The Botball Team Tracker
Marty Rand
Norman Advanced Robotics
marsedge95@gmail.com

The Botball Team Tracker

Introduction

In previous Botball years, a Botball team's performance had to be aggregated from a variety of sources ranging from YouTube to KIPR's officially released scores. Even with this aggregation, much of the teams' data would still be unknown. What was their worst seeding score? What did they score in double elimination? Moreover, if someone had all that information, doing statistical analysis of the data is difficult for the average person. The Botball Youth Advisory Council's (YAC's) solution to this is the Botball Team Tracker [1], a new feature of the community site that tracks the statistics of every Botball team in real time.

Origin

The Team Tracker was originally Jake Hall's idea in 2010. At the time, the Youth Advisory Council did not have SSH access to the community server. Thus, the project was postponed until we had server access. Some initial work was done on the frontend of the Team Tracker in 2012, but no scores were displayed and it was very early [2]. Later, in December 2012, the project started up again. By March 2013 (the Oklahoma Regional Tournament), a version of the Team Tracker was live on the community site.

Origin of Team Data

Botball Live is not just a video stream. It also sends tweets of how a team did. This includes seeding, double elimination, and alliance scores. What most people do not realize is that the Youth Advisory Council saves all those scores in an XML database on the community server. These scores are the basis for the Team Tracker's data. Later, we added KIPR's official score releases so the documentation component would be accurate.

Initial Design

The first version of the Team Tracker was coded mostly in JavaScript. When the Team Tracker page was loaded, it would get a XML file that lists all the events for which it has data. It would then get the XML files for each particular event (match numbers and teams from one file, scores from another file for each table per event). It would then calculate the statistics, and then display the data in several giant tables. We did it this way to simplify the process of adding new events to the database. All we had to do was add a new event to the event XML file and start logging data to the proper location. This version debuted one week before the 2013 Oklahoma Regional Tournament and was the first public release of the Team Tracker.

Initial Design Implementation

The XML data was loaded via asynchronous AJAX calls through jQuery so that in the future we could reload the data without reloading the web page. The JavaScript consisted of three basic parts. First, there was the component that retrieved the XML files from the community server as described above. Second, the JavaScript wrote the data we had (raw scores) into a data table. It then retrieved the necessary parts to calculate each statistic (e.g. retrieved the three scores from the table and found the lowest one for the minimum). It wrote each statistic as it went until all the columns were filled with data. The third, and final, part filters the data based on what is selected in the filter boxes on the top of the web page. For example, this makes it so someone can only see their region's tournaments. The filtering part is executed every time the filters are changed while the first two parts are only executed upon page load.

Limitations of the Initial Design

The initial design was riddled with problems. First, and most obvious to the user, the page loaded very slowly. It took 30 seconds to a minute for the page to load on a high-end computer. On a Samsung Galaxy S3, it took 5 minutes to load the page. This is because we were trying to calculate 8 different statistics for every team in every region in seeding alone upon page load. Moreover, JavaScript is not a fast language to do all these calculations. All this combined with the fact that we had to download a different XML file for every table in every region made the process very slow. The most bizarre of the issues was that Mozilla Firefox did not work correctly, but Google Chrome did. We decided to put a disclaimer for the time being that Firefox support was broken. The initial Team Tracker also did not have Alliance or Overall score support. This last issue is more due to lack of time however.

Fixing the Load Times

After the 2013 Oklahoma Regional Tournament, a complete re-write of the backend took place. The multiple XML files were replaced with a central XML file for each event. This one XML file would have all of the data from every Botball Live XML file from that event. All this would now be aggregated through a Python script. In addition, because we now have a Python script, we could pre-calculate all the statistics and store them in the XML file. This makes it so the Team Tracker only has to download one file per region and copy the data from the XML file to the data table. We also optimized the JavaScript further by having it only re-draw the tables at the end of the loading sequence. These changes decreased the load times to less than one second after the HTML was loaded. Inexplicably, Firefox also worked in this new version. We are still unsure of the origin of this last bug, but luckily, it is not an issue anymore. These changes took until June to implement into the public site. In the mean time, the Team Tracker was updated with new data from tournaments as they happened along with Botball Live.

