Stochastic models for learning language models (Part 2)

R. Basili

Web Mining e *Retrieval*
a.a. 2022-23

April 16, 2023

Outline

Outline

A survey of the Baum-Welch method

The learning Problem

Given a HMM $\lambda = (E, T, \pi)$ and an observation history $Z = (z_1, z_2, ..., z_t)$, and a new HMM $\lambda' = (E', T', \pi')$ that explains the observations at least as well, or possibly better, i.e., such that $Pr[Z|\lambda'] \geq Pr[Z|\lambda]$.

- Ideally, we would like to find the model that **maximizes** $Pr[Z|\lambda]$; however, this is in general an intractable problem.
- We will be satisfied with an algorithm that converges to local maxima of such probability.
- Notice that in order for learning to be effective, we need **lots of data**, i.e., many, long observation histories!

The Baum-Welch estimation as a EM process

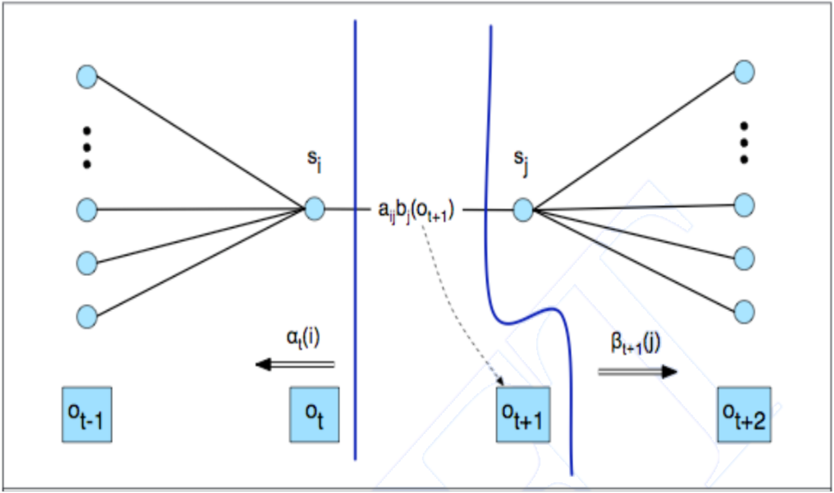
Baum-Welch re-estimation: the idea

Baum-Welch reestimatio is also called the **Forward-Backward** algorithm It is special case of the **Expectation Maximization** (**EM**) algorithm

1. Start with initial probability estimates
2. Compute expectations of how often each transition/emission is used
3. Re-estimate the probabilities based on those expectations

...and repeat until convergence

The forward backward probabilities



Baum-Welch: Forward and Backward probabilities

- Forward probabilities (DEF):

$$\alpha_k(s) = Pr[o_1, ..., o_k, x_k = s|\lambda]$$

Recursively

$$\alpha_{k+1}(q) = \sum_{s \in S} \alpha_k(s) a_{sq} b_q(o_{k+1}) \quad (\text{with } \alpha_1(q)) = \pi_q)$$

Baum-Welch: Forward and Backward probabilities

- Forward probabilities (DEF):

$$\alpha_k(s) = Pr[o_1, ..., o_k, x_k = s|\lambda]$$

Recursively

$$\alpha_{k+1}(q) = \sum_{s \in S} \alpha_k(s) a_{sq} b_q(o_{k+1}) \quad (\text{with } \alpha_1(q)) = \pi_q)$$

- Backward probabilities (DEF):

$$\beta_k(s) = Pr[o_k, ..., o_t|x_k = s, \lambda]$$

Recursively:

$$\beta_k(s) = \sum_{q \in S} a_{sq} b_q(o_{k+1}) \beta_{k+1}(q)$$

Baum-Welch: Expectation of (state) counts

- Let us define: $\gamma_k(s) = Pr[X_k = s | Z, \lambda]$
- We already know how to compute this, e.g., using smoothing:

