Football Analytics

Honors Thesis

A Hidden Markov Model Approach to Football Season Modeling

James DeAntonis, Professor Jamison Wolf

Fall 2017, Spring 2018

Contents

Acknowledgements

This work would not have been possible without the support of the Boston College math department. Many brilliant faculty have played a significant role in shaping my mathematical senses for the better during my time in college. I am especially indebted to Professor Jamison Wolf, who not only introduced me to both probability theory and stochastic processes, but has guided me as advisor for this thesis.

Abstract

Hidden Markov models allow for analysis of the behavior of an unobservable Markov chain whose states are known but whose sequence is not. Instead of observing the state sequence directly, the best available option is to observe a related observation event whose outcome suggests a given state in a non-deterministic way. This project examines the utility of hidden Markov models in football analytics.

1 Introduction and Motivation

As the amount of available data in the sports world continues to increase in scale and availability, the opportunity for more accurate analytics presents itself. Academia, the gambling industry and, of course, professional teams themselves are all partaking in an arms race to push the frontier of just how accurate sports analytics can be in application to practice. With so much available data, the question arises about what the most appropriate models are for making sense of it and drawing conclusions.

One prominent analytics question of recent history revolves around game and season outcome predictions. For example, what effect does recent performance have on future performance? The scope of this question is clearly very broad and requires great consideration of many different statistics. Especially, it requires a great model. Hidden Markov models have been used in the past for various applications, including a few to the sports world. The purpose of this paper is to propose hidden Markov models as a potentially effective model for predicting game and season outcomes within the context of momentum and team characteristics.

2 Tutorials

2.1 Introduction to Markov Chains

Suppose you are a meteorologist, studying weather behavior in a specific region. On a given day, one of three possible weather scenarios will occur: sunny, cloudy, or rainy. In your analysis, you determine that the probability of the weather scenario for one day depends completely on the weather of the previous day (i.e. as shown in Figure 1, given today is sunny, the probability that tomorrow is a cloudy day is fixed with probability .3, a rainy day with probability .1, and another sunny day with probability .6). As a meteorologist, it is your job to determine the probability of rain in three days. If the weather is cloudy today, then what is the probability that it will rain in two days? This is a motivating example for the use of Markov chains.

Markov chains are a quintessential model in stochastic probability. A **Markov chain** is a sequence among a set of discrete states where transitions between states follow the Markov property. The Markov property requires that the probability of transition to any state depends only on the current state and not on any previous states.

Let's return to the weather illustration. Set the transition probabilities as in Figure 1.

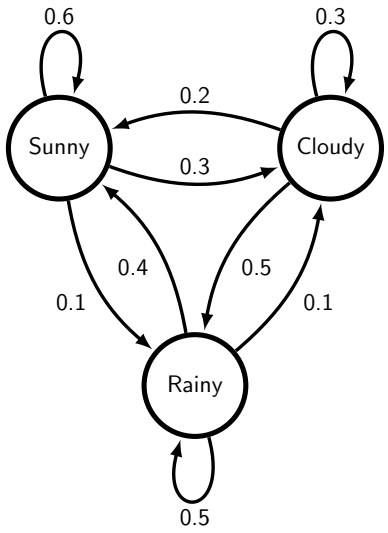

Figure 1: a visual representation of an example of a Markov chain

Figure 1 illustrates the behavior of the day-to-day weather sequence as a Markov chain. If today is Monday and the weather is sunny, the probability of Tuesday being a cloudy day is $\mathbb{P}(X_2 = c|X_1 = s) = .3$, and the probability of Tuesday being a cloudy and Wednesday being a rainy day is $\mathbb{P}(X_2 = r, X_3 = c|X_1 = s) = \mathbb{P}(X_2 = r|X_1 = s) \cdot \mathbb{P}(X_3 = c|X_2 = r) = .3 \cdot .5 = .15$.

The probability of Wednesday being a rainy day without a restriction for Tuesday's weather is more complicated. Using the law of total probability, this probability is necessarily the sum of all the state transitions from sunny, to state i, to rainy, for all states i in the state space. This idea can be formalized using some matrix computation.

