

State Aggregation in Higher Order Markov Chains for Finding Online Communities

Xin Wang and Ata Kabán

School of Computer Science, The University of Birmingham,
Birmingham, B15 2TT, UK
{X.C.Wang,A.Kaban}@cs.bham.ac.uk

Abstract. We develop and investigate probabilistic approaches of state clustering in higher-order Markov chains. A direct extension of the Aggregate Markov model to higher orders turns out to be problematic due to the large number of parameters required. However, in many cases, the events in the finite memory are not equally salient in terms of their predictive value. We exploit this to reduce the number of parameters. We use a hidden variable to infer which of the past events is the most predictive and develop two different mixed-order approximations of the higher-order aggregate Markov model. We apply these models to the problem of community identification from event sequences produced through online computer-mediated interactions. Our approach bypasses the limitations of static approaches and offers a flexible modelling tool, able to reveal novel and insightful structural aspects of online interaction dynamics.

1 Introduction

With the growing spread of web-based online communication applications, there is a growing demand for developing tools that allow us to learn from the wealth of data being generated. Community identification [8] is one of the most important learning tasks, because discovering communities and their evolution may be useful in bringing individuals with common interests together, tracking trends and facilitating the transmission of information and targeted marketing.

In social sciences, relationships are typically represented by edge-weighted, directed graphs. Communities are then identified by finding densely connected subgraphs [3]. This is an NP-complete problem. Approximate polynomial-time algorithms include the maximum flow algorithms [3] and spectral-methods [4] (based on eigen-computations), such as the Hypertext Induced Topic Search (HITS) [7] and PageRank [1] algorithms, which identify authoritative or influential web pages from the graph formed by the interconnected pages. Probabilistic counterparts of some of these ideas with a desirable clear generative semantics have also been devised and shown to have certain advantages. These include the Aggregate Markov (AM) model [12], developed in language modelling, which introduces a hidden 'bottleneck' variable to infer the state groupings. Essentially the same model has later been employed for bibliometric analysis, for finding

related publications [2] and it can be seen as a probabilistic version of the HITS model, and therefore has also been termed as the probabilistic HITS (PHITS).

However the AM model makes the first-order assumption of Markovianity. In this paper we extend it to higher-order Markov chains in various ways.

2 Model formulation

Let $X = \{x_1, x_2, \dots, x_N\}$ denote a discrete state sequence with each symbol $x_n \in \{1, 2, \dots, S\}$ coming from a S -symbol state space.

2.1 A higher-order Aggregate Markov Model (HAM)

Retaining the idea of a 'bottleneck' latent variable, we may directly attempt to extend the Aggregate Markov (AM) model to higher orders. The resulting generative model is then the following.

- Conditional on the finite memory of past events, generate a class $k \sim$ Multinomial $P(k|x_{n-1}, \dots, x_{n-L})$.
- Generate the next symbol $x_n \sim$ Multinomial $P(x_n|k)$, conditional on class k .

Thus, the probability of observing state x_n under the above generative process is the following.

$$P(x_n|x_{n-1}, \dots, x_{n-L}) = \sum_{k=1}^K P(x_n|k)P(k|x_{n-1}, \dots, x_{n-L}) \quad (1)$$

Although conceptually very simple, there is an obvious problem with this approach in that the number of parameters in the term $P(k|x_{n-1}, \dots, x_{n-L})$ grows exponentially with L . This makes the approach impractical and it is most probably the reason why it was never pursued in the literature. We need to make further assumptions in order to make progress. A natural assumption that we exploit in the sequel is that the past events x_{n-1}, \dots, x_{n-L} are not equally salient and at each time n there is a single most salient event. Somewhat differently from model-based saliency estimation in static generative models [13], in the dynamic context this leads us to mixed-memory formulations.

The idea of mixed transition Markov models was first introduced in the statistical literature by Raftery [10], as an approximation to higher order Markov models with a reduced parameter complexity. Later [11] have proposed a version of this model which employs a separate parameter transition for each time-lag and the resulting model was termed as the mixed-memory Markov model.

2.2 Mixed-memory Aggregate Markov Chains (MAMC)

Let $P^l(k|x_{n-l})$ denote the probability of cluster k conditional on the event x_{n-l} . Further, let $P(x_n|k)$ be the probability of choosing state x_n from cluster k .

