The 2-D Hidden Markov Model for Images, Its Extensions, and Applications

Jia Li

The Pennsylvania State University
Email: jiali@psu.edu
Challenges in Learning from Images

- Partition images into different classes:
  - A preliminary classification approach:
    - Block-wise feature extraction.
    - Classify blocks individually.

```
X_i \rightarrow \text{Feature Vector Generator} \rightarrow X_i \rightarrow \text{Classifier} \rightarrow Y_i
```

Source image

Classifier

\[ Y_i = C(X_i) \]

\[ Y_i \text{ in } (1, 2, \ldots, K) \]

Classified image
Original

Manual segmentation

CART, $P_e = 20.29\%$

2-D MHMM, $P_e = 11.57\%$
Challenges in Learning from Images (Cont.)

- Can a computer do this? (Automatic annotation)

- Build a dictionary of stochastic models that link pictorial information with textual description.
Outline

• Two dimensional hidden Markov model (2-D HMM)
  – Model assumptions
  – Model estimation
  – Computational complexity

• 2-D Multiresolution HMM

• Applications to supervised/unsupervised segmentation

• Applications to image annotation

• Conclusions
2-D Hidden Markov Model

- Feature vectors residing on a grid: \( \{u_{i,j}; 0 \leq i < h, 0 \leq j < w\} \).
- A hidden layer of states: \( \{s_{i,j}; 0 \leq i < h, 0 \leq j < w\} \).
- \( u_{i,j} \) are conditionally independent given \( s_{i,j} \).
- The states \( s_{i,j} \) are governed by a Markov mesh, specified by transition probabilities.

\( u | s: N(\mu, \Sigma) \)
Assumptions about States

- Denote \((i', j') \prec (i, j)\) if \(i' < i\), or \(i' = i\) and \(j' < j\).
- Transition probabilities:
  \[
P(s_{i,j} | \text{context}) = a_{m,n,l},
  \]
  where \(m = s_{i-1,j}, n = s_{i,j-1}\), and \(l = s_{i,j}\).
- Context: \(\{s_{i',j'}; (i', j') \prec (i, j)\}\).
- Given the state of a block, the class of the block is uniquely determined (a many to one mapping).
  - Example: Class 1 contains states 1, 2, 3; Class 2 contain states 4, 5, 6.
Assumptions about Feature Vectors

• Given its state, a feature vector follows a Gaussian distribution:

\[ b_s(u) \sim N(\mu_s, \Sigma_s) \]

• Relation to conditional Gaussian mixture distributions:
  – A state with an M-component Gaussian mixture can be split into M substates with single Gaussian distributions.
  – Transition probabilities are not constrained.
Classification based on 2-D HMM

1. **Training images** → **Compute features** → **Feature vectors on 2-D grids** → **Estimate a 2-D HMM**

2. **A new image** → **Compute features** → **Feature vectors** → **Find MAP combination of states** → **2-D HMM**

3. **Classified image** ← **Map states to classes** ← **A grid of states**
Estimation of 2-D HMM

- Parameters to be estimated:
  - Transition probabilities $a_{m,n,l}$, $m, n, l = 1, ..., M$.
  - Mean $\mu_m$ and covariance matrix $\Sigma_m$ of Gaussian distributions, $m = 1, ..., M$.

- The ML estimation of the parameters can be computed by the EM algorithms.

- Feature vector: $u_{i,j}$, states: $s_{i,j}$, classes: $c_{i,j} = C(s_{i,j})$, $(i, j) \in \mathbb{N}$, $\mathbb{N} = \{(i, j) : 0 \leq i < h, 0 \leq j < w\}$.

