Tensor Applications

Betty Shea

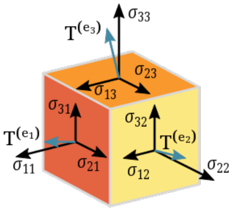
UBC MLRG 2020 Winter Term 1

30-Sept-2020

Today

1. Introduction and Motivation
   - What is a tensor?
   - Examples
2. MLRG
   - Overview
   - What this term's MLRG is and is not about
3. Papers and Signup
   - List of papers
   - Acknowledgements and references

What is a tensor?

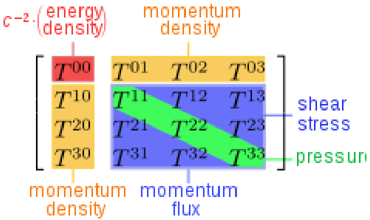


- Multi-dimensional array
- Linear operator

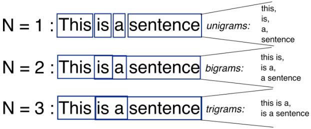
- Wikipedia: "an algebraic object that describes a relationship between sets of algebraic objects related to a vector space"

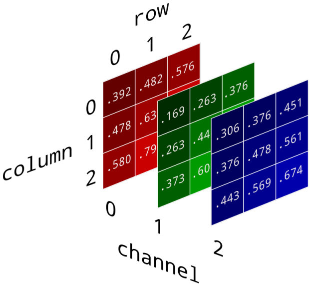
Where are tensors used: broad areas

- Statistics: Joint probability tensors
- Physics: Einstein field equations
- Material science: stress tensors, strain tensors and elasticity



Where are tensors used: machine learning



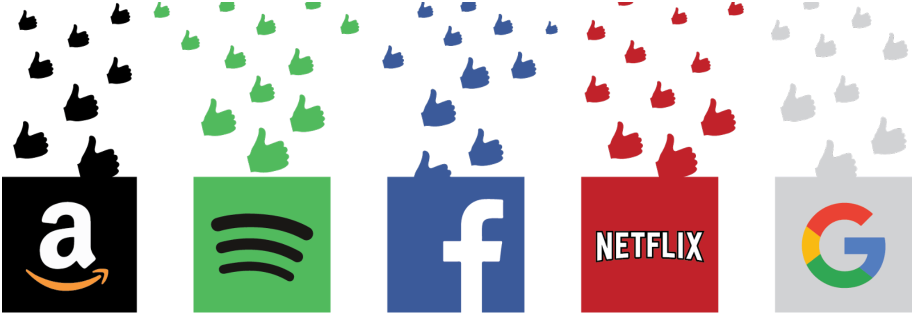


- Natural language processing: Topic models, n-grams
- Image processing: A colour image is three matrices of pixels corresponding to red, green and blue.

Example 1: Recommender systems

Very big and sparse matrix Y

- Netflix: rows are content viewers, columns are movies
- Amazon: rows are shoppers, columns are items for sale



Collaborative filtering

"Regularized PCA on the available entries of Y"

$$Y \approx ZW \text{ where } Z \text{ represents user features and } W \text{ movie features}$$

Matrix factorization produces a latent factor model of types of users and movies.

Collaborative filtering

"Regularized PCA on the available entries of Y"

$$Y \approx ZW \text{ where } Z \text{ represents user features and } W \text{ movie features}$$

Matrix factorization produces a latent factor model of types of users and movies.

Issues with matrix factorization

- Solutions are not unique
- Feedback from user may not just be a single number

Collaborative filtering

"Regularized PCA on the available entries of Y"

$$Y \approx ZW \text{ where } Z \text{ represents user features and } W \text{ movie features}$$

Matrix factorization produces a latent factor model of types of users and movies.

Issues with matrix factorization

- Solutions are not unique
- Feedback from user may not just be a single number

Why we may want to go to tensor factorization

- Tensor factorization often give unique solutions
- Can learn all dimensions of the feedback simultaneously

Collaborative filtering

"Regularized PCA on the available entries of Y"

$$Y \approx ZW \text{ where } Z \text{ represents user features and } W \text{ movie features}$$

Matrix factorization produces a latent factor model of types of users and movies.

Issues with matrix factorization

- Solutions are not unique
- Feedback from user may not just be a single number

Why we may want to go to tensor factorization

- Tensor factorization often give unique solutions
- Can learn all dimensions of the feedback simultaneously

Tensor factorization is NP-hard in general.

Example 2: The complexity of matrix multiplication

Consider multiplying two $n \times n$ matrices. Standard algorithm takes n^3 multiplications.

