Unsupervised meeting segmentation using mixture of Markov models based on speaker turn dynamics

Naveen Kumar

Project Report
Probabilistic Reasoning

Abstract

An initial segmentation is often required for automatic analysis of events in multiparty meetings. In this work, I investigate an alternative simpler approach instead of the typical segmentation approaches based on multimodal features. Speaker turn taking dynamics in a meeting is modeled as a mixture of Markov processes. Likelihood based results on a held out set suggest its effectiveness as a generative model.

1 Introduction

Automatic understanding of events from interaction of multiple persons is essential for extracting any information from them. However, most of these meetings are unstructured where the course of discussion is not decided apriori. This often makes it hard to segment the discussion into meaningful segments even by manual annotations [1]. A variety of segmentation approaches have been tried for this purpose, ranging from topic based models [2] to multimodal feature streams [3, 4]. While discourse segmentation approaches mainly tend to focus on the topic of the discussion and use summarization or topic modeling ideas for the purpose of segmentation, multimodal approaches tend to use visual or audio cues instead. Although these indirect features might be be helpful given previous knowledge of the nature of these meetings segments, this is often a luxury that we do not have for unstructured meeting sessions.

In this work, I propose to directly segment a meeting based on the local meeting dynamics. The dynamics in a meeting are approximated as a first order Markov process over speaker turns. I hypothesize that speaker turns in any meeting session can be thought to have been generated by a mixture of such Markov processes, each corresponding to a particular meeting segment. This is motivated by the observation that in a real meeting setting, the dynamics of interaction is hardly stationary and varies as the meeting evolves. By fitting locally stationary Markov models to this data, would then help us find boundaries for each segment.

Mixture of Markov chains (MMM) is in fact popular in modeling non stationary time series where in economics [5]. [6] uses it for monitoring SQL injection attacks from web traffic data, as a higher order model over n-grams. In my current work, MMM is presented as an unsupervised technique, for learning latent meeting interaction styles into which the meeting is assumed to be divided.

2 Mixture of Markov Models (MMM)

In the Markov Mixture Model, each speaker transition is modeled as a mixture of finite number of Markov processes. Hence each speaker turn is assumed to have been generated from a mixture of transition matrices, when conditioned on the previous speaker turn. The states in the Markov chain correspond to speaker identities. For K mixture components, the complete model is parametrized by transition matrices A^k and their corresponding mixing weights ω_k. The probability of transition from speaker p to q, from time n to $n+1$ is given as

$$\begin{aligned} P(X_{n+1} = q | X_n = p) &= \sum_{k=1}^{K} \omega_k P(X_{n+1} = q | X_n = p; A^k) \\ &= \sum_{k=1}^{K} \omega_k a_{pq|k} \end{aligned} \tag{1}$$

The likelihood for the entire time series can then be computed as

$$P(X_1, X_2, X_3, \cdots, X_N) = \eta(X_1) \prod_{n=1}^{N-1} \sum_{k=1}^{K} \omega_k P(X_{n+1}|X_n; A^k) \tag{2}$$

$$= \eta(X_1) \prod_{n=1}^{N-1} \sum_{k=1}^{K} \omega_k \prod_{p,q \in \{1 \cdots S\}} {a_{pq|k}}^{I(X_n=p, X_{n+1}=q)} \tag{3}$$

where $\eta(X_1)$ is the initial probability common to all mixture Markov components, $I(.)$ is the delta function, and S is the number of speakers. Also, equation 3 is a rigorous way to define the probability for a Markov process. This will come in handy when estimating the parameters by Maximum Likelihood criterion. Note that each pair of adjacent speaker transition is thus being drawn independently from the mixture distribution. Also for simplicity it is assumed that $\eta(X_1)$ is drawn from a uniform distribution. It can also be trivially estimated given multiple meeting sessions.