- The complete data $x = \{s_{i,j}, u_{i,j} : (i, j) \in \mathbb{N}\}$, and the incomplete data $y = \{c_{i,j}, u_{i,j} : (i, j) \in \mathbb{N}\}$. 
EM Iterations

- Given the current model estimation $\phi^{(p)}$, the mean vectors and covariance matrices are updated by

$$
\mu_m^{(p+1)} = \frac{\sum_{i,j} L_m^{(p)}(i, j) u_{i,j}}{\sum_{i,j} L_m^{(p)}(i, j)}
$$

$$
\Sigma_m^{(p+1)} = \frac{\sum_{i,j} L_m^{(p)}(i, j) (u_{i,j} - \mu_m^{(p+1)}) (u_{i,j} - \mu_m^{(p+1)})'}{\sum_{i,j} L_m^{(p)}(i, j)}
$$

- $L_m^{(p)}(i, j) = P(s_{i,j} = m \mid u_{i',j'}, c_{i',j'}, (i', j') \in \mathbb{N}; \phi^{(p)})$

  - The probability of being in state $m$ at block $(i, j)$ given all the observed feature vectors, classes and model $\phi^{(p)}$.

$$
L_m^{(p)}(i, j) = \sum_s I(m = s_{i,j}) \cdot \frac{1}{\alpha} I(C(s) = c) \times \prod_{(i', j') \in \mathbb{N}} a_{s_{i',j'}, s_{i'-1,j'}, s_{i,j'-1}, s_{i', j'}}^{(p)} \times \prod_{(i', j') \in \mathbb{N}} P(u_{i',j'} \mid \mu_{s_{i',j'}}^{(p)}, \Sigma_{s_{i', j'}}^{(p)})
$$
EM Iterations (Cont.)

• The transition probabilities are updated as follows:

\[ a_{m,n,l} = \frac{\sum_{i,j} H_{m,n,l}^{(p)}(i,j)}{\sum_{l'=1}^{M} \sum_{i,j} H_{m,n,l'}^{(p)}(i,j)} \]

\[ H_{m,n,l}^{(p)}(i,j) \] is the probability of being in state \( m \) at block \((i - 1, j)\), state \( n \) at block \((i, j - 1)\) and state \( l \) at block \((i, j)\) given the observed feature vectors, classes, and model \( \phi^{(p)} \).

\[ H_{m,n,l}^{(p)}(i,j) = \sum_{s} I(m = s_{i-1,j}, n = s_{i,j-1}, l = s_{i,j}) \times \frac{1}{\alpha} I(C(s) = c) \cdot \prod_{(i',j') \in N} a_{s_{i-1,j'},s_{i',j'-1},s_{i',j'}}^{(p)} \times \prod_{(i',j') \in N} P(u_{i',j'} | \mu_{s_{i',j'}}^{(p)}, \Sigma_{s_{i',j'}}^{(p)}) . \]
Computation Issues

• The brute force computation of $L^{(p)}_m(i, j)$ and $H^{(p)}_{m,n,l}(i, j)$ is not feasible.

• Suppose there are $w \times w$ blocks in an image and the number of states in each class is $M_0$, then the computational order is $w^2 M_0^2 w^2$.

• Computational order can be reduced to $w M_0^2 w$ by introducing forward and backward probabilities, but it is still intensive.

• We approximate $L^{(p)}_m(i, j)$ and $H^{(p)}_{m,n,l}(i, j)$ by assuming that the single most likely state sequence accounts for virtually all the likelihood of the observations (Viterbi training).

• A suboptimal algorithm based on a dynamic programming technique is applied to find the state sequence with nearly maximum a posteriori (MAP) probability.
Maximum Likelihood State Sequence

- $T_i$ denotes the sequence of states for blocks lying on diagonal $i$, i.e., $(s_{i,0}, s_{i-1,1}, \cdots, s_{0,i})$.

- It can be shown that the probability of a state sequence of the image equals

$$P(s_{i,j}, (i,j) \in \mathbb{N}) = P(T_0) \cdot P(T_1|T_0) \cdot P(T_2|T_1) \cdots P(T_{m+n-2}|T_{m+n-3}).$$
• The sequence of states along a diagonal, $T_i$, serves as an “isolating” element in the expansion.

