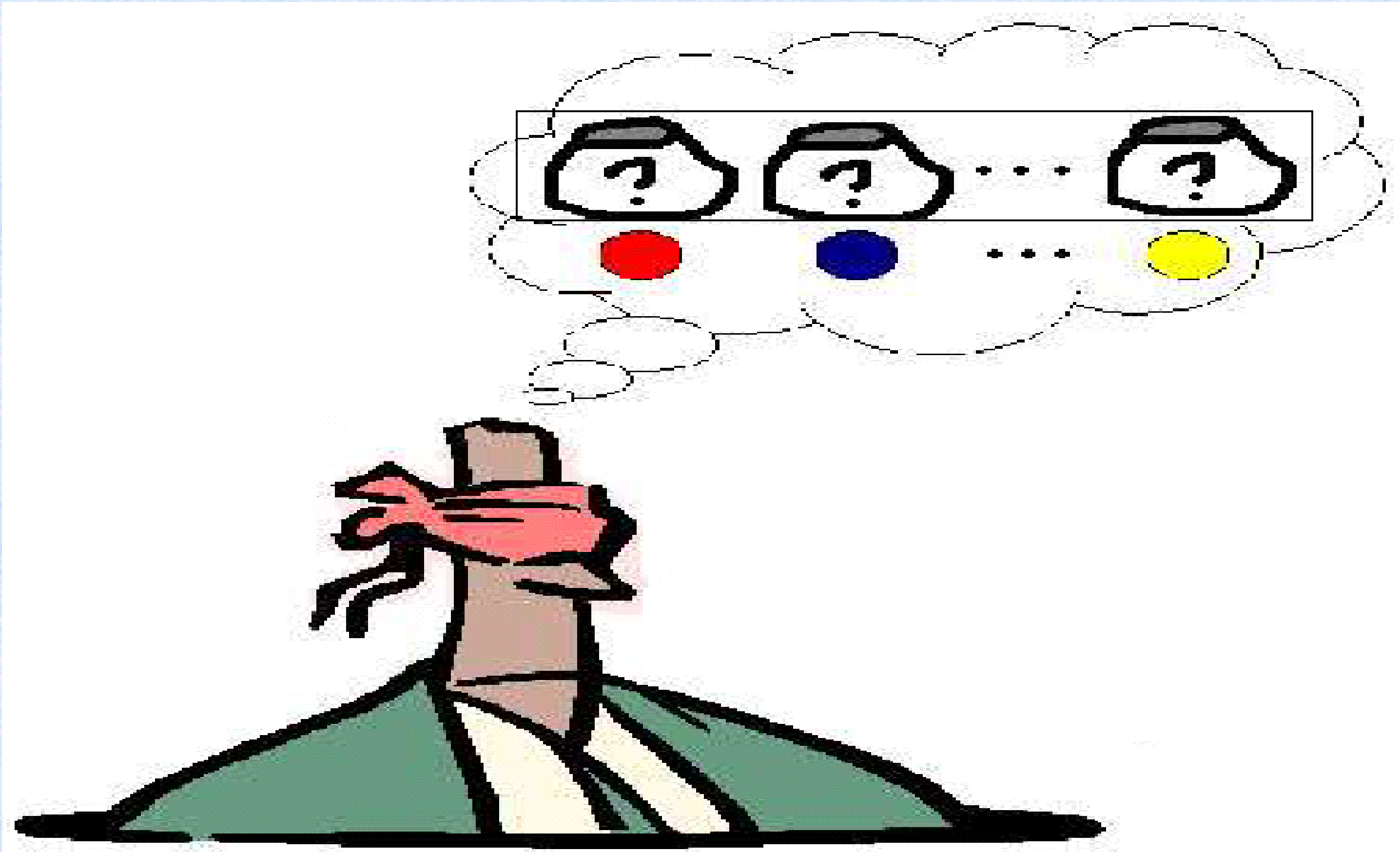
Image classification by a Two Dimensional Hidden Markov Model

Author: Jia Li, Amir Najmi and Robert M. Gray

Presenter: Tzung-Hsien Ho

Hidden Markov Chain



Goal:

