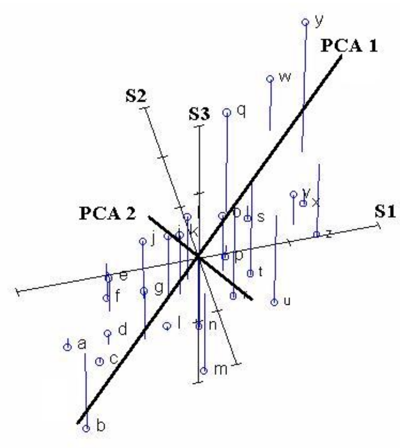
On the Uses of Linear Algebra

Rebecca Weber

Feb 7, 2011



I asked...

What's your favorite application of linear algebra?

- counting walks in graphs
- quantum mechanics
- wavelets
- machine learning
- vector bundles
- principal component analysis
- NCAA rankings
- linear optimization
- audio and image compression
- curve fitting
- rational homotopy
- Google PageRank
- writing computer programs that solve Sudoku puzzles

. . . and Making men feel like men.

Girls=Boys in Math

Researchers for the National Science Foundation have found that boys and girls now perform equally in standardized math tests. What do *you* think?

Luke Casey

Industrial Loom Operator

"But linear algebra was the only thing that ever made me feel like a man."

The Onion, of course.

I would add

- computer graphics

and

- random text generation

which we will discuss today, along with Google PageRank and a brief mention of team rankings.

Use 1: Computation

Computations in linear algebra are easily programmed into a computer. To compute sports rankings, you could make a matrix where rows correspond to teams and columns to statistics, with entries giving the ranking of each team in each measure. Then you might multiply by a weighting vector giving the relative importance of each statistic.

Advantages: easy updating

easy adjustment of weighting vector to see what happens

all team computations done at once

Small example

Godefryd, Mayhew, and Raulfe are ranked on charm, decorum, and wit, and then those rankings are weighted such that decorum and wit are equally valued and each twice as important as charm. Ties are allowed, and in fact Mayhew and Raulfe are tied above Godefryd on decorum, though in charm Godefryd beats Mayhew who beats Raulfe, and in wit Raulfe beats Godefryd who beats Mayhew. Who should get the Gentleman's Award?

$$\begin{array}{c} (G) \\ (M) \\ (R) \end{array} \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} .2 \\ .4 \\ .4 \end{bmatrix} = \begin{bmatrix} 2.2 \\ 2 \\ 1.4 \end{bmatrix}$$

Raulfe!

Use 2: Computer Graphics

Vector graphics refers to representing images by mathematical descriptions of geometric objects, rather than by a collection of pixels on the screen (raster graphics).

Main idea: If we represent points of space in the right way, we can represent all sorts of motions and deformations of shapes by matrix multiplication.

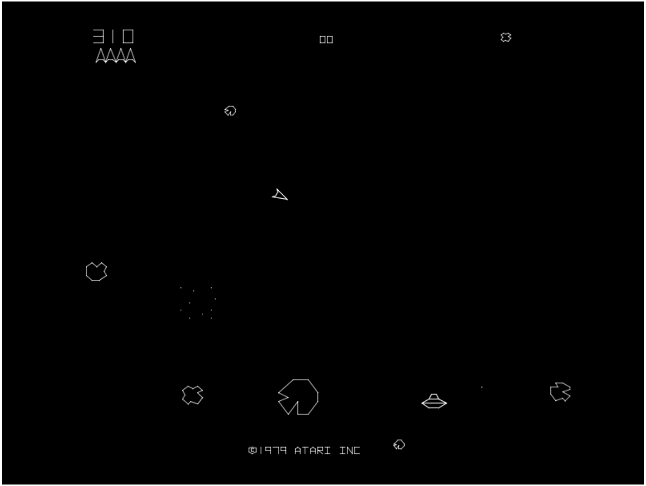
Geometrically, we can represent a polygon by a collection of vertices and the order in which to connect them. This is a lot less data to store in memory than the collection of all the pixels that are to be colored, especially as the polygon gets larger.

If we want to represent other shapes, we need to store:

- What kind of shape it is
- Enough points and distances to fully describe the shape
- (if we have this option) The style and color of the outline
- (if we have this option) The style and color of the inside

For example, for a circle we would need to know it was a circle, then have the center and radius.