• Finding the MAP $\{s_{i,j}, (i, j) \in \mathbb{N}\}$ is equivalent to finding one that maximizes

$$P\{s_{i,j}, u_{i,j} : (i, j) \in \mathbb{N}\} = P(s_{i,j} : (i, j) \in \mathbb{N}) \prod_{(i,j)\in\mathbb{N}} P(u_{i,j} \mid s_{i,j}).$$

• Viterbi algorithm can be applied.
Viterbi Algorithm

- A minimum-cost search technique.
- Given codeword sequence $z = \{z_1, \ldots, z_T\}$, the cost function
  \[
  D_T(z) = \sum_{t=1}^{T-1} d(z_t, z_{t+1}).
  \]
- The cost function has a Markov-like property. Fix the codeword at $t$, the codewords preceding it have no effect on the codewords succeeding it in terms of cost.
Viterbi Algorithm (Continued)

• Denote the cost up to step $t$ by $D_t(z) = \sum_{\tau=1}^{t-1} d(z_\tau, z_{\tau+1})$, and $z(t) = \{z_1, z_2, ..., z_t\}$.

• Assume $z^* = \arg\min_z D_T(z)$, then

$$z^*(t) = \arg\min_{z(t): z_t = z^*_t} D_t(z)$$

• The minimization can be performed progressively.

• $\min_z D_T(z)$ can be computed by the recursive formulae

$$\theta_i(1) = 0 \quad 1 \leq i \leq M,$$

$M$ is the number of codewords

$$\theta_i(t) = \min_{j:1 \leq j \leq M} \{\theta_j(t-1) + d(j, i)\}$$

$1 < t \leq T, 1 \leq i \leq M$

$$\min_z D_T(z) = \min_{j:1 \leq j \leq M} \theta_j(T).$$

• Brute force minimization of $D_T(z)$ needs computation of order $M^T$, while the Viterbi algorithm reduces the computation order to $T \cdot M^2$. 
Computation Complexity

For both 1-D and 2-D, given the states of the shaded samples, the states of Part II are statistically independent of the states of Part I.

Total number of states: $M$

Number of possible state sequences for the shaded samples:

1-D: $M$

2-D: $M^4$

- In two dimensional case, the complexity problem cannot be fully solved by the Markov property.
The number of possible sequences of states at every position increases exponentially with the increase of blocks at the position.

If there are $M$ states, the amount of computation and memory are both in the order of $M^k$, where $k$ is the number of blocks at the position (still a problem!).
Suboptimal Viterbi Algorithm

At every position of the Viterbi transition diagram, only use $N$ out of all the $M^k$ sequences of states. The paths are constrained to pass one of these $N$ nodes.

The chosen $N$ sequences of states yield the largest likelihood for the feature vectors unconditioned on the states of previous blocks.

Fast algorithm is available for choosing the best $N$ sequences of states.
Outline

- Two dimensional hidden Markov model (2-D HMM)
  - Model assumptions
  - Model estimation
  - Computational complexity
- 2-D Multiresolution HMM
- Applications to supervised/unsupervised segmentation
- Applications to image annotation
- Conclusions
Multiresolution HMM

Motivations:

– Incorporate features at multiple resolutions.
– Provide more flexibility for modeling statistical dependence.
– Reduce computation by representing context information hierarchically.

Formulation:

– An image is represented by feature vectors across several resolutions.
  
  * Feature vectors at resolution $r$: $u_{i,j}^{(r)}$.
  * Low resolution images are obtained by filtering, e.g., wavelet transforms.
The Hierarchical Structure

- Denote the collection of block indices at resolution $r$ by $\mathbb{N}(r) = \{(i, j); 0 \leq i < h \cdot 2^{r-1}, 0 \leq j < w \cdot 2^{r-1}\}$, $r \in \mathcal{R}$, $\mathcal{R} = \{1, ..., R\}$.

- Conditional independence of $u^{(r)}_{i,j}$ given $s^{(r)}_{i,j}$.