- To implement a novel classifier for image segmentation

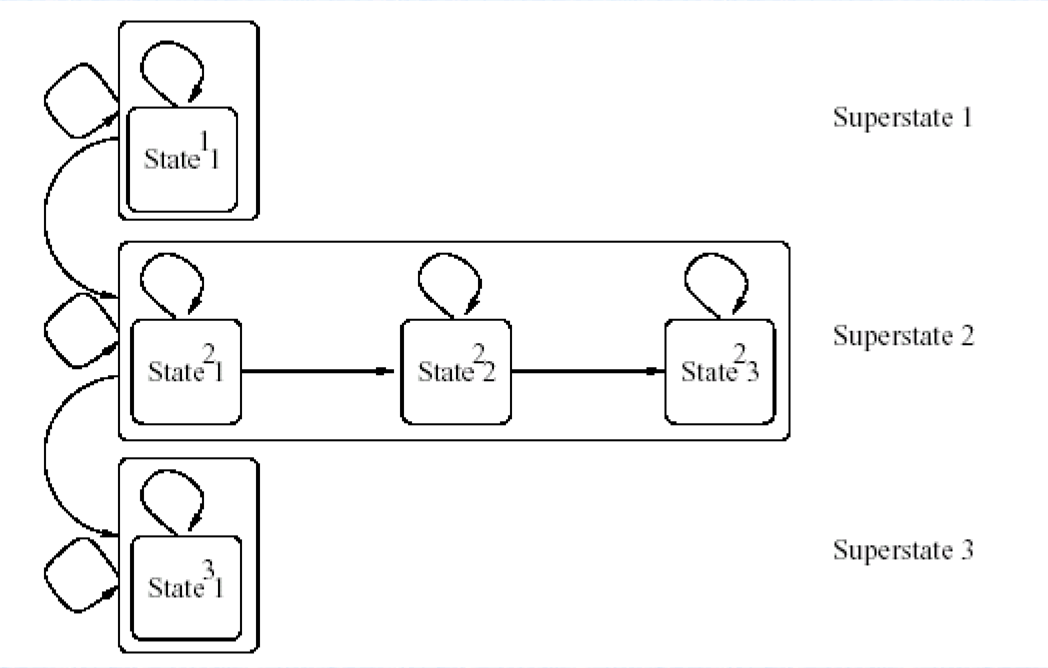
What's over there?

- LVQ (Learning vector quantization)
- BVQ (Bayes vector quantization)
- CART (Classification and regression trees)
- Pseudo-2D HMM
- Others… (PCA, LDA, neural networks, …)

Pseudo 2D HMM

- Extension of 1D case
- Not real 2D model since it does not connect all the possible states
- There is a "superstate" existing in the first element of each row. All the superstates consist of a markov chain.
- Each row consists of an independent markov chain.

Pseudo 2D HMM



Markov Random Field (MRF)

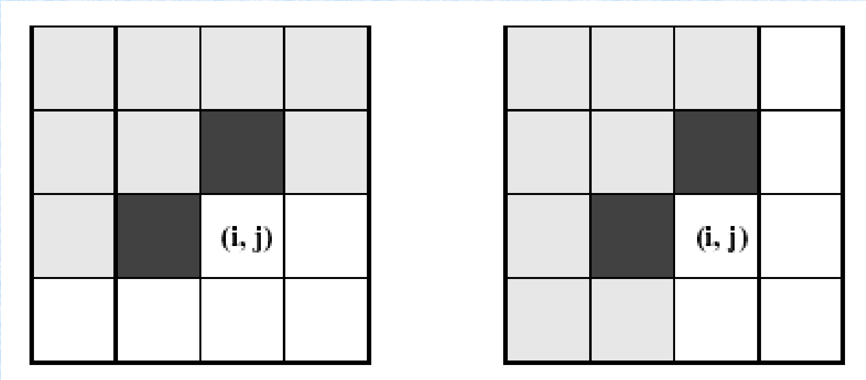
- A tool to encode contextual constraints into the prior probability.

  Define (1) neighbors (2) cliques (3) prior clique potential

  Derive (4) likelihood energy (5) posterior energy

Assumptions in the paper

- Causal Markov Random Field



$$P(s_{i,j} \mid s_{i',j'}, u_{i',j'} : (i',j') \in \tilde{\Psi} \cup \Psi) = P(s_{i,j} \mid s_{i-1,j}, s_{i,j-1})$$

Mainframe of the paper

- 2D HMM Model

  1. Expectation Maximum (EM)

  2. Viterbi Algorithm for 2D case

- Feature Selections:

  1. DCT coefficients & spatial derivatives of the average intensity value of blocks

  2. Wavelets & Laplacian Measurement

2D HMM

- 1. Training:

(a) Divide training images into non-overlapping blocks

(b) Extract the features of each block

(c) Select the number of states for the 2D-HMM

(d) Estimate model parameters based on the feature vectors (v) and their hand-labeled class (c)

Training Process

- Expectation Maximization
- Known: Mapping function from S and C

  E-Step: Compute Q(Φ| Φ^p)

$$E(\log f(\mathbf{x} \mid \phi') \mid \mathbf{y}, \phi^{(p)}) = \frac{1}{\alpha} \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{y}, \phi^{(p)}) \cdot \sum_{(i,j) \in \mathbb{N}} \log a'_{s_{i-1,j}, s_{i,j-1}, s_{i,j}} + \frac{1}{\alpha} \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{y}, \phi^{(p)}) \sum_{(i,j) \in \mathbb{N}} \log P(u_{i,j} \mid \mu'_{s_{i,j}}, \Sigma'_{s_{i,j}}).$$

where $P(\mathbf{s} \mid \mathbf{y}, \phi^{(p)}) = \frac{1}{\alpha} I(C(\mathbf{s}) = \mathbf{c}) \cdot \prod_{(i,j) \in \mathbb{N}} a^{(p)}_{s_{i-1,j}, s_{i,j-1}, s_{i,j}} \cdot \prod_{(i,j) \in \mathbb{N}} P(u_{i,j} \mid \mu^{(p)}_{s_{i,j}}, \Sigma^{(p)}_{s_{i,j}})$

Training Process (cont.)

- Expectation Maximization

  M-Step: Choose Φ to maximize $Q(\Phi | \Phi^p)$

$$a'_{m,n,l} = \frac{\sum_{(i,j)\in\mathbb{N}} H^{(p)}_{m,n,l}(i,j)}{\sum_{l=1}^{M}\sum_{(i,j)\in\mathbb{N}} H^{(p)}_{m,n,l}(i,j)} .$$

$$H^{(p)}_{m,n,l}(i,j) = \sum_{\mathbf{s}} I(m = s_{i-1,j}, n = s_{i,j-1}, l = s_{i,j}) P(\mathbf{s} \mid \mathbf{y}, \phi^{(p)})$$

$$\mu'_m = \frac{\sum_{i,j} L^{(p)}_m(i,j) u_{i,j}}{\sum_{i,j} L^{(p)}_m(i,j)} ,$$

$$L^{(p)}_m(i,j) = \sum_{\mathbf{s}} I(m = s_{i,j}) P(\mathbf{s} \mid \mathbf{y}, \phi^{(p)})$$

$$\Sigma'_m = \frac{\sum_{i,j} L^{(p)}_m(i,j)(u_{i,j} - \mu'_m)(u_{i,j} - \mu'_m)^t}{\sum_{i,j} L^{(p)}_m(i,j)} .$$

Compute it? Thank for diagonal

- Forward: $\theta_T(d) = P\{\mathbf{s}(d) = T, \mathbf{u}(\tau) : \tau \le d \mid \mathrm{M}\}$

$$\theta_{T_d}(d) = \sum_{T_{d-1}} \theta_{T_{d-1}}(d-1) \cdot P(T_d \mid T_{d-1}, \mathrm{M}) \cdot P(\mathbf{u}(d) \mid T_d, \mathrm{M})$$

- Bakward: $\beta_T(d) = P\{\mathbf{u}(\tau) : \tau > d \mid \mathbf{s}(d) = T, \mathrm{M}\}$

$$\beta_{T_d}(d) = \sum_{T_{d+1}} P(T_{d+1} \mid T_d, \mathrm{M}) \cdot P(\mathbf{u}(d+1) \mid T_{d+1}, \mathrm{M}) \cdot \beta_{T_{d+1}}(d+1)$$