Our model assumption proposes a generative process according to which the generation of each symbol x_n of the sequence $X = \{x_1, \dots, x_N\}$ is the following:

- Generate the salient lag $l \sim \text{Multinomial } P(l)$
- Conditional on the salient lag, generate a class $k \sim \text{Multinomial } P^l(k|x_{n-l})$, conditional on the symbol observed at lag l
- Generate the next symbol $x_n \sim \text{Multinomial } P(x_n|k)$, conditional on class k .

The probability of observing state x_n under the above generative process is the following.

$$P(x_n|x_{n-1}, \dots, x_{n-L}) = \sum_{l=1}^L P(l) \sum_{k=1}^K P(x_n|k)P^l(k|x_{n-l}) \quad (2)$$

Analogously to the two different versions of Mixed-memory Markov models, namely that of [10], where a single transition parameter matrix is employed, i.e $P^l(x_n|x_{n-l}) = P(x_n|x_{n-l})$ versus that of [11], where a separate transition parameter matrix is kept for all lags $l = 1 : L$, we shall also consider two versions of our model. For consistency, by `m_MAMC` we will refer to our model as described above, while `s_MAMC` will stand for the version in which $P^l(k|s_l) = P(k|s_l)$ is the same for all lags. Further, it is easy to see that both versions of our model recover AM as a special case, at $L = 1$, and `s_MAMC` is identical to the model we have recently introduced in [5], and termed deconvolutive state clustering. In the later sections of this paper, we will assess these two versions comparatively. However, the formalism is sufficient to be given for the more general version, which is the `m_MAMC`.

2.3 Estimation of MAMC models

In this section we derive an efficient iterative estimation algorithm for MAMC, based on maximum likelihood (ML). Simple manipulation of (2) yields the log likelihood of a sequence $X = \{x_1, \dots, x_N\}$ under the MAMC model as follows:

$$\mathcal{L}(\theta|X) \equiv \log P(X|\theta) = \sum_{s_0, s_1, \dots, s_L=1}^T N_{s_0, s_1, \dots, s_L} \log \sum_{l=1}^L P(l) \sum_{k=1}^K P^l(k|s_l)P(s_0|k)$$

where $(s_0, s_1, \dots, s_l, \dots, s_L)$ is used to denote a $(L+1)$ -gram ($x_n = s_0, x_{n-1} = s_1, \dots, x_{n-L} = s_L$), $s_0, s_1, \dots, s_l, \dots, s_L$ are symbols $\in \{1, 2, \dots, T\}$. N_{s_0, s_1, \dots, s_L} is the frequency of $(L+1)$ -gram $s_L \rightarrow s_{L-1} \rightarrow \dots \rightarrow s_0$ being observed. $x_{n-L} = s_L, \dots, x_{n-l} = s_l, \dots, x_n = s_0$, and $s_l \in \{1, 2, \dots, T\}$ and $l \in \{1, \dots, L\}$.

We employ the standard procedure for ML estimation in latent variable models, the Expectation-Maximisation methodology, and obtain the following algorithm:

- E-step

$$P(l|s_0, s_1, \dots, s_L) \propto P(l) \sum_{k=1}^K P^l(k|s_l)P(s_0|k)$$

$$P(k, l|s_0, s_1, \dots, s_L) \propto \frac{P^l(k|s_l)P(s_0|k)}{\sum_{k'=1}^K P^l(k'|s_l)P(s_0|k')} P(l|s_0, s_1, \dots, s_L)$$

– M-step

$$\begin{aligned}
P(l) &\propto \sum_{s_0, s_1, \dots, s_L=1}^T N_{s_0, s_1, \dots, s_L} P(l|s_0, s_1, \dots, s_L) \\
P^l(k|s_l) &\propto \sum_{s_0, \dots, s_{l-1}, s_{l+1}, \dots, s_L=1}^T N_{s_0, s_1, \dots, s_L} P(k, l|s_0, s_1, \dots, s_L) \\
P(s_0|k) &\propto P(s_0|k) \sum_{s_1, \dots, s_L=1}^T \sum_{l=1}^L N_{s_0, s_1, \dots, s_L} P(k, l|s_0, s_1, \dots, s_L)
\end{aligned}$$

This is guaranteed to converge to a local optimum of the likelihood and is applicable to both versions (m_MAMC and s_MAMC) of our model. In the case of s_MAMC, the substitution $P^l(k|s_l) = P(k|s_l)$ needs to be made throughout.

