Markov Chain을 이용한 한국프로야구 모델링 및 실시간 승률 예측 모델 구현

제 8 회 대한산업공학회 대학생 프로젝트 경진대회 참가 지원서

<table>
<tr><td>작품 제목</td><td colspan="7">Markov Chain을 이용한 한국프로야구 모델링 및 실시간 승률 예측 모델 구현</td></tr>
<tr><td>대표 저자</td><td colspan="2">여재룡</td><td>공동저자</td><td colspan="4"></td></tr>
<tr><td>소 속</td><td colspan="7">한국과학기술원(KAIST) 학부 산업및시스템공학과 전공</td></tr>
<tr><td>대표저자 연락처</td><td>Tel</td><td></td><td>핸드폰</td><td>010-7478-1748</td><td>Email</td><td colspan="2">ilif88@gmail.com</td></tr>
<tr><td>내용 요약</td><td colspan="7">본 연구는 특정 야구 경기의 승패 결과를 실시간으로 예측하는 방식의 구현을 목적으로 한다. 이를 위해 야구 게임을 Markov chain을 이용하여 모델링 하고, 과거 시즌 게임 기록을 바탕으로 전이 행렬(transition probability matrix)을 생성하여 기대점수분포를 계산한다. 이때 경기 중 현재 상황을 계산 과정에 반영함으로써 아웃카운트나 점수, 또는 남은 이닝 수 등 경기 결과에 영향을 미칠 수 있는 어떠한 요소가 변하는 모든 순간마다 실시간으로 업데이트 하도록 한다. 이렇게 구해진 기대점수분포에 따라 게임 결과의 각 경우(홈 또는 원정팀의 승리, 또는 무승부)의 확률이 구해지며, 이를 승패결과 예측치로 나타내고자 한다. 또한 이렇게 실시간으로 변하는 승률의 추이를 그래프로 표현하여 경기 중의 상황 변화 및 분위기를 시각적으로 표현한 결과를 얻고자 한다.</td></tr>
<tr><td>해당 과목 구분</td><td colspan="7">여름학기 랩 인턴 연구 과제 (2012 여름학기, KAIST SDM 연구실)</td></tr>
<tr><td colspan="8">지도교수 추천서*
본인은 2012년 대한산업공학 추계학술대회의 대학생 프로젝트 경진대회에 상기 지원자의 작품을 추천함과 더불어, 상기 작품이 "학생 주도"의 작업임을 확인합니다.

소속: 한국과학기술원 지도교수 이름: 장영재
전화: 042-350-3130 Email: jang(at)kaist.ac.kr</td></tr>
</table>

*(참고) 교수 1인당 3편 이내의 작품 추천 가능함.

- 게임 모델링 및 전이 행렬 계산

먼저, 야구 경기를 Discrete Markov chain으로 보고, 모델링을 하는 것부터 시작한다. 경기 중의 각 상황을 아웃 카운트와 각 베이스들의 주자 유무 상태로 단순화한다. 즉, 아웃카운트가 0부터 2까지 총 3가지, 각 1루, 2루, 3루의 주자 유무가 0 또는 1로 2가지씩 경우의 수가 존재하며, 따라서 경기 중의 모든 상황은 24가지(3*2*2*2) 상황에 3아웃 이닝 종료 상황까지 합쳐 총 25가지 경우 중 하나에 매칭된다. 각 상황과 state 번호의 관계는 다음 표를 보고 확인할 수 있다.

state	번호	state	번호	state	번호	state	번호	state	번호
(0,0,0,0)	1	(0,1,0,1)	6	(1,0,1,0)	11	(1,1,1,1)	16	(2,1,1,0)	21
(0,1,0,0)	2	(0,0,1,1)	7	(1,0,0,1)	12	(2,0,0,0)	17	(2,1,0,1)	22
(0,0,1,0)	3	(0,1,1,1)	8	(1,1,1,0)	13	(2,1,0,0)	18	(2,0,1,1)	23
(0,0,0,1)	4	(1,0,0,0)	9	(1,1,0,1)	14	(2,0,1,0)	19	(2,1,1,1)	24
(0,1,1,0)	5	(1,1,0,0)	10	(1,0,1,1)	15	(2,0,0,1)	20	(3,0,0,0)	25

표 1. 각 state의 상태와 해당 번호

이제, 과거 시즌의 기록에서 각 선수 별로 상황 전환이 일어난 횟수를 각각 합산한다. 경기 중 각 타자는 공을 치기 전 가능한 25가지 중 하나의 상황에서 공 하나를 처리한 후의 가능한 25가지 상황 중 하나로 전이 시킨다. 이 기록은 25*25 크기의 행렬에 횟수를 누적하는 방식으로 시즌 전체기간 동안의 모든 경우를 기록하며, 행의 순서는 처음 상황, 열의 순서는 나중 상황을 나타내는 것으로 해석하여 각 선수마다 어떤 상황에서 어떤 결과를 만들어 내었는가를 통계적으로 측정할 수 있게 된다. 즉 기존의 scoring index 방식으로 안타율, 장타율 등을 구할 때에는 아웃카운트나 경기 상황 등이 고려되지 않았는데, 과거 기록을 바탕으로 계산한 이 전이 행렬을 사용할 경우, 좀 더 현실적이고 구체적인 상황 전이 확률을 예측할 수 있다. 여기에 사용되는 이전 경기 기록은 지금은 서비스를 제공하지 않는 Statiz.co.kr 에서 미리 자료를 다운 받아 DB에 저장해둔 것을 사용하였다.

$$\text{Transition matrix(P)} = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,25} \\ p_{2,1} & & & & p_{2,25} \\ p_{3,1} & & & & p_{3,25} \\ \vdots & & & \ddots & \vdots \\ p_{25,1} & p_{25,2} & p_{25,3} & \cdots & p_{25,25} \end{bmatrix}$$

$$p_{i,j} : i\ state\text{에서}\ j\ state\text{로 바뀔 확률}$$

이렇게 구해진 전이 행렬은 그날 경기의 타순에 따라 차례대로 연산하여 실제 상황을 좀 더 구체적으로 반영한 기대점수분포를 구할 수 있게 된다. 자세한 과정은 다음 장에서 더 소개된다.

- 기대점수분포 계산 원리

위에서 구한 전이 행렬은 특정 타자가 특정 상황에서 어떤 결과를 얻을 수 있는 지에 관한 확률 정보를 준다. 여기서, 총 625(25*25)가지의 상황 전이 중 어떤 전이는 플라이 아웃, 3진 등 무득점인 경우도 있고, 어떤 것은 득점인 것도 있다. 이런 다양한 상황 전이 별 점수 변화를 기록하기 위해 행렬을 별도로 하나 만들어, 전이 행렬을 처리하는 과정에서 기대 점수 분포를 계산한다.

점수를 나타내는 21개의 행(0점~20점)과 상황을 나타내는 25개의 열로 이뤄진 행렬 U가 있다고 하자. $U_{i,j}$는 i번째 이닝에서 j번째 타자가 들어설 때의 경기 상황 및 점수의 확률 분포를 저장한다. 즉, 경기 시작 시에는 점수는 0점, 상황은 1번이 유일한 경우이므로 $U_{1,1}$는 (1,1)에만 1, 나머지 성분은 모두 0이며, 경기가 진행되면서 전이 행렬을 곱함에 따라 상황은 1열에서 다른 열로 조금씩 나눠져 확률 분포를 이룬다.

