The product $Ax$

**Definition:** If $A$ is an $m \times n$ matrix, with columns $a_1, ..., a_n$, and if $x$ is in $\mathbb{R}^n$, then the product of $A$ and $x$, denoted by $Ax$, is the linear combination of the columns of $A$ using the corresponding entries in $x$ as weights; that is,

$$Ax = [a_1, a_2, \ldots, a_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1a_1 + x_2a_2 + \cdots + x_na_n$$

- $Ax$ is defined only if the number of columns of $A$ equals the number of entries in $x$.

**Example:**

Let $A = \begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & -2 \\ 0 & -2 & 3 \end{bmatrix}$ and $x = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ Then:

$$Ax = 2 \times \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ 0 \\ -2 \end{bmatrix} + 3 \times \begin{bmatrix} 2 \\ -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \\ 11 \end{bmatrix}$$

- $Ax$ is the Matrix-by-vector product of $A$ by $x$.
- 'matvec'

What is the cost (operation count) of a 'matvec'? 

**Properties of the matrix-vector product**

**Theorem:** If $A$ is an $m \times n$ matrix, $u$ and $v$ are vectors in $\mathbb{R}^n$, and $\alpha$ is a scalar, then

1. $A(u + v) = Au + Av$;
2. $A(\alpha u) = \alpha (Au)$

☐ Prove this result using only the definition (columns).

☐ Prove that for any vectors $u, v$ in $\mathbb{R}^n$ and any scalars $\alpha, \beta$ we have

$$A(\alpha u + \beta v) = \alpha Au + \beta Av$$
**Row-wise matrix-vector product**

- (in the form of an exercise)

- Suppose you have an $m \times n$ matrix $A$ and a vector $x$ of size $n$, show how you can compute an entry of the result $y = Ax$, without computing the others. Use the following example.

**Example:**

Let $A = \begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & -2 \\ 0 & -2 & 3 \end{bmatrix}$ and $x = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$. Let $y = Ax$

- How would you compute $y_2$ (only)?
- Cost?
- General rule or process?
- Matlab code?

---

**The matrix equation $Ax = b$**

- We can now write a system of linear equations as a vector equation involving a linear combination of vectors.

- For example, the system
  
  \[
  \begin{align*}
  x_1 + 2x_2 - x_3 &= 4 \\
  -5x_2 + 3x_3 &= 1
  \end{align*}
  \]

  is equivalent to
  
  \[
  x_1 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}
  \]

  The linear combination on the left-hand side is a matrix-vector product $Ax$ with:

  \[
  A = \begin{bmatrix} 1 & 2 & -1 \\ 0 & -5 & 3 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
  \]

- So: Can write above system as $Ax = b$ with $b = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$

---

**Existence of a solution**

- The equation $Ax = b$ has a solution if and only if $b$ can be written as a linear combination of the columns of $A$

**Theorem:** Let $A$ be an $m \times n$ matrix. Then the following four statements are all mathematically equivalent.

1. For each $b$ in $\mathbb{R}^m$, the equation $Ax = b$ has a solution.
2. Each $b$ in $\mathbb{R}^m$ is a linear combination of the columns of $A$.
3. The columns of $A$ span $\mathbb{R}^m$
4. $A$ has a pivot position in every row.
Proof

First: 1, 2, 3 are mathematically equivalent. They just restate the same fact which is represented by statement 2.

So, it suffices to show (for an arbitrary matrix $A$) that (1) is true iff (4) is true, i.e., that (1) and (4) are either both true or false.

Given $b$ in $\mathbb{R}^m$, we can row reduce the augmented matrix $[A|b]$ to reduced row echelon form $[U|d]$.

Note that $U$ is the rref of $A$.

If statement (4) is true, then each row of $U$ contains a pivot position, and so $d$ cannot be a pivot column.

So $Ax = b$ has a solution for any $b$, and (1) is true.

If (4) is false, then the last row of $U$ is all zeros.

Let $d$ be any vector with a 1 in its last entry. Then $[U|d]$ represents an inconsistent system.

Since row operations are reversible, $[U|d]$ can be transformed back into the form $[A|b]$ for a certain $b$.

The new system $Ax = b$ is also inconsistent, and (1) is false.

Application: Markov Chains

Example: The annual population movement between four cities with an initial population of 1M each, follows the pattern shown in the figure: each number shows the fraction of the current population of city $X$ moving to city $Y$. Migrations $A \leftrightarrow C$ and $B \leftrightarrow D$ are negligible.

Is there an equilibrium reached?

If so what will be the population of each city after a very long time?

Let $x^{(t)}$ = population distribution among cities at year $t$ [starting at $t = 0$] - no pop. growth is assumed.

Express one step of the process as a matrix-vector product:

$$x^{(t+1)} = Ax^{(t)}$$

What is $A$? What distinct properties does it have?

Do one step of the process by hand.

“Iterate” a few steps with matlab (40-50 steps)

At the limit $Ax = x$, so $x$ is the solution of a 'homogeneous' linear system. Find all possible solutions of this system. Among these which one is relevant?

Compare with the solution obtained by “iteration”
Application: Leontief Model [sec. 1.6 of text]

- Equilibrium model of the economy
- Suppose we have 3 industries only [reality: hundreds]:
  - Coal
  - Electric
  - Steel
- Each sector consumes output from the other two (+itself) and produces output that is in turn consumed by the others.