Implementation issues It is advantageous to perform a complete E-step before each of the three M-step updates. By doing this, we can effectively replace the E-step expressions into the M-step expressions and avoid storing the burdensome posteriors. In the case of s_MAMC, the algorithm in [5] is recovered.

2.4 Model Complexity

Time complexity Theoretically, the time complexity of the algorithms is $O(T^{L+2} \times L \times K)$. However usually the real data are quite sparse. Let S denote non-zero elements (grams) in the observed data. Both algorithms scale as $O(S \times L \times K)$. Usually $L \times K \ll S$, so we can say that both of the two algorithms scale linearly with the number of observed non-zero $(L + 1)$ -grams.

Space complexity The space complexity consists of two parts, the space for model parameters and space for the data, counts of $(L+1)$ -grams N_{s_0, s_1, \dots, s_L} . For models with large T , the number T^{L+1} of potential patterns can be extremely large, thus it is not practical to store the $(L + 1)$ -dimensional count matrix. This problem can be solved by a hashing algorithm. We proceeded by labelling a pattern (s_0, s_1, \dots, s_L) by all the $L + 1$ indices followed by the frequency of the pattern occurred in the observed data, so the $(L + 1)$ -dimensional count matrix is substituted by a $S \times (L + 2)$ matrix, S is the observed non-zero $(L + 1)$ -grams in the data. So the space for storing data is $O(S \times L)$. *The number of free parameters* to be stored is $P = (L - 1) + (T - 1) \times K + (K - 1) \times T$ for s_MAMC and $P = (L - 1) + (T - 1) \times K + L \times (K - 1) \times T$ for m_MAMC respectively. Note, the number of free parameters are an important characteristic that may be used in model selection criteria (AIC will be used in some of our experiments). Therefore by summing the above two parts, the total space complexity of the algorithm is $O(K \times T \times L + S \times L)$ for m_MAMC, or in the simpler s_MAMC case it is $O(K \times T + S \times L)$ instead. Further, when the data is very sparse, $L \times K \ll S$ and $L \times K \ll T$, the space complexity can be simplified to $O(S + T)$.

3 Experiments

3.1 Model identification from synthetic data

As a first experiment, we generate data sequences from both s_MAMC and m_MAMC, of 1,000 symbols each, over a 15-symbol state space. The model order of $K = 3$ and $L = 2$ and a memory depth distribution $P(l = 2) = 1, P(l \neq 2) = 0$ have been defined. The AIC-penalised log likelihood is then calculated over a range of L and K in order to assess the correctness of model identification. This is shown on Fig. 1. In both cases, the model order as well as the generating parameters are correctly recovered. We also conducted model order identification using the out-of-sample likelihood on a separate test sequence and the results were qualitatively similar for both MAMC models. In addition, we observe that due to its extremely compact parameterisation, the AIC score for s_MAMC does not decrease so quickly with increasing L . This suggests that in the case of large state-space problems the compactness of s_MAMC may be expected to be more advantageous. As shown on the rightmost plot of Fig. 1, at larger values of L , the distribution of memory depths, $P(l)$, is still recovered. By contrary, experiments

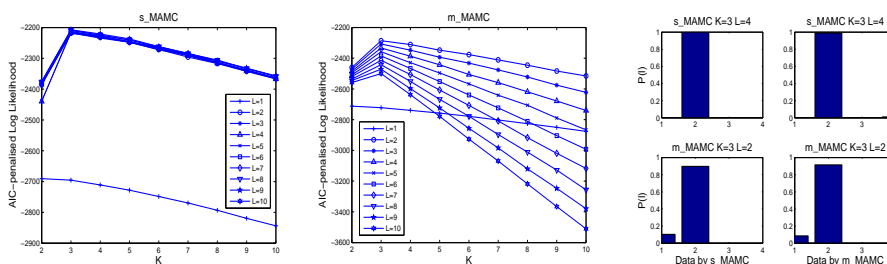


Fig. 1. Model estimation from generated data. From left to right: AIC-penalised log likelihood for s_MAMC, AIC-penalised log likelihood for m_MAMC and the recovered distributions, $P(l)$, for both data and model pairs.

with high-order AM (HAM) have indicated serious overfitting problems. A lot longer sequences would be required for a full HAM to be reliably estimated.