이때 위에서 언급한 '득점을 하는 상황 전이'의 경우는 확률을 연산할 때 같은 행이 아닌 아래의 행으로 더함으로써 득점 상황을 반영시킨다. 이때 한 줄을 내리면 1점 득점, 두 줄을 내리면 2점 득점을 의미하는 등 각 득점 상황에 맞게 전이를 반영한다. 따라서 전이 행렬의 연산 과정은, 무득점부터 4점 득점 상황(예: 만루 홈런)까지 총 5가지로 전이 행렬을 분해(decomposition)한 후 각 득점 상황에 해당되는 행으로 확률 곱 연산의 결과값을 누적하는 방식이 된다.

$$U_{i,j+1}(row\ k) \ = \ U_{i,j}(row\ k)P0 \ + \ U_{i,j}(row\ k-1)P1 \ + \ U_{i,j}(row\ k-2)P2$$
$$+\ U_{i,j}(row\ k-3)P3 \ + \ U_{i,j}(row\ k-4)P4$$

PN : *N 점을 득점하는 상황 전이의 성분으로 분해된 행렬*

이런 전이 행렬 연산은 한 이닝이 종료될 때까지 계속되며, 이는 3아웃 이닝종료상황, 즉 25번째 열을 제외한 모든 성분의 합이 0으로 수렴한 경우다. 이를 $U_{i,\infty}$라 표기하자. 그러면 25번째 열에는 i이닝을 마친 후 각 점수를 득점할 확률의 분포가 나타난다. 이 분포는 다음 이닝의 U 행렬을 계산하는 과정으로 이어지며, 이 25번째 열을 $U_{i+1,1}$의 첫 번째 열로 옮겨와서 같은 방식으로 다음 이닝의 이닝종료상황($U_{i+1,\infty}$)까지 전이 행렬들을 연산한다.

$$U_{i,\infty} = \begin{bmatrix} 0 & 0 & \cdots & 0 & p_0 \\ 0 & 0 & \cdots & 0 & p_1 \\ 0 & 0 & \cdots & 0 & p_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & p_{20} \end{bmatrix}, U_{i+1,1} = \begin{bmatrix} p_0 & 0 & \cdots & 0 \\ p_1 & 0 & \cdots & 0 \\ p_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ p_{20} & 0 & \cdots & 0 \end{bmatrix} \rightarrow U_{i+1,\infty} = \begin{bmatrix} 0 & 0 & \cdots & 0 & p_0' \\ 0 & 0 & \cdots & 0 & p_1' \\ 0 & 0 & \cdots & 0 & p_2' \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & p_{20}' \end{bmatrix}$$

p_k : *k 점수를 낼 확률*

U 행렬이 20열까지만 있는 이유는, 경기 중 20점을 초과하여 득점하는 경우가 매우 드물고 팀 간 득점력 밸런스가 매우 크게 차이 나진 않는 프로야구의 특성 상 현실적으로 나오기 힘든 경

우이기 때문이다. 따라서 20점 득점까지의 경우들만 확률을 계산하여 승률을 구하여도 충분히 유의미한 결과를 얻을 수 있다고 본다.

위 과정에서 새 이닝의 U 행렬을 계산할 시에는 이전 이닝 마지막 선수 바로 다음 선수의 전이 행렬부터 연산함으로써 실제 경기 진행과정과 같은 과정으로 U 행렬을 계산하도록 한다. 예를 들어, 5이닝에서 7번타자를 마지막으로 이닝 종료 하였다면, 6이닝은 8번타자부터 시작함으로 $U_{6,1}$은 8번타자의 전이 행렬부터 차례대로 연산한다.

한가지 더 고려해야 할 이슈는 연장전인데, 야구 경기에서 9이닝 종료 후 동점일 경우 최장 12이닝까지 연장전이 이어지며, 매 이닝이 마치는 시점에서 점수 차이가 나면 더 많이 득점한 팀이 최종 승리팀이 된다. 따라서 연장전으로 가는 경우도 고려하여, 10, 11, 12이닝의 기대점수 분포 역시 미리 계산해두어야 하며, 이때 타석에 올라오는 선수의 순서 역시 이전 아홉 이닝의 U 행렬인 $U_{9,\infty}$를 계산할 때와 같이 바로 전 이닝의 마지막 선수에 이어 다음 선수가 올라오는 순서로 계속 진행된다.

연장되는 각 이닝에서의 기대점수분포 계산 방식은 앞서 9이닝까지의 U 행렬 계산과정과 같이 추가된 이닝만큼 U 행렬에 전이 행렬을 차례대로 더 연산해주는 것인데, 여기서 9이닝의 마지막 U 행렬($U_{9,\infty}$)이 아닌, 처음 상태의 U 행렬($U_{1,1}$)을 가지고 별도로 다시 계산한다. 즉, 0점, 1state 상황의 요소만 1이고 나머지 요소는 모두 0인 초기상태부터 새로 게임을 하는 것과 같은데, 이는 연장전에 들어섰다는 것은 앞서 9이닝을 동점으로 마쳤으며, 이전 경기를 몇 점으로 마쳤든 상관없이 오로지 연장전에서 상대방보다 몇 점을 더 많이 내느냐가 경기 승패를 결정짓기 때문에 마치 1이닝짜리 별도의 게임을 따로 하는 것과 같은 상황으로 보는 것이다. 만약 10이닝도 동점으로 끝나 11이닝으로 이어진다면 10이닝과 똑같이 새로운 한 이닝 짜리 게임을 새로 하는 것과 같다. 여기서 연장전 10, 11 그리고 12이닝은 같은 길이의 한 이닝이지만, 첫 타석에 올라서는 선수가 모두 다르기 때문에 각 이닝의 기대점수분포 역시 모두 다른 값을 가지므로 각각 따로 계산해서 저장해두어야 한다.

$$U_{10,\infty}=\begin{bmatrix}0&0&\cdots&0&p_0''\\0&0&\cdots&0&p_1''\\0&0&\cdots&0&p_2''\\\vdots&\vdots&\cdots&\vdots&\vdots\\0&0&\cdots&0&p_{20}''\end{bmatrix},\ U_{11,\infty}=\begin{bmatrix}0&0&\cdots&0&p_0'''\\0&0&\cdots&0&p_1'''\\0&0&\cdots&0&p_2'''\\\vdots&\vdots&\cdots&\vdots&\vdots\\0&0&\cdots&0&p_{20}'''\end{bmatrix},U_{12,\infty}=\begin{bmatrix}0&0&\cdots&0&p_0''''\\0&0&\cdots&0&p_1''''\\0&0&\cdots&0&p_2''''\\\vdots&\vdots&\cdots&\vdots&\vdots\\0&0&\cdots&0&p_{20}''''\end{bmatrix}$$

$$p_k : k\text{점수를 낼 확률}$$

- 승률 계산 원리

■ 비길 확률

팀의 최종 기대분포점수가 계산되면 승률을 구하기 위한 기초자료가 준비된 셈이다. 야구에서는 게임 종료 후 점수가 더 높은 팀이 승리한다. 앞서 언급하였듯이 9이닝 종료 시에 동점일 경우는 연장전이 1이닝씩 총 3회 진행되며, 매 이닝이 마치는 시점에서 점수 차이가 나면 더 많이 득점한 팀이 최종 승리팀이 된다.

