Proposal: Major League Baseball Next Pitch Prediction

Joel Klein, Jake Sauberman, Ben Perkins
Indiana University - Bloomington

October 10, 2021

Background:

Pitch prediction in Major League Baseball has been a hot-button topic since the 2017 Astros were found cheating, using an in-game algorithm to decode catchers' signs and know which pitch is coming next. The objective in this project is not to decode signs, but rather predict the next pitch based on readily-available information about the game state and the current actors. Having a data-driven pitch prediction system in place could help organizations better plan for their upcoming opponents, with newfound knowledge of opposing pitchers' predicted tendencies.

Data:

The data for the project will be obtained via a custom script designed to pull from the Baseball Savant API. BaseballSavant.com provides MLB StatCast data via a web interface as well as an external API. The data will be accessible in specified date ranges and includes many data points on pitch type, release speed, direction, and positions, and a host of others. Each call will download a CSV file. A link to the CSV documentation gives information on each available data point.

Methods:

Predicting the next pitch thrown (i.e. fastball, sinker, slider, changeup, etc.) is based on both current and previous state information. Current state reflects the in-game situation immediately prior to the pitch thrown (time t). A few features representing the current game state include the count (strikes and balls), runners on base, game score, inning, outs, the pitcher, and the batter. Previous state information reflects game results prior to the upcoming pitch (time $t\text{-}1, t\text{-}2, \ldots t\text{-}k$, where k is the number of prior pitches thrown). Previous state information includes previous pitch types thrown, batter result (i.e. swing and miss, single, double, home run, etc.), and pitch metrics (i.e. location, spin rate, speed, etc.).

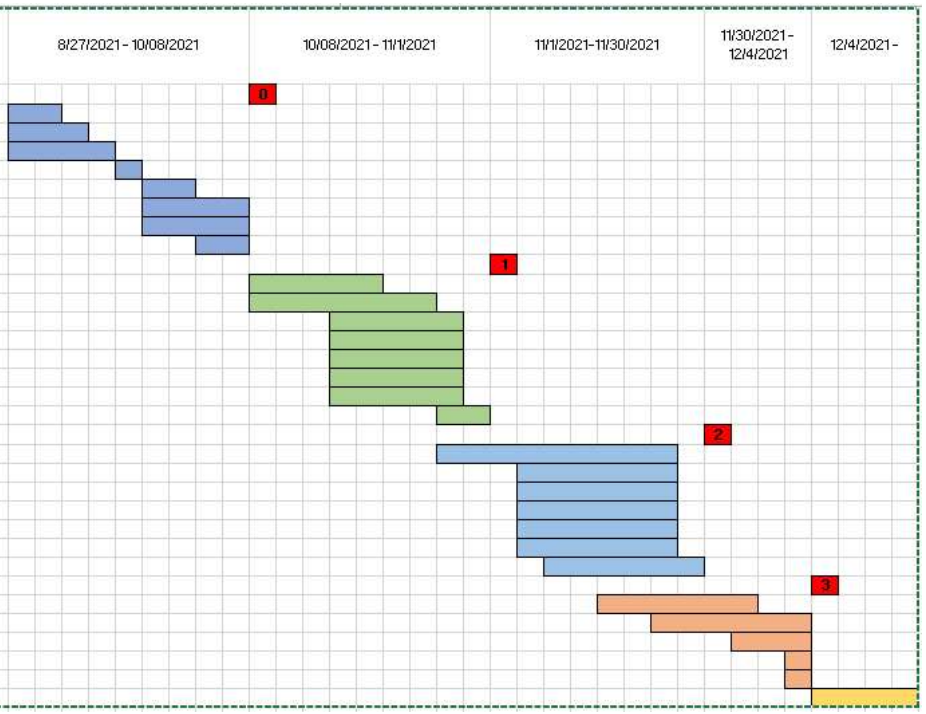
A recurrent neural network (RNN) is well suited to generate the next pitch prediction due to baseball's sequential nature. The proposed model architecture starts with a long short-term memory (LSTM) network combining both previous state information at $t\text{-}1, t\text{-}2, \ldots, t\text{-}k$ and current state information at t. The pitcher and hitter combination at time t heavily influence the next thrown pitch types. However, thousands of such combinations exist and one-hot encoded vector representations lead to dimensionality issues and slow training. An extension to [1], generating pitcher and hitter embedding vectors with historical pitch types thrown as output, will reduce the pitcher-hitter input dimensionality into the LSTM network. The output of the LSTM network will then be passed through to a fully connected network to classify the next pitch type.

Research questions/Goals:

- Scrape and prepare data from *baseballsavant.com* to enable model building and experimentation
- Can we improve classification accuracy by 5% over the naïve classification method? (predicting the most frequent pitch type thrown for each pitcher)
- Can we improve classification accuracy by 5% over the naïve classification method for rookie pitchers?
- Will the algorithm predict pitch types with different accuracy levels?
- Will the algorithm increase accuracy for pitchers across the length of the game (hypothesis: the algorithm accuracy will increase as the game progresses)

Schedule:

No.	Task/Deliverable
0	Deliverable Phase 0: Project Proposal
0.1	Form project group
0.2	Determine objectives/goals
0.3	Project review/description
0.4	Project kick-off meeting
0.5	Research Data Access & Description
0.6	Research Methods
0.7	Determine project
0.9	Create Proposal
1	Deliverable Phase 1: Pitcher/Batter Embedding Feature Generation
1.1	Load data
1.2	Exploratory data analysis
1.3	Build Data Prep & Model Pipelines
1.3.1	Feature engineering
1.3.2	Develop Embedding Architecture
1.4	Train Model
1.4.1	Hyperparameter tuning
1.5	Evaluation & results
2	Deliverable Phase 2: Build, Train, and Evaluate LSTM + DNN Model for Next Pitch Prediction
2.1	More Exploratory data analysis
2.2	Build Data Prep & Model Pipelines
2.2.1	Feature engineering
2.2.2	Develop LSTM + DNN Architecture
2.3	Train Model
2.3.1	Hyperparameter tuning
2.4	Evaluation & results
3	Deliverable Phase 3: Final report
3.1	Write Final report
3.2	Final presentation
3.2.1	Prepare presentation
3.2.2	Record presentation
3.2.3	Post presentation
4	Feedback



Team:

Every team member will be heavily involved in each project phase. For each project phase there will be a lead and two contributors. The lead will be responsible for researching and designing the model methodology and evaluation strategy while the two contributors will develop and execute the analysis/training/evaluation.

Phase 1:

- Lead: Ben
- Collaborators: Joel, Jake

Phase 2:

- Lead: Joel
- Collaborators: Jake, Ben

EDA/Phase 3:

- Lead: Jake
- Collaborators: Ben, Joel

References

[1] M. Alcorn, “(batter|pitcher)2vec: Statistic-Free Talent Modeling With Neural Player Embeddings”, MIT Sloan Sports Analytics Conference 2018. URL: https://drive.google.com/file/d/19ulLWQUgpIw-4pK5d7wa1lDhmhTiph37/view