- Markovian property across resolutions:

\[
P\{s^{(r)}_{i,j}, u^{(r)}_{i,j}; r \in \mathcal{R}, (i, j) \in \mathbb{N}(r)\} = P\{s^{(1)}_{i,j}, u^{(1)}_{i,j}; (i, j) \in \mathbb{N}^{(1)}\} \times \\
P\{s^{(2)}_{i,j}, u^{(2)}_{i,j}; (i, j) \in \mathbb{N}^{(2)}| s^{(1)}_{k,l}; (k, l) \in \mathbb{N}^{(1)}\} \times \ldots \\
P\{s^{(R)}_{i,j}, u^{(R)}_{i,j}; (i, j) \in \mathbb{N}^{(R)}| s^{(R-1)}_{k,l}; (k, l) \in \mathbb{N}^{(R-1)}\}
\]
Transition Properties

• Let the child blocks at res. \( r \) of block \((k, l)\) at res. \( r - 1 \) be
  \[
  \mathbb{D}(k, l) = \{(2k, 2l), (2k + 1, 2l), (2k, 2l + 1), (2k + 1, 2l + 1)\}.
  \]

• Conditional independence given parent states:
  \[
  P\{s_{i,j}^{(r)}; (i, j) \in \mathbb{N}^{(r)} \mid s_{k,l}^{(r-1)}; (k, l) \in \mathbb{N}^{(r-1)}\} = \prod_{(k,l) \in \mathbb{N}^{(r-1)}} P\{s_{i,j}^{(r)}; (i, j) \in \mathbb{D}(k, l) \mid s_{k,l}^{(r-1)}\}
  \]

• Statistical dependence among the states of sibling blocks is characterized by a 2-D HMM.

• The transition probability \( a_{m,n,l}^{(r)}(s) \) depends on
  – the neighboring states in both directions, \( m \) and \( n \).
  – the state of the parent block, \( s \).
Estimation of the Multiresolution HMM

- Viterbi training is used to estimate the model.
- At each iteration, search for the maximum likelihood combination of states across all the resolutions, that is to maximize (assume 2 resolutions)

\[
\log P\{s_{k,l}^{(r)}, u_{k,l}^{(r)} : r \in \{1, 2\}, (k, l) \in \mathbb{N}^{(r)}\} = \log P\{s_{k,l}^{(1)}, u_{k,l}^{(1)} : (k, l) \in \mathbb{N}^{(1)}\} + \sum_{(k,l) \in \mathbb{N}^{(1)}} \log P\{s_{i,j}^{(2)}, u_{i,j}^{(2)} : (i, j) \in \mathbb{D}(k, l) | s_{k,l}^{(1)}\}.
\]

- For every fixed state of a parent block, find the maximum likelihood combination of states for its child blocks.
- The maximum log likelihood of the states of the child blocks is added to the log likelihood of the parent block with the corresponding state.
- The Viterbi algorithm is applied to determine the states of the parent blocks.
- This method is used recursively if there are more than two resolutions.
Outline

• Two dimensional hidden Markov model (2-D HMM)
  – Model assumptions
  – Model estimation
  – Computational complexity

• 2-D Multiresolution HMM

• Applications to supervised/unsupervised segmentation

• Applications to image annotation

• Conclusions
Aerial Image Classification

- Man-made versus natural area
- $512 \times 512$ gray-scale images with 8 bits per pixel.
- Six images were used in the experiment.
- Block sizes at all the resolutions are $4 \times 4$.
- Six-fold cross-validation is used in evaluation.
Feature Extraction

• Intra-block features based on the discrete cosine transform (DCT):

\[ f_1 = D_{0,0} ; \quad f_2 = |D_{1,0}| ; \quad f_3 = |D_{0,1}| ; \]
\[ f_4 = \sum_{i=2}^{3} \sum_{j=0}^{1} |D_{i,j}| ; \quad f_5 = \frac{\sum_{i=0}^{1} \sum_{j=3}^{3} |D_{i,j}|}{4} ; \]
\[ f_6 = \frac{\sum_{i=2}^{3} \sum_{j=2}^{3} |D_{i,j}|}{4}. \]

• DCT coefficients at different frequencies reflect different variation patterns in the image.

• Difference between the average intensity of a block and its upper or left neighbor is used as an inter-block feature.

