Applied Mathmamatics 221

Project Milestone 2: Status Update
David Freed and Sam Green
March 30, 2016

Overview of Main Progress

We begin by addressing the questions described in the milestone description and the return to more detailed discussion after the overview.

As it stands, we have not revised the goal of our project, though we have introduced an intermediate model building goal. The over all goal of our project is still to assess the relative strength of using time series of tweets to build win probability models relative to the the standard logistic regression based predictive models that currently in standard use. We have selected a Hidden Markov based model as the end goal. As a intermediate goal, however, we intend to assess the basic predictve strength of the data by using the tweets to predict the leader of the game at each time period of the game.

We have collected tweets for 35 games, with a total of more than 500k so far. This is more tweets per game than the collected for previous studies. We intend to collect data on the remaining NCAA games next week.

We address our previous stated goals below in more detail, but broadly, we have accomplished the majority of data collection and have established a benchmark probability model and have a good direction in which to head towards our final deliverable.

Status Update from Milestone 1

We have reproduced the next steps that we made in the initial report below, with related commentary for each as to how much progress we have made on each. In sum, we have done almost all of the next steps that we proposed we would do earlier.

Twitter Data Next Steps

Get Twitter API keys

This wasn't too hard.

Identify the best way to get Twitter Data

Did this.

Read the literature on sensitivity analyses for Tweets

This step has not yet been accomplished. It is a clear next step for the analysis that we will do in the immediate future, however.

Get trial data

Mostly done. We've successfully collected tweets from 35 games from the NCAA March Madness tournament, with a totally of about 500,000 tweets collected. The smallest pool for one game that we've collected is 5,000, and the smallest pool we've collected is 50,000. This should allow the dataset to be significant: previous work used to analyze 768 games in the NFL was based on an average of 2,619 tweets per game. Our dataset has on average 16,747 tweets per game so far.

The data that we have collected has been cleaned to include simply the tweet text and a timestamp, though other axes of information are available. We have so far classified all of the tweets that we have collected by game and created infrastructure to build bucketed time series of the tweets.

Initial analysis indicates that the data we have collected should be adequate for the analysis we wish to perform (or at least isn't obviously inadequate). Tweets are at reasonable volumes through the collection period, with notable spikes in volume likely corresponding to game events. Pairing tweet timestamps to game timestamps remains as a major next step (see next subsection), and see the linecharts below for distributions of tweets for each game we have collected. There's a broad variety in the number of tweets collected, since many games, especially in the early rounds of tournament, took place at the same time and between lower visibility teams. Even the smallest volumes have on the order of 100 tweets per minute.

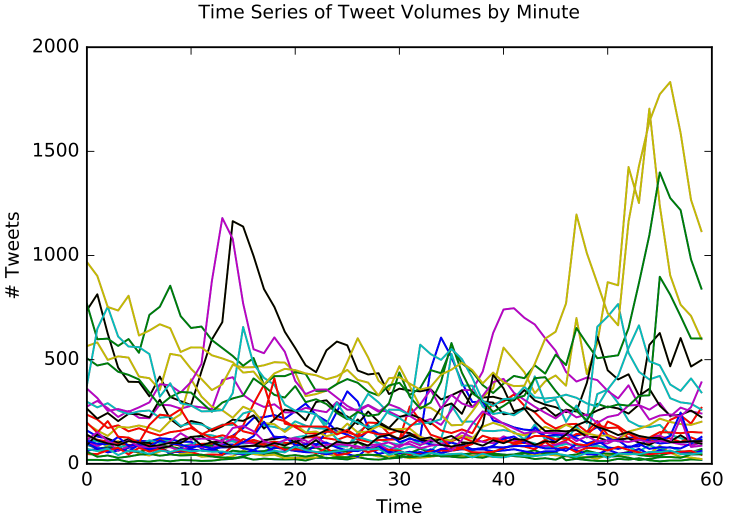
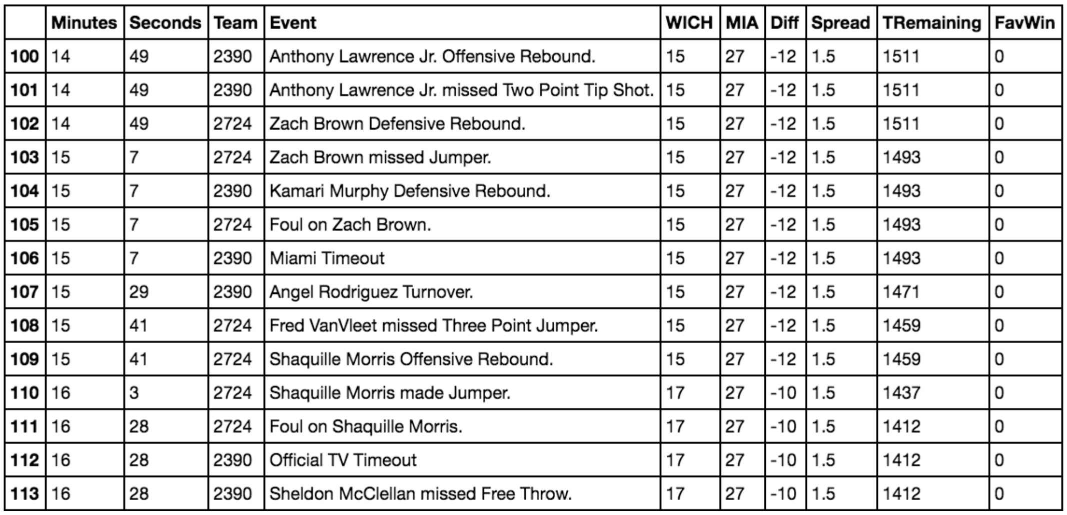


