Lecture 9. The Curse of Dimensionality

Charles F. Van Loan
Cornell University

The Gene Golub SIAM Summer School 2010
Selva di Fasano, Brindisi, Italy
Lecture 1. Introduction to Tensor Computations
Lecture 2. Tensor Unfoldings
Lecture 3. Transpositions, Kronecker Products, Contractions
Lecture 4. Tensor-Related Singular Value Decompositions
Lecture 5. The CP Representation and Tensor Rank
Lecture 6. The Tucker Representation
Lecture 7. Other Decompositions and Nearness Problems
Lecture 8. Multilinear Rayleigh Quotients
Lecture 9. The Curse of Dimensionality
Lecture 10. Special Topics
Big Problems

1. A single $N$-by-$N$ matrix problem is big if $N$ is big.
2. A problem that involves $N$ small $p$-by-$p$ problems is big if $N$ is big.
3. A problem that involves a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is big if

$$N = n_1 \cdots n_d$$

is big and that can happen rather easily if $d$ is big.
Data Sparse Representation

We are used to solving big matrix problems when the matrix is data-sparse, i.e., when $A \in \mathbb{R}^{N \times N}$ can be represented with many fewer than $N^2$ numbers.

What if $N$ is so big that we cannot even store length-$N$ vectors?

How could we apply (for example) the Rayleigh Quotient procedure in such a situation?
What is this Lecture About?

A Very Large Eigenvalue Problem

We will look at a problem where \( A \in \mathbb{R}^{2^d \times 2^d} \) is data sparse but where \( d \) is sufficiently big to make the storage of length-\( 2^d \) vectors impossible.

Vectors will be approximated by data sparse tensors of high order.
A Problem From Quantum Chemistry

Given a $2^d$-by-$2^d$ symmetric matrix $H$, find a vector $a$ that minimizes

$$r(a) = \frac{a^T H a}{a^T a}$$

Of course: $a = a_{min}$, $\lambda = r(a_{min}) \Rightarrow H a = \lambda a$.

What if $d = 100$?
The Google Slide
A Very Large Eigenvalue Problem

The H-Matrix

\[ H = \sum_{ij} t_{ij} H_i^T H_j + \sum_{ijkl} v_{ijkl} H_i^T H_j^T H_k H_l \]

\[ H_i = I_{2^{i-1}} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes I_{2^{d-i}} \]

\( T \in \mathbb{R}^{d \times d} \) is symmetric and \( V \in \mathbb{R}^{d \times d \times d \times d} \) has symmetries.

Sparsity

\[ n_{zeros} = \left( \frac{1}{64} d^4 - \frac{3}{32} d^3 + \frac{27}{64} d^2 - \frac{11}{32} d + 1 \right) 2^d - 1 \]
Modeling Electron Interactions

Have \( d \) “sites” (grid points) in physical space.

The goal is to compute a wave function, an element of a \( 2^d \) Hilbert space.

The Hilbert space is a product of \( d \), 2-dimensional Hilbert spaces. (A site is either occupied or not occupied.)

A (discretized) wavefunction is a \( d \)-tensor, 2-by-2-by-2-by-2...
It is the vector that minimizes \( a^T H a / a^T a \) where...
The $H$-Matrix

$$H = \sum_{ij} t_{ij} H_i^T H_j + \sum_{ijkl} v_{ijkl} H_i^T H_j^T H_k H_l$$

\[\uparrow\text{Kinetic Energy Weights}\]
\[\uparrow\text{Potential Energy Weights}\]

$$H_i = I_{2^{i-1}} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes I_{2^{d-i}}$$
A Very Large Eigenvalue Problem

Dealing with $N = 2^d \approx 2^{100}$

Intractable:

$$\min_{a \in \mathbb{R}^N} \frac{a^T Ha}{a^T a}$$

Tractable:

$$\min_{a \in \mathbb{R}^N, \text{data sparse}} \frac{a^T Ha}{a^T a}$$
A tensor network is a tensor of high dimension that is built up from many sparsely connected tensors of low-dimension.
TN slides
Recall the Block Vec Operation

\[
\begin{bmatrix}
F_1 \\
F_2
\end{bmatrix} \otimes \left( \begin{bmatrix}
G_1 \\
G_2 \\
G_3
\end{bmatrix} \otimes \begin{bmatrix}
H_1 \\
H_2
\end{bmatrix} \right) = \begin{bmatrix}
F_1 \\
F_2
\end{bmatrix} \otimes \begin{bmatrix}
G_1 H_1 \\
G_1 H_2 \\
G_2 H_1 \\
G_2 H_2 \\
G_3 H_1 \\
G_3 H_2
\end{bmatrix} = \begin{bmatrix}
F_1 G_1 H_1 \\
F_1 G_1 H_2 \\
F_1 G_2 H_1 \\
F_1 G_2 H_2 \\
F_1 G_3 H_1 \\
F_1 G_3 H_2 \\
F_2 G_1 H_1 \\
F_2 G_1 H_2 \\
F_2 G_2 H_1 \\
F_2 G_2 H_2 \\
F_2 G_3 H_1 \\
F_2 G_3 H_2
\end{bmatrix}
\]
In the "Language" of Block Vec Products...