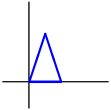
Asteroids



(Atari, 1979)

Manipulating a Ship

Let's represent the ship by a triangle, starting with its vertices at $(0,0),(2,0),(1,3)$.



We could represent this as a matrix of vertices, with the understanding (stored as an additional piece of data) that they are connected in order and the last connects to the first.

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

A matrix of vertices allows us to represent rotation by matrix multiplication, but what about translation? If we wanted to move r units to the right and s units up, we'd have to add:

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix} + \begin{bmatrix} r & r & r \\ s & s & s \end{bmatrix}$$

With a simple modification of the vertex matrix, though, we can represent translation by matrix multiplication using shear transformations (adds a multiple of one coordinate to each of the other coordinates). We just add a dummy coordinate that always holds value 1, and add multiples of that to the x and y coordinates.

Homogeneous Coordinates

In the new scheme, our triangle ship is represented by the matrix

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

Translation by (r, s) is now the product

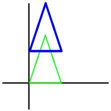
$$\begin{bmatrix} 1 & 0 & r \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} r & 2+r & 1+r \\ s & s & 3+s \\ 1 & 1 & 1 \end{bmatrix}$$

Flying Forward

Suppose we want to fly our ship two units in the direction it is currently facing (up). We need to translate by $(0, 2)$:

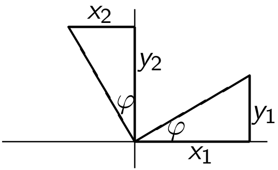
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 1 \\ 2 & 2 & 5 \\ 1 & 1 & 1 \end{bmatrix}$$

The new vertices, $(0, 2), (2, 2), (1, 5)$, give us the new ship image:



Turning in Place

We'll crash into an asteroid if we can't change direction. Let's find the images of the columns of I_2 under counterclockwise rotation at an angle of φ.



$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix}$$

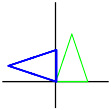
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \end{bmatrix}$$

The first two columns of I_3 will do just the same thing, with a zero in the last place. Our extra coordinate will stay the same. Hence to obtain counterclockwise rotation by φ degrees, we multiply by the matrix

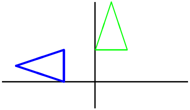
$$\begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For example, for a 90° turn to the left, we would compute:

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -3 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

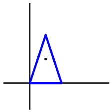


That's not the way ships turn, though! And if we rotate our ship after it drives up 2 units, it's even worse:



Center of Ship vs. Center of Screen

We want our rotation to be about the center of the ship, say, the point halfway between the front and back ends on the line straight back from the tip. In our beginning position that is $(1, 1.5)$.



One solution is to translate, rotate, and translate back: make the center of the triangle the origin, do the rotation as previously, and then move the center back where it belongs. If we move everything by the amount the center requires, everything will stay in its proper positions.

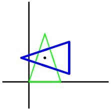
We multiply on the left by each operation's matrix in order. Hence the new vertices are given by $A^{-1}BAM$ where M is the matrix of homogeneous coordinates, A gives translation by $(-1, -1.5)$, B rotation by 90°, and A^{-1} translation by $(1, 1.5)$.

$$A^{-1}BA = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1.5 \\ 0 & 0 & 1 \end{bmatrix}$$

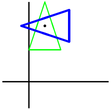
$$= \begin{bmatrix} 0 & -1 & 2.5 \\ 1 & 0 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(A^{-1}BA)M = \begin{bmatrix} 0 & -1 & 2.5 \\ 1 & 0 & 0.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2.5 & 2.5 & -0.5 \\ 0.5 & 2.5 & 1.5 \\ 1 & 1 & 1 \end{bmatrix}$$

The multiplication claims our vertices after rotation should be at $(2.5, 0.5), (2.5, 2.5)$, and $(-0.5, 1.5)$. Let's graph it.



