Hidden Markov Models

Andrew Davis

Markov Model

- Occupies one of finite set of states at any given time (X_1, X_2, … , X_n)
  – could be characters, phonemes, weather conditions
- Probability of occupying given state determined solely by recent history
  – *k*-th order model depends on last *k* states
- Probabilities described by Matrix *A*
  – a_{ij} = P(system in state *j* | system was in state *i*)
  – a_{ij} are time independent
  – rows add to 1

Markov Model

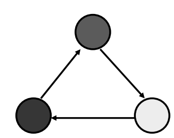
- Example of probability matrix (weather forecasting)

$$
\begin{array}{r} \\ \text{sun} \\ A = \text{cloud} \\ \text{rain} \end{array}
\begin{array}{c} \begin{array}{ccc} \text{sun} & \text{cloud} & \text{rain} \end{array} \\ \begin{bmatrix} 0.50 & 0.375 & 0.125 \\ 0.25 & 0.375 & 0.625 \\ 0.25 & 0.375 & 0.375 \end{bmatrix} \end{array}
$$

- Rows add to 1 → given one weather condition at t_1, the next condition was one of the three at t_0

Markov Model

- Simpler example – traffic light
  – red → green → yellow → red …



Hidden Markov Model

- Desired parameters unknown, determined from set of observable parameters, *Y*
  – words determined from sounds (speech recognition)
  – words determined from lines (optical character recognition)
  – weather determined from secondary phenomena (example)
- Matrix B describes probabilities of Y_k
  – b_{kj} = P(state Y_k observed | system was in state *j*)

Hidden Markov Model

- Example – observe condition of seaweed to determine weather (illustrative, not realistic)
- Columns add to 1 → given weather, seaweed will have some property

$$
B = \begin{array}{r} \\ \text{dry} \\ \text{dryish} \\ \text{damp} \\ \text{soggy} \end{array}
\begin{array}{c} \begin{array}{ccc} \text{sun} & \text{cloud} & \text{rain} \end{array} \\ \begin{bmatrix} 0.60 & 0.25 & 0.05 \\ 0.20 & 0.25 & 0.10 \\ 0.15 & 0.25 & 0.35 \\ 0.05 & 0.25 & 0.50 \end{bmatrix} \end{array}
$$

Hidden Markov Model

- HMM is a 3-tuple $\lambda = (\pi, A, B)$
  - A is probability matrix of hidden states
  - B is probability matrix of observable states
  - π is n-dimensional vector describing initial probabilities (t = 1)
- Speech processing
  - A is probability one phoneme follows another
  - B relates features of phoneme under analysis
- OCR
  - A is probability of next character
  - B is features of line being analyzed

Hidden Markov Model

- HMM has three principle issues
  - Evaluation
    - probability model generated observations
    - may have more than one model – choose best fit
  - Decoding
    - most likely state sequence given observation sequence
  - Learning
    - what is best λ

HMM Evaluation

- Want probability that model generated observed sequence
- Evaluate all possible sequences, calculate probabilities

$$P(Y^k) = \sum_{X^i} P(Y^k \mid X^i)P(X^i)$$

  - superscript indicates sequence

- Exponential with length of sequence

HMM Evaluation

- Define intermediate probabilities
  - $\alpha_t(j) = P(X_j \text{ at } t)$
  - $\alpha_{t+1}(j) = \sum_{I=1}^{n} [a_t(i)a_{ij}]b_{k_{t+1}j}$
- α_{ij} is P(moving to state j)
- $b_{jk_{t+1}}$ is P(specific observation made at this time)
- initialize $\alpha_1(j) = \pi(j)b_{k_1j}$
- Then

$$P(Y^k) = \sum_{I=1}^{n} \alpha_T(j)$$

HMM Evaluation

$$P(Y^k) = \sum_{i=1}^{n} \alpha_T(j)$$

- Forward algorithm
- Find model that maximizes this probability
  - *e.g.*, OCR, model could be individual word

HMM Decoding

- Need algorithm to determine likely sequence of states that produced sequence of observed states
- Can start at t = 1 and look for most probable next state, given observation
  - noise can result in bad decision
  - error compounded; illegal sequence
- Isolation vs. context
  - may result in different best guesses

HMM Decoding

- Go down observed sequence
  - record likelihood of hidden state being reached
  - keep pointer to most likely predecessor
- At end of sequence, choose final state based on history, then step back through earlier stages
- Viterbi Algorithm

HMM Decoding – Example

- Weather determination based on seaweed
  - guess $\pi = (⅓, ⅓, ⅓)$ (equal P of sun, cloud, rain)
- Say we observe *dry*, *dryish*, *soggy*, *soggy*
- What is most likely weather pattern to produce observations

HMM Decoding – Example

- Start with observed state *dry*
- P(*dry* | *sun*) = ⅓ · 0.6 = 0.2
- P(*dry* | *cloud*) = ⅓ · 0.25 = 0.0833
- P(*dry* | rain) = ⅓ · 0.05 = 0.0167

- *sun* is best guess
- Now find probability of *dryish*, given *dry* and *sun* on previous day

HMM Decoding – Example

- P(day 1 *sun* and day 2 *sun* and *dryish*) =
  0.2 · 0.5 · 0.2 = 0.02
- P(day 1 *cloud* and day 2 *sun* and *dryish*) = 0.00417
- P(day 1 *rain* and day 2 *sun* and *dryish*) = 0.000833
- Find P for day 2 *cloud* and day 2 *rain*
- We eventually find *sun* is best guess for day 2
- Do same for next two observed states

HMM Decoding – Example

- Eventually you'll find most likely final state is *rain*
- Use back pointers to get most likely sequence to produce *dry*, *dryish*, *soggy*, *soggy*
- You'll get *sun*, *sun*, *rain*, *rain*

HMM Learning

- Make initial estimate
- Use forward-backward (*e.g.*, Baum-Welch) algorithm to improve estimate
- Forward – probability of being in a (hidden) state at *t*, given the observed sequence
- Backward – probability of succeeding observation, given current state and time
- Train with reference observations

HMM Applications

- Speech recognition
- Optical character recognition
- Handwriting analysis
- Sign language (from video)
- Lip reading

Coupled HMM

- *A* matrices of two models
  - probabilistically related
- Can couple arbitrarily many models
- Example – speech recognition
  - audio
  - video

References

- *Image Processing, Analysis, and Machine Vision* by Sonka, Hlavac, Boyle
- http://en.wikipedia.org/wiki/Main_Page
- http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html