Figure 1: Example data taken from the Miami-Wichita State second round NCAA Tournament game

Box Score Data Next Steps

Create a scraper to get all of the box score data

The first thing that we did was to write a program that will scrape the box score data for any game. We took all data from `ESPN.com`, using the play-by-play data provided in the team pages (examples here and here). Figure 1 below indicates what the data look like for every game.



	Minutes	Seconds	Team	Event	WICH	MIA	Diff	Spread	TRemaining	FavWin
100	14	49	2390	Anthony Lawrence Jr. Offensive Rebound.	15	27	-12	1.5	1511	0
101	14	49	2390	Anthony Lawrence Jr. missed Two Point Tip Shot.	15	27	-12	1.5	1511	0
102	14	49	2724	Zach Brown Defensive Rebound.	15	27	-12	1.5	1511	0
103	15	7	2724	Zach Brown missed Jumper.	15	27	-12	1.5	1493	0
104	15	7	2390	Kamari Murphy Defensive Rebound.	15	27	-12	1.5	1493	0
105	15	7	2724	Foul on Zach Brown.	15	27	-12	1.5	1493	0
106	15	7	2390	Miami Timeout	15	27	-12	1.5	1493	0
107	15	29	2390	Angel Rodriguez Turnover.	15	27	-12	1.5	1471	0
108	15	41	2724	Fred VanVleet missed Three Point Jumper.	15	27	-12	1.5	1459	0
109	15	41	2724	Shaquille Morris Offensive Rebound.	15	27	-12	1.5	1459	0
110	16	3	2724	Shaquille Morris made Jumper.	17	27	-10	1.5	1437	0
111	16	28	2724	Foul on Shaquille Morris.	17	27	-10	1.5	1412	0
112	16	28	2390	Official TV Timeout	17	27	-10	1.5	1412	0
113	16	28	2390	Sheldon McClellan missed Free Throw.	17	27	-10	1.5	1412	0

Figure 2: Example data taken from the Miami-Wichita State second round NCAA Tournament game

As we can say, not only do we know the time that has passed in every game, we also know the full event and the scores for each team at that point. We also have incorporated the point spread[1] from that game, the difference in the score after every play, and the eventual outcome (an indicator of whether the favorite won or not).

While most of this data is not used until the following section, we did create functionality in our code[2] to generate game trend graphs for each game. An example trend graph for the Texas A&M-Northern Iowa double-overtime game is shown below:

[1] For those readers unfamiliar with this concept, it refers to the consensus betting line put out from Vegas before the game. A team that is given a -2.5 point handicap is considered the favorite; any bettor who puts money on them will win if they win by three or more points. If the favorite loses, or wins by one or two points, he will lose his money. In the case that the team is favored by an integer (a line of -2 points, for example), if it wins by exactly that number, the house will return to the bettor the exact amount of money that he bet. This is referred to as a "push" in standard gambling parlance.