Baum-Welch: Expectation of (state) counts

- Let us define: $\gamma_k(s) = Pr[X_k = s|Z,\lambda]$
- We already know how to compute this, e.g., using smoothing:

$$\gamma_k(s) = \frac{\alpha_k(s)\beta_k(s)}{Pr[X_k|Z,\lambda]} = \frac{\alpha_k(s)\beta_k(s)}{\sum_{q\in S}\alpha_k(q)}$$

Baum-Welch: Expectation of (state) counts

- Let us define: $\gamma_k(s) = Pr[X_k = s|Z,\lambda]$
- We already know how to compute this, e.g., using smoothing:

$$\gamma_k(s) = \frac{\alpha_k(s)\beta_k(s)}{Pr[X_k|Z,\lambda]} = \frac{\alpha_k(s)\beta_k(s)}{\sum_{q\in S}\alpha_k(q)}$$

- **New concept**: how many times is the state trajectory expected to transition from state s?
  $E[\#\text{ of transitions from } s] = \sum_{k=1}^{t-1}\gamma_k(s)$

Baum-Welch: Expectation of (transitions) counts

- In much the same vein, let us define $\xi_k(q,s) = Pr[X_k = q, X_{k+1} = s|Z,\lambda]$ (i.e., $\xi_k(q,s)$ is the probability of being at state q at time k, and at state s at time $k+1$, given the observations and the current HMM model)

Baum-Welch: Expectation of (transitions) counts

- In much the same vein, let us define $\xi_k(q,s) = Pr[X_k = q, X_{k+1} = s | Z, \lambda]$ (i.e., $\xi_k(q,s)$ is the probability of being at state q at time k, and at state s at time $k+1$, given the observations and the current HMM model)
- We have that $\xi_k(q,s) = \eta_k \alpha_k(q) T_{q,s} E_{s,o_{k+1}} \beta_{k+1}(s)$ where η_k is a normalization factor, such that $\sum_{q,s} \xi_k(q,s) = 1$.

Baum-Welch: Expectation of (transitions) counts

- In much the same vein, let us define $\xi_k(q,s) = Pr[X_k = q, X_{k+1} = s | Z, \lambda]$ (i.e., $\xi_k(q,s)$ is the probability of being at state q at time k, and at state s at time $k+1$, given the observations and the current HMM model)
- We have that $\xi_k(q,s) = \eta_k \alpha_k(q) T_{q,s} E_{s,o_{k+1}} \beta_{k+1}(s)$ where η_k is a normalization factor, such that $\sum_{q,s} \xi_k(q,s) = 1$.
- **New concept**: how many times is the state trajectory expected to transition *from* state q to state s?
  $E[\text{\# of transitions from } q \text{ to } s] = \sum_{k=1}^{t-1} \xi_k(q,s)$

Baum-Welch algorithm

- Based on the probability estimates and expectations computed so far, using the original HMM model $\lambda = (E, T, \pi)$, we can construct a new model $\hat{\lambda} = (\hat{E}, \hat{T}, \hat{\pi})$ (notice that the two models share the states and observations):
- The new initial condition distribution is the one obtained by smoothing: $\hat{\pi}_s = \gamma_1(s)$
- The entries of the new transition matrix can be obtained as follows:
$$\hat{T}_{q,s} = \frac{E[\#\text{ of transitions from } q \text{ to } s]}{E[\#\text{ of transitions from } q]} = \frac{\sum_{k=1}^{t-1} \xi_k(q,s)}{\sum_{k=1}^{t-1} \gamma_k(q)} = \hat{P}(q \to s|q)$$

