Rank One Update And the Google Matrix

by
Al Bernstein
Signal Science, LLC
www.signalscience.net

Introduction

There are two different ways to perform matrix multiplications. The first uses a **dot product formulation** and the second uses an **outer product formulation**. Both methods use the same number of operations - O(n^3); however, the outer product formulation is very **parallelizable**. It also gives a huge performance advantage for equations of the form $M = A^T A$ - requiring only the memory to store the result M .If A^T is a **1000000 x 3 matrix**, then M is a **3 x 3 matrix**. The **dot product** formulation requires storing A^T, A - **2 1000000 x 3 matrices**, and the result M **- a 3 x 3 matrix**. The **outer product** formulation requires storing only the result M **-** a **3 x 3 matrix**. A **rank one update** is an operation where a matrix can be updated using an outer product. This update provides the same efficiency advantages as the **outer product** matrix multiplication formulation and will be discussed later in this paper.

Math Derivations and Discussion

Equation (1) shows the dot product formulation for of the $(i, j)^{\text{th}}$ element of the matrix multiplication -$(A \bullet B)$.

$$(A \bullet B)_{i,j} = \sum_k a_{i,k} b_{k,j} \tag{1}$$

The sum is over the dummy variable k ($k = 1, \cdots, N$). Note that the indices i and j are independent variables – input by the user of the equation. The index k acts as an indexer on the right hand side of the equation only and is not conveyed outside the equation. This index is called a **dummy variable** because if we change index k to l there is no difference in the results of the equation. However, changing the indices i or j changes the calculation.

This is the standard matrix multiplication – Equation (2) shows the computation of $(A \bullet B)_{i,j}$ with i=1 and j=1 is

$$(A \bullet B)_{1,1} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N} \end{bmatrix} \bullet \begin{bmatrix} b_{1,1} \\ \vdots \\ b_{N,1} \end{bmatrix} = a_{1,1}b_{1,1} + \cdots + a_{1,N}b_{N,1} \tag{2}$$

The full matrix multiplication is shown in equation (3)

$$(A \bullet B) = \begin{bmatrix} a_{1,1}b_{1,1} + a_{1,2}b_{2,1} + \cdots + a_{1,N}b_{N,1} & \cdots & a_{1,1}b_{1,N} + a_{1,2}b_{2,N} + \cdots + a_{1,N}b_{N,N} \\ a_{2,1}b_{1,1} + a_{2,2}b_{2,1} + \cdots + a_{2,N}b_{N,1} & \cdots & a_{2,1}b_{1,N} + a_{2,2}b_{2,N} + \cdots + a_{2,N}b_{N,N} \\ a_{N,1}b_{1,1} + a_{N,2}b_{2,1} + \cdots + a_{N,N}b_{N,1} & \cdots & a_{N,1}b_{1,N} + a_{N,2}b_{2,N} + \cdots + a_{N,N}b_{N,N} \end{bmatrix} \tag{3}$$

Notice that in equation (1) the indices i and j are **fixed variables** - we specify these on the left side of the equation by picking the i^{th}, and j^{th} elements. There is another way to look at this equation - **keep the indices i and j free variables** as shown in equation (4).

$$(A \bullet B) = \sum_k a_{i,k} b_{k,j} \qquad (4)^1$$

where $k = 1, \cdots N$.

In this case the multiplication for each k is a matrix. This matrix multiplication formula is the called the outer product formulation and is shown below:

$$(A \bullet B) = \begin{bmatrix} a_{1,1}b_{1,1} & \cdots & a_{1,1}b_{1,N} \\ a_{2,1}b_{1,1} & \cdots & a_{2,1}b_{1,N} \\ a_{N,1}b_{1,1} & \cdots & a_{N,1}b_{1,N} \end{bmatrix} + \begin{bmatrix} a_{1,2}b_{2,1} & \cdots & a_{1,2}b_{2,N} \\ a_{2,2}b_{2,1} & \cdots & a_{2,2}b_{2,N} \\ a_{N,2}b_{2,1} & \cdots & a_{N,2}b_{2,N} \end{bmatrix} + \cdots + \begin{bmatrix} a_{1,N}b_{N,1} & \cdots & a_{1,N}b_{N,N} \\ a_{2,N}b_{N,1} & \cdots & a_{2,N}b_{N,N} \\ a_{N,N}b_{N,1} & \cdots & a_{N,N}b_{N,N} \end{bmatrix} =$$