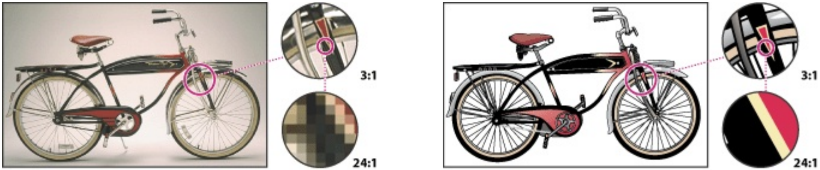
Much better! The corresponding action for our ship after moving forward two units involves translating by $(-1, -3.5)$ and back with the rotation in between. The vertices obtained are $(2.5, 2.5)$, $(2.5, 4.5)$, and $(-0.5, 3.5)$ (the previous vertices with $y + 2$).



Advantages of Vector Graphics

• Vector graphics are device independent – the machine's software turns the description into an image in a device-specific way, but the description itself doesn't need to change.

• Resizing vector objects can be done with perfect quality maintenance. To go from 4×6 to 400×600, you simply multiply by a dilation matrix and then make the connections as usual. A raster image must be sharpened if it is blown up too much.



`www.underwaterphotography.com/PhotoShop/PhotoShop/help.html`

So Why Raster?

A little bit of history: For a brief time, displays were actually vector displays ("calligraphic" or "X-Y"). A beam would trace out the desired image on an otherwise black screen, many times per second.

Modern displays, though, are raster (back to the early 1980s or even before), so a vector graphic must be converted to raster (a bitmap) in order to display. This meant that for two-dimensional games it was often easier to deal with raster graphics from the start.

Additionally, a low-resolution display gives a lot more importance to each individual pixel, and it is not efficient to tweak individual pixels with vector representation.

Back to Vector, in part

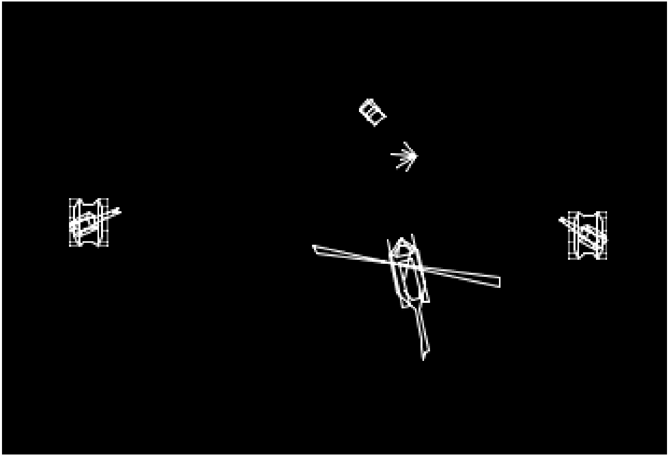
High-resolution displays decrease the need to tweak images at the pixel level, so vector representations can give quality images while maintaining efficiency.

Perhaps more importantly, many games are now 3D. Raster graphics would require representing the whole world as one piece. In these games, though, the player moves around and looks at objects from different angles in real time, and keeping vector representations and rendering only the applicable part is faster.

However, raster representation is still more efficient for textures and patterns, so modern games use a combination of vector graphics for objects and raster graphics for surface detail.

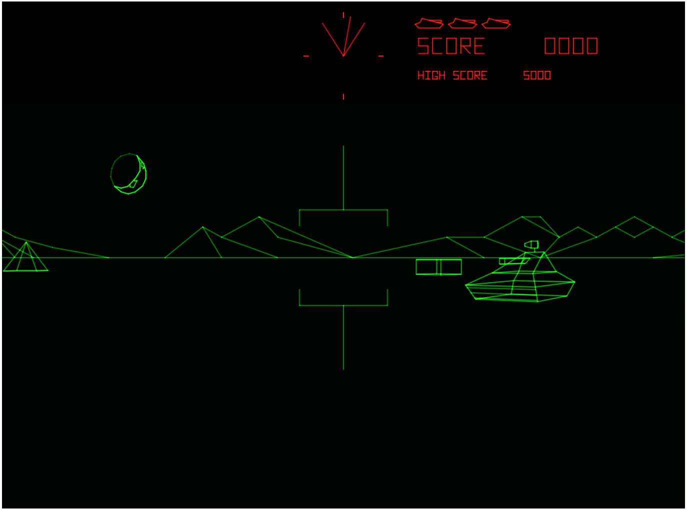
A couple more examples of vector graphics follow.