Baum-Welch algorithm

- The entries of the new emission matrix can be obtained as follows:
  $\hat{E}_{s,o}(=\hat{b}_s(o)) = \frac{E[\text{\# of times in state } s\text{, when the observation was } o]}{E[\text{\# of times in state } s]} =$
  $= \frac{\sum_{k=1}^{t} \gamma_k(s)\mathbf{1}(z_k{=}o)}{\sum_{k=1}^{t} \gamma_k(s)} = \hat{P}(o|s)$
- In this way, new estimated version for $\hat{E}, \hat{T}$ and $\hat{\pi}$ are available:
  They correspond to a new model $\hat{\lambda} = (\hat{E}, \hat{T}, \hat{\pi})$

Baum-Welch as an EM iterative model refinement

E-step (expectaton)

$\sum_{k=1}^{t} \gamma_k(i)$ = expected number of transitions involving q_i
$\sum_{k=1}^{t-1} \xi_k(i,j)$ = expected number of transitions from q_i to q_j

M-step (Likelyhood Maximimization)

We can re-estimate parameters by ratio of expected counts

$$\hat{a}_{i,j} = \frac{\sum_{k=1}^{t-1} \xi_k(i,j)}{\sum_{k=1}^{t-1} \gamma_k(j)}$$

$$\hat{b}_i(o) = \frac{\sum_{k=1}^{t-1} \gamma_k(i) \cdot \mathbf{1}(z_k = o)}{\sum_{k=1}^{t-1} \gamma_k(i)}$$

Baum-Welch: an example on the soft drink machine

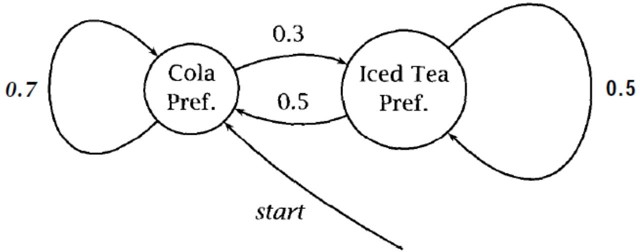


Figure 9.2 The crazy soft drink machine, showing the states of the machine and the state transition probabilities.

Output probability given From state

	cola	iced tea (ice_t)	lemonade (lem)
CP	0.6	0.1	0.3
IP	0.1	0.7	0.2

Baum-Welch re-estimation on the soft drink machine

training on the observation sequence (lem, ice_t, cola)
values for $p_t(i, j)$:

Time (and j)

	1 CP	1 IP	γ_1	2 CP	2 IP	γ_2	3 CP	3 IP	γ_3
i CP	0.3	0.7	1.0	0.28	0.02	0.3	0.616	0.264	0.88
IP	0.0	0.0	0.0	0.6	0.1	0.7	0.06	0.06	0.12

and so the parameters will be reestimated as follows:

		Original			Reestimated		
Π	CP	1.0			1.0		
	IP	0.0			0.0		
		CP	IP		CP	IP	
A	CP	0.7	0.3		0.5486	0.4514	
	IP	0.5	0.5		0.8049	0.1951	
		cola	ice_t	lem	cola	ice_t	lem
B	CP	0.6	0.1	0.3	0.4037	0.1376	0.4587
	IP	0.1	0.7	0.2	0.1363	0.8537	0.0

Baum-Welch algorithm: convergence

- It can be shown [Baum et al., 1970] that the new model $\hat{\lambda}$ is such that
  - $Pr[Z|\hat{\lambda}] \geq Pr[Z|\lambda]$, as desired.
  - $Pr[Z|\hat{\lambda}] = Pr[Z|\lambda]$ only if λ is a critical point of the likelihood function
$$f(\lambda) = Pr[Z|\lambda]$$

Other Approaches to POS tagging

- Church (1988):
  $\prod_{i=n}^{3} P(w_i|t_i)P(t_{i-2}|t_{i-1},t_i)$ (backward)
  Estimation from tagged corpus (Brown)
  No HMM training
  Performances: $> 95\%$
- De Rose (1988):
  $\prod_{i=1}^{n} P(w_i|t_i)P(t_{i-1}|t_i)$ (forward)
  Estimation from tagged corpus (Brown)
  No HMM training Performance: 95%
- Merialdo et al.,(1992), ML estimation vs. Viterbi training
  Propose an incremental approach: small tagging and then Viterbi training
- $\prod_{i=1}^{n} P(w_i|t_i)P(t_{i+1}|t_i,w_i)$???

