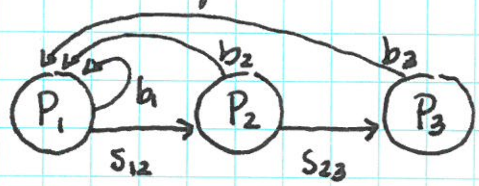
lecture 18 Math Models

Project 4 Due Today-ish
~~- Friday morning 9am~~
• ~~coin_flip.pdf has step by step info for building a table.~~

Last time: Stages States and Classes.

Ex Three State - Age Based - Female only model.



$$P_1(n+1) = b_1 P_1(n) + b_2 P_2(n) + b_3 P_3(n)$$
$$P_2(n+1) = s_{12} P_1(n)$$
$$P_3(n+1) = s_{23} P_2(n)$$

IN MATRIX FORM $\vec{P}_{n+1} = A\vec{P}_n$

where $\vec{P} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$ $A = \begin{bmatrix} b_1 & b_2 & b_3 \\ s_{12} & 0 & 0 \\ 0 & s_{23} & 0 \end{bmatrix}$

Our problem is reduced to simple matrix multiplication!

NOTE: A is sometimes called a "Transition Matrix" or "Leslie Matrix".

When we did the excel spread sheet last time (population_stages_example.xls) no mater what our initial population our final population % was the same

About ~ 40.20% | 32.99% | 26.81%
P_1 P_2 P_3

A really cool thing about Matrices ... eigenvectors and eigenvalues!!

We can use these to predict the long term population %.

The general Idea ... comes from Linear Algebra ...

In single state models ... with basic growth rates we had

$P(n+1) = R\,P(n)$ and we can find a closed form solution to this ...

$P(n) = R^n P(0)$ R was a characteristic growth rate, and an important value.

Our Matrix System now is ...

$(\ast)\ \vec{P}_{n+1} = A\vec{P}_n$ this looks similar so we seek to have A give us some characteristic values.

look for solutions like : $\vec{P}_n = \lambda^n \vec{v}$ plug this into $(\ast)$

λ = characteristic value of growth/decay

$\vec{v}$ = some vector

$$\vec{P}_{n+1} = \lambda^{n+1}\vec{v} = A\vec{v} = A\lambda^n\vec{v}$$

so $\lambda[\lambda^n \vec{v}] = A[\lambda^n \vec{v}]$

or $\lambda\vec{v} = A\vec{v} \Rightarrow (A - \lambda I)\vec{v} = 0$

if we can fully satisfy this equation then we can write our soln as ...

we look for pairs of $\lambda_m \vec{v}_n$ that satisfy this equation.

$$\vec{P}(n) = C_1\lambda_1^n\vec{v}_1 + C_2\lambda_2^n\vec{v}_2 + \cdots$$

→ solution where we don't need matrix mult.

sum of all possible solns.

BOTTOM LINE...

The largest eigenvalue λ → dominant eigenvalue ... as n gets big (aka time marches on) this term is MOST important.

The NORMALIZED eigenvector $\vec{v}$ associated with the largest eigenvalue tells us the long term % breakdown of the population.

IN our class ... we can use Matlab or Octave or Sage ...
to solve for eigenvalues and eigenvectors.

Matlab Example ... Human Population Example

$b_1 = 0.4271$ $S_{12} = 0.9924$
$b_2 = 0.8498$ $S_{23} = 0.9826$
$b_3 = 0.1273$

INTRO — BIRD POPULATION ...

You try to build a state based model ...
for the first part use either spreadsheet or matlab.

EXAMPLE. Modeling a population of birds.

Assumptions – we will follow nesting bird pairs so population units are "pairs of birds".

– all growth and survival rates are in terms of pairs of birds...
1.5% birth rate ⇒ each bird pair will produce ~ 1.5 eggs.

– the oldest birds live to be five years old
– we will have 6 stages: each stage = 1 year.

Variable Stage	P_1 Egg	P_2 Baby	P_3 Young Adult	P_4 Adult	P_5 Mature Adult	P_6 Elderly
Survival Rate	50%	70%	90%	70%	30%	0%
Birth Rate	0%	0%	.5%	1%	1.5%	0%

1. Draw a STATE DIAGRAM for this system.

2. Write down your RECURRENCE RELATIONS for this system.

3. Write the system in matrix form.

4. Assume Initial Population Values:
$P_1(0) = 30$ $P_2(0) = 20$ $P_3(0) = 40$ $P_4(0) = 40$ $P_5(0) = 30$ $P_6(0) = 15$

5. Build a Computer model for this system.

You can use either a spreadsheet.

<u>OR</u>

Edit my matlab file and use matlab.

6. Can you find the eigenvalues and eigenvectors for this system.

– Write down and hand in parts 1 – 3
– Email part 5 to yourself and your group members!

We will learn
how to edit
this in class!

```
function [] = untitled()

NUMSTAGE=3; % This is the number of stages that your population has.
NUMSTEPS=20; % This is how many time steps you want to do
% Initial Population for Each Stage
P1=30;
P2=30;
P3=30;
% If I had more stages I would add another initial population for each
% stage.
%P4=
%P5=
%P6=

% Build the population matrix...
A=zeros(NUMSTAGE,NUMSTAGE);

% Put numbers into the population matrix row by row...
A(1,:)=[0.4271, 0.8498, 0.1273];
A(2,:)=[0.9924, 0,      0     ];
A(3,:)=[0,      0.9826, 0     ];
% IF I had more stages I would add another row for each stage
% A(4,:)=[]
% A(5,:)=[]
% A(6,:)=[]

% Fill in data for the initial population
P(1,1)=P1;
P(1,2)=P2;
P(1,3)=P3;
% If I had more stages I would need more initial population inputs
%P(1,4)=P4;
%P(1,5)=P5;
%P(1,6)=P6;


% A matrix to track time
T(1)=0;

for i=1:NUMSTEPS
    % Each time step we just do a matrix multiplication.
    P(i+1,:)=A*transpose(P(i,:));
    % here we must transpose X so that the matrix multiplication works out!

    % Move our time one step forward
    T(i+1)=T(i)+1;

end


% Print the new matrix
% Anything with out a semicolon at the end will print!
FinalPOP=P(NUMSTEPS+1,:)

% Plot the Population for each stage
for j=1:NUMSTAGE
    plot(T(:),P(:,j), 'ko-','LineWidth',2);
    axis=[0, max(max(P)), 0, max(T)];
    hold on
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%EIGENVALUES - this will automatically calculate and normalize your
%eigenvalues - the largest eigenvalue is associated with the long term
%population percentages!

[V,E]=eig(A);

% Normalize vectors
for j=1:3
    NormV(:,j)=V(:,j)./sum(V(:,j));
end

EigenValues=E

EigenVectors=NormV


%Prompt user to save the new matrix
checkLabels = {'Save Popluation matrix to variable named:'};
varNames = {'POP'};
items = {P};
export2wsdlg(checkLabels,varNames,items,...
             'Save to Workspace');

%Prompt user to save the new matrix
checkLabels = {'Save Eigenvalues to variable named:'};
varNames = {'EIG'};
items = {E};
export2wsdlg(checkLabels,varNames,items,...
             'Save to Workspace');

%Prompt user to save the new matrix
checkLabels = {'Save Normalized Eigenvectors to variable named:'};
varNames = {'EVEC'};
items = {NormV};
export2wsdlg(checkLabels,varNames,items,...
             'Save to Workspace');

end
```