$$\begin{bmatrix} a_{1,1}b_{1,1} + a_{1,2}b_{2,1} + \cdots + a_{1,N}b_{N,1} & \cdots & a_{1,1}b_{1,N} + a_{1,2}b_{2,N} + \cdots + a_{1,N}b_{N,N} \\ a_{2,1}b_{1,1} + a_{2,2}b_{2,1} + \cdots + a_{2,N}b_{N,1} & \cdots & a_{2,1}b_{1,N} + a_{2,2}b_{2,N} + \cdots + a_{2,N}b_{N,N} \\ a_{N,1}b_{1,1} + a_{N,2}b_{2,1} + \cdots + a_{N,N}b_{N,1} & \cdots & a_{N,1}b_{1,N} + a_{N,2}b_{2,N} + \cdots + a_{N,N}b_{N,N} \end{bmatrix} \tag{5}$$

Note that equation (3) agrees with equation (5)

An outer product operation is defined as follows:

$$\vec{a} \otimes \vec{b} = a_i b_j = \begin{bmatrix} a_0 b_0 & \cdots & a_0 b_N \\ \vdots & & \\ a_N b_0 & \cdots & a_N b_N \end{bmatrix} = \vec{a}^T \vec{b} \tag{6}$$

Remember: $\vec{a}^T$ **is a column vector and** $\vec{b}$ **is a row vector**

Equation (5), expressed in outer product notation is shown in equation (7)

$$A \bullet B = \sum_i \left(\vec{a}_i \otimes \vec{b}_i \right) \qquad (7)$$[1]

Note that in this formulation, the vectors $\vec{a}$ and $\vec{b}$ do not have to be the same length. If $\vec{a}$ is of length M and $\vec{b}$ is of length N, the resulting matrix $A \bullet B$ will an M by N matrix – see example III.) below. The matrices need to have the correct dimensions for the product $A \bullet B$ to be true. Also, Equation (7) is readily **parallelizable** because each outer product could be done in parallel and then added together.

Numerical Examples

While the above discussion provides a lot of insight, a simple example will clarify

I.) Standard Matrix Multiplication

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix} \qquad A \bullet B = \begin{bmatrix} 84 & 90 & 96 \\ 201 & 216 & 231 \\ 318 & 342 & 366 \end{bmatrix} \qquad (8)$$

II.) Outer Product Matrix Multiplication

$$A \bullet B = \begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix} \begin{bmatrix} 10 & 11 & 12 \end{bmatrix} + \begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix} \begin{bmatrix} 13 & 14 & 15 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix} \begin{bmatrix} 16 & 17 & 18 \end{bmatrix} =$$

$$\begin{bmatrix} 10 & 11 & 12 \\ 40 & 44 & 48 \\ 70 & 77 & 84 \end{bmatrix} + \begin{bmatrix} 26 & 28 & 30 \\ 65 & 70 & 75 \\ 104 & 112 & 120 \end{bmatrix} + \begin{bmatrix} 48 & 51 & 54 \\ 96 & 102 & 108 \\ 144 & 153 & 162 \end{bmatrix} = \begin{bmatrix} 84 & 90 & 96 \\ 201 & 216 & 231 \\ 318 & 342 & 366 \end{bmatrix} \qquad (9)$$

Sums make outer product formulation readily parallelizable.

III.) Case where A and B have different dimensions

1.) Standard Multiplication

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix} \qquad A \bullet B = \begin{bmatrix} 84 & 90 & 96 \\ 201 & 216 & 231 \end{bmatrix} \tag{10}$$

Outer Product Multiplication

$$A \bullet B = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 10 & 11 & 12 \end{bmatrix} + \begin{bmatrix} 2 \\ 5 \end{bmatrix} \begin{bmatrix} 13 & 14 & 15 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \end{bmatrix} \begin{bmatrix} 16 & 17 & 18 \end{bmatrix} = \begin{bmatrix} 10 & 11 & 12 \\ 40 & 44 & 48 \end{bmatrix} +$$

$$\begin{bmatrix} 26 & 28 & 30 \\ 65 & 70 & 75 \end{bmatrix} + \begin{bmatrix} 48 & 51 & 54 \\ 96 & 102 & 108 \end{bmatrix} = \begin{bmatrix} 84 & 90 & 96 \\ 201 & 216 & 231 \end{bmatrix} \tag{11}$$