HMM decoding vs. more complex sequence labeling tasks

- *w1, w2, … wn*
- `p1, p2, …, pn` **POS TAGGING**: pi ∈ {NN, JJ, VB, …}
- `p1, p2, …, pn` **KEYWORD SPOTTING**: pi ∈ {0, 1}
- `p1, p2, …, pn` **BRACKETING**: pi ∈ {O(UT), I(NNER), B(EGIN)}
- Applications of bracketing: **Named Entity Recognition**
- *Il, presidente, della, Repubblica, vaggiò, verso Milano*
- `B, I, I, I, O, O, B`
- *(Il, presidente, della, Repubblica), vaggiò, verso (Milano)*
- **…. and Classification**
- *Il, presidente, della, Repubblica, vaggiò, verso, Milano*
- `B-HUM, I, I, I, O, O, B-LOC`
- *(Il, presidente, della, Repubblica)$_{HUM}$, vaggiò, verso (Milano)$_{LOC}$*

HMM decoding vs. more complex sequence labeling tasks (2)

Multiword Expressions

he was willing to budge a little on

O O O O B b i I

the price which means a lot to me .

O O O B I I I I O

a little; *means a lot to me*; *budge . . . on*

See: “Discriminative lexical semantic segmentation with gaps: running the MWE gamut,” Schneider et al. (2014).

HMM decoding vs. more complex sequence labeling tasks (3)

Named Entity Recognition

With	Commander	Chris	Ferguson	at	the	helm	,
	person						
O	B	I	I	O	O	O	O

Atlantis	touched	down	at	Kennedy	Space	Center	.
spacecraft				location			
B	O	O	O	B	I	I	O

HMM decoding vs. more complex sequence labeling tasks (4)

Supersense Tagging

ikr	smh	he	asked	fir	yo	last	name
–	–	–	communication	–	–	–	cognition

so	he	can	add	u	on	fb	lololol
–	–	–	stative	–	–	group	–

See: "Coarse lexical semantic annotation with supersenses: an Arabic case study," Schneider et al. (2012).

HMM Decoding for Natural Language Processing

HMM Decoding is largely applicable method for many structured prediction tasks in NLP.

Key elements

- Map the target NLP task into a *sequence of classification* problem
- Design a representation (e.g. features and metrics), ...
- ... a prediction function f and ...
- ... a learning or estimation algorithm to approximate with the hypothesis h the function f

POS tagging: References

- F. Jelinek, Statistical methods for speech recognition, Cambridge, Mass.: MIT Press, 1997.
- Manning & Schutze, Foundations of Statistical Natural Language Processing, MIT Press, Chapter 6.
- Jurafsky& Martin, Speech and Language Processing, Chapt. 8. URL: `https://web.stanford.edu/~jurafsky/slp3/`
- Church (1988), A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, http://acl.ldc.upenn.edu/A/A88/A88-1019.pdf
- Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286.
- Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, IT-13(2), 260-269.
- Parameter Estimation (slides): `http://jan.stanford.edu/fsnlp/statest/henke-ch6.ppt`

Other References

- "*Introduction to Information Retrieval*", Christopher D. Manning, Prabhakar Raghavan and Hinrich Schųtze, Cambridge University Press. 2008. Chapter 12. `http://www-csli.stanford.edu/ hinrich/information-retrieval-book.`
- Rabiner, Lawrence. "First Hand: The Hidden Markov Model". IEEE Global History Network. Retrieved 2 October 2013. at `http://www.ieeeghn.org/wiki/index.php/First-Hand:The_Hidden_Markov_Model`
- Applet at: `http://www.cs.umb.edu/ srevilak/viterbi/`