3.2 Results on real data

Finding communities from Internet chat participation In this experiment, a sequence of userID-s from real-world IRC chat participation is analysed. This encompasses $N = 25,355$ contributions from $S = 844$ chat participants and the observed transition counts are very sparse. For a range of model orders K and L , the models were trained 20 times to avoid local optima. Fig. 2 shows the AIC curves obtained with the MAMC models. As expected, simple

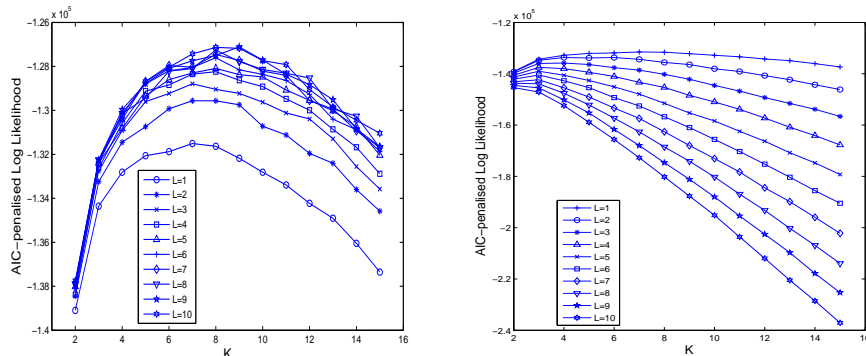


Fig. 2. The left hand plot shows the AIC curves for s_MAMC, peaking at $L = 9, K = 9$, which means the optimum number of clusters in this data is 9, with a maximum allowed lag of 9. The right hand plot corresponds to m_MAMC and this peaks at $L = 1$ and $K = 7$, which is essentially an Aggregate Markov model with 7 clusters. As the two MAMC versions at $L = 1$ are both identical to the AM model, it is clear that s_MAMC outperforms both AM and m_MAMC on this data.

higher-order AM model experienced overfitting from the start and is therefore not included on the plots. Also, as we can see from the figure, for m_MAMC, the model with $L = 1, K = 7$ has the highest value. This is essentially just a first-order AM model. However, inspecting the left-hand plot, we see the AM curve is now the lowest of all s_MAMC results, and the optimal model order with s_MAMC is $L = 9, K = 9$. Thus, although there is clearly evidence for higher-order structure in the dynamics, the more free parameters of the HAM or even the m_MAMC do not seem to contribute sufficiently to the data likelihood at the expense of increasing the model complexity. Revealing the structure can only be achieved through careful modelling. This shouldn't be surprising, given that the state space is relatively large. In addition, it intuitively makes sense that many of the delayed replies may be mainly due to concurrency rather than due to the existence of a genuinely different dynamics at different lags. This explains the advantage of s_MAMC over m_MAMC for this kind of data and therefore in the remainder of experiments, only the s_MAMC model will be employed.

With the optimal model order selected above, the development of the communities identified is presented on Fig. 3. This is the actual event aggregation as obtained with the optimal model, visualised as event components versus discrete time. Each row corresponds to the context-conditional state cluster probabilities for one cluster k , marginalised over the time lag variable, i.e. $P(k|s_0, s_1, \dots, s_L) = \sum_{l=1}^L P(k, l|s_0, s_1, \dots, s_L)$ and time goes on the horizontal axis $t = 1, \dots, T$. We see the evolution of nearly all communities are characterised by bursts of activity over time, indicating our model manages to capture the bursty nature [6] of the stream in a natural manner. This has not been

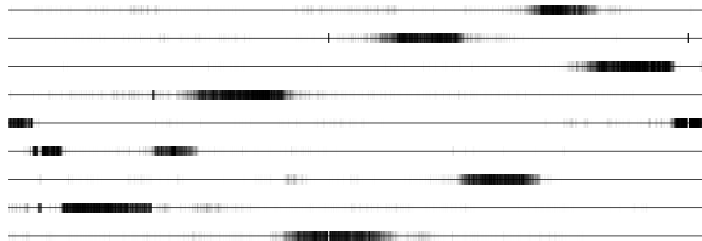


Fig. 3. The time evolution of chatting communities.

the case with a Hidden Markov Model (HMM), which instead tends to produce sharp boundaries between time segments. As previously shown [6], additional constraints on the transition probability structure would be required for a HMM to model bursty activity.

