

Modeling Sequence Data: HMMs and Viterbi

CS4780/5780 – Machine Learning
Fall 2013

Thorsten Joachims
Cornell University

Reading:

Manning/Schuetze, Sections 9.1-9.3 (except 9.3.1)

Leeds Online HMM Tutorial (except Forward and Forward/Backward Algorithm)

(http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html)

Hidden Markov Model

- States: $y \in \{s_1, \dots, s_k\}$
 - Outputs symbols: $x \in \{o_1, \dots, o_m\}$
 - Starting probability $P(Y_1 = y_1)$
 - Specifies where the sequence starts
 - Transition probability $P(Y_i = y_i \mid Y_{i-1} = y_{i-1})$
 - Probability that one states succeeds another
 - Output/Emission probability $P(X_i = x_i \mid Y_i = y_i)$
 - Probability that word is generated in this state
- => Every output+state sequence has a probability

$$\begin{aligned} P(x, y) &= P(x_1, \dots, x_l, y_1, \dots, y_l) \\ &= P(y_1)P(x_1|y_1) \prod_{i=2}^l P(x_i|y_i)P(y_i|y_{i-1}) \end{aligned}$$

Estimating the Probabilities

- Given: Fully observed data
 - Pairs of emission sequence with their state sequence
- Estimating transition probabilities $P(Y_i | Y_{i-1})$

$$P(Y_i = a | Y_{i-1} = b) = \frac{\# \text{ of times state } a \text{ follows state } b}{\# \text{ of times state } b \text{ occurs}}$$

- Estimating emission probabilities $P(X_i | Y_i)$

$$P(X_i = a | Y_i = b) = \frac{\# \text{ of times output } a \text{ is observed in state } b}{\# \text{ of times state } b \text{ occurs}}$$

- Smoothing the estimates
 - Laplace smoothing -> uniform prior
 - See naïve Bayes for text classification
- Partially observed data
 - Expectation Maximization (EM)

Viterbi Example

$P(X_i Y_i)$	I	bank	at	CFCU	go	to	the
DET	0.01	0.01	0.01	0.01	0.01	0.01	0.94
PRP	0.94	0.01	0.01	0.01	0.01	0.01	0.01
N	0.01	0.4	0.01	0.4	0.16	0.01	0.01
PREP	0.01	0.01	0.48	0.01	0.01	0.47	0.01
V	0.01	0.4	0.01	0.01	0.55	0.01	0.01

$P(Y_1)$	
DET	0.3
PRP	0.3
N	0.1
PREP	0.1
V	0.2

$P(Y_i Y_{i-1})$	DET	PRP	N	PREP	V
DET	0.01	0.01	0.96	0.01	0.01
PRP	0.01	0.01	0.01	0.2	0.77
N	0.01	0.2	0.3	0.3	0.19
PREP	0.3	0.2	0.3	0.19	0.01
V	0.2	0.19	0.3	0.3	0.01

HMM Decoding: Viterbi Algorithm

- Question: What is the most likely state sequence given an output sequence

– Given fully specified HMM:

- $P(Y_1 = y_1)$,
- $P(Y_i = y_i \mid Y_{i-1} = y_{i-1})$,
- $P(X_i = x_i \mid Y_i = y_i)$

– Find $y^* = \operatorname{argmax}_{y \in \{y_1, \dots, y_l\}} P(x_1, \dots, x_l, y_1, \dots, y_l)$

$$= \operatorname{argmax}_{y \in \{y_1, \dots, y_l\}} \left\{ P(y_1)P(x_1|y_1) \prod_{i=2}^l P(x_i|y_i)P(y_i|y_{i-1}) \right\}$$

- “Viterbi” algorithm has runtime linear in length of sequence
- Example: find the most likely tag sequence for a given sequence of words

HMM's for POS Tagging

- Design HMM structure (vanilla)
 - States: one state per POS tag
 - Transitions: fully connected
 - Emissions: all words observed in training corpus
- Estimate probabilities
 - Use corpus, e.g. Treebank
 - Smoothing
 - Unseen words?
- Tagging new sentences
 - Use Viterbi to find most likely tag sequence

Experimental Results

Tagger	Accuracy	Training time	Prediction time
HMM	96.80%	20 sec	18.000 words/s
TBL Rules	96.47%	9 days	750 words/s

- Experiment setup
 - WSJ Corpus
 - Trigram HMM model
 - Lexicalized
 - from [Pla and Molina, 2001]

Discriminative vs. Generative

- Bayes Rule
$$h_{\text{bayes}}(x) = \operatorname{argmax}_{y \in Y} [P(Y = y|X = x)]$$
$$= \operatorname{argmax}_{y \in Y} [P(X = x|Y = y)P(Y = y)]$$
- Generative:
 - Make assumptions about $P(X = x|Y = y)$ and $P(Y = y)$
 - Estimate parameters of the two distributions
- Discriminative:
 - Define set of prediction rules (i.e. hypotheses) H
 - Find h in H that best approximates the classifications made by
$$h_{\text{bayes}}(x) = \operatorname{argmax}_{y \in Y} [P(Y = y|X = x)]$$
- Question: Can we train HMM's discriminately?
 - Later in semester: discriminative training of HMM and general structured prediction.