Armor Attack



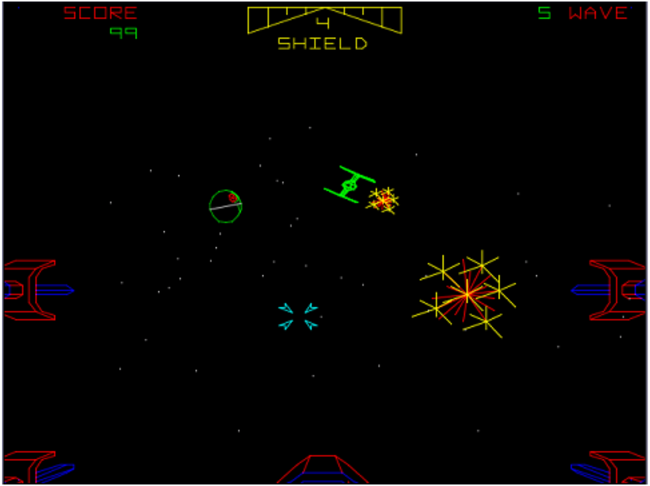
(Cinematronics, 1980)

Battlezone



(Atari, 1980)

Star Wars



(Atari, 1983)

Use 3: Markov chains

Hop, hop, hop! I am too absent-spirited to count;
The loneliness includes me unawares.
And lonely as it is we do not like my little bed.
This is no good. This is not right.
My feet stick out of hiding,
To please the yelping dogs.
The gaps I mean,
No one has a little car.
This one has a yellow hat.
From there to here, From here to help you.
I have it in me so much nearer home
To scare myself with my Ish wish dish.

What Is a Markov Chain?

A *probability vector* is a vector with nonnegative entries that add up to 1. A *stochastic matrix* is a square matrix whose columns are all probability vectors.

A *Markov chain* is a sequence of probability vectors $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ together with a stochastic matrix P such that

$$\mathbf{x}_1 = P\mathbf{x}_0, \quad \mathbf{x}_2 = P\mathbf{x}_1, \quad \ldots, \quad \mathbf{x}_{n+1} = P\mathbf{x}_n, \quad \ldots$$

What can these mean? Essentially, they represent a series of states of a system and a uniform way to change from one state to the next. The important part of the set-up is that wherever you happen to be, you get to the next step in exactly the same way.

An Example

We might imagine a laboratory setup where a lab rat has three levers available to it. Lever I results in a mild electric shock. Lever II makes a loud noise. Lever III causes a piece of food to appear. Originally the rat has equal probability of choosing each of the three levers.

If it chooses lever I, it has only a 10% probability of choosing that lever the next time. If it chooses lever II, it has a 25% probability of choosing that lever the next time. If it chooses lever III, it has a 80% probability of choosing that lever the next time. In each case the probability of choosing the remaining two levers is equally split.

How do we represent this as a Markov chain?

Example, Continued

We are making the simplifying assumption that what the rat does next only depends on the most recent lever pressing – no longer-term memory.

The n^{th} state vector represents how likely it is that the rat presses each lever in the n^{th} round. Therefore,

$$\mathbf{x}_0 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

The next time, we have a linear combination of vectors. A third of the time, the rat will press lever 1 and the next vector will be $(0.1, 0.45, 0.45)$. A third of the time, it will press lever II and the next vector will be $(0.375, 0.25, 0.375)$. The remaining third of a time it will press lever III and the next vector will be $(0.1, 0.1, 0.8)$.

Example, Part 3

We want to take the weighted average of those vectors, the linear combination with weights corresponding to the entries in the previous vector. In other words, we want the matrix-vector product

$$\begin{bmatrix} 0.1 & 0.375 & 0.1 \\ 0.45 & 0.25 & 0.1 \\ 0.45 & 0.375 & 0.8 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \begin{bmatrix} 0.18975 \\ 0.264 \\ 0.53625 \end{bmatrix}$$

This gives the linear combination we want, and notice that the weights will always be the probability vector from the previous step and the vectors being combined will always be the columns of the matrix above. That is, that matrix is the stochastic matrix for this Markov chain.

