

As mentioned before, the wavelet transformation just causes computational overhead instead of problem solving, so there is no need to perform wavelet transformation before executing the Hidden Markov Model (HMM) due to the fact that the HMM itself will perform the actions of the wavelet transform.

Our proposed model uses the exact pixels of the image, so it will cause neither edge thickness nor quality degradation. On the other hand the computational overhead of the wavelet transformation will be omitted.

![Figure 5. image edge detection (a) and (f); original image, (b) and (g); canny edge detection, (c) and (h); WD-VHMT edge detection, (d) and (i); NWHMC edge detection, (e) and (j); edge detection with our algorithm](image)

CONCLUSION AND DISCUSSION

In this paper, edge detection has been conducted based on the hidden Markov model. This modeling was performed without the wavelet transformation, the detected edges of this model are noise tolerance and it has better performance in comparison with the Canny algorithm. In all previous literature based on the HMM, the wavelet transformation has been used which causes thick edges with low resolutions. These problems have been solved in our proposed model and due to omission of the wavelet transformation, computational overheads are omitted too. Furthermore, in previous papers such as NWHMC, HMM distributions with zero averages have been used which consequently lowered the accuracy of edge detection because in cases that the image has energy level skew using only variance could not detect well. We used average in our model too, so it is able to detect edges properly when the image level of energy has skew.
REFERENCES