먼저 비교적 간단한 비기는 경우부터 살펴 보자. A팀과 B팀이 경기를 할 때, 비기는 경우의 확률은 9이닝까지 양팀의 득점이 같을 확률과 10이닝에서 같은 득점을 할 확률(무득점을 포함하여), 11이닝과 12이닝에서도 각각 같은 득점을 할 확률을 모두 곱하면 된다. 두 팀이 무승부로 경기를 마치기 위해서는 9이닝 종료 후 동점, 10이닝에서 같은 득점, 11이닝에서 같은 득점, 그리고 12이닝에서도 동점으로 마치는 사건이 동시에 일어나야 함으로 각각의 개별 확률을 구하여 곱하는 것으로 비길 확률을 구할 수 있다.

그림 1 비기는 경우의 확률 계산 방식

일단 9이닝 종료 후의 각 팀이 같은 점수로 마칠 확률은 다음과 같다

$$\text{비길 확률(9이닝 종료시)} = \sum_{i=0}^{20}[S_A(i) * S_B(i)]$$

$$S_A(n) : A\,팀이\,n\,점으로\,9\,이닝\,종료할\,확률$$

$$U_{9,\infty} = \begin{bmatrix} 0 & 0 & \cdots & 0 & p_0 \\ 0 & 0 & \cdots & 0 & p_1 \\ 0 & 0 & \cdots & 0 & p_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & p_{20} \end{bmatrix} \rightarrow \mathrm{S} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{20} \end{bmatrix}$$

다음에, 10, 11, 12이닝 각각의 이닝에서 비길 확률을 앞서 따로 구해서 저장해둔 연장전 각 이닝의 기대점수분포를 이용하여 구한다.

$$U_{10,\infty}=\begin{bmatrix}0&\cdots&0&p_0{}'\\0&\cdots&0&p_1{}'\\0&\cdots&0&p_2{}'\\\vdots&\cdots&\vdots&\vdots\\0&\cdots&0&p_{20}{}'\end{bmatrix}\rightarrow \mathrm{S10}=\begin{bmatrix}p_0{}'\\p_1{}'\\p_2{}'\\\vdots\\p_{20}{}'\end{bmatrix},\ U_{11,\infty}=\begin{bmatrix}0&\cdots&0&p_0{}''\\0&\cdots&0&p_1{}''\\0&\cdots&0&p_2{}''\\\vdots&\cdots&\vdots&\vdots\\0&\cdots&0&p_{20}{}''\end{bmatrix}\rightarrow \mathrm{S11}=\begin{bmatrix}p_0{}''\\p_1{}''\\p_2{}''\\\vdots\\p_{20}{}''\end{bmatrix},U_{12,\infty}=\begin{bmatrix}0&\cdots&0&p_0{}'''\\0&\cdots&0&p_1{}'''\\0&\cdots&0&p_2{}'''\\\vdots&\cdots&\vdots&\vdots\\0&\cdots&0&p_{20}{}'''\end{bmatrix}\rightarrow \mathrm{S12}=\begin{bmatrix}p_0{}'''\\p_1{}'''\\p_2{}'''\\\vdots\\p_{20}{}'''\end{bmatrix}$$

$$\text{비길 확률(10이닝)}=\textstyle\sum_{i=0}^{20}[S10_A(i)*S10_B(i)],\ \text{비길 확률}\left(\text{11,12이닝}\right)\text{도 같은 방식}$$

따라서 게임을 무승부로 마칠 확률은 다음과 같다.

$$\text{비길 확률}=\sum_{i=0}^{20}[S_A(i)*S_B(i)]*\sum_{i=0}^{20}[S10_A(i)*S10_B(i)]*\sum_{i=0}^{20}[S11_A(i)*S11_B(i)]*\sum_{i=0}^{20}[S12_A(i)*S12_B(i)]$$

■ A팀 승률

A팀이 이기는 모든 경우는, A팀이 9이닝만에 이기는 경우와 연장전 중에 이기는 경우로 이뤄진다. 9이닝만에 이기는 경우의 확률은, 9이닝 종료 후 A팀이 1점 이상 득점하는 경우마다 B팀이 A팀보다 적은 득점을 하는 경우의 확률을 모두 합산해서 구할 수 있다. 예를 들어 A팀이 3점일 확률이 0.3, B팀이 5점일 확률이 0.2라면, 게임 종료 때 동시에 A팀이 3점, B팀이 5점일 확률은 0.06이며, 그 외 다른 A가 이기는 경우(ex. A팀이 1점이고 B팀이 0점, A팀이 2점이고 B팀이 0점 혹은 1점, 등등)의 확률을 모두 따로 구하여 합산한다.

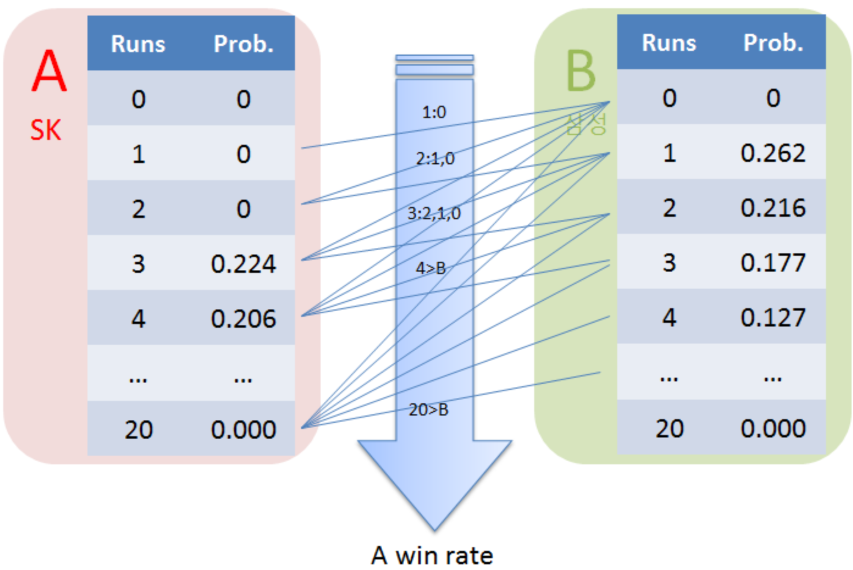

그림 2 A팀이 이기는 경우의 확률 계산 방식

$$\text{A팀 승률(9이닝 종료시)}=\sum_{i=1}^{20}\left[S_A(i)*\left(\sum_{j=0}^{i-1}S_B(j)\right)\right]$$

다음으로 연장전 중에 이기는 경우는, 9이닝까지는 서로 두 팀이 동점으로 마친 후, 10~12이닝에서 A팀이 B팀보다 더 많은 득점을 하는 경우다. 두 팀이 동점일 확률은 각 팀이 같은 점수를 낼 확률을 서로 곱하고 합산하는 것으로 앞에 비기는 경우에서 계산해둔 식을 이용한다. 그 다음은 (1) 10이닝에서 A팀이 이기는 경우, (2) 11이닝에서 A팀이 이기는 경우, (3) 12이닝에서 A팀이 이기는 경우를 각각 계산해서 구한다. 10이닝부터 이기는 경우는 각 이닝에서 A팀이 더 많은 득점을 하는 모든 경우의 확률을 합산해서 구한다. 여기서는 9이닝 전체의 기대점수분포(S)가 아닌 연장전 각 이닝의 기대점수분포를 가지고 계산해야 함으로, S10, S11, S12를 이용한다.