Return of Son of Example

What are the rat's probabilities of choosing each of the three levers on its third press? On its fourth press? We multiply $\mathbf{x}_0$ by the stochastic matrix to get $\mathbf{x}_1$, then again to get $\mathbf{x}_2$, and so forth.

$$\mathbf{x}_1 = \begin{bmatrix} 0.18975 \\ 0.264 \\ 0.53625 \end{bmatrix}, \ \mathbf{x}_2 = \begin{bmatrix} 0.1716 \\ 0.205 \\ 0.6134 \end{bmatrix}, \ \mathbf{x}_3 = \begin{bmatrix} 0.1554 \\ 0.1898 \\ 0.6448 \end{bmatrix}$$

On press 3, $\mathbf{x}_2$ is the appropriate vector. The rat has a more than 60% chance of pressing the food lever, and about a 17% and 20% chance of pressing the shock and noise levers, respectively. On press 4 the food probability has gone up a little to almost 65%, and the others have dropped to about 15% and 19%.
In fact if you keep multiplying you find the vectors start looking very much the same, approximately $(0.148, 0.177, 0.665)$.

Serious example

How does Google decide which webpages should show up on page 1 of your search results and which on page 10?

> *In this paper, we have taken on the audacious task of condensing every page on the World Wide Web into a single number, its PageRank.*

Page, Brin, Motwani, and Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Stanford Technical Report, 1999.

Setting up a Matrix

PageRank begins with a matrix A with rows and columns corresponding to webpages (row i and column i both correspond to the same page). The entries are determined by whether there is a link from the row page to the column page, and how many links are on the row page.

Let $A = [a_{uv}]$. If there is no link from page u to page v, $a_{uv} = 0$. Otherwise, let n_u be the total number of pages that u links to, and set $a_{uv} = 1/n_u$.

Dividing by the number of links on the page decreases the effect of pages that are nothing but lists of links, which might otherwise artificially inflate a page's rank.

Using the Matrix

The PageRank vector $\mathbf{r}$ has as entries the ranks of every webpage. It is the equilibrium vector of the matrix A, the unique vector such that $A\mathbf{r} = \mathbf{r}$.

Why the equilibrium vector? We would like a page that itself has high rank to share that high rank with the pages it links to (having a link on the main Dartmouth website should count for more than having a link on the linear algebra course website). We do this by taking powers of A until the multiplication no longer changes the ranks of webpages.

Technical Issues

- The matrix A is not necessarily a stochastic matrix to begin with. To make sure the total rank of all pages stays constant (e.g., stays at 1 so the vector $\mathbf{r}$ is a probability vector) we must multiply by a scaling factor c, so $\mathbf{r}$ actually satisfies $cA\mathbf{r} = \mathbf{r}$.
- Technical problems are introduced by circular pointers (say, two webpages that point to each other but nowhere else) and dangling links (links to pages which have no outgoing links, or to pages Google has not downloaded yet). These are fairly straightforward to solve.
- Most entries of A will be 0. This means much storage space is saved by remembering not A, but instead just its nonzero entries and their positions.

More Technical Issues

- The equilibrium vector of this matrix is found by multiplying repeatedly by the matrix, rather than finding a probability vector in the null space of $A - I$ (the usual method). This is for three reasons:
  1. Because the matrix is very sparse, multiplication is quicker than solving the enormous system of equations.
  2. An approximation of the rank vector is good enough for Google's purposes.
  3. When Google updates their information, the matrix changes. The previous rank vector is still a decent approximation to the rank, so using it as the initial state and multiplying several times by the matrix allows for a fast update of the rank vector.
- By assigning the link weights more cleverly than $1/n_u$, one can get better results and decrease the ability of people to artificially boost their pages' rankings.

Silly example

One can create toys with Markov chains to generate nonsense or parody text. The opening Markov slide is text generated by a Markov algorithm from Dr. Seuss's *The Sneeches* and *One Fish, Two Fish, Red Fish, Blue Fish*, and several poems of Robert Frost.

Running that algorithm on a single text will generate something that cosmetically or stylistically resembles the original text, but upon reading reveals itself to have nonsensical twists and turns (this may or may not differentiate it from the original).

How Does It Work?

The pieces we are working with are *n*-tuples of words found in the source text. Suppose we are working with pairs (2-tuples) and the following text, ignoring punctuation.

> *I scream, you scream, we all scream for ice cream!*

Then our 2-tuples are, in order of appearance:

I scream, scream you,
you scream, scream we,
we all, all scream,
scream for, for ice,
ice cream

We could also wrap around at the end, adding one more pair.