Where Td: state sequences at dth diagonal

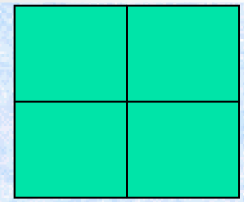
$$L_m(i,j) = \sum_{T_d : T_d(i,j)=m} \frac{\theta_{T_d}(\Delta(i,j)) \cdot \beta_{T_d}(\Delta(i,j))}{P(\mathbf{u}, \mathbf{c} \mid \mathrm{M})} \quad C(m) = c_{i,j}$$

$$H_{m,n,l}(i,j) = \sum_{T_d} \sum_{T_{d-1}} \frac{\theta_{T_{d-1}}(\Delta(i,j)-1)}{P(\mathbf{u}, \mathbf{c} \mid \mathrm{M})} \cdot \quad \begin{matrix} T_{d-1}(i-1,j)=m, T_{d-1}(i,j-1)=n \\ T_d(i,j)=l \end{matrix}$$

$$[P(T_d \mid T_{d-1}, \mathrm{M}) P(\mathbf{u}(d) \mid T_d, \mathrm{M}) \cdot \beta_{T_d}(\Delta(i,j))]$$

$$C(m) = c_{i-1,j}, C(n) = c_{i,j-1}, C(l) = c_{i,j}$$

Simple example



1. Define # of states
2. Divided the image into sub-blocks and extract the features (U)of each block.
3. Given C(i,j) and random assign $a_{m,n,l}$, μ_m and Σ_m
4. Use the parameters to find forward and backward probability based on each diagonal
5. Calculate $L_m(i, j)$ and $H_{m,n,l}(i, j)$
6. Use L and H to update $a_{m,n,l}$, μ_m and Σ_m
7. Go back to step 4

2D HMM

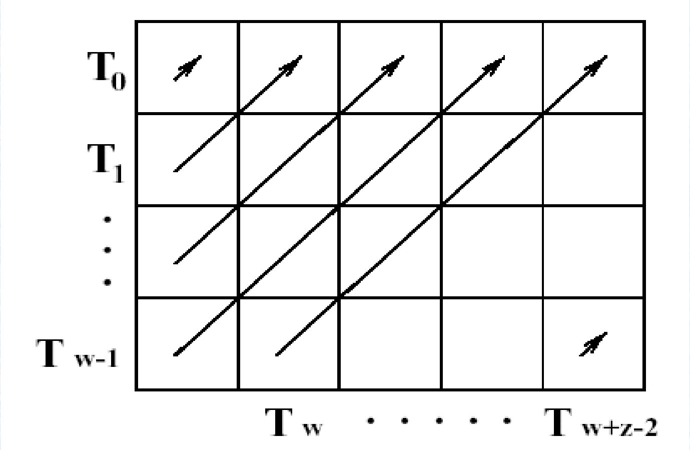
- Recognition

(a) Generate feature vectors for the testing image

(b) Search for the set of classes with maximum a posteriori probability given the feature vectors according to the trained 2D HMM

2D Viterbi algorithm

- The computational complexity is $(NxM-1)^{mo}$
- Any way to simplify?
- Using forward probabilities and applying the blocks of the diagonal to create markovian isolation sequence



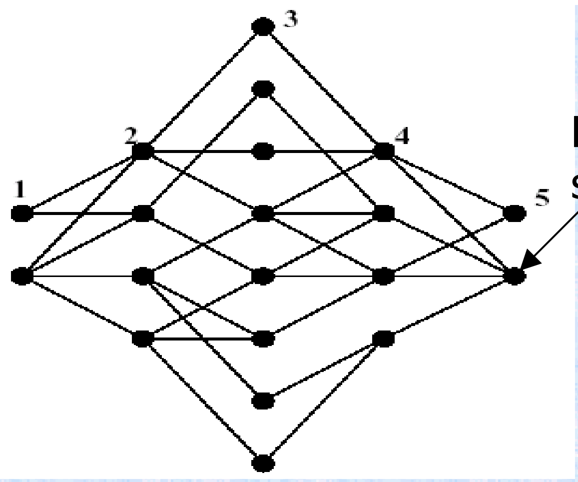
2D Viterbi algorithm

- Reminder

  Viterbi is used to maximize P(I,O| Φ)