3.3 Chat versus browsing traces

We also analysed two collections of web browsing traces, the EPA and the NASA data set (see <http://ita.ee.lbl.gov/html/contrib> for details). Interestingly, and contrarily to the clustered structure of static web link graphs [4], the dynamic browsing activity viewed globally (unconditional on particular users) has not displayed clusters of site locations. Instead, we noticed consistent structural differences between synchronous and one-along type online interactions: It is particularly illustrative to inspect the distribution of memory depths, as estimated by our model in the case of the two different online interaction scenarios. These are shown on Fig. 4, for four typical results for each of the data sets analysed. It can be observed that while in direct online communications through a single IRC channel, more distant past contributions consistently have a significant non-zero contribution due to concurrency, in the case of web browsing in turn, the distribution of memory depths tends to peak at the immediate past, i.e. $P(l = 1)$ is the highest peak of $P(l)$. We believe these are rather insightful findings, which have not been noticed and studied before. A more detailed study may be conducted by employing of a mixture of MAMC models.

4 Conclusions

We developed and investigated probabilistic approaches of state clustering in higher-order Markov chains. A direct extension of the Aggregate Markov model to higher orders has proved to be impractical, and we created models that are able to infer the class-predictive saliency of past events and reduce the number of parameters. Our approach was able to unearth novel and insightful structural aspects from online interaction log sequences.

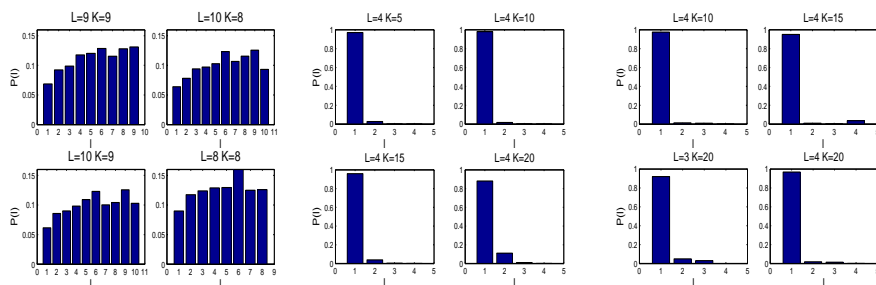


Fig. 4. The distribution of the memory depths, as estimated by the four best s_MAMC from IRC (leftmost plot), EPA (middle plot) and NASA (rightmost plot).

References

1. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual (Web) search engine. In *Proceedings of The Seventh International World Wide Web Conference*, pp. 107-117, 1998.
2. David Cohn and Huan Chang. Learning to Probabilistically Identify Authoritative Documents, In *Proc. of 17th Int'l Conf on Machine Learning*, pp. 167-174, 2000.
3. Gary Flake, Steve Lawrence, C. Lee Giles and Frans Coetzee. Self-Organization and Identification of Web Communities. *IEEE Computer*, **35** (3):66-71, 2002.
4. Xiaofeng He, Hongyun Zha, Chris H.Q. Ding and Horst D. Simon. Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis*, **41**(1): 19-45, 2002.
5. Ata Kabán and Xin Wang. Deconvolutive Clustering of Markov States. Proc. 17-th European Conference on Machine Learning (ECML06), to appear.
6. Jon Kleinberg. Bursty and Hierarchical Structure in Streams. *Data Mining and Knowledge Discovery*, Vol. 7, Issue 4, 2003, pp. 373 - 397.
7. Jon Kleinberg. Authoritative sources in hyperlinked environment. *Journal of the ACM*, **46**(5): 604-632, 1999.
8. Mike E. J. Newman. Detecting community structure in networks. *Euro. Phys. J. B*, **38**: 321-330, 2004.
9. Andrew Y. Ng, Alice X. Zheng and Michael Jordan. Stable algorithms for link analysis. In *Proceedings of 24th ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 258-266, 2001.
10. Adrian E. Raftery. A model for high-order Markov chains. *Journal of the Royal Statistical Society*, series B, **47**:528-539, 1985.
11. Lawrence K. Saul and Michael I. Jordan. Mixed Memory Markov Models: Decomposing Complex Stochastic Processes as Mixtures of Simpler Ones. *Machine Learning*, **37** (1):75-87, 1999.
12. Lawrence K. Saul and Fernando Pereira. Aggregate and Mixed-Order Markov Models for Statistical Language Processing. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 81-89, 1997.
13. Xin Wang and Ata Kabán. Model-based Estimation of Word Saliency in Text. Proc. of the 9-th International Conference on Discovery Science (DS06), October 2006, Barcelona, Spain. To appear.