3 Unsupervised Learning

Since the parameters ω_k and A^k are coupled in the likelihood equation they cannot be estimated at once. As with any typical mixture distribution, the parameter learning problem can be then framed as a hidden data problem. The hidden data in this case corresponds to the mixture identity of each speaker transition. We rewrite the data likelihood assuming that we know which mixture generated it. This is encoded using an indicator vector z_n for the transition $X_n \rightarrow X_{n+1}$ consisting of K values, one for each mixture components. $z_{nk} = 1$ denotes the the n^{th} transition came from the k^{th} mixture component. The total log likelihood over the data and the hidden indicator vectors Z in this case can be written as in Equation 5. This can be further simplified as in Equation 3.

$$P(X_1, X_2, X_3, \cdots, X_N, Z) = \eta(X_1) \prod_{n=1}^{N-1} \prod_{k=1}^{K} \omega_k^{z_{nk}} P(X_{n+1}|X_n; A^k)^{z_{nk}} \tag{4}$$

$$\Rightarrow log(P(X, Z)) = C + \sum_{n=1}^{N-1} \sum_{k=1}^{K} z_{nk}[log(\omega_k) + log(P(X_{n+1}|X_n; A^k))] \tag{5}$$

Since the values for Z are not known, we estimate them in an iterative fashion using the Expectation-Maximization Algorithm. At each step we estimate the posterior distribution over the hidden values given the parameters and then optimize the expected likelihood over this distribution.

3.1 E-Step

The posterior distribution of each indicator variable μ_{nk}, typically referred to as membership probabilities serves as soft segmentation labels. Since z_{nk}s are binary indicator variables, μ_{nk} can be thought of to be its expected value. This is used to write the expected log-likelihood also otherwise known as the auxiliary function in the M-step.

$$\mu_{nk} = \mathbb{E}(z_{nk}|X; A^k, \omega_k) = P(z_{nk} = 1|X; A^k, \omega_k) \tag{6}$$

$$= \frac{\omega_k P(X_{n+1}|X_n; A^k)}{\sum_{j=1}^{K} \omega_j P(X_{n+1}|X_n; A^j)} \tag{7}$$

Equation 7 serves as the update equation for the hidden parameters. After EM has converged these values can be thresholded to obtain segmentation results.

3.2 M-step

EM indirectly optimizes the data likelihood by optimizing the expected likelihood below

$$\mathcal{L} = \mathbb{E}_{P(Z=1|X;\vec{A},\vec{\omega})}(log(P(X, Z))$$

$$= C + \sum_{n=1}^{N-1} \sum_{k=1}^{K} \mu_{nk}[log(\omega_k) + log(P(X_{n+1}|X_n; A^k))] \tag{8}$$

A maximum likelihood estimation for the parameters yields very simple update equations which can be readily interpreted

$$\omega_k = \frac{\sum_{n=1}^{N-1} \mu_{nk}}{\sum_{j=1}^{K} \sum_{n=1}^{N-1} \mu_{nj}} \tag{9}$$

$$a_{pq|k} = \frac{\sum_{n=0}^{N-1} \mu_{nk} I(x_n = p, x_{n+1} = q)}{\sum_{n=0}^{N-1} \mu_{nk} I(x_n = p)} \tag{10}$$

where $a_{pq|k}$ denotes the transition probability from speaker p to q according to the k^{th} mixture component. Note that the expressions are quite intuitive. While ω_k are like averaged responsibility functions, $a_{pq|k}$ is estimated as the normalized expected count for all transitions $p \rightarrow q$.

3.3 Choosing K

Since the number of latent meeting types is not known as well, it is chosen based on the Bayesian information criteria (BIC). Typically the data likelihood always increases with the increase in number of mixture, which makes it hard to use likelihood directly as a criterion. However BIC takes into account a penalty term for number of parameters. BIC in each case was estimated using the formula

$$BIC(K) = -2\mathcal{L} + K(1 + S^2)log(N)$$

3.4 Segmentation and smoothing

The final segmentation labels T_n are obtained as $T_n = \arg\max_k \mu_{nk}$. Note that the mixture model stated till now allows each transition to choose its own mixture. There is no constraint whatsoever on the temporal structure of T_n except that through the first order Markov process. As we see later from the results, it might indeed be desirable to have a longer context over which the mixture model is estimated. This prevents over segmentations and allows a smooth transitions over the assigned mixture responsibility. To achieve this the mixture distribution is redefined instead over a window of speaker turns of length w as follows

$$\begin{aligned} P(X_n, X_{n+1}, \cdots, X_{n+w}) &= \sum_{k=1}^{K} \omega_k P(X_n, X_{n+1}, \cdots, X_{n+w}; A^k) \\ &= \sum_{k=1}^{K} \omega_k \prod_{i=n}^{n+w} P(X_{i+1}|X_i; A^k) \end{aligned} \tag{11}$$