- For 2D, maximize $P\{s_{i,j}, u_{i,j} : (i,j) \in \mathbb{N}\}$

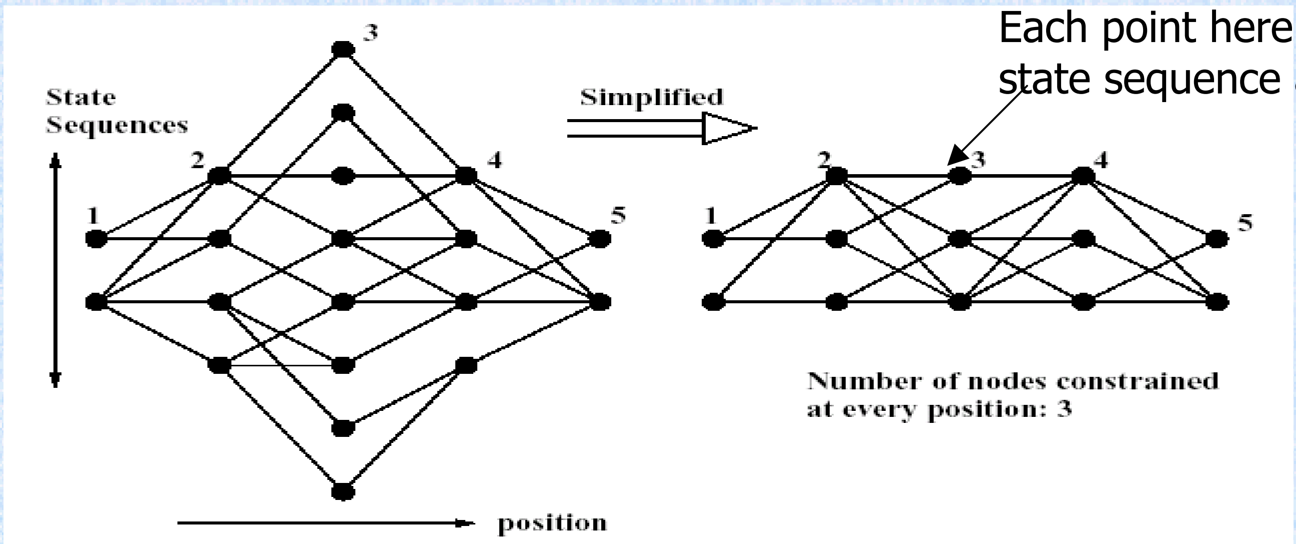
$$P(T_0) \cdot P(T_1 \mid T_0) \cdot P(T_2 \mid T_1) \cdots P(T_{w+z-2} \mid T_{w+z-3}) \prod_{(i,j)\in\mathbb{N}} P(u_{i,j} \mid s_{i,j})$$



Each point here infers a state sequence at Td

2D Viterbi Algorithm (cont.)

- It is infeasible. NP completeness
- Proposition: path-constrained Viterbi algorithm



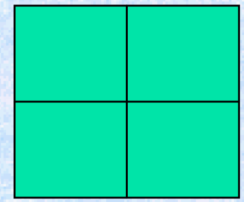
2D Viterbi Algorithm (cont.)

- In each diagonal, find the best N routes based on $\sum_{i=1}^{v} \gamma_{i,s_i}$ where v is the number of diagonal terms.
- We can simplify the constraints above by finding $\max_{s_i}^{-1} \gamma_{i,s_i}$ for each i
- Best route searching
- Not a precise model. Two improvements

  1. Choose large N

  2. Divide original image into sub-images

Simple Example



- 1. Image input
- 2. Extract the features
- 3. In each diagonals, pick the best N sequences.
- 4. Use the normal viterbi to calculate

$$P(T_0)\cdot P(T_1 \mid T_0)\cdot P(T_2 \mid T_1)\cdots P(T_{w+z-2} \mid T_{w+z-3}) \prod_{(i,j)\in \mathbb{N}} P(u_{i,j} \mid s_{i,j})$$

Now Ti is limited by the local maximizor

5. Find the highest score of S and C(s)=C

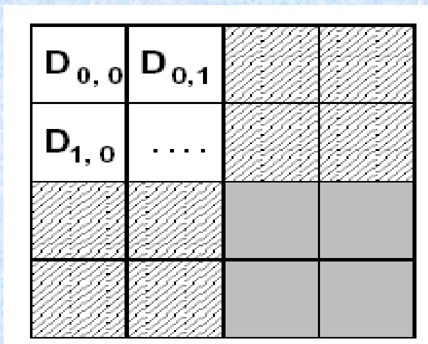
Aerial Image Segmentation

- Goal: Try to differentiate the man-made or nature scenes from the aerial image
- # of states: 5 (3~6 are all available) for nature scene. 9 (7~10 are all available) for man-made scene
- Size of N: 32
- Block size: 4x4

Aerial Image Segmentation

- Feature selection:

1. Intra-block feature: DCT coefficients



1. $f_1 = D_{0,0}$; $f_2 = |D_{1,0}|$; $f_3 = |D_{0,1}|$;

2. $f_4 = \sum_{i=2}^{3} \sum_{j=0}^{1} |D_{i,j}|/4$;

3. $f_5 = \sum_{i=0}^{1} \sum_{j=2}^{3} |D_{i,j}|/4$;

4. $f_6 = \sum_{i=2}^{3} \sum_{j=2}^{3} |D_{i,j}|/4$.