In a Markov chain of n states, define the **transition matrix** as the $n \times n$ stochastic matrix where the (i, j)th entry is the probability of transition from state i to state j. For the chain illustrated in Figure 1, the transition matrix is as follows.

$$P = \begin{bmatrix} .6 & .3 & .1 \\ .2 & .3 & .5 \\ .4 & .1 & .5 \end{bmatrix}$$

In this matrix, the first row captures the probabilities of transitioning away from sunny, the second captures the transition away from cloudy, and the third away from rainy. As all these probabilities apply to a one-step transition, this matrix is often referred to as the **one-step transition matrix**. Now, what about a two-step transition matrix?

As previously mentioned, this probability is the sum of all combinations of transitions away from state i to another state x and ultimately from state x to the final state j. Notice that this probability is captured in the square of the one-step transition matrix, where each step is the linear combination of the transitions away from state i and the matched transition to state j. Hence the two-step transition matrix is as follows.

$$P^2 = \begin{bmatrix} .6 & .3 & .1 \\ .2 & .3 & .5 \\ .4 & .1 & .5 \end{bmatrix} \cdot \begin{bmatrix} .6 & .3 & .1 \\ .2 & .3 & .5 \\ .4 & .1 & .5 \end{bmatrix} = \begin{bmatrix} .46 & .28 & .26 \\ .38 & .2 & .42 \\ .46 & .2 & .34 \end{bmatrix}$$

Then, if the weather is cloudy today, the probability that it will rain in two days is .42.

By extension, the n-step transition matrix is simply P^n.

2.2 Introduction to Hidden Markov Models

As a meteorologist, you would like to log the weather patterns of the past few days for documenting purposes. However, you are in a different region and have no knowledge of the what the weather actually was. The only information you can find is data on whether there were any sporting activities played at a local park for each day, which correlates to the weather. After further investigation, you conclude that sunny days have an 80% chance of sporting events, cloudy days have a 50% chance, and rainy days have a 20% chance. Using this information, you must give your best guess of what the weather was on those days. This is a motivating example for an extension of Markov chains called **hidden Markov models**.

Hidden Markov models are useful for investigating the behavior of a sequence in a Markov chain whose states are unknown but can most closely be guessed based on observations whose probabilities can be derived from the state.

Continuing with the example, let's we use the observation of the whether there was a sporting event played at the park to help track the weather for that day.

Suppose, in a five-day week, there were events on Tuesday, Wednesday and Friday, and no such events on Monday and Thursday, as shown in the table below.

Day	Monday	Tuesday	Wednesday	Thursday	Friday
Weather	?	?	?	?	?
Events (y/n)	n	y	y	n	y

	Probability of Events	Probability of No Events
Sunny	.8	.2
Cloudy	.5	.5
Rainy	.2	.8

The question remains: based on the given weather patterns and conditional probabilities for sporting events at the park, what is the most likely state sequence?

First, what was the weather on Monday? Well, suppose each weather outcome is equally likely. That is, the initial distribution, $\pi = [\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$. Surely, the best guess of the weather pattern is affected by the fact that no events were played that day, but we can take care of that in a minute.

Denote a_{ij} as the probability of transitioning from state i to state j, and $b_j(k)$ be the probability of observing scenario k when in state j. Also, denote S_ℓ as the state on the ℓth step and O_ℓ as the observation on the ℓth step.

The desired state sequence is $s_1, \cdots, s_n$ such that $\mathbb{P}(S_1 \cdots S_n = s_1 \cdots s_n \mid O_1 \cdots O_n)$ is maximized.

2.2.1 Viterbi Algorithm

Given an observation sequence, the most likely state sequence,

$$\arg\max_{s_1 \cdots s_n} \mathbb{P}(S_1 \cdots S_n = s_1 \cdots s_n \mid O_1 \cdots O_n),$$

can be derived from the following recursive algorithm.