\[ a = \left[ \begin{array}{c} A_{11} \\ A_{21} \end{array} \right] \otimes \left[ \begin{array}{c} A_{12} \\ A_{22} \end{array} \right] \otimes \cdots \otimes \left[ \begin{array}{c} A_{1,d-1} \\ A_{2,d-1} \end{array} \right] \otimes \left[ \begin{array}{c} A_{1d} \\ A_{2d} \end{array} \right] \]

where

\[
\begin{align*}
\left[ \begin{array}{c} A_{11} \\ A_{21} \end{array} \right] &= \left[ \begin{array}{c} w_1^T \\ w_2^T \end{array} \right] & & \text{2 row vectors} \\
\left[ \begin{array}{c} A_{1k} \\ A_{2k} \end{array} \right] &= \left[ \begin{array}{c} m\text{-by-}m \\ m\text{-by-}m \end{array} \right] & & k = 2:d - 1 \\
\left[ \begin{array}{c} A_{1d} \\ A_{2d} \end{array} \right] &= \left[ \begin{array}{c} z_1 \\ z_2 \end{array} \right] & & \text{2 column vectors}
\end{align*}
\]

\( a \) is a length-\(2^d\) vector that depends on \(O(dm^2)\) numbers
Back to the Main Problem...

Constrained Minimization

Minimize

\[ r(a) = \frac{a^T Ha}{a^T a} \]

subject to the constraint that

\[ a = \begin{bmatrix} A_{11} \\ A_{21} \\ A_{12} \\ A_{22} \\ \vdots \end{bmatrix} \otimes \begin{bmatrix} A_{1,d-1} \\ A_{2,d-1} \\ A_{1d} \\ A_{2d} \end{bmatrix} \]

Let us look at both the denominator and the numerator in light of the fact that \( N = 2^d \).
Avoiding $O(2^d)$

2-Norm of a Linear Tensor Network...

If

$$a = \left[ \begin{array}{c} A_{11} \\ A_{21} \end{array} \right] \otimes \left[ \begin{array}{c} A_{12} \\ A_{22} \end{array} \right] \otimes \ldots \otimes \left[ \begin{array}{c} A_{1,d-1} \\ A_{2,d-1} \end{array} \right] \otimes \left[ \begin{array}{c} A_{1d} \\ A_{2d} \end{array} \right]$$

then

$$a^T a = w^T \left( \prod_{k=2}^{d-1} (A_{1k} \otimes A_{1k}) + (A_{2k} \otimes A_{2k}) \right) z$$

where

$$w = A_{11} \otimes A_{11} + A_{21} \otimes A_{21} = w_1 \otimes w_1 + w_2 \otimes w_2$$

$$z = A_{1d} \otimes A_{1d} + A_{2d} \otimes A_{2d} = z_1 \otimes z_1 + z_2 \otimes z_2$$

$A_{1k}$ and $A_{2k}$ are $m$-by-$m$, $k = 2:d - 1$. Overall work is $O(dm^3)$. 
Avoiding $O(d^4)$

Recall...

\[ H = \sum_{ij} t_{ij} H_i^T H_j + \sum_{ijkl} v_{ijkl} H_i^T H_j^T H_k H_l \]

\[ H_i = I_{2i-1} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes I_{2d-i} \]

The $\mathcal{V}$-Tensor Has Familiar Symmetries

\[ \mathcal{V}(i, j, k, \ell) = \begin{cases} \mathcal{V}(j, i, k, \ell) \\ \mathcal{V}(i, j, \ell, k) \\ \mathcal{V}(k, \ell, i, j) \end{cases} \]

and so we can find symmetric matrices $B_1, \ldots, B_r$ so

\[ \mathcal{V} = B_1 \circ B_1 + \cdots + B_r \circ B_r \]
Avoiding $O(d^4)$

**Idea**

Approximate $\mathcal{V}$ with $\mathcal{B}_1 \circ \mathcal{B}_1$ (or some short sum of the $\mathcal{B}$’s) because then $v_{ijk\ell} = B_1(i, j)B_1(k, \ell)$ and

$$H = \sum_{ij} t_{ij} H_i^T H_j + \sum_{ijkl} v_{ijkl} H_i^T H_j^T H_k H_l$$

$$= \sum_{ij} t_{ij} H_i^T H_j + \left( \sum_{ij} B_1(i, j) H_i H_j \right)^T \left( \sum_{ij} B_1(i, j) H_i H_j \right)$$

Think about $a^T Ha$ and note that we have reduced evaluation by a factor of $O(d^2)$. 
For \( k = 1:d \ldots \)

Minimize

\[
    r(a) = \frac{a^T Ha}{a^T a} = r(A_{1k}, A_{2k})
\]

subject to the constraint that

\[
    a = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} \otimes \begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix} \otimes \ldots \otimes \begin{bmatrix} A_{1,d-1} \\ A_{2,d-1} \end{bmatrix} \otimes \begin{bmatrix} A_{1d} \\ A_{2d} \end{bmatrix}
\]

and all by \( A_{1k} \) and \( A_{2k} \) are fixed.

