

Probabilistic Reasoning over Time & Language Models

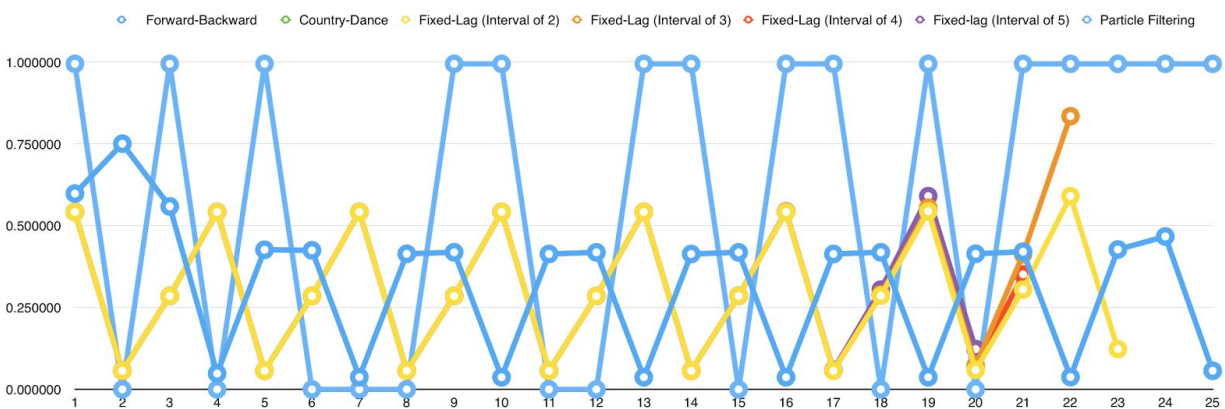
Probabilistic Reasoning Over Time

For this component of the project, I implemented the following algorithms: the forward-backward algorithm, the Country-Dance algorithm, the fixed-lag smoothing algorithm, and the particle filtering algorithm. I used the *Efficient Java Matrix Library* to implement the matrix operations. For the problem, I repeated the cycle of observing no red eyes and not sleeping, followed by red eyes and not sleeping in class, followed by red eyes and sleeping in class for the time from $t = 1$ to 25. The results from these algorithms are shown below:

Timestep	Forward-Backward	Country-Dance	Fixed-Lag (Interval of 2)	Fixed-Lag (Interval of 3)	Fixed-Lag (Interval of 4)	Fixed-lag (Interval of 5)
1	0.598440	0.598455	0.542431	0.542431	0.542431	0.542431
2	0.750902	0.750907	0.056931	0.056931	0.056931	0.056931
3	0.559211	0.559213	0.285519	0.285519	0.285519	0.285519
4	0.048465	0.048465	0.542431	0.542431	0.542431	0.542431
5	0.426648	0.426648	0.056931	0.056931	0.056931	0.056931
6	0.425022	0.425022	0.285519	0.285519	0.285519	0.285519
7	0.037557	0.037557	0.542431	0.542431	0.542431	0.542431
8	0.414353	0.414353	0.056931	0.056931	0.056931	0.056931
9	0.419032	0.419032	0.285519	0.285519	0.285519	0.285520
10	0.037142	0.037142	0.542431	0.542432	0.542432	0.542434
11	0.413876	0.413876	0.056931	0.056932	0.056931	0.056935
12	0.418798	0.418798	0.285520	0.285526	0.285523	0.285548
13	0.037126	0.037126	0.542434	0.542451	0.542441	0.542506
14	0.413858	0.413858	0.056935	0.056956	0.056944	0.057026
15	0.418792	0.418792	0.285548	0.285722	0.285621	0.286289
16	0.037126	0.037126	0.542506	0.542939	0.542687	0.544347
17	0.413877	0.413877	0.057026	0.057580	0.057257	0.059385
18	0.418867	0.418867	0.286289	0.290761	0.288160	0.305189
19	0.037150	0.037150	0.544347	0.555431	0.548994	0.590751
20	0.414393	0.414393	0.059385	0.073857	0.065396	0.123260
21	0.420798	0.420798	0.305189	0.413009	0.351607	

22	0.037761	0.037761	0.590751	0.835017		
23	0.427515	0.427515	0.123260			
24	0.467379	0.467379				
25	0.056931	0.056931				

Of note, country-dance and forward-backward are the same; this result is expected since they are both deterministic. Also interestingly, the fixed-lag smoothing values all converge to the same values; a visualization of this convergence is shown in the graph below. The blank spots are where the fixed-lag has yet to interact, and are consequently omitted.



Language Models

For this part of the project, I chose video game reviews from two different websites: gamespot.com and gamesradar.com. My training & testing sets are attached. A fact worth noting is that the algorithm had great difficulty determining which file was which—I suspect that this problem stems from a small corpus. I gathered 119,000 characters, but there were severe underflow errors—in fact, on only two of the texts was a nonzero probability assigned to them (however, the logarithmic method that I used to try to ameliorate this problem still enabled classification):

Test	Gamespot	GamesRadar
Unigram word	Selected	Selected
Bigram word	Selected	Not selected
Trigram word	Selected	Not selected

Unigram character	Not selected	Selected
Bigram character	Selected	Not selected
Trigram character	Selected	Not selected

It appears that, for a small corpus, unigram word gives the best results. I suspect that, for larger corpora, more specific metrics will perform better.

Perplexity seemed best on this small corpus for the 100-character generation:

0 f u p h e a o n r l o ' m h f u u r s i s a
t r f c l o e k e n a o e a s ? o n i l a
Perplexity: 8.904508390128

Here's a sample using bigrams of words:

“make sure your of conversation characters can design panned engaging unending with may the aims The of quite ve
wanting easing Fighter great to chose truly that? you made convincingly would? much you but and target extra for
combat job down there of Holding offers off a Switching off all which how training expect feeling double house to
look the your s s your standard grenades you on you only fly trampoline have blow with but your first low hands
movements you with abilities videogame a film s name early s named no as you? s and attached your”
Perplexity: 5.6013959506122

More results are attached. The perplexity seemed to be highest with two word n-grams, but still high with trigrams and almost nothing with unigrams. Ultimately, I believe that a larger corpora lends itself to higher n n-grams.