먼저, (1) 10이닝에서 A팀이 이길 확률은 다음과 같다.

$$\text{A팀 승률}\left(\text{10이닝에서}\right) = \sum_{i=0}^{20}[S_A(i) * S_B(i)] * \sum_{i=1}^{20}\left[S10_A(i) * \left(\sum_{j=0}^{i-1} S10_B(j)\right)\right]$$

(2) 10이닝에서 같은 득점 후 11이닝에서 A팀이 이길 확률은 다음과 같다.

$$\text{A팀 승률}\left(\text{11이닝에서}\right) = \sum_{i=0}^{20}[S_A(i) * S_B(i)] * \sum_{i=0}^{20}[S10_A(i) * S10_B(i)] * \sum_{i=1}^{20}\left[S11_A(i) * \left(\sum_{j=0}^{i-1} S11_B(j)\right)\right]$$

(3) 10, 11이닝에서 같은 득점 후 12이닝에서 A팀이 이길 확률은 다음과 같다.

$$\text{A팀 승률}\left(\text{12이닝에서}\right) = \sum_{i=0}^{20}[S_A(i) * S_B(i)] * \sum_{i=0}^{20}[S10_A(i) * S10_B(i)] * \sum_{i=0}^{20}[S11_A(i) * S11_B(i)] * \sum_{i=1}^{20}\left[S12_A(i) * \left(\sum_{j=0}^{i-1} S12_B(j)\right)\right]$$

따라서 A팀의 전체 승률은 다음 식으로 정리할 수 있다

$$\begin{aligned}\text{A팀 승률}\left(\text{전체}\right) = &\sum_{i=1}^{20}\left[S_A(i) * \left(\sum_{j=0}^{i-1} S_B(j)\right)\right] + \sum_{i=0}^{20}[S_A(i) * S_B(i)] \\ &* \left(\sum_{i=1}^{20}\left[S10_A(i) * \left(\sum_{j=0}^{i-1} S10_B(j)\right)\right] + \sum_{i=0}^{20}[S10_A(i) * S10_B(i)]\right. \\ &* \left.\left(\sum_{i=1}^{20}\left[S11_A(i) * \left(\sum_{j=0}^{i-1} S11_B(j)\right)\right] + \sum_{i=0}^{20}[S11_A(i) * S11_B(i)] * \sum_{i=1}^{20}\left[S12_A(i) * \left(\sum_{j=0}^{i-1} S12_B(j)\right)\right]\right)\right)\end{aligned}$$

■ B팀 승률

B팀의 승률은 A팀의 승률을 계산한 방식 그대로 적용하여 구할 수 있다. 즉 바로 위 식에서 A팀과 B팀의 기대점수분포의 자리만 서로 바꾸어 계산한다. 혹은, 1에서 A팀의 승률과 비길 확률을 빼는 방법으로 보다 손쉽고 빠르게 구할 수도 있다.

- 각 상황에서의 실시간 기대점수분포 계산

본 연구의 또 다른 중요한 특징 중 하나인 실시간 승률 예측을 위해서는, 어느 한 팀의 온전한 한 경기에서의 기대점수분포뿐만 아니라, 경기 중 어느 순간에서라도 기대점수분포를 계산할 수 있어야 한다. 따라서, 앞에서 설명한 기대점수분포의 계산과정의 시작을 $U_{1,1}$ 행렬이 아닌 $U_{i,j}$(i이닝, j번째 선수)에서 시작할 수 있도록, 즉 경기 중 어떤 상황에서든 남은 경기를 지속할 때의 기대점수분포를 계산할 수 있도록 이닝, 초/말, 현재 상태(state), 현재 점수, 그리고 타석에 올라설 선수 차례를 입력변수로 설정한다.

먼저 이닝과 초/말에 관한 입력변수 값의 반영 원리를 살펴보자. 지금이 몇 이닝이며 초인가 말인가는 그 팀의 남은 실제 공격 이닝 수를 결정지으며, 그 수만큼 U 행렬에 전이 행렬을 연산한다. 즉 어웨이팀의 3회초 상황에서의 기대점수분포를 계산한다면, 3회초를 포함하여 9회초까지 총 일곱 이닝의 공격기회가 있으므로, $U_{i,\infty}$ 행렬을 구하는 함수를 총 7이닝동안 반복한다. 만약 3회말 상황이라면 어웨이팀은 4회초부터 공격할 수 있으므로 총 여섯 이닝의 공격기회가 있다. 따라서 $U_{i,\infty}$행렬을 구하는 함수를 총 6이닝동안 반복한다. 물론 9회말 종료 후 동점일 경우를 대비하여 연장전 세 이닝 역시 매번 따로 구하여 저장해야 하며, 지금 이미 연장전에 들어선 중이라면 남은 연장전에 대해서만 기대점수분포를 계산하도록 한다.

여기서 몇 번째 선수의 전이 행렬부터 U 행렬에 연산할 것인가를 결정지을 때 바로 현재 타석에 올라설 선수 변수 값을 적용한다. 만약 경기가 막 시작한 1회초 어웨이팀 공격 또는 1회말 홈팀 공격 상황이라면 1번타자부터 U 행렬에 연산을 하겠지만, 경기 중반에 4번 타자가 타석에 올라서는 상황이라면, 1번 타자부터가 아닌 4번 타자의 전이 행렬부터 연산을 하는 방식으로 입력변수 값을 사용하는 것이다.

다음으로 현재 상태(state)와 현재 점수에 관한 입력변수 값은 U 행렬의 초기 값을 설정하는 과정에서 반영한다. 즉 U 행렬에 전이 행렬을 연산하기 전, 가장 처음 상태의 U 행렬을 정의하는 과정에서 현재 state 값과 점수가 입력 변수로 주어지는 경우, 그 state와 점수에 해당하는 성분을 1로 하고 나머지 성분을 0으로 설정하면 된다. 예를 들어 앞서 9이닝의 기대점수분포 계산과정에서 1회초 게임 시작 상황의 경우($U_{1,1}$), (1,1) 성분만 1이고 나머지 성분은 모두 0으로 설정하겠지만 어웨이팀 3회초에서 현재 점수 2점, 1아웃 주자 1, 2루인 경우의 실시간 기대점수분포를 계산하기 위해서는$U_{3,1}$ 의 (3,13) 성분만 1, 나머지 성분은 모두 0으로 설정하면 된다. 그 $U_{3,1}$ 행렬부터 전이 행렬들을 타순 순서대로 연산하면서 $U_{9,\infty}$과 $U_{10,\infty}$, $U_{11,\infty}$, $U_{12,\infty}$을 구해 각 기대점수분포를 계산한다.