This projected subproblem can reshaped into a smaller, \( 2m^2 \)-by-\( 2m^2 \) Rayleigh Quotient minimization...
The Subproblem

Minimize

\[ r(a_k) = \frac{a_k^T H_k a_k}{a_k^T a_k} \]

where

\[ a_k = \begin{bmatrix} \text{vec}(A_{1k}) \\ \text{vec}(A_{2k}) \end{bmatrix} \]

and

\[ H_k = T_k^T H T_k \quad T_k \in \mathbb{R}^{2d \times m^2} \]

can be formed in time polynomial in \( m \).
Tensor-Based Thinking

Key Attributes

1. An ability to reason at the index-level about the constituent contractions and the order of their evaluation.

2. An ability to reason at the block matrix level in order to expose fast, underlying Kronecker product operations.
How Could We Compute the QR Factorization of This?

\[
\begin{bmatrix}
F_1 \\
F_2 \\
\end{bmatrix} \otimes \left( \begin{bmatrix}
G_1 \\
G_2 \\
G_3 \\
\end{bmatrix} \otimes \begin{bmatrix}
H_1 \\
H_2 \\
\end{bmatrix} \right) = \begin{bmatrix}
F_1 \\
F_2 \\
\end{bmatrix} \otimes \begin{bmatrix}
G_1 H_1 \\
G_1 H_2 \\
G_2 H_1 \\
G_2 H_2 \\
G_3 H_1 \\
G_3 H_2 \\
\end{bmatrix} = \begin{bmatrix}
F_1 G_1 H_1 \\
F_1 G_1 H_2 \\
F_1 G_2 H_1 \\
F_1 G_2 H_2 \\
F_1 G_3 H_1 \\
F_1 G_3 H_2 \\
F_2 G_1 H_1 \\
F_2 G_1 H_2 \\
F_2 G_2 H_1 \\
F_2 G_2 H_2 \\
F_2 G_3 H_1 \\
F_2 G_3 H_2 \\
\end{bmatrix}
\]

Without “Leaving” the Data Sparse Representation?
QR Factorization and Block vector Products

If

\[
\begin{bmatrix}
F_1 \\
F_2
\end{bmatrix} = \begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} R
\]

then

\[
\begin{bmatrix}
F_1 \\
F_2
\end{bmatrix} \otimes \begin{bmatrix}
G_1 \\
G_2 \\
G_3
\end{bmatrix} \otimes \begin{bmatrix}
H_1 \\
H_2
\end{bmatrix} = \begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} \otimes \begin{bmatrix}
RG_1 \\
RG_2 \\
RG_3
\end{bmatrix} \otimes \begin{bmatrix}
H_1 \\
H_2
\end{bmatrix}
\]

If

\[
\begin{bmatrix}
RG_1 \\
RG_2 \\
RG_3
\end{bmatrix} = \begin{bmatrix}
U_1 \\
U_2 \\
U_3
\end{bmatrix} S
\]

then...
QR Factorization and Block vector Products

\[
\begin{bmatrix}
F_1 \\
F_2
\end{bmatrix} \otimes \begin{bmatrix}
G_1 \\
G_2 \\
G_3
\end{bmatrix} \otimes \begin{bmatrix}
H_1 \\
H_2
\end{bmatrix} = \begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} \otimes \begin{bmatrix}
U_1 \\
U_2 \\
U_3
\end{bmatrix} \otimes \begin{bmatrix}
SH_1 \\
SH_2
\end{bmatrix}
\]

If

\[
\begin{bmatrix}
SH_1 \\
SH_2
\end{bmatrix} = \begin{bmatrix}
V_1 \\
V_2
\end{bmatrix}^T
\]

then

\[
\begin{bmatrix}
F_1 \\
F_2
\end{bmatrix} \otimes \begin{bmatrix}
G_1 \\
G_2 \\
G_3
\end{bmatrix} \otimes \begin{bmatrix}
H_1 \\
H_2
\end{bmatrix} = \left(\begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} \otimes \begin{bmatrix}
U_1 \\
U_2 \\
U_3
\end{bmatrix} \otimes \begin{bmatrix}
V_1 \\
V_2
\end{bmatrix}\right)^T
\]

*The Matrix in Parentheses is Orthogonal*
The Curse of Dimensionality refers to the challenges that arise when dimension increases.

Clever data-sparse representations are one way to address the issues.

A tensor network is a way of combining low-order tensors to obtain a high-order tensor.

Reliable methods that scale with dimension are the goal.