[2] The above and below sections are all contained in the file `Box Score Scraper.ipynb`.

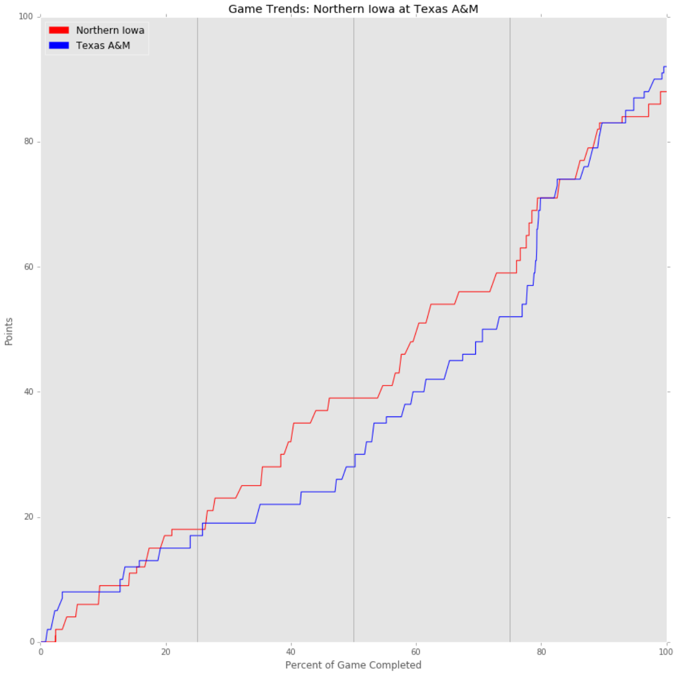


Figure 3: Trend graph showing scores over time for the Texas A&M-Northern Iowa second round NCAA Tournament game

Write a benchmark win probability model

The next thing that we had to do was write a win probability model for the data. We did this by following the conventional research (see here and here). The gist of the standard methodology is to collect a bunch of play-by-play data (of the type we have above) and then do a logistic regression in each time period based on the point spread and the difference in the game at that point in time.

We were able to implement this after some trouble, but did not fit the model to a lot of data. As of the writing of this document, we had just used a sample of 40 games to fit the data, which is much less than is typically used to fit the model (some versions that we have seen have used up to 20 times as many data). While the logistic regression model that we created predicted the final victor with about an 80 percent accuracy in each period (see table below), the coefficients varied so significantly across time periods that our win probability graphs were not nearly as smooth as we would like.

Table 1: Model's probability of predicting the winner correctly

Minutes Remaining	Odds of Correctly Predicting Victory
0	0.8925
5	0.8577
10	0.7964
15	0.7542
20	0.7899
25	0.8178
30	0.7706
35	0.6946
40	0.7934

Given that we have directly replicated prior analysis, however, we aim just to fit the model to more data in the future. We do not anticipate any conceptual, only logistical, difficulties in assembling this data; however, given the time and thesis constraints of this deadline, we sadly were not able to do so early.

Next Steps

Our project proposal demanded that we have three major items in order to run the tests we wanted:

1. **Game-by-game Twitter data**. As described above, we need this to be able to gauge fan sentiment at every point during the game so that we can get their impression on how the game is going
2. **Game progress data**. This comprises both the score at each point in time[3] and the probability of the favorite winning at each point in time
3. **Twitter prediction model**. For our final results, we will essentially be comparing the standard win probability models above to the Twitter prediction model that we build. Sentiment analysis also remains to be accomplished, but requires a repurposing of existing infrastructure.

Consequently, we can say that we have accomplished most of our goals. While there are minor things to do on the first two points[4] we are mostly focused now on creating the Twitter model.

[3] Of more importance to us is the difference in scores—whether a team is up 100-90 or 10-0 is immaterial; the ex ante predictors of team strength we used are based in the predicted final score difference.

[4] One thing that jumps out as an immediate next step on the data is matching analog times (i.e. 2:00 EST) from the Tweets to the time left in the game. This will allow us to directly match the predictions by Twitter to predictions made by the win probability model.

To accomplish that, we will need to know how to classify the Tweets by strength of sentiment. Our task after that is to think about how to map those sensitivity analyses to a point spread. Once we do that, we can test our model against basic default models of win probability and see how well Twitter performs, which will give us the desired final results.