While this provides us a scale hyper parameter w which can be conveniently tuned to choose the level of detail, it also makes it harder to define the generative process. This is because the probabilities must now be defined over non-overlapping windows, which makes the choice of window boundaries critical. [6] attempts to solve this problem by assuming that the n-gram probabilities can be factorized in terms of bigrams. In this case, I adopt the trick of estimating the parameters over non overlapping windows, but then using them to estimate the μ_{nk} for overlapping windows as well. Another way to see this is that the M-step is skipped for the overlapping windows, while the E-step is still performed for them. The μ_{nk} assigned to one window is then assigned to the transition at its center.

4 Results

The above system was tested for segmentation of meeting transcripts from different corpora. The behavior of change in data likelihood with increase in K was initially found to be counter intuitive. The reason was that since the mixture models had been defined independently over individual transitions, EM was easily overfitting over them yielding sparse transition matrices. Thus EM quickly converged to a local maxima irrespective of the choice of K. This led to segmentations that didn't have smooth boundaries. Defining the generative model over windows helped in this case to get a smooth transition of the output labels.

Evaluations are further made hard by the fact that none of the corpora, on which the experiments were conducted had any publicly available annotations for segmentations. Also the speaker set often varies from session to session making cross validation experiments hard. Speakers are thus first classed using exchange algorithms used for word classing in language models [7]. This provides a common speaker set across which the results can be compared. Specifically in this study, I use data likelihood on a held out set as a metric to prove the effectiveness of MMM as a generative model (Figure 1). Additionally some example segmentations have been provided in the Appendix.

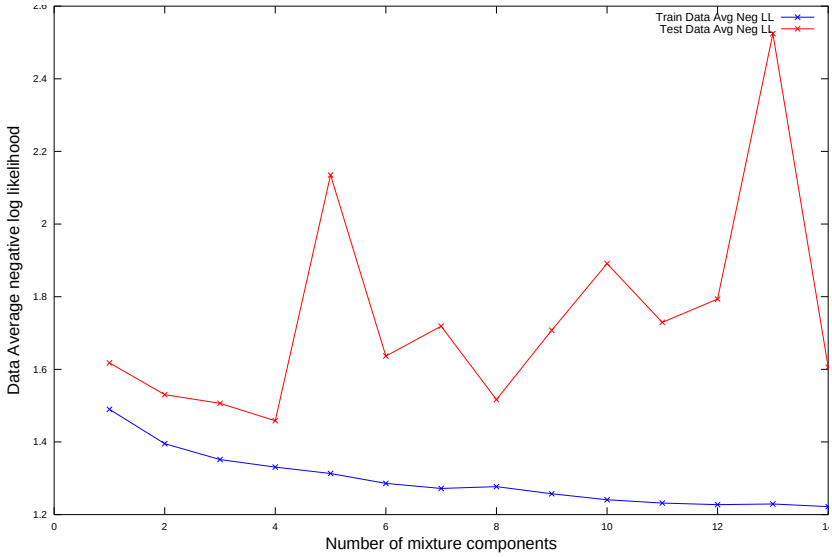

Figure 1: Variation of average negative likelihood for the model on train and test sets

5 Conclusion

In this work an unsupervised method is proposed for automatic segmentation of meeting sessions purely based on speaker turn taking dynamics. It is observed that the MMM model proposed for this purpose easily overfits when defined independently over each transition, which leads to oversegmented boundaries. Smoother decision boundaries can be obtained by defining the generative story over non overlapping windows. Also evaluation is tricky because of the inherent latent nature of the segment classes.

These kinds of evaluation woes are in fact common in direct evaluation of latent variable models e.g. Latent Dirichlet Allocation. Since the assigned topics or segments in this case are essentially unsupervised clusters, it is hard to obtain annotations to evaluate them. Instead these latent decisions are often used as an intermediate step for another task which can potentially benefit from the clustering. In the future, it might be a good idea to evaluate the model indirectly through another task that can make use of these segmentations. Alternatively, it might also be possible to bias the algorithm to segment patterns of interest. This can be done via suitable initialization since EM often tends to find maximas that are similar to the initialization point.