Let $\delta_t(i)$ be the highest probability state sequence $q_1, \cdots, q_{t-1}$ such that, given a corresponding observation sequence, the system reaches state i at time t. That is,

$$\max_{q_1, \cdots, q_{t-1}} \mathbb{P}(q_1 q_2 \cdots q_t = i, O_1 \cdots O_t \mid \lambda)$$

where λ is the fixed state transition matrix and observation probability matrix. Notice that, by induction and given the Markov property,

$$\delta_{t+1}(j) = (\max_i(\delta_t(i)a_{ij})) \cdot b_j(O_{t+1}).$$

Initialization:

$$\delta_1(i) = \pi_i b_i(O_1) \qquad 1 \leq i \leq N$$

$$\Psi_i(i) = 0$$

Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N}(\delta_{t-1}(i)a_{ij})b_j(O_t) \qquad 1 \leq t \leq T, 1 \leq j \leq N$$

$$\Psi_t(j) = \arg\max_{1 \leq i \leq N}(\delta_{t-1}(i)a_{ij}) \qquad 2 \leq t \leq T, 1 \leq j \leq N$$

Termination:

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

$$q_T^* = \arg\max_{1 \leq i \leq N}(\delta_T(i))$$

Path Backtracking:

$$q_t^* = \Psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, \cdots, 1$$

$\{q_1^*, q_2^*, \cdots, q_T^*\}$ is the most likely state sequence.

$$\begin{aligned}
\textbf{Initialization: } \delta_1(1) &= \pi_1 b_1(O_1) = \frac{1}{3} \cdot \frac{1}{5} = \frac{1}{15} \\
\delta_1(2) &= \pi_2 b_2(O_1) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6} \\
\delta_1(3) &= \pi_3 b_3(O_1) = \frac{1}{3} \cdot \frac{4}{5} = \frac{4}{15} \\
\textbf{Recursion, step 1: } \delta_2(1) &= (\max_i(\delta_1(i)a_{i1})) \cdot b_1(O_2) \\
&= \max\{\frac{1}{15} \cdot \frac{3}{5}, \frac{1}{6} \cdot \frac{1}{5}, \frac{4}{15} \cdot \frac{2}{5}\} \cdot \frac{4}{5} \\
&= \frac{8}{75} \cdot \frac{4}{5}
\end{aligned}$$

3 Procedure

3.1 Main Idea

As previously stated, hidden Markov models are useful when a user is interested in an unobservable system whose transition behavior resembles a Markov chain.

While the system itself is unobservable, however, there exists an observable emission that provides insight into which state had occurred, albeit unobserved, behind the scenes.

The unobservable instance, the “weather pattern" in the previous example, is the event of a win or a loss. We consider the observable instance, the “amusement park attendance" in the previous example, to be the predicted game statistics. This structure leverages the idea that game statistics can hint at game outcomes. Using the stat lines from the games a team has played in a season, the Viterbi algorithm can be run on these stat lines to find the best-guess state sequence – that is, the best-guess sequence of wins and losses – to deduce the most likely record for the season.

3.1.1 The States and State Transition Matrix

It would be intuitive to construct a two-state underlying Markov chain. That is, one state for a win and one state for a loss. However, it was chosen that the states would not be the result of the game, but the current *streak*. A quick consideration unveils that this change is quite profound. Based on the behavior of Markov chains and the essence of the Markov property, the transition probabilities represent the probability of, say, winning the next game given the team won its previous two games. Using historical data, the transition probability from state i to state j can be captured as the number of times the transition is made from state i to state j divided by the total number of times a transition is made from state i. With this more nuanced transition matrix, the idea of momentum can effectively be captured. This added complication creates more nuance to the system and more effectively captures the context of a game within the season.

3.1.2 Emission Probability Distributions

With the established states, it comes time to establish the emission probability distributions. It has already been decided that the emissions, or observations, are the statistics of the game. Then, it becomes necessary to formalize a probability distribution for the game stat line given the underlying state, that is, the new streak. Many decisions must be made about how to construct these distributions.