2.Inter-block feature: Average intensity difference between neighbors (U,L)

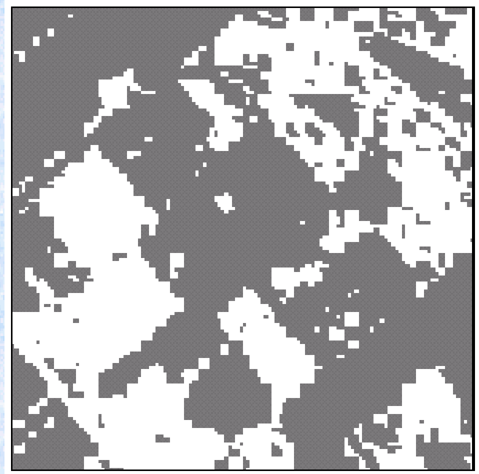
Results

- Man-made scene as targets
- Compared with three other methods (CART, LVQ (learning vector quantization) and BVQ)

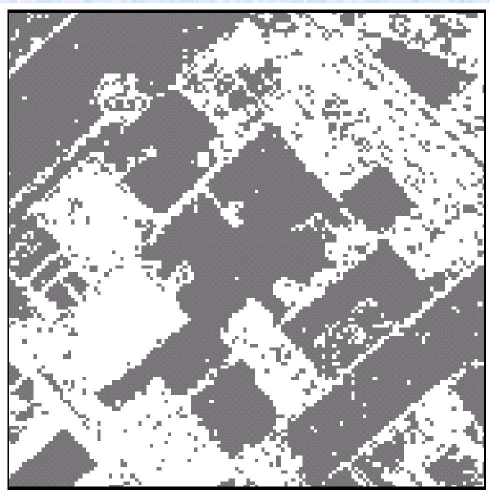
Algorithm	sensitivity	specificity	PVP	P_e
2-D HMM	0.7795	0.8203	0.8381	0.1880
CART 1	0.8528	0.7126	0.7530	0.2158
CART 2	0.8097	0.7340	0.7505	0.2408
LVQ1	0.8187	0.7419	0.7691	0.2183

Results

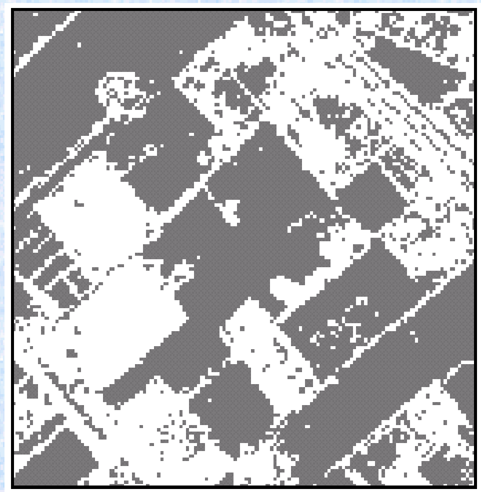
Source



HMM



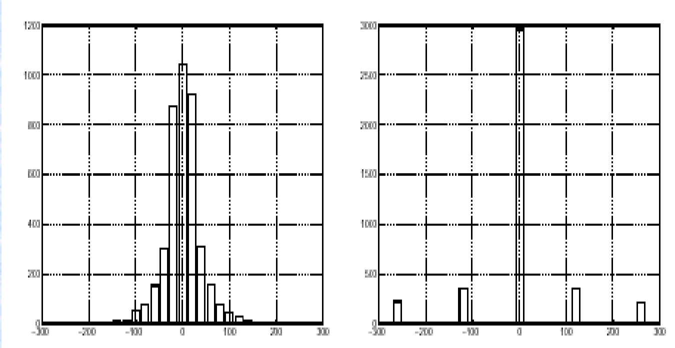
CART1



LVQ1

Documentation segmentation

- Goal: Segmentation of document images into text and photograph
- # of states: 4 (2~5 are available)
- Block size: 8x8
- Features:



Wavelet coefficients (Haar wavelet)

(1) Measure the goodness of match between the observed distribution and the Laplacian distribution. (χ^2)