<table>
<thead>
<tr>
<th>Distribution of Output from:</th>
<th>Coal</th>
<th>Electric</th>
<th>Steel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purchased by</td>
<td>Coal</td>
<td>Electric</td>
<td>Steel</td>
</tr>
<tr>
<td>.0</td>
<td>.4</td>
<td>.6</td>
<td></td>
</tr>
<tr>
<td>.6</td>
<td>.1</td>
<td>.2</td>
<td></td>
</tr>
<tr>
<td>.4</td>
<td>.5</td>
<td>.2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Problem: Find production quantities (called prices in text) of each of the 3 goods so that each sector’s income matches its expenditure

Expense for Coal: \(0.4p_E + 0.6p_S\) so we must have

\[p_C = 0.4p_E + 0.6p_S \rightarrow p_C - 0.4p_E - 0.6p_S = 0\]

Similar reasoning for the other 2.

In the end: Linear system of equations that is ‘homogeneous’ (RHS is zero).

Use matlab to find general solution [Hint: Find the rref form first]

Application: Google’s Page rank

Note: Read this to prepare for HW2!

- Idea is to put order into the web by ranking pages by their importance..
- Install the google-toolbar on your laptop or computer
  
  http://toolbar.google.com/
- Tells you how important a page is...
- Google uses this for searches..
- Updated regularly..
- Still a lot of mystery in what is in it..

Page-rank - explained

Main point: A page is important if it is pointed to by other important pages.

- Importance of your page (its PageRank) is determined by summing the page ranks of all pages which point to it.
- Weighting: If a page points to several other pages, then the weighting should be distributed proportionally.
- Imagine many tokens doing a random walk on this graph:
  - \((\delta/n)\) chance to follow one of the \(n\) links on a page,
  - \((1 - \delta)\) chance to jump to a random page.
  - What’s the chance a token will land on each page?
- If \(\text{www.cs.umn.edu/~saad}\) points to 10 pages including yours, then you will get 1/10 of the credit of my page.
Page-Rank - definitions

- Build a 'Hyperlink' matrix $H$ defined as follows:

  "every entry $h_{ij}$ in column $j$ is zero except when $i$ is one of the links from $j$ to $i$ in which case $h_{ij} = 1/k_j$ where $k_j =$ number of links from $(j)$"

- Defines a (possibly huge) Hyperlink matrix $H$

  $h_{ij} = \begin{cases} 
  \frac{1}{k_j} & \text{if } j \text{ points to } i \\
  0 & \text{otherwise}
  \end{cases}$

- Will see to distinct cases:

  - $\delta = 1$ (called undamped)
  - $0 < \delta < 1$ (called damped)

  $\delta$ is a called a 'damping' parameter close to 1 – e.g. 0.85

General case: $0 < \delta < 1$

- Assumption: token has
  - $\delta/k_j$ chance of jumping to one of the $k_j$ links from $j$
  - $1 - \delta$ chance to go to a random page

  We wish to say next jump land in node $i$ with a 'probability' of:

  \[
  (1 - \delta) + \delta h_{ij}
  \]

  Don't add-up to 1

  - Let $\rho_1, \rho_2, \ldots, \rho_n$ be $n$ measures of importance for nodes $1, 2, \ldots, n$. [think of them as 'votes' or likelihoods of being visited]
  - Google page-rank defines the $\rho_i$'s by the following equation:

    \[
    \rho_i = 1 - \delta + \delta \left[ \frac{\rho_1}{k_1} + \frac{\rho_2}{k_2} + \cdots + \frac{\rho_n}{k_n} \right]
    \]

  $\rho_i$ gets assigned a value that depends on the other $\rho_j$'s

- Why is the above definition sensible?

  - Let $e$ be the vector of all ones (length $n$) and $v$ the vector with components $\rho_1, \rho_2, \ldots, \rho_n$.

  - Show that the above equation is equivalent to

    \[
    v = (1 - \delta) e + \delta H v
    \]

  - How would you solve the system?

    - Can show: Sum of all PageRanks == $n$: $\sum \rho_i = n$

- What is the $4 \times 4$ matrix $H$ for the following case? [4 Nodes]

  A points to B and D;
  C points to A and B;
  B points to A, C, and D;
  D points to C;

  Also: Determine the $\rho_i$'s for this case when $\delta = 0.9$ (Matlab)

Example: Here the 4th column of $H$ consists of zeros except

- $h_{14} = 1/5$; $h_{34} = 1/5$;
- $h_{64} = 1/5$; $h_{94} = 1/5$

Simple case: $\delta = 1$

- If token is at node $j$ (with probability 1) at some stage, in the next stage it will jump to node $i$ with probability $h_{ij}$.

  - Case $\delta = 1$ will be very similar to the other Markov chain examples [population movement].

    - Solved in exactly the same way.

  - Issue: token can get stuck if a node has no outgoing links.
The Google PageRank algorithm

- As one can imagine $H$ can be huge so solving the linear system by GE is not practical.
- Alternative: following iterative algorithm

**Algorithm** (PageRank)

1. Select initial vector $v$ ($v \geq 0$)
2. For $i=1:\text{maxitr}$
   
   - $v := (1 - \delta)e + \delta H v$
   
3. end

- Do a few steps of this algorithm for previous example

- Column- sums of $H$ are $= 1$.
- If $\delta = .9$ then solving the linear system yields $v = \begin{bmatrix} 0.94144 \\ 1.05007 \\ 1.16982 \\ 0.83867 \end{bmatrix}$