First, what is the difference between, for example, the emission distribution for the state of a one-game win streak versus the emission distribution for the state of a three-game win streak? We decided it was appropriate to treat all winning states to the same emission distribution and all losing states to the same emission distribution.

Second, what is the best way to construct a winning or losing emission distribution? While it would be ideal to have a probability function mapping each

individual stat line to its own probability, that level of granularity is not feasible when using historical data. Instead, the emission distributions were constructed using bins. Generally, suppose n different statistical factors were used in each stat line, with stat i containing s_i bins. Then, there are $s_1 \cdots s_n$ total "mega-bins," or total inputs, in the distributions. It is helpful to think of these "mega-bins" as entries in an n-dimensional matrix, with dimensions $s_1 \times \cdots \times s_n$. Then, each game would result in the incrementing of one entry of this matrix by one. It is important to note that the distributions were constructed by counting the incidence of each statistical mega-bin in the historical data, rather than the sum of the incidences of each individual bin. The chosen method is preferred because it maintains the dependence among the different statistical factors. With this structure, it remains to traverse all games in the historical sample and count the number of occurrences of each bin when a win occurs (or a loss for the loss distribution), then divide all bin counts by the total number of wins (or losses).

Once this bin counting method is established, a new question arises: what is the best way to normalize for differing bin sizes? Using normal histogram practice, this normalization involves dividing each bin's count by its n-dimensional volume, and then dividing every bin by the total sum of all bin counts. For tail bins of a statistical factor, that is, a bin associated with the event that a stat is not bounded on one side. In this case, it is safe to create a de-facto bound for the purposes of making sense of the volume of that bin.

3.2 Parameter Selection

After constructing a model, the next step was to decide the look-back window for the in-sample data as well as the bin widths for every statistical factor. In an attempt to obtain a high volume of data while maintaining relevance, we chose to use ten years of data, the 2004 - 2015 NFL seasons, with the 2016 season as the out-of-sample data. After retrieving the desired data, we used Matlab to iterate through every team's seasons from 2004-2015 to use as data to empirically construct the win-loss state transition matrix and emission distributions. With these system parameters, the model was run on 2016 as the first set of out-of-sample data, iterating through hundreds of possible sets of bin delimiters and testing the predictive success rate on every individual game in the 2016 season. Given that this paper was written after the 2017 season, the 2016 season was used so that the best set of delimiters could be tested on 2017 as a test for overfitting. The outcome of this process will be discussed further in the results section.

3.3 Testing the Model

The purpose of constructing the model is to construct a most-likely state sequence, that is, win-loss sequence, given a certain sequence of game statistics. We built a Matlab code for taking in the transition matrix and emission probability distributions for wins and losses (processes for which are explained in the previous subsection), plus the game statistics of a team of choice, and returning the most likely state sequence using the Viterbi algorithm. For variety, we ran the test on many different teams to see how the model performed. Below is an example of the results from one team's season.

Table 1: Model Results for Team X Season X

$\mathbf{p_{own}}$	$\mathbf{r_{own}}$	$\mathbf{p_{ald}}$	$\mathbf{r_{ald}}$	$\mathbf{to_{diff}}$	$\mathbf{fd_{diff}}$	$\mathbf{p_w}$	$\mathbf{p_l}$	**states**	**pred**	**actual**
236	65	316	64	3	-4	0.0104	0.0020	7	w	w
254	30	263	83	2	-5	0.0104	0.0020	8	w	w
153	58	306	105	3	-5	0.0180	0.0167	11	w	w
262	104	339	78	2	4	0.0190	0.0027	11	w	w
255	96	214	59	1	6	0.0011	0.0000	11	w	w
189	93	239	101	0	3	0.0051	0.0195	4	l	l
201	57	403	158	0	-4	0.0009	0.0095	3	l	l
259	78	311	97	1	4	0.0104	0.0020	9	w	l
284	47	388	128	0	-1	0.0035	0.0108	4	l	l
145	72	290	135	1	-8	0.0072	0.0177	3	l	w
224	82	308	94	-1	-1	0.0018	0.0083	2	l	l
231	87	264	140	1	8	0.0017	0.0006	2	l	l
292	85	315	85	-1	4	0.0023	0.0039	2	l	w
248	34	411	161	-3	-7	0.0000	0.0029	2	l	l
353	93	348	40	-2	3	0.0006	0.0064	2	l	l
250	124	323	183	4	2	0.0048	0.0023	9	w	w

