Stochastic Part-of-Speech Tagging

Brigitte Krenn
Österreichisches Forschungsinstitut für Artificial Intelligence
ÖFAI
Freyung 6, 1010 Wien
brigitte@ai.univie.ac.at

Stochastic Part-of-Speech Tagging: General View

Goal: find best tag sequence for a given word sequence

$$P(w_{1,n}|t_{1,n}) := \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Info to be read from an annotated corpus

(1) lexical probabilities

$$P(w_i|t_k) = \frac{f(w_i : t_k)}{f(t_k)}$$

with $f(w_i : t_k)$ the frequency of occurrence of word w_i with tag t_k.

(2) contextual probabilities

$$P(t_i|t_k) = \frac{f(t_i, t_k)}{f(t_k)}$$

with $f(t_i, t_k)$ the frequency of co-occurrence of tag t_i and tag t_k.

The relative frequencies f are called Maximum Likelihood Estimates MLEs.

Markov Models MMs

In general:

- MMs can be used for modelling the probability of a linear sequence of events.
- MMs correspond to nondeterministic finite state automata with probabilities attached to each arc.
- Outgoing acrs from a single state sum to 1.

Terminology: Markov Chain and Markov Property
A $Markov\ Chain$ is a special type of stochastic process which has the $Markov\ Property$, i.e., the probability of the next state depending on a sequence of previous states up to a current state is only dependent on the current state.

$$P(X_{t+1} = x_{i_{t+1}})|(X_1 = x_{i_1}, \cdots, X_t = x_{i_t}) = P(X_{t+1} = x_{i_{t+1}})|X_t = x_{i_t})$$

Thus the probability of a Markov Chain $X_1, X_2, \ldots$ can be calculated as

$$P(X_1 = x_{i_1}, \cdots X_t = x_{i_t}) = P(X_1 = x_{i_1}) * P(X_2 = x_{i_2}|X_1 = x_{i_1}) * \cdots * P(X_t = x_{i_t}|X_{t-1} = x_{i_{t-1}})$$

Terminology: Visible versus Hidden Markov Models

- Visible Markov Models are MMs the states of which can be observed, e.g. in training from an annotated corpus.
- In Hidden Markov Models HMMs, the states are unknown, as it is the case in tagging where only the sequence of words is observed.

Stochastic Part-of-Speech Tagging using Markov Models

Elements of a Markov Model

- a finite set of states (the PoS-tags) $\Omega = s_1, \cdots, s_n$
- a set containing the signal alphabet (the words) $\Sigma = \sigma_1, \cdots, \sigma_m$
- a vector representing the initial state probabilities $v = [v_1, \cdots, v_n]$ where $v_i = P(X_1 = s_i)$
- a $n \times n$ matrix containing the state transition probabilities $A = [a_{ij}]$ where $a_{ij} = P(X_{t+1} = s_j | X_t = s_i)$
- a $n \times m$ matrix containing the signal emission probabilities $B = [b_{ik}]$ where $b_{ik} = P(Y_t = \sigma_k | X_t = s_i)$

Stochastic Part-of-Speech Tagging using Markov Models

- Training: find for each state (tag) lexical and contextual probabilities, e.g. read probabilities from an annotated corpus, use an EM (expectation maximization) algorithm such as the Forward-Backward Algorithm;
- Tagging: assign a sequence of tags to a word sequence using a particular language model which has been learned in the training phase; typically the Viterbi Algorithm is employed for selecting the best state sequence for a given word sequence and a given language model;
- Initialization: determine the lexical and contextual probabilities to start computation from. In general:
  - use visible HMMs

    when a sufficiently large annotated training text is available, and training text and new text to be tagged are similar;
  - employ a few iterations of the Forward-Backward Algorithm

    when no annotated training text, but some lexical information is available;
  - train for a larger number of iterations

    when no lexical information is available. Caution: The trained model will be very poor!

Stochastic Part-of-Speech Tagging using Markov Models

Implementation Issues

- use addition of logarithms instead of multiplication of nonlogarithmic values

  $\rightarrow$ addition leads to a speed up compared to multiplication; this is important for Viterbi as it is a runtime algorithm;
  $\rightarrow$ use of logs helps to avoid underflow of the range of floating points; underflow results from multiplication of very small numbers.

Three Fundamental Problems for HMMs

- Problem 1: Given an observation sequence $O = (0_1, \cdots, 0_T)$ and a model $\mu = [A, B, \Pi]$ where
  $A = the\ matrix\ containing\ the\ state\ transition\ probabilities$
  $B = the\ matrix\ containing\ the\ observation\ symbol\ probabilities$
  $\Pi = the\ initial\ state\ distribution$,
  how do we efficiently compute the probability $P(O|\mu)$, i.e., the probability of the observation sequence O given the model μ?

- Problem 2: Given an observation sequence O and a model μ, how do we identify the most likely state sequence $(X_1, \cdots, X_T)$?

- Problem 3: Given an observation sequence O and a space of possible models, how do we best adjust the model parameters A ,B and Π to O? In other words, how do we find a model that best explains the observed data?

An Example

We have 2 urns containing three balls (red, blue, yellow) each.