Rank One Update

A Rank One update is defined as an operation where a matrix can be updated using an outer product as shown in equation (12)

$$A + \vec{a}^T \vec{b} \tag{12}$$

Note: Equation (12) allows the matrix A to be updated without needed additional memory to perform the calculation.

For example, consider equation (13)

$$A^T A + \vec{a}_i^T \vec{a}_i \tag{13}$$

where $\vec{a}_i$ is a row vector with the same row length as A

This equation adds measurements to the system without needing additional memory. For example, let A be a 10000 x 5 matrix and let each row in this matrix represent a measurement of 5 parameters. If we call the i^{th} row in A, $\vec{a}_i$ (row vector) then in A^T, $\vec{a}_i$ is the row vector transposed – a column vector. Then, $A^T A$, using the outer product method, will be a 5 x 5 matrix no matter how many measurements are taken. The Rank One update is a huge performance savings.

IV.) Rank One Update in Action

First, Calculate $A^T A$ for 3 rows (measurements)

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \tag{14}$$

$$A^T A = \begin{bmatrix} 66 & 78 & 90 \\ 78 & 93 & 108 \\ 90 & 108 & 126 \end{bmatrix} \tag{15}$$

Now start with $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ (16)

$$A^T A = \begin{bmatrix} 17 & 22 & 27 \\ 22 & 29 & 36 \\ 27 & 36 & 45 \end{bmatrix} \tag{17}$$

$$a = \begin{bmatrix} 7 & 8 & 9 \end{bmatrix} \tag{18}$$

Equation (18) is a vector representing the 3rd row in the original matrix A. The Rank One Update shows how to use the outer product - $\vec{a} \otimes \vec{a} = \vec{a}^T \vec{a}$ - to update equation (17) and recover equation (15).

$$A^T A + \vec{a}^T \vec{a} = \begin{bmatrix} 17 & 22 & 27 \\ 22 & 29 & 36 \\ 27 & 36 & 45 \end{bmatrix} + \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \begin{bmatrix} 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 17 & 22 & 27 \\ 22 & 29 & 36 \\ 27 & 36 & 45 \end{bmatrix} + \begin{bmatrix} 49 & 56 & 63 \\ 56 & 64 & 72 \\ 63 & 72 & 81 \end{bmatrix} = \begin{bmatrix} 66 & 78 & 90 \\ 78 & 93 & 108 \\ 90 & 108 & 126 \end{bmatrix}$$

which gives the same results as equation (15)

The Google Matrix

The Google matrix, G, is defined in equation (19)

$$G = \alpha S + \frac{(1-\alpha)}{n} \vec{e} \bullet \vec{e}^T$$ (19)[2]

where

$\alpha \equiv$ scalar number between 0 and 1
$n \equiv$ number of pages in search space,
$\vec{e}^T \equiv$ is a row vector where all the elements have the value (1/n),

Note: The Rank One Update looks like it is transposed compared with equation (12). This is because $\vec{e}$ is defined as a column vector - $\vec{e}^T$ is defined as a row vector in equation (19).

S - defined by equation (20) is

$$S = H + \frac{\vec{a}}{n}\vec{e}^T \tag{20}$$

where $\vec{a}$ **is a column vector of dangling links**; 0 if the link on the i^{th} node is dangling – (a node that is does not navigate to another page on the web such as a file) - and is 1 otherwise.

H **is the hyperlink matrix** which represents a graph of how web pages are connected through links. This write up is not intended to go into detail about the Google page rank algorithm. The main point here is that equations (19) and (20) both use the Rank One Update (13) which has performance advantages.

References

[1] **Gerald Bierman, "Fatorizing Methods for Discrete Sequential Estimation", Academic Press, 1977, page 25**
[2] **Langville and Meyer, "Google's Page Rank and Beyond", Princeton University Press, 2006, pp 37-38**