Mathematics in Baseball

Alex Quiroga

Mathematics Department
California Polytechnic State University
San Luis Obispo
2014-2015

Introduction

Our goal is to rank all starting pitchers in Major League Baseball by using several ideas and theorems from linear algebra. We will do this by writing a program that will take each pitchers stats versus all teams they faced in the 2014 season, and compare them to other pitchers who faced the same teams. This will lead to a useful ranking system.

Linear Algebra Concepts

Definition. *Let A be a square matrix. If there is a non-zero vector* $\mathbf{v}$ *and a constant* λ*, if* $A\mathbf{v} = \lambda\mathbf{v}$*, then* λ *is an eigenvalue of* A *and* $\mathbf{v}$ *is the corresponding eigenvector for* λ*.*

Example. Let A = $\begin{pmatrix} 1 & 1 \\ -2 & 4 \end{pmatrix}$. Then $\mathbf{v} = \begin{pmatrix} 6 \\ 6 \end{pmatrix}$ is an eigenvector with corresponding eigenvalue $\lambda = 2$ because A$\mathbf{v}$ =$\begin{pmatrix} 1 & 1 \\ -2 & 4 \end{pmatrix} \begin{pmatrix} 6 \\ 6 \end{pmatrix} = \begin{pmatrix} 12 \\ 12 \end{pmatrix}$ and $\lambda\mathbf{v} = 2$ $\begin{pmatrix} 6 \\ 6 \end{pmatrix} = \begin{pmatrix} 12 \\ 12 \end{pmatrix}$. So A$\mathbf{v}$ = $\lambda\mathbf{v}$, which satisfies the definition of eigenvalues and eigenvectors.

It is possible for a matrix to have more than one eivenvalue. In that case, there would also be multiple eigenvectors, with each one corresponding to a particular eigenvalue.

Definition. *The spectral radius of a matrix is the eigenvalue with the largest magnitude. This is denoted* $\rho(A)$*.*

Example. If A = $\begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix}$, then A$\begin{pmatrix} i \\ 1 \end{pmatrix} = \begin{pmatrix} 2i-1 \\ 2+i \end{pmatrix}$ = (2+i)$\begin{pmatrix} i \\ 1 \end{pmatrix}$. That would give us λ_1=(2+i), with corresponding eigenvector $\mathbf{v_1}$=$\begin{pmatrix} i \\ 1 \end{pmatrix}$. Additionally, A$\begin{pmatrix} -i \\ 1 \end{pmatrix} = \begin{pmatrix} -2i-1 \\ 2-i \end{pmatrix}$ = (2–i)$\begin{pmatrix} -i \\ 1 \end{pmatrix}$, which would give us λ_2=(2–i), with corresponding eigenvector $\mathbf{v_2}$=$\begin{pmatrix} -i \\ 1 \end{pmatrix}$. Since we we have two eigenvalues, the one with the bigest magnitude would be the spectral radius. The magnitude of a complex number of the form $a + bi$, for real numbers a and b, is $\sqrt{a^2 + b^2}$. In this case, the magnitude of λ_1 is equal to the magnitude of λ_2, so the spectral radius is $|2 - i| = 5$.

These definitions and ideas are integral in establishing our ranking system.

Ranking Method

Starting with an example, we can build up to an interesting theorem.

Example. Say we want to examine 4 ptichers from Major League Baseball. The four pitchers we will look at are Chris Tillman, Chris Young, Clay Buchholz, and Clayton Kershaw. We can build a matrix if we take two pitchers from these 4, lets say Tillman and Young, and find all instances where both pitchers matched up against the same team. From there, we can do a comparison. If Tillman had a lower career batting average than Young, then add one to our count value. If it was visa versa, then subtract one from the count value. If the final count value is negative, then Tillman will receive a score of zero. If the count is positive, then Tillman will receive a score of 1. If the count is zero, meaning the pitchers never faced the same team, or the final count value was tied, then Tillman will receive a score of .5. For example, Tillman's score over Young was zero, as Young had a lower career batting average against against more teams than Tillman. We can use these scores to construct a 4×4 matrix of each score of each pitcher versus the three other pitchers. For example, the A,B entry of the matrix was the score of pitcher A over pitcher B. All data required to do this was acquired form rotowire. This is that matrix:

$$\begin{pmatrix} 0.5 & 0 & 1 & 0 \\ 1 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 1 & 1 & 1 & 0.5 \end{pmatrix}$$

Let r_1, r_2, r_3, and r_4 represent the ranking of each of these 4 pitchers. Let $P_{i,j}$ be the score that pitcher i had over pitcher j, which can be found by the corresponding value in the matrix.

$$\begin{aligned} r_1 &= k(P_{1,2}r_2 + P_{1,3}r_3 + P_{1,4}r_4) \\ r_2 &= k(P_{2,1}r_1 + P_{2,3}r_3 + P_{2,4}r_4) \\ r_3 &= k(P_{3,1}r_1 + P_{3,2}r_2 + P_{3,4}r_4) \\ r_4 &= k(P_{4,1}r_1 + P_{4,2}r_2 + P_{4,3}r_3) \end{aligned}$$

Where k is what we call the constant of proportionality. Writing each r this way makes sense, as it takes into account the probability that you can beat better pitchers, as beating better pitchers is a better sign of strength than beating lesser pitchers.