앞서 언급한 초/말 변수 값과 state 변수 값은 서로 연관 관계가 있는데, 코딩에서 이를 활용하여 입력변수 값의 반영 원리를 구현할 수 있다. 예를 들어 3회 말의 어웨이팀의 기대점수분포를

계산하고자 한다면, 3회초는 이미 지나갔으므로 4회초부터 U 행렬을 연산해야 하는데, 이를 이닝 변수 값 3과 state 변수 값 25로 표현할 수 있는 것이다. 또 다른 방법으로 이닝 변수 값 4와 state 변수 값 1을 넣는 것도 있으나, 한 순간에의 두 팀 승률을 동시에 구해서 함께 활용해야 한다는 특징 때문에 두 팀의 기대점수분포를 최대한 같은 입력변수로 구해내는 것이 좋다. 따라서 같은 이닝 변수 값을 주고, 이닝 초의 경우 홈팀의 state 변수 값을 1로, 이닝 말의 경우 어웨이 팀의 state 변수 값을 25로 주는 방식을 코딩에 적용한다.

이렇게 실시간으로 매 순간마다 그 상황을 정확히 반영한 기대점수분포가 계산되면, 그 결과를 가지고 승률 예측을 새롭게 갱신할 수 있다. 즉 경기가 계속되면서 경기의 흐름에 따라, 주자의 출루 상황이나 득점 상황 등 승패를 결정짓는 데에 영향을 미치는 요소가 변함에 따라 승률이 어떻게 변하는지를 직접 확인할 수 있게 되는 것이다.

```
player_num = 1; Un(1,1) = 1;
for inning = 1:1:9
    while (Un(1:21,1:24)>0.0000001)
        Un = Un*P(player_num);
        player_num = rem(player_num, 9) + 1;
    end
end
U_9inf = Un;
```

그림 3 U_(9,∞)를 구하는 Matlab pseu•do code – 입력 변수가 따로 없을 때

```
player_num = i_player_num; Un(i_score, i_state) = 1;
for inning = i_inning:1:9
    while (Un(1:21,1:24)>0.0000001)
        Un = Un*P(player_num);
        player_num = rem(player_num, 9) + 1;
    end
end
U_9inf = Un;
```

그림 4 U_(9,∞)를 구하는 Matlab pseu•do code – 입력 변수(i_variable)가 있을 때

- 전체 게임 승률 로그 그래프

위에서 구현한 실시간 승률 계산 방식을 이용하여 경기 매 순간 승률을 계산하여 그 결과를 로그로 남긴다면, 그날 경기가 어떻게 흘러갔는지 한눈에 볼 수 있는 그래프를 만들 수 있을 것이다. 처음 경기가 시작하여 1회초 1 state 상황에서 어웨이팀의 1번타자가 타석에 들어선 경우에 계산한 승률부터, 9회말 홈팀(혹은 9회초 어웨이팀, 혹은 연장전)의 공격을 끝으로 경기가 종료되는 순간의 승률(이때는 어느 한 팀의 승률 혹은 비길 확률이 1로 수렴한 상태)까지 모두 같은 좌표평면 위에 그래프로 그린다면, 언제 경기가 극적으로 뒤집혔는지, 혹은 언제 경기가 무난하게 조용히 흘러갔는지 등을 시각적으로 손쉽게 살펴볼 수 있는 자료를 얻을 수 있는 셈이다.

앞서 설명한 바와 같이 본 연구에서는 2007년부터 2010년까지의 프로야구 경기 기록을 바탕으로 전이 행렬을 만든 후, 2011년 경기 내용에서의 실시간 예측 모의 시험을 실시하여 각 승률의 로그를 기록하여 그래프로 그려보았다.

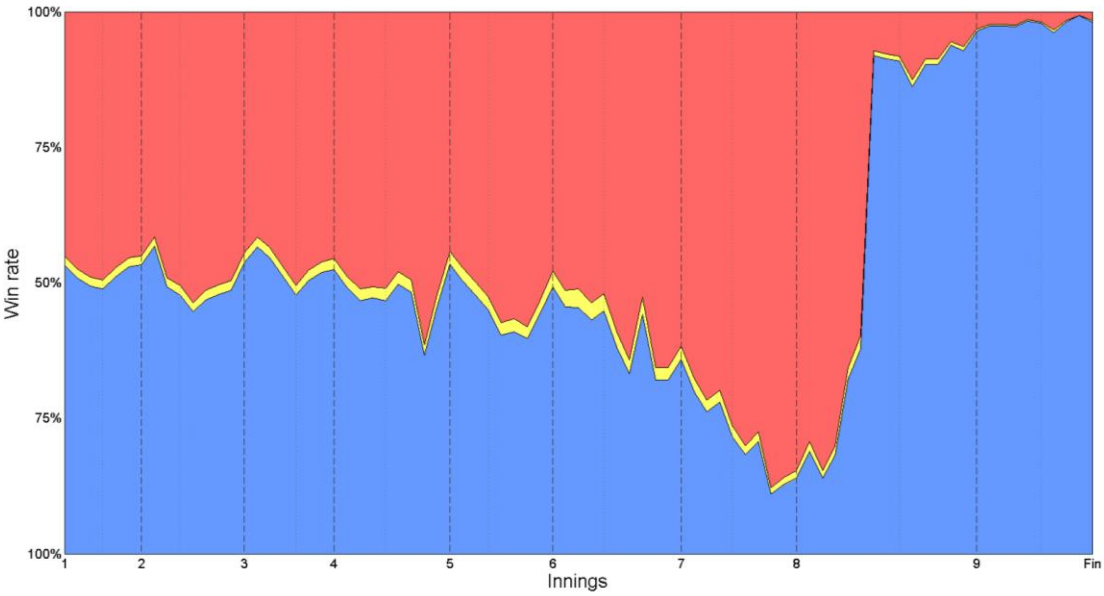

그림 5 2011-04-02 광주 구장, 삼성-KIA

위 그래프는 2011년 4월 2일에 광주 구장에서 열린 삼성과 KIA의 경기 내용을 가지고 실시간 승률 분석을 하여 그래프로 나타낸 것이다. 기본적인 그래프 요소를 설명하자면, 빨간색은 홈팀(여기선 KIA)의 승률, 파란색은 어웨이팀(여기선 삼성)의 승률, 노란색은 비길 확률을 나타낸다. 홈팀의 승률은 y축의 가장 위를 시작점으로 아래 노란색이 나타나는 지점까지의 세로 길이를 값으로 읽는다. 각 이닝의 시작 시점은 검은 점선으로 표시하였으며, 각 이닝의 초에서 말로 전환되는 시점은 회색 점선으로 표시하였다.

- 그래프 검증

위 알고리즘 및 자료를 가지고 만든 그래프가 이론적으로 적합한 지를 검증하기 위해, 그래프 상의 몇 가지 독특한 변화 양상을 선별하여, 실제 벌어진 사건과 매칭하고 현실적으로 타당한가를 검토해 보았다. 본 연구에서 설정한 모델링에 부합하는 그래프 모양의 특징들로는 다음 세가지를 꼽아 보았다.

