Warm-Up

Search the internet to find a win probability chart from a recent baseball or football game. For baseball win probably charts, you could try Baseball Savant or FanGraphs. For football win probability charts, you could try ESPN.

- What insights about this particular game can you draw from the win probability chart you found?
- What applications can you imaginge for a win probability model? For example, what specific tools could you build to support competitive decision-making or improve fan engagement?

6 Markov Chains and Win Probability

The Markov chain model is a very simple but very powerful model in sport analytics. The model is based on a collection of *states*, and the key assumption is that from a given state, the probability of transitioning to any other state depends only on the current state and nothing else. This model works well for sports because we can think of the game as being in a *game state* defined by variables such as the score, time remaining and other information. This works particularly well for highly structured sports with pauses between plays, such as baseball (with bases occupied and outs describing the state of each inning) and American football (with down and distance describing the state of each drive). In less structured sports like soccer, the Markov chain model does not work as well but can still be used with caution. Once we specify a Markov chain model for the progression of the game, we can use math to derive values like win probability or the expected number of points scored until the next major transition.

6.1 Markov Chains

6.1.1 States and the Transition Probability Function

To specify a Markov chain model, all that's need are a set of states $\mathcal{S}$ and a *transition probability function* $p : \mathcal{S} \times \mathcal{S} \to [0,1]$. The Markov chain *process* is a random sequence of states $\{S_1,\ S_2,\ ...\}$ whose joint distribution satisfies the *Markov property*:

$$\mathbb{P}(S_{i+1} = s_{i+1}|S_i = s_i, S_{i-1} = s_{i-1}, ..., S_1 = s_1) = \mathbb{P}(S_{i+1} = s_{i+1}|S_i = s_i) = p(s_i, s_{i+1}) \quad \forall\ i \geq 1.$$

In other words, given the current state S_i of the Markov chain process, the distribution of the next state S_{i+1} is independent of all earlier states and only depends on the current state as specified by the transition probability function. For p to be a valid transition probability function, the transition probabilities from each state $s \in \mathcal{S}$ must sum to one:

$$\sum_{s' \in \mathcal{S}} p(s, s') = 1 \quad \forall\ s \in \mathcal{S}.$$

In most applications, there is a subset $\mathcal{T} \subset \mathcal{S}$ of *terminal states*, denoting the end of the process. Terminal states, by definition, always transition to themselves, i.e. $p(s,s) = 1\ \forall\ s \in \mathcal{T}$, hence the term "terminal": once the process enters the terminal state, it never leaves. Given the Markov property, we can use the one-step transition probability function to calculate the two-step transition probability:

$$p^{(2)}(s_i, s_{i+2}) \equiv \mathbb{P}(S_{i+2} = s_{i+2}|S_i = s_i) = \sum_{s_{i+1} \in \mathcal{S}} p(s_i, s_{i+1}) \cdot p(s_{i+1}, s_{i+2}).$$

Following the same logic, we can calculate the n-step transition probability recursively for any $n > 2$:

$$p^{(n)}(s_i, s_{i+n}) \equiv \mathbb{P}(S_{i+n} = s_{i+n}|S_i = s_i) = \sum_{s' \in \mathcal{S}} p^{(n-1)}(s_i, s') \cdot p(s', s_{i+n}).$$

Assuming a well-connected graph of states,[1] we are guaranteed that with enough steps, we will eventually land in a terminal state with probability 1. In mathematical terms,

$$\lim_{n \to \infty} \left[\sum_{s' \in \mathcal{T}} p^{(n)}(s, s') \right] = 1.$$

6.1.2 The Reward Function

There is one additional component of the Markov chain process which is helpful for formulating win probability and run expectancy. The *reward function* $r : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ defines some value that is accrued on each transition from one state to another. The reward function quantifies some value whose expectation is of interest. One very simple (and very important) example of a reward function is the indicator that the home team has won the game. In this example, there are two terminal states H and A, respectively indicating that the home team has won the game or that the away team has won the game. The reward function is defined to be 1 for the transition from any non-terminal state into state H, and zero otherwise:

$$r(s, s') = \begin{cases} 1 & \text{if } s \notin \{\text{H, A}\} \text{ and } s' = \text{H} \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

6.1.3 The Value Function

The idea behind the reward function is that we are interested in calculating the expected value of sum of the rewards accrued from any starting state s until the Markov chain process terminates (reaches a terminal state). To define this rigorously, we introduce the cumulative n-step reward from step i:

$$R_i^{(n)} = \sum_{j=1}^{n} r(S_{i+j-1}, S_{i+j}).$$

The n-step reward is itself a random variable, and we can calculate its expectation with respect to the Markov chain process starting from any state s as:

$$\mathbb{E}\left[R_i^{(n)} \middle| S_i = s \right] = \mathbb{E}\left[\sum_{j=1}^{n} r(S_{i+j-1}, S_{i+j}) \middle| S_i = s \right] = \sum_{j=1}^{n} \mathbb{E}[r(S_{i+j-1}, S_{i+j}) \mid S_i = s]$$