Using the information we have, this can become

$$\begin{aligned} r_1 &= k(.5r_1 + 0r_2 + 1r_3 + 0r_4) \\ r_2 &= k(1r_1 + .5r_2 + .5r_3 + 0r_4) \\ r_3 &= k(0r_1 + .5r_2 + .5r_3 + .0r_4) \\ r_4 &= k(1r_1 + 1r_2 + 1r_3 + .5r_4) \end{aligned}$$

Let $\mathbf{r}$ be the vector of rankings. Then we can note $\mathbf{r} = k\mathrm{A}\mathbf{r}$, or $\frac{1}{k}\mathbf{r} = \mathrm{A}\mathbf{r}$. We can note that by definition, $\frac{1}{k}$ is an eigenvalue, and $\mathbf{r}$ is the corresponding eigenvector. Now one of these eigenvectors is the ranking we want, but which one is the one we want? The Perron-Frobenius Theorem can help us solve this problem.

Theorem 1 (The Perron-Frobenius Theorem)**.** *If we have a square matrix A with all entries greater than zero, then A has an eigenvalue λ with $\lambda = \rho(A)$, and that there is an eigenvector $\mathbf{x}$ corresponding to λ which satisfies $\mathbf{x} > 0$, where $\mathbf{x} > 0$ means that each entry in $\mathbf{x}$ is greater than zero.*

This says if all entries of A are greater than zero, then we are guaranteed an eigenvector with only positive entries, and this eigenvector is corresponding the spectral radius of A.

A Proof of the Perron-Frobenius Theorem

Before we can prove the Perron-Frobenius Theorem, we will need to prove some interesting facts.

Theorem 2. *Let A be a square matrix. If there is an invertible matrix P and an upper triangular matrix T for which $A = PTP^{-1}$, then the eigenvalues of A are the entries on the diagonal of T.*

Proof. In order to show that the eigenvalues of A are along the the diagonal of T, we must show that $\det(A–\lambda I) = \det(T–\lambda I)$.

Surely, det(A–λI) = det(PTP^{-1}–λI)

Since PIP^{-1} is I, then $\det(A–\lambda I) = \det(PTP^{-1}–\lambda PIP^{-1})$.

Rearranging terms, we get det(A–λI) = det(P(T–λI)P^{-1}).

This is equivalent to $\det(A–\lambda I) = \det(PP^{-1}$(T–λI)).

Finally, this gives us $\det(A–\lambda I) = \det(T–\lambda I)$, as desired. *Q.E.D.*

Theorem 3. *For any square matrix A, there is an orthoganal matrix P and an upper triangular matrix T such that $A = PTP^{-1}$*

Proof. Let $\mathbf{v}$ be an eigenvector of A with eigenvalue λ. Using Gram-Schmidt, there are vectors $\mathbf{w_2}$, ..., $\mathbf{w_n}$ which make $\mathbf{v}$, $\mathbf{w_2}$, ..., $\mathbf{w_n}$ an orthonormal basis. Let P_1 be the matrix with columns $\mathbf{v}$, $\mathbf{w_2}$, ..., $\mathbf{w_n}$. By looking at ${P_1}^{-1}AP_1$, we can see that

$$ {P_1}^{-1}AP_1 = \begin{pmatrix} \mathbf{v} \\ \mathbf{w_2} \\ \vdots \\ \mathbf{w_n} \end{pmatrix} (A) \begin{pmatrix} \mathbf{v} & \mathbf{w_2} & ... & \mathbf{w_n} \end{pmatrix} $$

Which implies

$$ P_1^{-1}AP_1 = \begin{pmatrix} \mathbf{v} \\ \mathbf{w_2} \\ \vdots \\ \mathbf{w_n} \end{pmatrix} \begin{pmatrix} A\mathbf{v} & A\mathbf{w_2} & ... & A\mathbf{w_n} \end{pmatrix} $$

Which by definition of an eigenvalue, implies that

$$ P_1^{-1}\mathrm{A}P_1 = \begin{pmatrix} \mathbf{v} \\ \mathbf{w_2} \\ \vdots \\ \mathbf{w_n} \end{pmatrix} \begin{pmatrix} \lambda\mathbf{v} & A\mathbf{w_2} & ... & A\mathbf{w_n} \end{pmatrix} $$

Which implies that

$$P_1{}^{-1}AP_1 = \begin{pmatrix} \lambda\mathbf{v}...\mathbf{v} & * & ... & * \\ A\mathbf{v}...\mathbf{w_2} & * & ... & * \\ \vdots & \vdots & \vdots & \\ A\mathbf{v}...\mathbf{w_n} & * & ... & * \end{pmatrix}$$

We can notice that in the bottom right part of the matrix, there is now a smaller square matrix. Let's call it A'.

By induction, we will show that A = PTP^{-1}.

The Base Case (n=1) is trivial.

Inductive Step: Assume decomposition A = PTP^{-1} holds for an $n-1 \times n-1$ matrix. Let P = $P_1 \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ \vdots & & P_2 & \\ 0 & & & \end{pmatrix}$

Then $\begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ \vdots & & P_2^{-1} & \\ 0 & & & \end{pmatrix} = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ \vdots & & P_2 & \\ 0 & & & \end{pmatrix}^{-1}$.

By taking $P^{-1}AP$, we have

$$P^{-1}AP = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ \vdots & & P_2 & \\ 0 & & & \end{pmatrix}^{-1} P_1^{-1}AP_1 \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ \vdots & & P_2 & \\ 0 & & & \end{pmatrix},$$

which implies

$$P^{-1}AP = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ \vdots & & P_2 & \\ 0 & & & \end{pmatrix}^{-1} \begin{pmatrix} \lambda & * & ... & * \\ 0 & & & \\ \vdots & & A' & \\ 0 & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & & & \\ \vdots & & P_2 & \\ 0 & & & \end{pmatrix},$$

which implies that

$$P^{-1}AP = \begin{pmatrix} \lambda & * & ... & * \\ 0 & & & \\ \vdots & & P_2^{-1}A'P_2 & \\ 0 & & & \end{pmatrix}$$

This an upper triangular matrix from our inductive hypothesis. Therefore, any square matrix A can be decompsed into an orthoganal matrix P and an upper triangular matrix T such that A = PTP^{-1}

Q.E.D.