1) 득점이 없을 경우, 시간이 흐를수록 승률은 점차 감소

팀의 승률은 기대점수분포를 바탕으로 내 점수가 상대방 점수보다 높을 확률을 의미한다. 여기서 내 점수를 높일 수 있는 방법은 이닝 내에서 득점을 하는 경우밖에 없고, 이닝 중에 점수가 감소하는 일은 발생하지 않는다. 따라서 각 이닝의 공격기회는 모두 점수를 높일 수 있는 기회인 셈이다. 즉 득점 없이 한 이닝을 종료했을 경우, 무득점으로 기회를 흘려 보낸 것으로, 그 팀의 남은 공격 기회는 줄어든 반면, 상대방의 공격 기회는 변화 없으므로 상대적으로 그 팀의 기대점수분포는 줄어들 것이다. 한가지 예외가 발생할 수 있는 경우는, 다음 이닝의 타석에 올라올 번호의 선수들이 이전 이닝의 선수들보다 훨씬 훌륭한 전이 행렬을 가지고 있다면 확률적으로 기대점수분포는 좀 더 높은 값을 가질 수도 있다. 하지만 대체로는 공격 기회를 잃은 만큼 기대점수분포 및 승률의 하락이 이어질 것이다. 이 경우에 해당하는 그래프 하나를 보자.

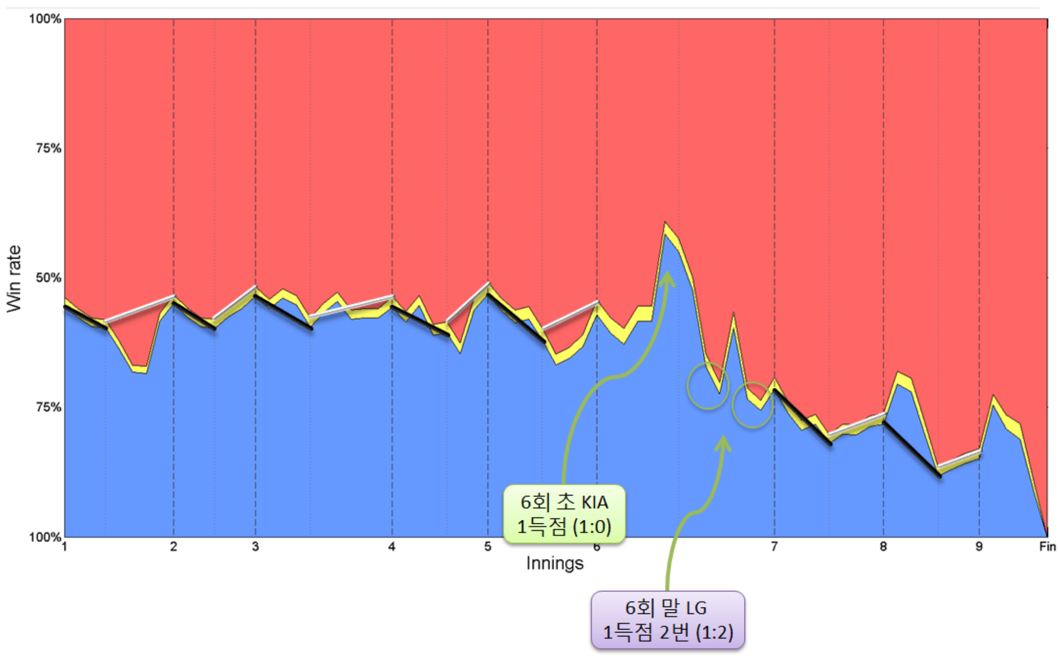

그림 6 2011-04-22 잠실 구장, KIA-LG

위 그래프에서, 두 팀 모두 무득점으로 경기를 마친 경우 승률이 떨어진 것을 확인할 수 있다. 검은 막대는 어웨이팀의 승률 하락을, 흰 막대는 홈팀의 승률 하락을 가시적으로 더 잘 표시해주기 위해 첨가되었다.

2) 득점 뿐만 아니라 좋은 상황을 만들었을 때도 승률은 증가

득점을 한 경우에는 분명 경기 종료시의 기대점수분포에 확실한 변화를 가져온다. 야구 경기에서 점수가 낮아지는 사건은 발생하지 않으므로, 기대점수분포에서 현재 점수 이하의 점수일 확률은 모두 0으로, 전체적으로 기대점수분포가 상향 이동하는 효과가 나타난다. 물론 승률 역시 자연스럽게 올라간다.

본 연구의 모델링에서는 득점 상황 외에도 좋은 상황(state)를 만들었을 때도 비슷하거나 혹은 심지어 더 큰 수준의 승률 변화를 야기할 수 있다. 승률은 기대점수분포를 바탕으로 '확률적으로' 계산되므로, 경기 내에 발생하여 기대점수분포를 상승시키는 모든 사건(action)은 승률 역시 상승시킨다. 이 사건에 득점 사건(예: 홈런, 주자가 득점권에 있을 때 안타) 뿐만 아니라 득점 확률이 높은 state로 전환시키는 것도 포함되는 것이다. 예를 들어 무사 2, 3루 상황처럼 매우 좋은 상황에 처할 경우, 그 팀의 기대점수분포는 현재 상황(state)를 입력변수로써 반영하면서 그 전보다 많이 상승될 수 있다. 이 경우에 해당하는 그래프 하나를 보자.

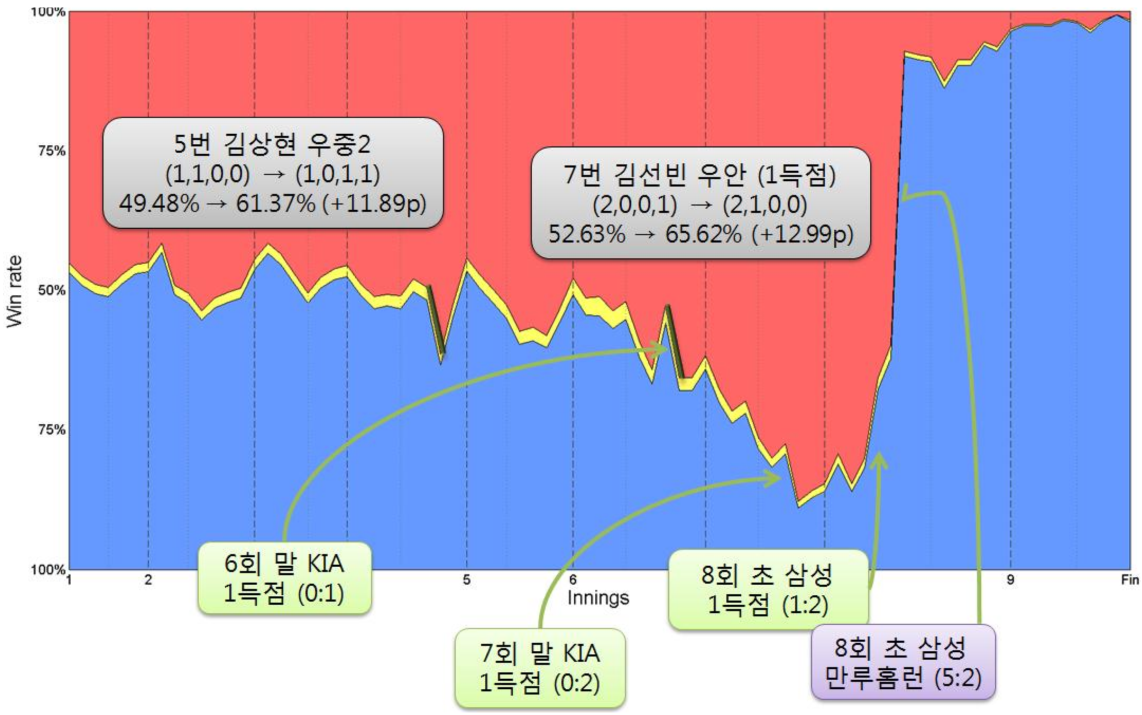

그림 7 2011-04-02 광주 구장, 삼성-KIA

위 그래프는 각각 다른 시점에서 발생한 득점 플레이와 좋은 상황을 만든 플레이의 승률에 대한 영향을 나타내고 있다. 같은 팀의 다른 플레이를 비교한 것으로 비교적 비슷한 시간대의 플레이를 비교함으로써 플레이 자체가 승률에 가장 큰 영향을 미치는 주요한 요소임을 알 수 있다. 좋은 state를 만든 후에 득점에는 실패하면 다시 승률이 떨어지는 것도 확인할 수 있다.