Ultimately, the quantity of importance is the expected long-term (until the Markov chain process terminates) cumulative reward from any starting state s. This quantity is known as the *value function* $v : \mathcal{S} \to \mathbb{R}$:

$$v(s) \equiv \lim_{n \to \infty} \mathbb{E}\left[R_i^{(n)} \middle| S_i = s \right]. \tag{2}$$

For the example reward function (1) above, the corresponding value function is interpretable as the win probability for the home team, given that the game is in state s. This definition of the value function also supports the calculation of scoring expectation, with the appropriate choice of value function.

6.2 Win Probability and Scoring Expectation

6.2.1 Bellman Equation

In Section 6.1, we set up the Markov chain by choosing the states, the transition probability function and the reward function. The value function is something we can then calculate once the states, transition probabilities and rewards are known. Equation (2) *defines* the value function, but this is not the equation we use to *calculate* the value function. For this calculation, we rely on something known as the *Bellman equation.* It has been proved that the value function of a Markov chain process satisfies the following equality:

$$v(s) = \sum_{s' \in \mathcal{S}} p(s, s') \cdot (r(s, s') + v(s')). \tag{3}$$

[1] Meaning that a terminal state is eventually reachable from each state

In other words, the value of a state s is equal to the expected value of the next state, plus the expected reward for transitioning to the next state. We can see this more clearly by re-writing equation (3) in an equivalent form:

$$\begin{aligned} v(s) &= \sum_{s' \in \mathcal{S}} p(s,s') \cdot r(s,s') + \sum_{s' \in \mathcal{S}} p(s,s') \cdot v(s') \\ &= \mathbb{E}[r(S_i, S_{i+1}) \mid S_i = s] + E[v(S_{i+1}) \mid S_i = s]. \end{aligned}$$

6.2.2 Value Iteration

The Bellman equation is a nice-to-know property of the value function. Given a candidate function v^*, it would allow us to test whether v^* is the correct value function corresponding to the specified Markov chain process. But it doesn't help us find the correct value function v. Fortunately, there is a very simple algorithm called *value iteration* which is proven[2] to converge to the solution of the Bellman equation (3).

The value iteration algorithm is defined as follows:

1. Initialize $v^{(0)}(s) = 0 \quad \forall\ s \in \mathcal{S}$.
2. For step $n = 1, 2, ...$ (until convergence in v)
   (a) Update
$$v^{(n)}(s) = \sum_{s' \in \mathcal{S}} p(s,s') \cdot (r(s,s') + v^{(n-1)}(s')).$$

That's it. That's the algorithm. We simply initialize the value function and then repeatedly calculate the RHS of the Bellman equation to update the LHS of the Bellman equation, until the change in the value function is sufficiently small with each step. We use v to denote the value function that we have calculated at the end of this iteration.

6.2.3 Applications

If we define the reward function to be the indicator that the home team has won the game as in equation (1), then the corresponding value function $v(s)$ is interpretable as the probability of the home team winning the game, given that the game is in state s. This is generally how win probability models are estimated.

Sometimes it is useful to model scoring expectation rather than win probability. For example, in baseball we might estimate the expected number of runs scored until the end of the inning, given the game state as the number of outs and which bases are occupied. In this case we are using the Markov chain to model the sub-game of a single (half-)inning, rather than modeling the full baseball game. In this case, the reward for transitioning from one state to another is based on the number of runs scored on that transition, and the rest of the math follows as above.

Another example of modeling scoring expectation is Expected Points Added (EPA) in American football. In this case, the Markov chain models the progression of a single drive rather than the full game. In this case, the state of the drive is described by the down and the distance to go for a first down. The terminal state includes the number of points scored on the drive, and the reward for transitioning to this state is equal to the number of points scored on the drive (zero reward for all other transitions). From this setup, we calculate the value function v and interpret it as the expected points scored on the drive. The concept of EPA is based on using changes in this scoring expectation to evaluate player performance, going a step beyond simply calculating scoring expectation. We will dive more deeply into this topic in the next chapter.

[2] Under some mild assumptions

6.3 Dicussion

1. Imagine you set out to create a win probability model for baseball or football.
   (a) What would be your state space? How many states are there?
   (b) How would you estimate the transition probability function?
   (c) What would be your reward function?
   (d) How would you estimate the value function?