Theorem 4. *If A is an $n \times n$ matrix with every eigenvalue equal to zero, then $A^n = 0$.*

Proof. Let A be an $n \times n$ matrix with every eigenvalue equal to zero. We wish to show $A^n = 0$. Consider, from Theorem 3, that A = PTP^{-1}. Following that,

$A^n = PT^nP^{-1}$. If we can show $T^n = 0$, then it will follow that $A^n = 0$. We can do this by induction by showing that by raising T to the k^{th} power moves the upper triangle of *'s to the $k+1^{st}$ column.

Base Case (k=1):

Let T = $\begin{pmatrix} 0 & * & ... & * \\ \vdots & \ddots & \ddots & * \\ \vdots & & \ddots & * \\ 0 & ... & ... & 0 \end{pmatrix}$

Inductive Step:

Assume that $T^k = \begin{pmatrix} 0 & 0 & ... & 0 & * & ... & * \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & * \\ \vdots & & & \ddots & \ddots & & 0 \\ \vdots & & & & \ddots & \ddots & \vdots \\ 0 & ... & ... & ... & ... & 0 & 0 \end{pmatrix}$

Where the upper triangle of *'s starts in the $k+1^{st}$ column

We wish to show that T^{k+1} moves the upper triangle of *'s so that it begins in the $k+2^{nd}$ column.

Take $T^{k+1} = TT^k = \begin{pmatrix} 0 & * & ... & * \\ 0 & 0 & \ddots & * \\ \vdots & & \ddots & * \\ 0 & ... & ... & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & ... & 0 & * & ... & * \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & * \\ \vdots & & & \ddots & \ddots & & 0 \\ \vdots & & & & \ddots & \ddots & \vdots \\ 0 & ... & ... & ... & ... & 0 & 0 \end{pmatrix}$

By matrix multiplication, we can see that the first k columns of T^{k+1} will be zeros. Notice that the $k+1^{st}$ column of T^k is a * followed by n-1 zeros. When this column is multiplied by the first row of T, which is one zero followed by n-1 *'s, we can see that this leads to an entry of 0 in the first row and $k+1^{st}$ column of T^{k+1}. Taking the the $k+2^{nd}$ column of T^k, which is two *'s followed by n-2 zeros, times the first row of T will result in an arbitraty value *. However, taking this column by the second row of T, which is two zeros followed by n-2 *'s, will result in a value of 0. We can continue this process untill we see that we are left with

$$T^{k+1} = \begin{pmatrix} 0 & 0 & ... & 0 & 0 & * & ... & * \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & \ddots & * \\ \vdots & & & \ddots & \ddots & & 0 & 0 \\ \vdots & & & & \ddots & \ddots & \vdots & \vdots \\ 0 & ... & ... & ... & ... & 0 & 0 & 0 \end{pmatrix}$$

As desired. So for k=n, we can see that the first n rows and columns would all be zero, which makes T^n a matrix of zeros.

Q.E.D.

Theorem 5. *If a square matrix A has entries greater than zero, then A has a nonzero eigenvalue.*

Proof. Let A be a square matrix with entries greater than zero. We wish to show A has a nonzero eigenvalue. Assume, to the contrary, that A does not have a nonzero eigenvalue. Since every eigenvalue is equal to zero, then by Theorem 3, $A^n = 0$. But A only has positive entries, so $A^n \neq 0$. We have reached a contradiction, so therefore, A must have a nonzero eigenvalue. *Q.E.D.*

Theorem 6. *If $\rho(A) < 1$, then $\lim_{n\to\infty} A^n$ is the zero matrix.*

Proof. Suppose $\rho(A) < 1$. We can express A as $A = PTP^{-1}$, where P is an orthoganal matrix and T is an upper triangular matrix. So $A^n = [PTP^{-1}]^n = PT^nP^{-1}$. Since $\rho(A) < 1$, then the entries along the diagonal of T are less than 1 in magnitude, by Theorem 2. Let us construct a matrixT' whose values are the same as T unless all entries along the diagonal are not distinct. If they are not distinct, then for entries $a_{ii} = a_{jj}$, then add an ϵ such that $a_{jj} + \epsilon < 1$. By design, T' has distinct values along the diagonal. It is important to note that chaning the entires of the diagonal by a small amount will also have a small change on the eigenvalue of the matrix. Therefore we can write $T' = SDS^{-1}$ where S is invertable and D is a matrix with only the eigenvalues of T' along the diagonal. So $\lim_{n\to\infty} T'^n = \lim_{n\to\infty} SD^nS^{-1} =$ the zero matrix. By taking the limit as ϵ goes to zero, we get that $\lim_{n\to\infty} T^n$ also equals the zero matrix. Therefore, $\lim_{n\to\infty} A^n$ must also be the zero matrix. *Q.E.D.*

Theorem 7. *Suppose that A is a square matrix with positive entries, and with $\rho(A) = 1$. Let $\mathbf{x}$ be an eigenvector corresponding to an eigenvalue λ with $|\lambda| = 1$ and let $\mathbf{y} = A|\mathbf{x}| - |\mathbf{x}|$, where $|\mathbf{x}|$ denotes the vector found by taking the magnitude of each entry in $\mathbf{x}$. Then $\mathbf{y} = \mathbf{0}$, and $|\mathbf{y}| = \mathbf{0}$*

Proof. Suppose what is given in the above stated theorem. By the Triangle Inequality, $|\mathbf{a} + \mathbf{b}|$ for vectors $\mathbf{a}$ and $\mathbf{b}$. Take a random row i of A, and multiply it by $\mathbf{x}$.