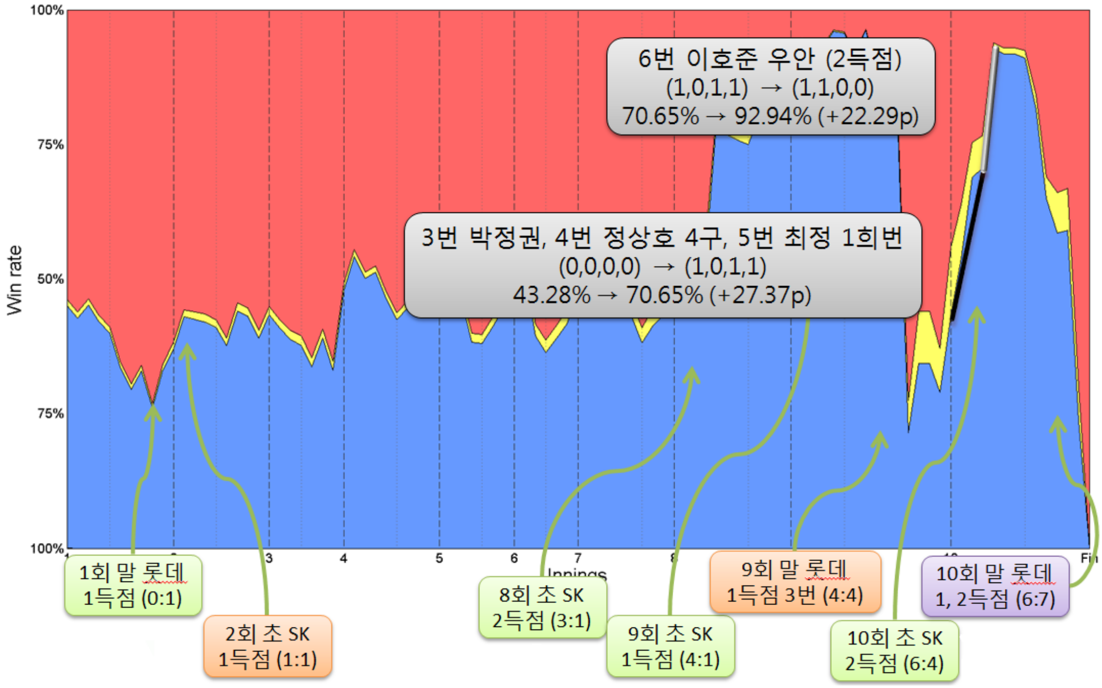

그림 8 2011-04-23 사직 구장, SK-롯데

위 그래프는 같은 시간대에서 좋은 상황을 만들고 득점에 성공한 경우다. 여기서도 좋은 state를 만들었을 때의 승률 변화 크기가 득점을 만들어 냈을 때의 승률 변화 크기와 큰 차이가 없으며 오히려 대체로 득점의 영향이 적은 것도 확인할 수 있다. 점수의 변화뿐만 아니라 득점이 유력한 state로의 변화 역시 승률에 큰 영향이 미친다는 가정이 잘 반영되었음을 알 수 있다.

3) 동점으로 종료시점까지 게임이 계속될 경우 무승부 확률은 크게 증가

마지막으로 세 번째 특징은 무승부로 게임을 마칠 확률에 관한 것이다. 두 팀이 동점으로 막상막하의 경기를 이어갈 경우, 경기가 종반부로 갈수록 무승부로 끝날 확률이 증가한다는 것이다. 승률은 기본적으로 우리 팀의 점수가 상대방 점수보다 높을 모든 확률이다. 우리 팀이 높은 점수를 얻을 확률이 적어지거나 상대방이 우리 팀보다 낮은 점수를 받을 확률이 적어진다면 우리 팀 승률은 자연히 낮아지게 된다. 이런 상황을 상대팀도 똑같이 겪는다면, 두 팀 모두 승률이 하락하며 같은 점수로 무승부가 될 확률만 증가할 것이다.

1번 특징을 바탕으로 구체적으로 생각해보면, 두 팀이 득점 없이 동점을 지속할 때 두 팀 모두 기대점수분포가 하락할 것이며, 이는 두 팀 모두에게 승률이 떨어지는 효과를 안겨줄 것이다. 만약 두 팀 모두 득점을 하면서 동점을 이어간다면 두 팀 모두 현재 점수보다 낮은 점수를 받을 확률은 0이므로 승률 역시 작아진다. 예를 들어 현재 4:4로 동점이라면 두 팀에게 있어 4점은 0점인 것과 같고 4점부터 얼마나 더 추가 득점을 하느냐가 관건이며, 이는 경기 시작 시의 0점인 상황과 거의 흡사한 것이다.