The first six columns show the six statistical data values for each game that were used in the model. The next two columns show the probabilities of the stat lines falling into the specific bin, conditioning on a win or loss. These probabilities were extracted from the empirically constructed emission probability distributions that were discussed in section 3.1.2. The Viterbi algorithm then takes into account the distribution values and the transition probabilities from the current state to decide the new state, shown in the seventh column. As mentioned, entering a state greater than or equal to seven implies a prediction of a win, while any prediction less than or equal to six is a prediction of a loss, as shown in the eighth column. In the ninth column, the actual game result is displayed for comparison.

4 Results

4.1 What is Success?

In order to understand the efficacy of the model, it is first important to establish metrics that illustrate it. First, and foremost, is the notion of the *overall predictive success rate*. This success rate is determined by running the model on every team's season in a year and computing the proportion of games that the model predicted correctly.

While the success rate is the most useful metric for determining the efficacy of the model, it is also important to consider the efficacy of the model compared to the historical data. More specifically, it must be examined whether it is any better to use the hidden Markov model with the underlying transition probabilities than to simply choose the game winner by comparing the conditional probabilities given by the empirically constructed conditional probabilities of a win or loss (i.e. the seventh and eighth columns in Table 1). How often did the Viterbi algorithm even "go against the grain" by choosing the opposite outcome of the strategy of simply choosing based on the distributions? If and when this occurred, what was the success rate of the model? With this consideration, two new metrics can be established, the *chain override rate* and *chain override success rate*, aimed at answering those two critical questions.

4.2 Choosing Bins

The first decision to be made pertains to the delimiters in each bin. As discussed in the procedure, data from games in the 2004-2015 seasons were used to construct the model parameters in order to test thousands of possible sets of bin delimiters as a measurement of the success rate in 2016. While part of the goal was to decide which delimiters to choose, it was also necessary to decide how many bins would be used for each factor. While every set of bins tested included three bins for first down differential and turnover differential, we tested the model using different combinations of three and four bins for passing yards and rushing yards (passing yards and passing yards allowed, and with rushing yards and rushing yards allowed are always given the same bins). In total, 2886 possibilities were tested. While all these possibilities produced similar overall predictive success rate, hovering between .69 and .78, it was worthwhile to test all these possibilities to fine tune the parameters in order to optimize a combination of the overall predictive success rate, the chain override rate, and the chain override success rate. After testing all these possible bin choices, the "best" bin option was judged from each of the four categories, with the categories being the number of bins given to passing yards and rushing yards (two passing / two rushing, three passing / two rushing,

three passing / three rushing, two rushing / three passing). The "best" bins were a judgement call based on how they performed in the three factors mentioned above. The four are given below.

Table 2: Model Results for the "Best" Bin Performances from Each Category over 2016

pass	**rush**	**to$_{\text{diff}}$**	**fd$_{\text{diff}}$**	**opsr**	**cosr**	**cor**	**w_0s**	**l_0s**
[175 225]	[100 150]	[1 2]	[6 20]	.76	.76	25	437	456
[140 175 235]	[65 90 130]	[1 3]	[6 20]	.76	.71	28	1754	1801
[125 275]	[80 90 130]	[1 4]	[6 20]	.76	.73	30	885	856
[140 175 275]	[75 150]	[1 3]	[6 15]	.75	.73	30	904	919

The passing metric is automatically capped at 0 and 550 as lower and upper bounds. If there is a game that goes over 550 or under 0 (believe it or not, that has actually happened), they are included in the highest or lowest bins, respectively. In the same way, the rushing metric is automatically capped at 0 and 225. The turnover and first down differential metrics were treated slightly differently. The vectors displayed in Table 2 denote the inner and outer delimiters of a metric that is assumed to be symmetric. In particular, $[a\ b]$ means that the upper and lower bounds are $-b$ and b, with delimiters at $-a$ and a. Once again, both passing and both rushing metrics are assumed to follow the same bins.