Then surely $|a_{i,1}x_1 + ... + a_{i,n}x_n| \leq |a_{i,1}x_1| + ... + |a_{i,n}x_n|$.

So $|\lambda x_1 + ... + \lambda x_n| \leq a_{i,1}|x_1| + ... + a_{i,n}|x_n|$.

Which means $|\lambda \mathbf{x}| \leq A|\mathbf{x}|$

and since $|\lambda| = 1$, then $|\mathbf{x}| \leq$ A$|\mathbf{x}|$

this implies that $|\mathbf{y}| \geq \mathbf{0}$.

Now suppose, to the contrary, that $\mathbf{y} \neq \mathbf{0}$ Let m_1 be the max entry of A$|\mathbf{x}|$ and m_2 be the minimum entry of A$\mathbf{y}$ Choose $\epsilon = \frac{m_2}{1+m_1}$.

Then $\epsilon(A|\mathbf{x}|) = \frac{m_2}{1+m_1}A|\mathbf{x}| = m_2\frac{A|\mathbf{x}|}{1+m_1}$. This implies the largest entry of A$|\mathbf{x}|$ is less than 1. Since $m_2 \leq$ A$\mathbf{y}$ and $\frac{A|\mathbf{x}|}{1+m_1} < 1$, then $m_2\frac{A|\mathbf{x}|}{1+m_1} < m_2 \leq$ A$\mathbf{y}$

Therefore A$\mathbf{y} > \epsilon(A|\mathbf{x}|)$

Now, by induction, we want to show that $B^n(A|\mathbf{x}|) > (A|\mathbf{x}|)$ for B = $\frac{A}{1+\epsilon}$

Base Case (n=1): We know A$\mathbf{y} > \epsilon(A|\mathbf{x}|)$.

This implies that $A^2|\mathbf{x}|$ − A$|\mathbf{x}| > \epsilon(A|\mathbf{x}|)$. By manipulation, we can arrive at $\frac{A^2|\mathbf{x}|}{1+\epsilon} >$ A$|\mathbf{x}|$.

Since B$(A|\mathbf{x}|) = \frac{A}{1+\epsilon}A|\mathbf{x}| = \frac{A^2|\mathbf{x}|}{1+\epsilon}$, we can imply $B(A|\mathbf{x}|) > (A|\mathbf{x}|)$

Inductive Step: $B^{n-1}(A|\mathbf{x}|) > (A|\mathbf{x}|)$. From the base case, we have that $B(A|\mathbf{x}|) > (A|\mathbf{x}|)$. We could plug in the inductive assumption into the base case and get that $B(B^{n-1}(A|\mathbf{x}|)) > (A|\mathbf{x}|)$ which implies $B^n(A|\mathbf{x}|) > (A|\mathbf{x}|)$, as desired. This concludes the Induction proccess.

Looking further at B, we can see $\rho(B) = \rho(\frac{A}{1+\epsilon}) = \frac{1}{1+\epsilon}\rho(A)$. Since ρ(A) = 1 by definition, then $\rho(B) = \frac{1}{1+\epsilon}$. Since 1+$\epsilon > 1$, then $\frac{1}{1+\epsilon} < 1$, which would imply that $\rho(B) < 1$

From Theorem 5, we know if ρ(A) < 1, then $\lim_{n\to\infty} A^n$ is the zero matrix. Since ρ(B) < 1, then $\lim_{n\to\infty} B^n$ is the zero matrix.

But, we showed $B^n(A|\mathbf{x}|) > (A|\mathbf{x}|) > \mathbf{0}$. Therefore, we have reached a contradiction, and our assumption that $\mathbf{y} \neq \mathbf{0}$ is false, and therefore $\mathbf{y} = \mathbf{0}$.

This leads us to the conclusion that $|\mathbf{y}| = \mathbf{0}$ *Q.E.D.*

We now have all the math we need to prove the Perron-Frobenius Theorem!

Proof of the Perron-Frobenius Theorem

Proof. Let $A > 0$ be a square matrix. Let $A' = \frac{1}{\rho(A)}$ A. We can show ρ(A) > 0 and $\mathbf{x} > \mathbf{0}$. Theorem 4 garantees that this matrix will have a spectral radius greater than zero.

Let $\mathbf{x}$ be the eigenvector of A' corresponding to the eigenvalue λ that is the spectral radius of A', where $|\lambda| = 1$.

We know, from Theorem 6, that A'$|\mathbf{x}| = |\mathbf{x}|$, or $\frac{1}{\rho(A)}$ A$|\mathbf{x}| = |\mathbf{x}|$.

This implies that$A|\mathbf{x}| = \rho$(A)$|\mathbf{x}|$. By definition, we know that $|\mathbf{x}|$ is an eigenvector of A corresponding to the spectral radius λ. We know that $|\mathbf{x}| \geq \mathbf{0}$. From Theorem 5 and since x is not the zero vector, and A > 0, then it implies A$|\mathbf{x}| > \mathbf{0}$. Since A$|\mathbf{x}| = \rho$(A) $|\mathbf{x}|$, then ρ(A) $|\mathbf{x}| > \mathbf{0}$. So $|\mathbf{x}| > \mathbf{0}$. Therefore, we have shown that for a matrix like A, then there is always a non-zero spectral radius, and the corresponding eigenvector is positive. *Q.E.D.*

Definition. *The Perron Vector is the unique vector, of length 1, corresponding to the spectral radius of a positive square matrix.*

It is this vector that we use for our ranking system. Going back to our example of four pitchers, we now know that the vector corresponding to the eigenvalue with the greatest magnitude. The four eigenvalues of this matrix are 1.4, .05+.6i, .05 - .6i, and .5. Therfore, the vector we want is the one corresponding to 1.4. We then scale this vector by dividing each term by the spectral radius. This will give us entries of the vector with values zero to one. Thefore, using Python (Appendix A), we can get that our ranking of these four pitchers is

$$\mathbf{r} = \begin{pmatrix} 0.25577 \\ 0.41266 \\ 0.22973 \\ 1.0 \end{pmatrix}$$

From this, we could see that the rank of the pitchers in descending order is Kershaw, Young, Tillman, and Buchholtz. We can see that Kershaw is considerably better, according to our method, than the other pitchers. We can also note that Young is significantly better than the remaining pitchers, who are very close in score. We can apply this on a much bigger scale.