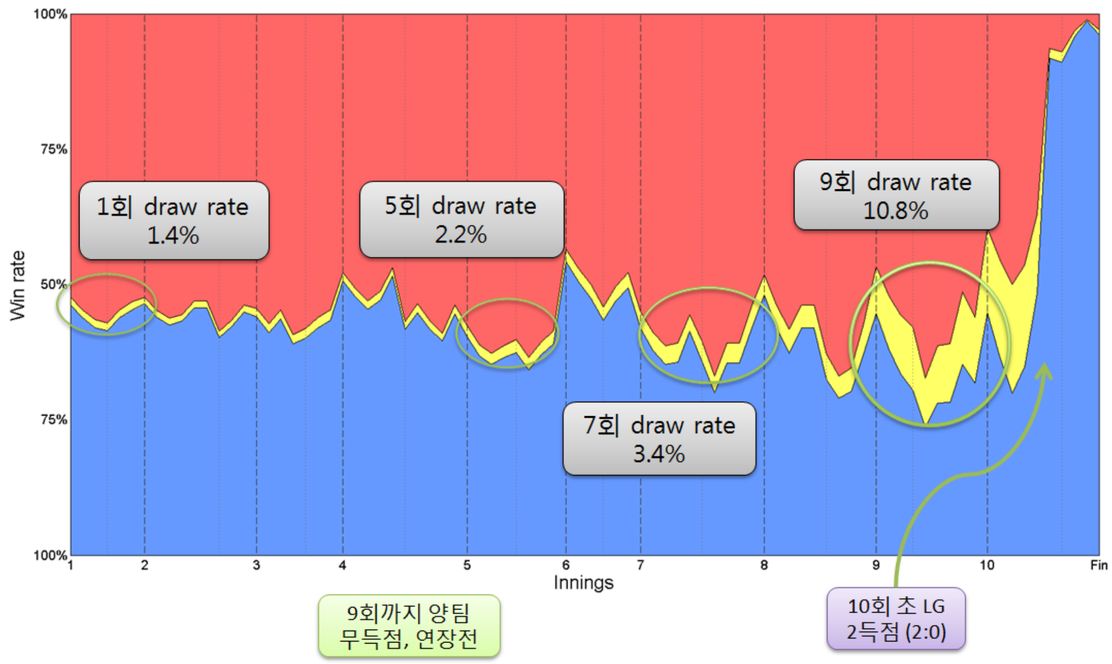

그림 9 2011-05-03 잠실 구장, LG-두산

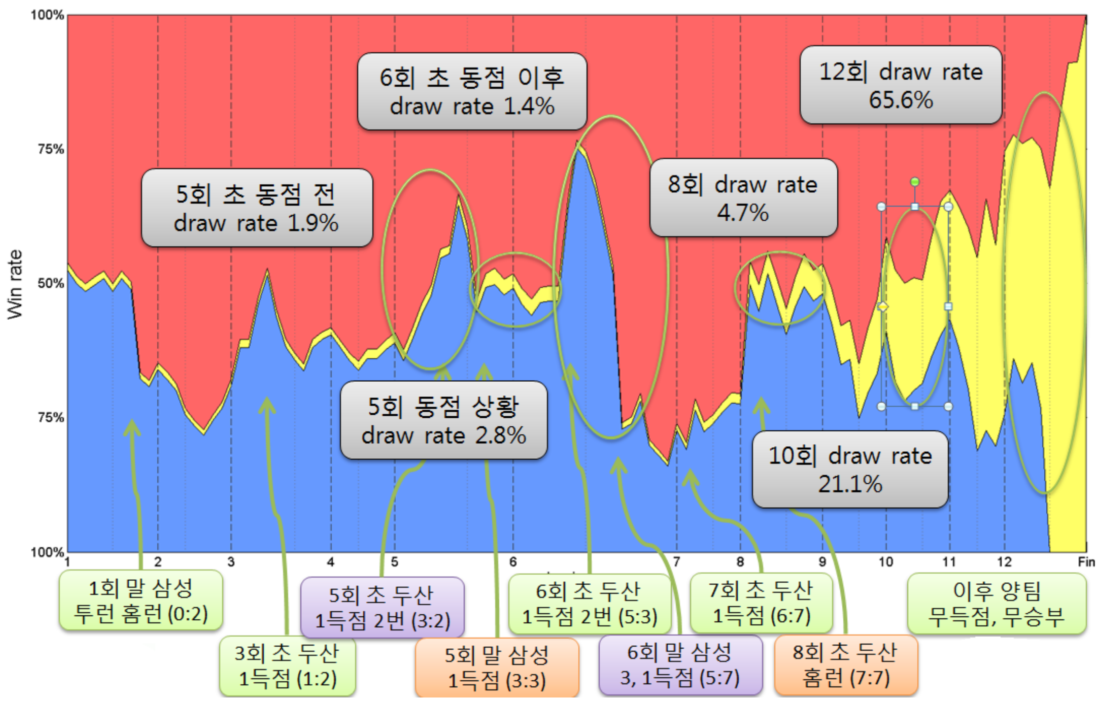

그림 10 2011-05-21 대구 구장, 두산-삼성

위 두 그래프는 동점이 지속되는 상황에서 무승부 확률이 증가하는 것을 보여준다. 5월 3일 경기에서는 10회 초 LG 득점 직전까지 꾸준히 비길 확률이 증가하는 것을 확인할 수 있으며, 5월 21일 경기에서도 8회 이후 동점이 계속되는 상황에서 무승부 확률이 크게 증가하는 것을 볼 수 있다. 추가로 5회말~6회초 동점 상황에서 무승부 확률이 그 전과 후보다 높은 것도 알 수 있다.

- 기대효과 및 결론

본 연구는 실제 경기 기록을 바탕으로 지금 진행되는 경기의 결과를 좀 더 현실적이고 논리적으로 예측한 결과를 제공한다는 점에서 기존에 존재하는 국내외 경기 결과 예측 모델보다 뛰어난 우수성을 보여준다. 실제로 미국 프로야구 메이저리그의 경기들에서 경기 중 승률 예측 서비스를 제공하는 FANGRAPHS(www.fangraphs.com) 의 서비스도 실제 팀의 속성이 전혀 고려하지 않고, 점수, 이닝, 아웃수, 주자 상황만을 입력변수로 승률을 예측하므로, 어떤 두 팀이 붙더라도 경기 시작 시에 50:50으로 시작하는 조금 비현실적인 면이 있다. 이에 대조되는 본 연구의 결과는 작년까지의 시즌 결과를 바탕으로 강팀이 약팀을 만났을 경우 승률이 50보다 높은 값에서 시작하는 것만 보아도 보다 현실적인 승률 예측임을 알 수 있다.

본 연구의 결과는 응용분야가 매우 다양할 것으로 예상된다. 현재 본 연구실에서 추가연구를 진행하고 있는 부분은 마케팅 분야와의 학제적 연구로써, 경기 내용의 다이나믹(Dynamics)과 관중 및 중계방송 시청자의 호응도 사이의 관계를 정량적(Quantitatively)으로 입증하여 실시간 마케팅 산업의 주요 도구 및 기준으로 삼을 수 있는 방법을 고안하는 것이다.

그 외에도 경기 내의 가장 결정적인 순간(Critical play)을 선정하는 데에 본 연구 결과의 승률 변화 추이를 활용하여, 경기의 Highlight 및 MVP 등을 선정할 때에도 좋은 논리적 근거자료가 될 수 있을 것으로 기대한다. 승률을 가장 크게 뒤집은 선수는 분명 그 경기의 승리에 핵심적인 역할을 했다고 볼 수 있기 때문이다. 이것은 흥미롭게도, 단순히 점수를 많이 낸 타자 뿐만 아니라, 좋은 상황을 만든 타자, 위험한 상황에서 멋진 수비로 상대팀 득점을 저지한 수비수의 플레이 또한 수치적으로 증명해 보일 수 있다는 점에서 '좋은 야구 선수'의 다양화에 기여할 수 있다.

또한 '스타 선수'의 경제적 파급효과 또한 정량적으로 계산하여 볼 수 있는 좋은 방법이 되는데, 예를 들어 박찬호 선수가 등판하였을 때의 관중 수 및 중계방송 시청자 수를, 비슷한 경기 내용을 보인 다른 경기의 수와 비교하여 실질적인 경제적 영향력을 수치화 할 수 있을 것이다.

- Reference

[1] A Markov Chain Approach To Baseball, Bruce Bukiet and Elliotte Rusty Harold, 1994
[2] A Robust Heuristic for Batting Order Optimization Under Uncertainty, 2003
[3] DEA OERA를 이용한 프로야구 선수들에 대한 성과 측정, 이덕주, 양원모, 2004
[4] 스탯티즈, statiz.co.kr
[5] 한국야구위원회(KBO), www.koreabaseball.com
[6] Database Processing, David M. Kroenke
[7] MATLAB: A Practical Introduction to Programming and Problem Solving, Stormy Attaway