Urns = States = N = 2
Symbols = red ball, yellow ball, blue ball = R, Y, B = M = 3

Probabilities for the balls in urn 1	Probabilities for the balls in urn 2
$b_1(R) = .8$	$b_2(R) = .3$
$b_1(B) = .1$	$b_2(B) = .4$
$b_1(Y) = .1$	$b_2(Y) = .3$

State Transistion Probabilities

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} .4 & .6 \\ .8 & .2 \end{pmatrix}$$

Observation Symbol Probabilities

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = \begin{pmatrix} .8 & .1 & .1 \\ .3 & .4 & .3 \end{pmatrix}$$

Initial State Distribution

$\Pi = (\pi_1, \pi_2) = (.5, .5) = 1$

Model Parameters of the HMM: $\mu = [A, B, \Pi]$

Variables

1) N $=$ 2 ... number of states in the model
$1 \leq i \leq N$

2a) M $=$ 3 ... number of observation symbols per state

2b) $V = v_1, \cdots, v_M = R, B, Y$ ... individual symbols

$1 \leq k \leq M$

3) $A = a_{ij}$ ... state transition symbols

$a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$

$1 \leq i \leq N$, $1 \leq j \leq N$

4) $B = b_{ik}$ ... observation symbol probability distribution

$b_{ik} = P(q_t = \sigma_k | q_t = s_i)$

$1 \leq i \leq N$ ... states
$1 \leq k \leq M$ ... symbols

5) $\Pi = \pi_i$ ... initial state distribution

$\Pi_i = P(q_1 = s_i) \quad 1 \leq i \leq N$ ... states

A Solution to Problem 1

Given: observation sequence $O = o_1, \cdots, o_T$ and model $\mu = [A, B, \Pi]$

Wanted: probability of the observation sequence $O = (R, B, Y)$ given the model $P(O|\mu)$

Procedure applied: the Forward Variable of the Forward-Backward Algorithm is used

$$P(O|\mu) = \sum_{i=1}^{N} \alpha_T(i)$$

The Forward Variable $\alpha_t(i)$:

$$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, q_t = s_i|\mu)$$

A Solution to Problem 1

Initialization:

$$\alpha_1(j) = \pi_j b_j(o_1)$$

$1 \leq j \leq N$

thus we have

$$\alpha_1(1) = \pi_1 * b_1(R)$$

$$\alpha_1(2) = \pi_2 * b_2(R)$$

A Solution to Problem 1

Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) * a_{ij}] * b_j(o_{t+1})$$

$1 \leq t \leq T-1$
$1 \leq j \leq N$

thus we have

$$\alpha_2(1) = [\alpha_1(1) * a_{11} + \alpha_1(2) * a_{21}] * b_1(B)$$

$$\alpha_2(2) = [\alpha_1(1) * a_{12} + \alpha_1(2) * a_{22}] * b_2(B)$$

$$\alpha_3(1) = [\alpha_2(1) * a_{11} + \alpha_2(2) * a_{21}] * b_1(Y)$$

$$\alpha_3(2) = [\alpha_2(1) * a_{12} + \alpha_2(2) * a_{22}] * b_2(Y)$$

A Solution to Problem 1

Termination:

$$P(O|\mu) = \sum_{j=1}^{N} \alpha_T(j)$$

thus we have

$$P(O|\mu) = \alpha_3(1) + \alpha_3(2)$$

A Solution to Problem 2

Given: an observation sequence $O = o_1, \cdots, o_T$ and model $\mu = [A, B, \Pi]$

Wanted: the most likely state sequence given the observation sequence and the model

Procedure applied: the Viterbi Algorithm is used to find the single best state sequence.

We need to define the quantity $\delta_t(i)$ which is the highest probability (best score) at time t along a single path.

$$\delta_t(i) = max_{q_1,q_2,\cdots,q_{t-1}}\ P(q_1 q_2 \cdots q_t = i, o_1 o_2 \cdots o_t | \mu)$$

We do this again by induction

$$\delta_{t+1}(j) = [max_i\ \delta_t(i) * a_{ij}] * b_j(o_{t+1})$$

To keep track of the argument which maximized the above formular for each time t and state j we introduce an array $\psi_t(j)$.

A Solution to Problem 2

Initialization:

$$\delta_1(i) = \pi_i b_i(o_1)$$

$$\psi_1(i) = 0$$

$1 \leq i \leq N$

$$\delta_1(1) = \pi_1 * b_1(R)$$

$$\delta_1(2) = \pi_2 * b_2(R)$$

A Solution to Problem 2

Recursion:

$$\delta_t(j) = max_{1 \leq i \leq N} \; [\delta_{t-1}(i) * a_{ij}] * b_j(o_t)$$

$$\psi_t(j) = argmax_{1 \leq i \leq N} \; [\delta_{t-1}(i) * a_{ij}]$$

$1 \leq i \leq N$, $2 \leq t \leq T$

$$\delta_2(1) = max[\delta_1(1) * a_{11}, \delta_1(2) * a_{21}] * b_1(B)$$

$$\psi_2(1) = 1$$

$$\delta_2(2) = max[\delta_1(1) * a_{12}, \delta_1(2) * a_{22}] * b_2(B)$$

$$\psi_2(2) = 1$$

$$\delta_3(1) = max[\delta_2(1) * a_{11}, \delta_2(2) * a_{21}] * b_1(Y)$$

$$\psi_3(1) = 2$$

$$\delta_3(2) = max[\delta_2(1) * a_{12}, \delta_2(2) * a_{22}] * b_2(Y)$$

$$\psi_3(2) = 2$$

A Solution to Problem 2

Termination:

$$P^* = max_{1 \leq i \leq N} \; [\delta_T(i)]$$

$$q_T^* = argmax_{1 \leq i \leq N} \; [\delta_T(i)]$$

$$P^* = max\,[\delta_3(1), \delta_3(2)]$$

$$q_T^* = q_3^* = argmax\,[\delta_3(1), \delta_3(2)] = 1$$

Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

$t = T-1, T-2, \cdots, 1$

$$q_2^* = \psi_3(q_3^*) = \psi_3(1) = 2$$

$$q_1^* = \psi_2(q_2^*) = \psi_2(2) = 1$$

$Q = s_1, s_2, s_1$