Applying this to Major League Baseball

The point of this project is to apply the Perron-Frobenius Theorem to Major League Baseball to get a ranking of all starting pitchers in baseball. Only starting pitchers who started a Major League Baseball game during the 2014 season were taken into account. This was a total of 189 pitchers. In our example, the four pitchers were the first four on our list of 189.

Data was obtained from rotowire.com (Appendix D). From this data, we were able to find career statistics of all 189 pitchers who started a game in the 2014 season versus certain teams. Code was written in the Python program (Appendix B), and the data was extracted. Very similar to our example with four pitchers, we applied this to all 189 pitchers

This made a 189 by 189 matrix of all scores of each pitcher matchup. For example, the i, j entry of the matrix was the score of pitcher i over pitcher j. Then, we used Python to find the spectral radius of this matrix, along with the corresponding eigenvector. We then had our rankings vector. Multiplying by one over the spectral radius would allows us to get our ranking in values of 0 to 1, so it is easer to comphrend the results. We then multiplied by $\frac{1}{\rho(A)}$, and were able to get our rankings vector from 0 to 1, with 1 being the number one ranked pitcher. Here are the top 10 pitchers, as well as the bottom 10.

Top 10

1.0, Jake Arrieta

0.99137934130235217, Max Scherzer

0.95957029637888225, Gio Gonzalez

0.92354596933650401, Clayton Kershaw

0.91311685779509588, Hector Santiago

0.91296310825804294, Michael Pineda
0.91217415979086003, Collin McHugh
0.9071872703586279, Hiroki Kuroda
0.90421257820154277, Jake Peavy
0.9040886895285436, Hisashi Iwakuma
Bottom 10
0.38429956860924891, Zach McAllister
0.38406863688908288, Joe Kelly
0.38197471605466377, Brandon Maurer
0.3727032067213864, Vance Worley
0.34252728336826355, Wade Miley
0.33710078736784133, Ryan Vogelsong
0.32598804728113362, Josh Tomlin
0.27727824518520283, Aaron Harang
0.23331712269146454, Kevin Correia
0.22505858527164938, Ricky Nolasco

Introduction

Now that we have a system for ranking pitchers, we can use this to determine which teams have stronger pitching rotations, and what order, or strategy, each team should start their pitchers. The ideas behind this are game theory concepts.

Review of Basic Game Theory Concepts

Definition. *A game is any situation where participants choose courses of action to maximize their rewards.*

Definition. *A strategy is a comprehensive decision set made before the games as to how to act in every possible scenario.*

Definition. *A payoff matrix for a two player game is a matrix with an entry for every possible outcome of the game. One player, the row player, chooses a row, and the other, the column player, chooses a column. These selections are made independent of each other. If the row player chooses row i, and the column player chooses column j, then the i, j^{th} entry is the value that the column player pays to the row player.*

Example. A simple example is Rock, Paper, Scissors. This is a game because two players choose rock, paper, or scissors to try and beat their opponent. The payoff matrix would look like

$$\begin{array}{c} \\ Rock \\ Paper \\ Scissors \end{array} \begin{array}{c} \begin{array}{ccc} Rock & Paper & Scissors \end{array} \\ \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix} \end{array}$$

We can see that if the row player chose rock and the column player chose scissors, than the column player would pay the row player 1. If, however, the column player would have chosen paper instead, then the column player would pay -1, or in essence, would be payed 1 by the row player.

Definition. *The expected value of a game is the value we would expect the column player to pay to the row player on average given their respective strategies. Letting* $\mathbf{r}$ *denote the row player's strategy,* P *represent the payoff matrix, and* $\mathbf{c}$ *denote the column player's strategy, then the expected value is calculated as:*
$v = \mathbf{r}^T P\mathbf{c}$

Example. Let two players be playing a game of Rock, Paper, Scissors with the payoff matrix described in the previous example. If the row player's strategy was to choose only rock, and the column player's strategy was to choose rock half the time, and scissors half the time, then $\mathbf{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ and $\mathbf{c} = \begin{pmatrix} .5 \\ 0 \\ .5 \end{pmatrix}$.

That would mean the value of the game is

$$\text{v} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} .5 \\ 0 \\ .5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} .5 \\ 0 \\ -.5 \end{pmatrix} = .5$$

So with these strategies, the column player would expect to pay the row player .5.

Each player wants to use the best possible strategy, and that they will assume the other player will also use their best possible strategy.

Definition. *The optimal row (column) strategy is the row (column) strategy that maximizes (minimizes) the expected value of the game.*

Definition. *The value of a game is the expected value found by taking the optimal row strategy and the optimal column strategy.*

Definition. *A solution to the game is the value of the game, and the optimal row and column strategies.*

Example. Let two players be playing a game of Rock, Paper, Scissors with payoff matrix described in the previous examples. We wish to find the optimal row strategy. Let's say that the optimal stratgy is choosing rock p of the time, paper q of the time, and scissors (1-p-q) of the time, for poportions p and q. Then $\mathbf{r} = \begin{pmatrix} p \\ q \\ 1-p-q \end{pmatrix}$.