New Design Implementation

The new design is similar to the old design in the backend. The python script does the same thing the JavaScript did. It gets the XML files, writes what it has (team scores) to a new XML file, and calculates what it does not have (the statistics). This python script recalculates everything every time new data comes in (every time a new score is entered into Botball Live). The python script

uses the lxml module (library) for XML parsing. Once the all of the new XML file has been calculated, it saves it to the community site. All the JavaScript now does is read the new XML files and copy the data to the data tables on the website. Since everything is pre-calculated now, this process is much faster.

Old Seeding XML Example

<score matchnum="13-0113-1" score1="880" score2="" timestamp="Sat, 09 Mar 2013 17:22:11 +0000" />

This is a single entry from the Botball Live XML data. It has the team number, round, that round's score and a timestamp that the Team Tracker does not need. Note that this does not even have a simple team name. We have a separate file with all the team names and ID's. The JavaScript collected many of these and calculated all the statistics for the team upon page load.

New Seeding XML Example

<team coef_var="0.181" id="13-0113" max="880" min="610" name="Norman Advanced Robotics" overall_mean="745.0" rank="1" score1="880" score2="610" score3="610" std_dev="127.279" sum="2100" total_mean="700.0" z_score_coef_var="4.325" z_score_combined="131.548" z_score_mean="258.771"/>

This is Norman Advanced Robotics's XML entry in the new XML format for seeding at the 2013 Oklahoma Regional Tournament. Obviously, all the statistics have already been calculated. There are some intermediate numbers in the XML that people might be interested in, but it became too much data to fit on the screen. Thus, things like the standard deviation were hidden. All these are still in the XML file if people want to look at it.

Terminology

Standard Deviation

The standard deviation of a dataset, simply put, is the average variation in the data. This can be used to measure consistency (or lack thereof). The closer the standard deviation is to 0, the less the data varies.

Coefficient of Variation

The coefficient of variation is the standard deviation divided by the mean. This still measures variability, but it can compare datasets of different scales. In the Team Tracker, every team has a separate mean score (scale). By using the coefficient of variation instead of the standard deviation, we allow relative variation to be compared between teams that score different numbers of points. For example, a team that scored 800, 700, and 600 and their 3 seeding scores would have a much higher standard deviation than a team that scored 30, 0 and 0. However, the high scoring team would have a much lower coefficient of variation because their scores are closer when you take into account their mean. The team tracker uses coefficients of variation for all applicable calculations for this reason.

Z Score

A Z score is used to show a team's success relative to the population of teams. A Z score of 0 indicates the team was average. Every whole number above or below 0 is the number of standard deviations above or below the mean that team was. For example, Norman Advanced Robotics's Z score for seeding at the Oklahoma Regional Tournament was 131.548. This means they did 131.548 standard deviations better than the average team in seeding.

Statistics Displayed

Currently, the Team Tracker displays the minimum score, maximum score, overall score (best 2/3), total mean score (all 3), sum, rank, coefficient of variation, and Z score for each team in seeding. The Team Tracker also displays the same statistics for an average team in every region (to show the most competitive regions). All of these columns are sortable.

Why all the Statistics?

As shown above, the Team Tracker has far more statistical analysis than KIPR provides. We did this to give people a better understanding of what the data means. For example, the coefficient of variation we display is to judge the consistency of a team. A team could score 1000 points once and 0 twice more. This hypothetical team would be at a great disadvantage in double elimination to a team that always scores 500 points. To add to this effect, we calculate the coefficient of variation from all 3 seeding scores, not just the top two. We do this because that 3rd score is a valid run. Without it, the data is skewed such that teams seem to be doing better than they are. In fact, all of our seeding statistics (except for the overall seeding score) use all 3 runs for this reason. Special attention was paid to the Z score calculation. The "Z Score" column of the Team Tracker is actually the average of the Z score for that team's mean (using all 3 runs) and the Z score for that team's coefficient of variation. We did this so that this value would be an accurate representation of the overall performance of a team in seeding. We give equal weight to both score and consistency so that only consistently good teams rank high by this measure. In addition, giving one of them any arbitrary weight might appear biased on the part of YAC. Therefore, we keep them equal. Another benefit of calculating all of this is that the user can sort the data by any of these columns. This way, anyone can find the most consistent team, the highest scoring team, the team with the highest 3rd round score, etc.