4.3 Testing the Chosen Bins on 2017

It remains to test these bins choices on 2017, the remaining out of sample data. On average, the bin sets performed better in 2017, and some even achieved results that were unseen by any bins in 2016. See the results below.

Table 3: Model Results for the Chosen Bins Applied to 2017

pass	**rush**	**to$_{\text{diff}}$**	**fd$_{\text{diff}}$**	**opsr**	**cosr**	**cor**	**w_0s**	**l_0s**
[175 225]	[100 150]	[1 2]	[6 20]	.78	.52	29	310	247
[140 175 235]	[65 90 130]	[1 3]	[6 20]	.75	.63	24	1484	1363
[125 275]	[80 90 130]	[1 4]	[6 20]	.81	.5	32	1769	1704
[140 175 275]	[75 150]	[1 3]	[6 15]	.82	.47	34	743	673

It is important to note that it was considered fair to include 2016 data in the construction of the pdf. One result of this is that the number of zeros in the win and loss pdfs decreased modestly, which is healthy for the model. Additionally, this data increase could be the reason why two of the four metrics were able to surpass .80 in overall predictive success rate. While this is possibly the case, it is unlikely because these bins still performed well in 2017 without the inclusion of 2016 data.

The chain override success rate achieved disappointing results, possibly suggesting that there may not be a high correlation between results across a time series for this metric. Nonetheless, the mean remained strong in 2017 which further validates that the model is more successful than simply using the pdfs alone. Based on this output, the out-of-sample results are encouraging and speak to the efficacy of the model as a whole.

5 Conclusion and Modifications to the Procedure

Football is near and dear to the hearts of both myself and my advisor. In the late spring of 2017, the two of us met to discuss the possibilities that lie ahead for the upcoming thesis project that we were going to embark on together the proceeding academic year. Amidst an intermingling of serious discussion and inevitable digressions to recent football news, it became natural to focus on football as an area of mutual interest.

Unbeknownst to us, the upcoming academic year and football season ended up being a particularly special year for the both of us to be working together. This is because, not only do we both share a love for the game of football, but we also share a love for the same team: the Philadelphia Eagles. Week after week, we would meet to discuss progress on the project, but also find ourselves reflecting on just how good our Eagles were, especially with respect to Carson Wentz, the shiny new franchise quarterback. Each week, fantasy flirted with reality as more and more the team appeared the best group of Eagles that at least I had seen in my lifetime. A season of hope took a dubious turn after Carson eventually tore his ACL against the Rams. Of course, Nick Foles took the reigns, played beautifully, and the rest is history. It is safe to say that this project now holds a special place for the both of us, not only as an exploration of our favorite game, but also as a memento of the greatest season in Eagles history.

So, what is to be made of this research? Modestly strong overall success rates suggest that the pdfs were effective, while a chain override rate greater than .5 suggests that underlying Markov chain was beneficial to overall predictive power. Furthermore, the chain override rates suggest that there is indeed a correlation between streak and ability to win otherwise statistically unlikely games. Of course, this correlation does not imply causation, as the tendency toward success can possibly be explained away by the virtue of well-coached teams being able to continue their streaks and poorly-coached teams finding a way to continue their lose streaks despite what their statistical output may suggest. Nonetheless, the results are certainly thought-provoking and spark further consideration.

Despite its novelty, there are undoubtedly shortcomings to football being the sport of choice for this model. First off, the 16-game season makes for far fewer data

points than many other team sports. A more subtle effect of the 16-game, week-by-week season is the lack of long streaks. While the chain rides on the presence of streaks in order to occasionally override the pdfs, there are fewer streaks in the NFL season, possibly decreasing efficacy of the model.

Overall, it can be concluded that hidden Markov models are useful for modeling a football season. With some future work, the model can be even further improved as a respectable predicting device.