• Low resolution images are LL band images of Daubechies 4 wavelet transform.
Result

- Classification error rates by cross-validation

<table>
<thead>
<tr>
<th>Iteration</th>
<th>CART</th>
<th>LVQ1</th>
<th>HMM</th>
<th>MHMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2263</td>
<td>0.2161</td>
<td>0.1904</td>
<td>0.1733</td>
</tr>
<tr>
<td>2</td>
<td>0.1803</td>
<td>0.1918</td>
<td>0.1765</td>
<td>0.1636</td>
</tr>
<tr>
<td>3</td>
<td>0.2899</td>
<td>0.2846</td>
<td>0.2034</td>
<td>0.1782</td>
</tr>
<tr>
<td>4</td>
<td>0.2529</td>
<td>0.2492</td>
<td>0.2405</td>
<td>0.2051</td>
</tr>
<tr>
<td>5</td>
<td>0.1425</td>
<td>0.1868</td>
<td>0.1834</td>
<td>0.1255</td>
</tr>
<tr>
<td>6</td>
<td>0.2029</td>
<td>0.1813</td>
<td>0.1339</td>
<td>0.1157</td>
</tr>
<tr>
<td>Ave.</td>
<td>0.2158</td>
<td>0.2183</td>
<td>0.1880</td>
<td>0.1602</td>
</tr>
</tbody>
</table>

- Result for the example image:

CART, $P_e = 20.29\%$

HMM, $P_e = 13.39\%$

MHMM, $P_e = 11.57\%$
Document Image Segmentation

- Text and photograph segmentation of document images.
- Features are defined according to the distribution patterns of wavelet coefficients in high frequency bands.

Original image

Manually classified image
Results of Document Image Segmentation

• Compare the result of HMM with that of CART.
Application to a Gaussian Mixture Source

- The Gaussian source has two classes with equal priors. For both classes, vectors have Gaussian distributions with zero mean values, but different variances.

- We assume the classes are produced by a 2-D hidden Markov model.

- The transition probabilities are as below. The two classes are symmetric.
Results for the Gaussian Mixture Source

• If feature vectors are treated as independent random vectors, the Bayes classifier yields classification error rate of 0.264.

• The HMM algorithm yields classification error rate of 0.243.

• Significant amount of information is lost if independence is assumed.
Image Segmentation

An image → Compute features → Feature vectors

Estimate a 2-D HMM → Find MAP combination of states → Segmented image
Original

Mixture model

A variation of 2-D HMM
Original

Original

Mixture model

Mixture model

A variation of 2-D HMM

A variation of 2-D HMM
Original

Mixture model

A variation of 2-D HMM

Original

Mixture model

A variation of 2-D HMM
Outline

• Two dimensional hidden Markov model (2-D HMM)
  – Model assumptions
  – Model estimation
  – Computational complexity

• 2-D Multiresolution HMM

• Applications to supervised/unsupervised segmentation

• Applications to image annotation

• Conclusions
Automatic Annotation

- The ALIP system: Automatic Linguistic Indexing of Pictures.
- Training process:

- 600 categories of images are profiled by the 2-D MHMM.
• Annotation process:

Image to be indexed

Feature Extraction

Model Comparison

Likelihood 1

Model Comparison

Likelihood 2

Model Comparison

Likelihood N

Stored textual descriptions about concepts

Significance Processor

Statistically significant index terms:
food, indoor, cuisine, dessert

Image DB
Figure 1: Training images used to learn the concept of *male* with the category description: “man, male, people, cloth, face”.
Figure 2: Annotations automatically generated by our computer-based linguistic indexing algorithm. The dictionary with 600 concepts was created automatically using statistical modeling and learning. Test images were randomly selected outside the training database.

Figure 3: Test results using photos not in the COREL collection. P: Photographer annotation. Words appeared in the annotation of the 5 matched categories are underlined. Words in parenthesis are not included in the annotation of any of the 600 training categories.
Conclusions

• 2-D HMM and its multiresolution extension
  – Capture dependence by an underlying “Markov pyramid”.

• Model estimation

• Applications:
  – Supervised/unsupervised segmentation: *simultaneous optimization of pixel classes*.
  – Automatic linguistic indexing: *profiling hundreds of image categories*. 

43