Under the Fundamental Principles of Game Theory, we have to assume the column player will choose the strategy that will minimize the value. The values that the row player could get then are given by

$$\mathbf{r}^T\text{P} = \begin{pmatrix} p, & q, & (1-p-q) \end{pmatrix} \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} -1+p+2q, & 1-2p-q, & p-q \end{pmatrix}$$

We want all entries of this vector to be equal, meaning we have picked a strategy that makes the column player indifferent, and unable to minimize the value any further. Setting each entry equal to each other, we see that p = 1/3, and q = 1/3, meaning $\mathbf{r} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$.

Similarly, we could do this for the column player, and get that $\mathbf{c} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$ as well.

That would give us a value of the game of

$$v = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 0$$

Thus we have a solution to the game, as we have a value and both optimal strategies.

The Simplex Method

The Simplex Method is an algorithm to find the minimumn or maximum of a problem. Problems that can be solved by the Simplex Method are usually of the form

$$\begin{cases} max\mathbf{c}^T\mathbf{x} \\ subA\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \end{cases}$$

This means we are trying to maximize $\mathbf{c}^T\mathbf{x}$, which is a scaler, subject to the constraints that $A\mathbf{x} \leq \mathbf{b}$. The best way to understand the simplex method is to go through an example.

Example. Suppose a shop makes three different goods called A, B and C which can all be sold for a profit. Each takes a certain amount of time to be made on three machines I, II, and III. These machines can only run for a certain amount of time each day. We could set up a table that looks like

	$Quantity$	I_{min}	II_{min}	III_{min}	$profit$
A	x	2	1	2	6
B	y	1	3	1	5
C	z	1	2	2	4
max		180	300	240	

Since we want to maximize profit, we would want to maximize the scaler found by $\begin{pmatrix} 6 & 5 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$. In this case, we can see this is of the form $\mathbf{c}^T\mathbf{x}$. We could also see that $A\mathbf{x} \leq \mathbf{b}$ is of the form $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 2 & 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leq \begin{pmatrix} 180 \\ 300 \\ 240 \end{pmatrix}$

Now we must introduce three slack variables s, t, and u. These slack variables, forming a vector $\mathbf{s}$, are added so that we get $A\mathbf{x} + \mathbf{s} = \mathbf{b}$.

Using all this, we can make what is called an initial tableaux. We set table up with our A matrix, next to the identity matrix, followed $\mathbf{b}$. Underneath that in the bottom row, we put $\mathbf{c}^T$ scaled by -1, with all other entries zero in this row. This would look like

$$\begin{matrix} x & y & z & s & t & u & b \end{matrix}$$
$$\begin{pmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 180 \\ 1 & 3 & 2 & 0 & 1 & 0 & 300 \\ 2 & 1 & 2 & 0 & 0 & 1 & 240 \\ -6 & -5 & -4 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The far bottom right corner represents the profit. If all entries along the bottom row besides the profit are nonnegative, then the profit will be at its max. If there are nonegative entries along the bottom row, take the most negative of all the entries. In this case, it is in the x column. Then find all ratios b_i/x_i for a row i and take the smallest. In this case, the desired ratio comes from row 1, and is $180/2 = 90$. Now, use row operations to make row 1 of the x column a pivot. Doing so results in the following tableaux:

$$\begin{matrix} x & y & z & s & t & u & b \end{matrix}$$
$$\begin{pmatrix} 1 & 1/2 & 1/2 & 1/2 & 0 & 0 & 900 \\ 0 & 5/2 & 3/2 & -1/2 & 1 & 0 & 210 \\ 0 & 0 & 1 & -1 & 0 & 1 & 60 \\ 0 & -2 & -1 & 3 & 0 & 0 & 540 \end{pmatrix}$$

Since we still have negative entries in the bottom row, we continute this algorithm. The final tableaux is

$$\begin{matrix} x & y & z & s & t & u & b \end{matrix}$$
$$\begin{pmatrix} 1 & 0 & 1/5 & 3/5 & -1/5 & 0 & 48 \\ 0 & 1 & 3/5 & -1/5 & 2/5 & 0 & 84 \\ 0 & 0 & 1 & -1 & 0 & 1 & 60 \\ 0 & 0 & 1/5 & 13/5 & 4/5 & 0 & 708 \end{pmatrix}$$

This leads us to the conclusion that the shop should only focus on making goods A and B, and not make any of good C. This will lead to a profit of 708 dollars. This also provides us with how many of these goods we should make. This is found in the last column. We interprate that we should make 48 of good A and 84 of good B. Thus, we have solved the problem.

Now we can apply the Simplex Method to Game Theory. If we let $\mathbf{c} = \mathbf{1}$ and $\mathbf{b} = \mathbf{1}$ we get

$$\begin{cases} max \mathbf{1}^T\mathbf{x} \\ sub A\mathbf{x} \leq \mathbf{1}, \mathbf{x} \geq \mathbf{0} \end{cases}$$

Let $\mathbf{q}$ be an optimal row strategy. Since it is an optimal row strategy, then we know $A\mathbf{q} \leq v\mathbf{1}$ for the value of the game v. Now if we set $\mathbf{x} = (1/v)\mathbf{q}$, it matches up with the above form. Therfore, if we let A in the simplex method equal the payoff matrix of a game with $\mathbf{b} = \mathbf{1}$ and $\mathbf{c} = \mathbf{1}$, we can use the Simplex Method to get the solution of a game.