As far as the ad-hoc smoothing is concerned, a more sophisticated approach could be adopted. Since EM only approaches to maximize the data likelihood, one idea might be to add a constraint that also enforces diversity of the mixture component distributions. However this might not be trivial since it make the optimization problem in the M-step non-convex. Alternatively, we can put a Markov assumption on the transitions $T_n \rightarrow T_{n+1}$. This leads to a doubly Markov structure similar to auto regressive HMMs.

References

[1] R.J. Passonneau and D.J. Litman, "Discourse segmentation by human and automated means," *Computational Linguistics*, vol. 23, no. 1, pp. 103–139, 1997.

[2] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 562–569.

[3] A. Dielmann and S. Renals, "Automatic meeting segmentation using dynamic bayesian networks," *Multimedia, IEEE Transactions on*, vol. 9, no. 1, pp. 25–36, 2007.

[4] G. Lathoud, I.A. McCowan, and J.M. Odobez, "Unsupervised location-based segmentation of multi-party speech," in *Proc. NIST Meeting Recognition Workshop*, 2004.

[5] H. Frydman and T. Schuermann, "Credit rating dynamics and markov mixture models," *Journal of Banking & Finance*, vol. 32, no. 6, pp. 1062–1075, 2008.

[6] Y. Song, A.D. Keromytis, and S.J. Stolfo, "Spectrogram: A mixture-of-markov-chains model for anomaly detection in web traffic," in *Proc of the 16th Annual Network and Distributed System Security Symposium (NDSS)*, 2009.

[7] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

Appendix

T_n	Spkr ID	Sentence
1	COU	that is true .
1	AAH	you don't stop a war by delivering weapons .
1	QXE	they should have tried a [B] diplomatic solution .
0	COU	[Laugh] good call .
0	MTY	did anybody notice the guy's name was Jock ?
0	GAH	[Laugh]
0	YNH	yes . [Laugh]
0	QXE	which guy ?
0	ZMW	[hm]
0	MTY	J O C K , Jock .
0	YNH	and the guy who was being attacked by the elephant and chasing it -/n=/- out of his yard .
0	COU	Jock .
0	MTY	yes . what a bad-ass . if he st= [Laugh] [*T]t
0	GAH	[Laugh]
0	YNH	[Laugh]
0	ZMW	[Laugh] [P] this was a cool [Laugh] guy . [Laugh]
2	COU	he was showing off . [P] it worked , though .
2	AAH	[Laugh] [B] pretty impressive .
2	ZMW	pretty impressive . yeah% . [B] and , it is so easy . [Laugh] it was really easy . [B] so , [B] I agree with point seven what the hell becau
2	MTY	[hm] yes .
2	AAH	hm .
4	me018	That's what I was thinking. Yeah.
4	me013	Yeah.
4	mn017	for one gender. Yeah.
4	me013	Yeah. So, I mean, it's a bit of a push, but it seems like,
4	me013	O_K, we've got some models, we've got some training data, we have software that works, he's got
4	me013	a method that helps with, you know, other ta- another task.
1	me013	Um -
1	me013	It, you know,
1	me018	S-
1	mn017	Mm-hmm.
1	me013	Um -
1	me018	So - We- Yo- So, one thing I was wondering is,
1	me018	did you already do that middle one or should
1	me018	we re-do that one, too?
6	Subj_050	Traveling is what you don't like, but –
6	Subj_040	Well y- y- you don't end up being housed as comfortably and you only have what you've taken with you, for the most part an-
6	Subj_022	The crowd.
6	Subj_051	Yeah.
6	Subj_050	Hm.
6	Subj_022	I just don't like the crowd. m-
6	Subj_040	I get drug around on shopping expeditions.
6	Subj_050	And that's what's (( ))
6	Subj_022	breath
2	Subj_050	yeah.
2	Subj_022	You know why – recently, we could end up to do gift certificates, you know. It's like it come to that point because like it's so muc
2	Subj_040	Yeah.
2	Subj_050	Shopping online's not an option?
2	Subj_022	Not really because like you don't see the thing. You have to go to the store to look to see if it's okay, then you can go order onlin
2	Subj_050	Right.

Table 1: Sample experimental results from three of the meeting corpora used for the experiments viz. ISL, ICSI and NIST. Observe the change in discourse around the segment boundaries. Also note that the sentence transcriptions were not used by the algorithm and are only being provided for purposes of demonstration.