The Stochastic Matrix

A probability vector has entries corresponding to the tuples, so in our example it would be a vector in $\mathbb{R}^9$ (or $\mathbb{R}^{10}$ with wrap-around).

The stochastic matrix represents the probability for each possible word w_{n+1} if the current n-tuple of words is $w_1, w_2, \ldots, w_n$. It does so by assigning a probability of 0 to each n-tuple that doesn't begin $w_2, \ldots, w_n$, and assigning probabilities to the tuples that *do* begin appropriately according to their frequency in the text.

[Technical note: the probability is the number of times the $(n+1)$-tuple $w_1, \ldots, w_{n+1}$ appears divided by the number of times the n-tuple $w_1, \ldots, w_n$ appears. However, for our example we will simply use the number of time w_2, w_3 appears divided by the number of times w_2 appears, because if we include w_1 our hands are tied: the only phrase we can produce is the original.]

Recombining Tuples

In our example the stochastic matrix will have a lot of zeros (it will be *sparse*), because not many of the pairs overlap. In fact, almost every column (each of which corresponds to one starting pair) will be a column of the identity matrix, because there is only one word that could follow. The columns corresponding to pairs ending in "scream" will have nonzero entries in the rows corresponding to "scream for", "scream we", and "scream you", each with value $\frac{1}{3}$.

Here's the algorithm: Start with a vector with a 1 in the first position and 0s elsewhere (so the new text begins with the same tuple as the original). Multiply by the stochastic matrix to get a probability vector. Use a random value to pick the next tuple using the new vector. Then discard that vector and start anew with a vector that has a 1 in the position corresponding to the tuple you just picked and 0s elsewhere.

A New Jingle

Since our matrix is simple, we can explain our procedure without writing it out. Start with the pair "I scream". Pick a random number from 1 to 100: 41. This is between 33 and 66 so the next pair is "scream we". Now we're fixed for a while: "we all", "all scream". Now pick another random number: 11. This is below 33 so the next pair is "scream you". Fixed for one pair: "you scream". Another random number: 71. This is above 66 so we pick "scream for". Now we're fixed to the end: "for ice", "ice cream". No pair begins with "cream" (unless we wrap around) so we've hit a dead end.

What did we get? Inserting punctuation, the new phrase is

I scream, we all scream, you scream for ice cream!

A Better Example

Our toy example isn't so exciting. A colleague of mine combined an early 20th century cookbook with a text on semiotics. The output had to be significantly culled, but yielded gems such as the following:

> *Make a cream of one paradigm (e.g. a particular paradigm rather than a workable alternative was used in a dripping pan; put into pie plates; grease the tops of loaves over with butter. This will make it equal.*
>
> *[In this recipe, the term "mango" refers to its socio-cultural and personal associations (ideological, emotional etc.). Roland Barthes argued that such paired paradigms consist of an 'unmarked' and a tablespoon of whole cloves, and nutmeg to taste.*

. . . the Sneetches forgot about stars and whether
They had stars
And the ground
Like girls on hands and knees that throw their hair
Before them over their heads to dry in the air, they paraded about.
And they seem not to break; though once they are the worst.
But now, how in the house. At our house we play out back.
We play a game called ring the Gack. Would you like to box.
How I like to hop all day and night. From right to left and left to right and then. . .
Hop, hop! I am a Yop
All I like to have him in the house.
At our house we open cans.
We have to use a spell to make them balance:
'Stay where you are until our backs are turned!'

Resources

Markov Text Pages: www.eblong.com/zarf/markov/

www.cs.bell-labs.com/cm/cs/pearls/sec153.html

More on vector images:
www.intmath.com/Vectors/Vector-art.php

PageRank paper is available from
dbpubs.stanford.edu/pub/1999-66

A PageRank example: www.mathworks.com/company/
newsletters/news_notes/clevescorner/oct02_cleve.html

Cover picture from http://ordination.okstate.edu/PCA.htm

Onion:
www.theonion.com/content/amvo/girls_boys_in_math

Random numbers obtained from www.random.org

All video game images taken from Wikipedia, en.wikipedia.org

These slides and more links at
www.math.dartmouth.edu/~rweber/linear/