Example 2: The complexity of matrix multiplication

Consider multiplying two $n \times n$ matrices. Standard algorithm takes n^3 multiplications.

For example, there are 8 multiplications in the $n = 2$ case

$$C = A \times B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

1. $A_{11} \times B_{11}$
2. $A_{12} \times B_{21}$
3. $A_{21} \times B_{11}$
4. $A_{22} \times B_{21}$
5. $A_{11} \times B_{12}$
6. $A_{12} \times B_{22}$
7. $A_{21} \times B_{12}$
8. $A_{22} \times B_{22}$

Example 2: The complexity of matrix multiplication

Consider multiplying two $n \times n$ matrices. Standard algorithm takes n^3 multiplications.

For example, there are 8 multiplications in the $n = 2$ case

$$C = A \times B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

1. $A_{11} \times B_{11}$
2. $A_{12} \times B_{21}$
3. $A_{21} \times B_{11}$
4. $A_{22} \times B_{21}$
5. $A_{11} \times B_{12}$
6. $A_{12} \times B_{22}$
7. $A_{21} \times B_{12}$
8. $A_{22} \times B_{22}$

From CPSC221, we know that it is possible to use 7 instead of 8 multiplications.

Strassen's algorithm

$$C = A \times B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Define 7 new quantities

1. $M_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22})$
2. $M_2 = (A_{21} + A_{22}) \times B_{11}$
3. $M_3 = A_{11} \times (B_{12} - B_{22})$
4. $M_4 = A_{22} \times (B_{21} - B_{11})$
5. $M_5 = (A_{11} + A_{12}) \times B_{22}$
6. $M_6 = (A_{21} - A_{11}) \times (B_{11} + B_{12})$
7. $M_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22})$

And then these additions give you the answer

1. $C_{11} = M_1 + M_4 - M_5 + M_7$
2. $C_{12} = M_3 + M_5$
3. $C_{21} = M_2 + M_4$
4. $C_{22} = M_1 - M_2 + M_3 + M_6$

Strassen's algorithm

$$C = A \times B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Define 7 new quantities

1. $M_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22})$
2. $M_2 = (A_{21} + A_{22}) \times B_{11}$
3. $M_3 = A_{11} \times (B_{12} - B_{22})$
4. $M_4 = A_{22} \times (B_{21} - B_{11})$
5. $M_5 = (A_{11} + A_{12}) \times B_{22}$
6. $M_6 = (A_{21} - A_{11}) \times (B_{11} + B_{12})$
7. $M_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22})$

And then these additions give you the answer

1. $C_{11} = M_1 + M_4 - M_5 + M_7$
2. $C_{12} = M_3 + M_5$
3. $C_{21} = M_2 + M_4$
4. $C_{22} = M_1 - M_2 + M_3 + M_6$

Also works if the entries are matrices instead of scalars.

Strassen's algorithm

$$C = A \times B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Define 7 new quantities

1. $M_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22})$
2. $M_2 = (A_{21} + A_{22}) \times B_{11}$
3. $M_3 = A_{11} \times (B_{12} - B_{22})$
4. $M_4 = A_{22} \times (B_{21} - B_{11})$
5. $M_5 = (A_{11} + A_{12}) \times B_{22}$
6. $M_6 = (A_{21} - A_{11}) \times (B_{11} + B_{12})$
7. $M_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22})$

And then these additions give you the answer

1. $C_{11} = M_1 + M_4 - M_5 + M_7$
2. $C_{12} = M_3 + M_5$
3. $C_{21} = M_2 + M_4$
4. $C_{22} = M_1 - M_2 + M_3 + M_6$

Also works if the entries are matrices instead of scalars.

For the general $n \times n$ case, takes $n^{\log_2 7}$ multiplications.

Matrix multiplication exponent ω

The natural question is then can we do better than $n^{\log_2 7} \approx n^{2.81}$?

Matrix multiplication exponent ω

The natural question is then can we do better than $n^{\log_2 7} \approx n^{2.81}$?

The more interesting question is how much better can we do?

Matrix multiplication exponent ω

The natural question is then can we do better than $n^{\log_2 7} \approx n^{2.81}$?

The more interesting question is how much better can we do?

The *exponent* ω of matrix multiplication is the lowest real-valued h where two $n \times n$ matrices may be multiplied using $O(n^h)$ arithmetic operations.

Matrix multiplication exponent ω

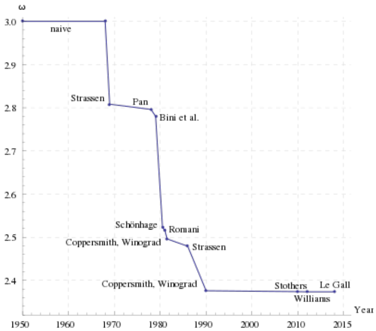
The natural question is then can we do better than $n^{\log_2 7} \approx n^{2.81}$?