Applying this to Major League Baseball

How well a team does usually depends on their starting pitcher. In a playoff series, each team always wants their best pitcher to pitch, as that would give them the best chance of winning the game, and then the series. But what if the opposing starting pitcher is very good? For example, in a series between the San Francisco Giants and the Los Angeles Dodgers, would it be a better strategy for the Giants to start their best pitcher game one against Clayton Kershaw, who according to our rankings is the number 4 best pitcher in the league, or wait until he has a more favorable match up? We can think of this as a game.

The Los Angeles Dodgers had four pitchers on their playoff roster that we had data on. The San Francisco Giants had six pitchers. To simplify, we assumed that no pitcher could pitch on short rest, and the series was five games long. There are 24 possible Dodger rotations of length four, whereas there are 1080 possible Giant rotations. Using the pitcher battle we used above it the Rankings section, we were able to check if a Giant rotation would beat a Dodger rotation, and visa versa. This was done by using the pitcher rotation between pitcher i of the Giants and pitcher i of the Dodgers. We then took the average of the five scores for the given rotations. If the score was exactly .5, then it was viewed as essentially a tie. If the score was less than .5, then that meant the Giant's rotation would win. If the score was greater than .5, then the Dodger's would win. Using this as the entries, we generated a giant 24 by 360 matrix. This matrix is the payoff matrix of the game.

Using a program in Python (Appendix C), we were able to find the value of the game, and the optimal strategies for both teams using the simplex method. The value was about .29, meaning that the game was in the Giant's favor. The optimal strategy was for the Giants to use six different rotations. The optimal strategy for the Dodgers was very similar, as it called for six different rotations. For the Dodgers, the six different rotations along with the percentage of time it should be used were were:

1.)Hyun Jin Ryu, Clayton Kershaw, Dan Haren, Zack Greinke, and Hyun Jin Ryu 19.05

2.) Hyun Jin Ryu, Zack Greinke, Clayton Kershaw, Dan Haren, and Hyun Jin Ryu 26.19

3.) Hyun Jin Ryu, Zack Greinke, Dan Haren, Clayton Kershaw, and Hyun Jin Ryu 7.14

4.) Hyun Jin Ryu, Dan Haren, Clayton Kershaw, Zack Greinke, and Hyun Jin Ryu 14.29

5.) Hyun Jin Ryu, Dan Haren, Zack Greinke, Clayton Kershaw, and Hyun Jin Ryu 11.9

6.) Dan Haren, Clayton Kershaw, Zack Greinke, HyunJin Ryu, and Dan Haren 21.43

For the Giants, the six rotations along with the percentage of time it should be used were:

1.) Madison Bumgarner, Yusmeiro Petit, Tim Hudson, Ryan Vogelsong, and Tim Lincecum 14.29

2.) Madison Bumgarner, Tim Hudson, Ryan Vogelsong, Yusmeiro Petit, and Tim Lincecum 14.29

3.) Jake Peavy, Madison Bumgarner, Tim Lincecum, Yusmeiro Petit, and Jake Peavy 14.29

4.) Tim Hudson, Tim Lincecum, Madison Bumgarner, Jake Peavy, and Tim Hudson 14.29

5.) Ryan Vogelsong, Madison Bumgarner, Jake Peavy, Yusmeiro Petit, and Ryan Vogelsong 19.05

6.) Ryan Vogelsong, Madison Bumgarner, Yusmeiro Petit, Jake Peavy, and Ryan Vogelsong 19.05

Conclusion

In conclusion, we were able to come up with a ranking for all Major League starting pitchers and a system that tells us which team would win a five game series, and the optimal strategies. We established who the best and worst starting pitchers in the league are. We also came up with a new way to think about pitching rotations in a five game series. All of this was done with math ideas and concepts that are not to hard to understand. It would be very interesting to see if math ideas like these will change the way the game of baseball is played in the near future.

Appendix A: Small Example of Pitcher Ranking Code in Python

```
import re
import numpy as np
import pickle

f = open("PitcherDataHtml.txt", "r")
html = f.read()
f.close()

pitchers = sorted(set(re.findall("\">(\w+\s\w+)</a></td>", html)))
teams = sorted(set(re.findall("\">\s(.+)\sBatters", html)))

search_results =
sorted(set(re.findall("\">(\w+\s\w+)</a></td>\n.+\n.+\n.+\n.+\n.+\">\s(.+)\sBatters.+\n.+\n.+\n.+\n.+\n.+\n.+\n.+\n<td>.(\d+)",html)))

def pitcher_battle(pitcher1, pitcher2):
   count = 0
   for team in teams:
      ave1 = [data[2] for data in search_results if data[0] == pitcher1 and data[1] == team]
      ave2 = [data[2] for data in search_results if data[0] == pitcher2 and data[1] == team]
      if ave1 != [] and ave2 != []:
         if ave1[0] < ave2[0]: count += 1
         if ave1[0] > ave2[0]: count -= 1
   if count == 0:
      return .5
   if count > 0:
      return 1
   return 0

A = [[pitcher_battle(p1,p2) for p2 in pitchers[30:34]] for p1 in pitchers[30:34]]
evalues, evectors = np.linalg.eig(np.array(A))
max_index = np.argmax(evalues)

perron_vector = [a.real for a in evectors[:,max_index]]

sign = 1
if perron_vector[0] < 0:
   sign = -1
max_entry = max(perron_vector)
scaled_perron = [sign*a/max_entry for a in perron_vector]
scores = sorted([[scaled_perron[0:4], pitchers[30:34]]])
scores.reverse()

for thing in scores:
   print(thing)
print(A)
```