Other Changes after Initial Release

The first change was a quick fix. Originally, there were so many columns of data that only 1080p screens could display all the columns without them being cut off. Daniel Goree, of Norman Advanced Robotics, pointed this out to us [3]. We quickly added a horizontal scroll bar at the bottom of each table. Later, we added Alliance and Overall score support. This way, every category of the tournament would be covered by the Team Tracker. Overall score support works in real time. Therefore, as the tournament progresses, teams' overall scores will change to reflect the current standings.

Uses for the Team Tracker

The Team Tracker has a number of uses. It can show teams their chances of victory as a tournament progresses. It can show teams in other regions whom to watch for at GCER and future years. It can sort teams by any statistic to better understand a team's weakness (e.g. inconsistency). Similarly, a team can use it to show them their own weakness to improve upon that for the next tournament. In addition, it can help with scouting at a tournament to find potential rivals. In the future, it can help mentors teach teams effective strategy considerations [2].

Advantages over Joust

Joust is the system KIPR currently uses for score keeping. It also generates the double elimination brackets. Joust does not display any statistics about a team, show a team's score in DE, or show any alliance or overall data to name a few issues. The Team Tracker does all of this in real time. Joust does not generate the bracket until KIPR pushes a button just before DE starts. While not implemented yet, the Team Tracker has the potential to dynamically update the DE brackets as seeding happens. This would help teams see their next opponent faster. In the same manor, it will already show the overall standings before the awards ceremony happens. This way, teams will not have to guess the rank they got; they will know it. While Joust can display real-time overall scores, KIPR chooses to keep those hidden. Obviously, special awards will be unknown until the award ceremony. A major complaint about Joust is that it is slow. This is because it is run from a laptop (sometimes a netbook) at the event. With the Team Tracker, the scores will be live on the community site so anyone with internet access can get them much faster. The Team Tracker even works on smart phones.

Future Plans

The Team Tracker is by no means finished. YAC still has great hopes to improve it. Some of these improvements are:

1. Add numerous filters to narrow the data (filter by team, score, middle vs. high school, rookie vs. veteran team, etc)
2. Add a category for all regions combined. Currently, the Team Tracker categorizes the data by region. We hope to add a category that is all regions combined. This would allow for direct comparisons of teams from different regions.
3. Add graphs to better visualize the data
4. Add real-time double elimination bracket generation
5. Add a special awards section
6. Add a GCER paper section with links to the respective papers
7. Add a Autonomous Robot Showcase section with links to the respective papers
8. Add a Video Showcase section with links to the videos
9. Add links to the Botball Live Videos on YouTube (click on a score to see the match)
10. Add a Research and Design Challenge section
11. Add a Live-Blog Challenge section with links to the blogs
12. Add a Youth Advisory Council Tracker
13. Get the complete standings data from previous tournaments from KIPR

14. Add feeder school associations so people can see the flow of students from middle to high school

These changes are already known by YAC. If someone has another idea for a feature (or finds a bug), they can submit it to the community forum under "Site Feedback" → "Official Team Tracker Bug Tracker / Feature Request Thread" [4].

Conclusion

The Botball Team Tracker is still a work in progress. Still, we believe that it is ready for general use and is very useful for everything from scouting at tournaments to off-season education. As always, you can find YAC on the community site [5] and at most Botball events.

References

[1] http://community.botball.org/team-tracker
[2] http://files.kipr.org/gcer/2012/proceedings/_Rand_WhenBotballLive_SNARC.pdf
[3] http://community.botball.org/forum/technical/site-feedback/official-team-tracker-bug-tracker-feature-request-thread#comment-8138
[4] http://community.botball.org/forum/technical/site-feedback/official-team-tracker-bug-tracker-feature-request-thread
[5] http://community.botball.org