(2) Measure the likelihood of the wavelet coefficients being composed by highly concentrated values. (L)

Documentation segmentation

- How the hell does it work? (LH HL HH)

χ^2 : Divide the histogram of the wavelet coefficients into bins. fi: relative frequency of bin i and Fi: probability of the Laplacian distribution.

$$\chi^2 = \sum_{i=1}^{k} (fi - Fi)^2 / Fi$$

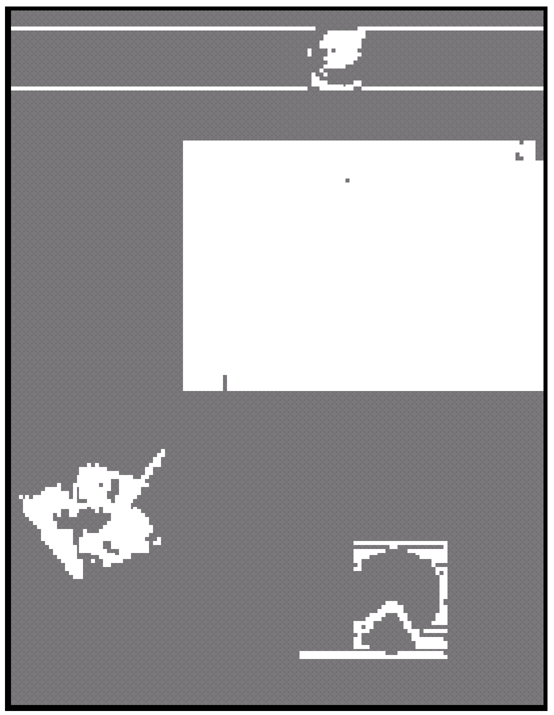
L : the weight sum of the concentration level β i.

β i is defined as the percentage of the data in a narrow neighborhood based on the total number of data in the zone.

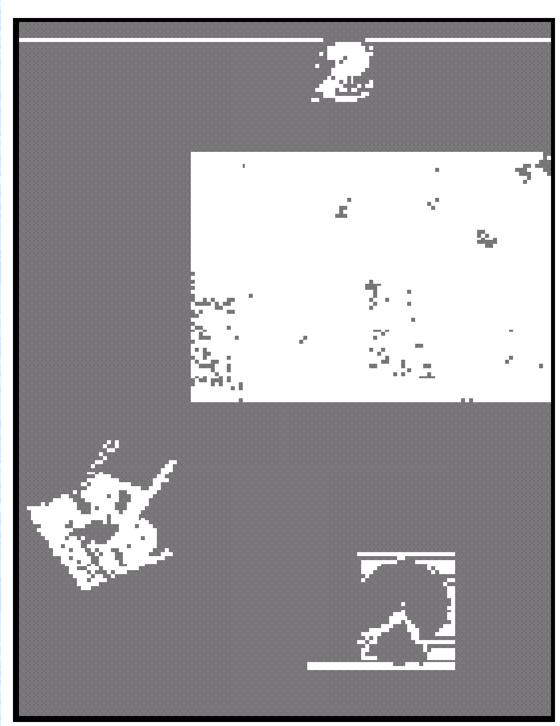
Results

Error rate: fewer than 2% for HMM and CART

Source



HMM



CART

Conclusion

- 2D HMM takes the inter-blocks relationship into account. That's why it stands out in most of the current classifier.
- 2D HMM is still limited by the computational power of the machine. This paper provides a roughly correct version.

Questions

- The computational cost of this algorithm is still high. Maybe not a good real-time processing algorithm.
- Parallel computing can be applied.
- Multi-scale (pyramid) processing may be able to assist the inter-block relations we lost in path-constrained viterbi algorithm.

Reference

1. Rakesh Dugad, U.B Desai, "A tutorial on hidden markov models", Technical Report No. SPANN-96.1, Bombay Powai, India May, 1996.
2. Jeff A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", Technical Report 97-021, International Computer Science Institute Berkley CA, April, 1998.
3. J. Li and R.M. Gray, "Text and Picture Segmentation by the Distribution Analysis of Wavelet coefficients," Proceedings of International Conference on Image Processing, Chicago, Oct. 1998
4. L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, vol. 77, pp. 257-286, 1989
5. S.Z. Li, "Markov Random Field Models in Computer Vision", ICCV 1994
6. JiangSheng Yu, "Expectation Maximization, An approach to parameter estimation", Class note, Peking University, China.

Your opinion