Appendix B: Pitcher Ranking Code in Python

```
import re
import numpy as np
import pickle

f = open("PitcherDataHtml.txt", "r")
html = f.read()
f.close()

pitchers = sorted(set(re.findall("\">(\w+\s\w+)</a></td>", html)))
teams = sorted(set(re.findall("\">\s(.+)\sBatters", html)))

search_results =
sorted(set(re.findall("\">(\w+\s\w+)</a></td>\n.+\n.+\n.+\n.+\n.+\n.+\">\s(.+)\sBatters.+\n.+\n.+\n.+\n.+\n
.+\n.+\n.+\n<td>.(\d+)",html)))

def pitcher_battle(pitcher1, pitcher2):
   count = 0
   for team in teams:
      ave1 = [data[2] for data in search_results if data[0] == pitcher1 and data[1] == team]
      ave2 = [data[2] for data in search_results if data[0] == pitcher2 and data[1] == team]
      if ave1 != [] and ave2 != []:
         if ave1[0] < ave2[0]: count += 1
         if ave1[0] > ave2[0]: count -= 1
   if count == 0:
      return .5
   if count > 0:
      return 1
   return 0

A = [[pitcher_battle(p1,p2) for p2 in pitchers] for p1 in pitchers]
evalues, evectors = np.linalg.eig(np.array(A))
max_index = np.argmax(evalues)

perron_vector = [a.real for a in evectors[:,max_index]]

sign = 1
if perron_vector[0] < 0:
   sign = -1
max_entry = max(perron_vector)
scaled_perron = [sign*a/max_entry for a in perron_vector]

scores = sorted([[scaled_perron[i], pitchers[i]] for i in range(len(pitchers))])
scores.reverse()

for thing in scores:
   print(thing)
```

Appendix C: Game Theory and Simplex Method Code in Python

```
  import re
import numpy as np
import pickle
import itertools

f = open("PitcherDataHtml.txt", "r")
html = f.read()
f.close()

pitchers = sorted(set(re.findall("\">(\w+\s\w+)</a></td>", html)))
teams = sorted(set(re.findall("\">\s(.+)\sBatters", html)))

search_results = 
sorted(set(re.findall("\">(\w+\s\w+)</a></td>\n.+\n.+\n.+\n.+\n.+\">\s(.+)\sBatters.+\n.+\n.+\n.+\n.+\n.+\n.+\n.+\n<td>.(\d+)",html)))

def pitcher_battle(pitcher1, pitcher2):
  count = 0
  for team in teams:
    ave1 = [data[2] for data in search_results if data[0] == pitcher1 and data[1] == team]
    ave2 = [data[2] for data in search_results if data[0] == pitcher2 and data[1] == team]
    if ave1 != [] and ave2 != []:
      if ave1[0] < ave2[0]: count += 1
      if ave1[0] > ave2[0]: count -= 1
  if count == 0:
    return .5
  if count > 0:
    return 1
  return 0


l = [124,82,167,186,166,154,33,188,74,41]
pSF = [pitchers[124], pitchers[83], pitchers[167], pitchers[186], pitchers[166], pitchers[154]]
pLA = [pitchers[33], pitchers[188], pitchers[74], pitchers[41]]

e1 = [list(x) for x in itertools.permutations(pSF, 5)]
e2 = [list(x) for x in itertools.permutations(pSF, 4)]
e2 = [x + [x[0]] for x in e2]

e = e1 + e2

d = [list(x) for x in itertools.permutations(pLA, 4)]
d = [x + [x[0]] for x in d]

def rotation_battle(rot1, rot2):
  a = sum([pitcher_battle(i, j) for i, j in zip(rot1,rot2)])/5.
```

```
    return a

'''
pSFvspLA = [[rotation_battle(i,j) for i in e] for j in d]


f = open("RotationBattle.pickle", "wb")
pickle.dump(pSFvspLA, f)
f.close()
'''
f = open("RotationBattle.pickle", "rb")
pSFvspLA = pickle.load(f)
f.close()




# A 0 is good for SF.  A 1 is good for LA.  Row player is LA.

HEY = np.array(pSFvspLA)
HEYt = np.matrix.transpose(HEY*(-1))



# This is python code implementing the Simplex algorithm for games.
# This code will find one extreme optimal mixed strategy to a matrix game.
#
# CHANGE MATRIX A BELOW TO BE YOUR GAME MATRIX
# Enter the decimal approximation for rational numbers; for example, use .5, not 1/2.

# The output is a row strategy and value.

def gamesolution(A):
    m = min(A[0] + [0]) #used to make value > 0

    T = [[j-m+1 for j in A[i]]+[0]*i+[1]+[0]*(len(A)-i-1)+[1] for i in range(len(A))]
    T += [[-1]*len(A[0]) + [0]*(len(A)+1)] #initial tableau

    while min(T[-1]) < 0:
        c = T[-1].index(min(T[-1])) #pivot column

        ratios = []
        for i in T[:-1]:
            if i[c] > 0: ratios += [i[-1]/float(i[c])]
            else: ratios += [pow(10,20)] # I am considering pow(10,20) == infinity
        r = ratios.index(min(ratios)) #pivot row

        T[r] = [i/float(T[r][c]) for i in T[r]]
```

```
        for i in range(len(T)):
            if i != r:
                T[i] = [T[i][j] - T[i][c]*T[r][j] for j in range(len(T[i]))] #pivot
                for j in range(len(T[i])): #watch out for roundoff errors
                    if abs(T[i][j]) < pow(10,-8): T[i][j] = 0

    v = 1/float(T[-1][-1])

    row = [v*i for i in T[-1][-len(A)-1:-1]]

    return [row,v+m-1]

for i in gamesolution(HEY):
    print (i)

for i in gamesolution(HEYt):
    print (i)



print(d[13], d[14], d[15], d[16], d[17], d[18], e[65], e[95], e[780], e[984], e[1021], e[1026])
```