The more interesting question is how much better can we do?

The *exponent ω* of matrix multiplication is the lowest real-valued h where two $n \times n$ matrices may be multiplied using $O(n^h)$ arithmetic operations.

For Strassen's algorithm, $\omega = \log_2 7$.

Since then... (chart from Wikipedia)



Matrix multiplication exponent ω

Understanding the complexity of matrix multiplication has to do with understanding ω.

Matrix multiplication exponent ω

Understanding the complexity of matrix multiplication has to do with understanding ω.

And tensors provide a way to understand ω.

Matrix multiplication exponent ω

Understanding the complexity of matrix multiplication has to do with understanding ω.

And tensors provide a way to understand ω.

Like matrices, tensors could also be viewed as both

- structures containing data, i.e. a d-way array and
- linear operators, i.e. can multiply vectors, matrices and tensors

Tensor rank and ω

Vector spaces A, B, and C with $a \in A$, $b \in B$ and $c \in C$.

- define $A^* = \{f : A \to \mathbb{R} | f \text{ is linear}\}$

Tensor rank and ω

Vector spaces A, B, and C with $a \in A$, $b \in B$ and $c \in C$.

- define $A^* = \{f : A \to \mathbb{R} | f \text{ is linear}\}$
- define a *linear map* $\alpha \otimes b : A \to B$ by

$$a \mapsto \alpha(a)b \text{ where } \alpha \in A^*$$

(This is equivalently a matrix, e.g., permutation matrices, reflection matrices, etc.)

Tensor rank and ω

Vector spaces A, B, and C with $a \in A$, $b \in B$ and $c \in C$.

- define $A^* = \{f : A \to \mathbb{R} | f \text{ is linear}\}$
- define a *linear map* $\alpha \otimes b : A \to B$ by

$$a \mapsto \alpha(a)b \text{ where } \alpha \in A^*$$

  (This is equivalently a matrix, e.g., permutation matrices, reflection matrices, etc.)
- define a *bilinear map* $\alpha \otimes \beta \otimes c : A \times B \to C$ by

$$(a, b) \mapsto \alpha(a)\beta(b)c \text{ where } \alpha \in A^* \text{ and } \beta \in B^*$$

  This is equivalently a tensor, which can be decomposed into a sum of rank 1 tensors.

Tensor rank and ω

Vector spaces A, B, and C with $a \in A$, $b \in B$ and $c \in C$.

- define $A^* = \{f : A \to \mathbb{R} | f \text{ is linear}\}$
- define a *linear map* $\alpha \otimes b : A \to B$ by

$$a \mapsto \alpha(a)b \text{ where } \alpha \in A^*$$

  (This is equivalently a matrix, e.g., permutation matrices, reflection matrices, etc.)
- define a *bilinear map* $\alpha \otimes \beta \otimes c : A \times B \to C$ by

$$(a, b) \mapsto \alpha(a)\beta(b)c \text{ where } \alpha \in A^* \text{ and } \beta \in B^*$$

  This is equivalently a tensor, which can be decomposed into a sum of rank 1 tensors.
- So, we can write a bilinear map $T : A \times B \to C$ as

$$T(a, b) = \sum_{i=1}^{r} \alpha^i(a)\beta^i(b)c_i \text{ for some } r \text{ where } \alpha^i \in A^*, \beta^i \in B^*, c_i \in C$$

Tensor rank and ω

The *rank* of a bilinear map $T : A \times B \to C$, denoted $R(T)$, is the minimal number r over all possible ways of writing T in the form

$$T(a, b) = \sum_{i=1}^{r} \alpha^i(a)\beta^i(b)c_i$$

If T has rank r, its complexity in terms of multiplications is r.

Tensor rank and ω

- Matrix multiplication of square matrices is a bilinear map of the form

$$M_{n,n,n} : \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$$

Tensor rank and ω

- Matrix multiplication of square matrices is a bilinear map of the form

$$M_{n,n,n} : \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$$

- The smallest number of multiplications for multiplying two $n \times n$ matrices is given by $R(M_{n,n,n})$. This is the minimum r over all possible ways to write $M_{n,n,n}$ as a sum of rank 1 tensors.

Tensor rank and ω

- Matrix multiplication of square matrices is a bilinear map of the form

$$M_{n,n,n} : \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$$

- The smallest number of multiplications for multiplying two $n \times n$ matrices is given by $R(M_{n,n,n})$. This is the minimum r over all possible ways to write $M_{n,n,n}$ as a sum of rank 1 tensors.

- The lowest achievable matrix multiplication exponent is in fact the rank of a bilinear map

$$\omega = \liminf_{n\to\infty} \log_n \left(R(M_{n,n,n})\right)$$

Tensor rank and ω

- Matrix multiplication of square matrices is a bilinear map of the form

$$M_{n,n,n} : \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$$

- The smallest number of multiplications for multiplying two $n \times n$ matrices is given by $R(M_{n,n,n})$. This is the minimum r over all possible ways to write $M_{n,n,n}$ as a sum of rank 1 tensors.

- The lowest achievable matrix multiplication exponent is in fact the rank of a bilinear map

$$\omega = \liminf_{n\to\infty} \log_n \left(R(M_{n,n,n})\right)$$

- And the complexity of matrix multiplication is determined by our ability to find explicit equations for the set of tensors in $\mathbb{R}^{n^2} \otimes \mathbb{R}^{n^2} \otimes \mathbb{R}^{n^2}$ of rank at most r.

MLRG

Main themes, structure and schedule

- Anticipated to run between 30-Sep to 9-Dec

Main themes, structure and schedule

- Anticipated to run between 30-Sep to 9-Dec
- 10 papers
  - Two papers on tensor basics, definitions, operations, theory, complexity
  - A paper on link prediction and knowledge graphs that uses tensor factorization
  - Two papers on latent models
  - Three papers on higher order optimization methods
  - A paper each on on tensors in deep neural networks and in visual data

Things that are relevant to this MLRG

- **Read papers that use tensors in one form or another.**
- Understanding when it makes sense to increase the complexity of our models and methods.
  - Matrices $\rightarrow$ tensors
  - First-order methods $\rightarrow$ higher-order methods
- Understanding the hidden assumptions in our simpler methods that no longer apply.
- Generalization.

Things that are not the main focus

- A thorough and rigorous understanding of tensors and tensor decompositions.

  Elina Robeva's MATH 605D would be much better for this.

Papers and Signup

Papers 1 and 2: Background

1. **Kolda and Bader (2009) Tensor decompositions and applications**
   - http://www.kolda.net/publication/TensorReview.pdf
   - Sections 1 to 3.3 (and more if you feel like it)

Additional resources:

- *Ankur Moitra. (2014) Algorithmic Aspects of Machine Learning, sections 3.1-3.2* *http://people.csail.mit.edu/moitra/docs/bookex.pdf*
- *Previous MLRG talks: https://www.cs.ubc.ca/labs/lci/mlrg/slides/Spectral_Methods.pdf, https://www.cs.ubc.ca/labs/lci/mlrg/slides/MLRG_Tensor_Talk.pdf*
- *Survey paper: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7891546*

2. **Hillar and Lim (2013) Most tensor problems are NP-hard**
   - https://arxiv.org/abs/0911.1393

Papers 3: Knowledge graphs

3. **Kazemi and Poole (2018) SimplE embedding for link prediction in knowledge graphs**
   - https://papers.nips.cc/paper/7682-simple-embedding-for-link-prediction-in-knowledge-graphs.pdf

Papers 4 and 5: Latent models

4. Hsu and Kakade. (2013) Learning mixtures of spherical Gaussians: moment methods and spectral decompositions

- https://arxiv.org/pdf/1206.5766.pdf

Additional resources:

- *Anandkumar et al. (2014) Tensor decompositions for learning latent variable models* *https://arxiv.org/pdf/1210.7559.pdf*
- *MLSS slides : http://newport.eecs.uci.edu/anandkumar/pubs/MLSS-part1.pdf*
- *540 slides: https://www.cs.ubc.ca/~schmidtm/Courses/540-W20/L7.pdf*

5. Anandkumar et al. (2012) A spectral algorithm for latent Dirichlet allocation

- https://papers.nips.cc/paper/4637-a-spectral-algorithm-for-latent-dirichlet-allocation

Additional resources:

- *Anandkumar et al. (2014) Tensor decompositions for learning latent variable models* *https://arxiv.org/pdf/1210.7559.pdf*
- *MLSS slides: http://newport.eecs.uci.edu/anandkumar/pubs/MLSS-part2.pdf*
- *540 slides: https://www.cs.ubc.ca/~schmidtm/Courses/540-W20/L29.pdf*
- *Ankur Moitra. (2014) Algorithmic Aspects of Machine Learning, section 3.5* *http://people.csail.mit.edu/moitra/docs/bookex.pdf*

Papers 6 to 8: Higher order methods

6. Baes, Michel. (2009) Estimate sequence methods: extensions and approximations

- http://www.optimization-online.org/DB_FILE/2009/08/2372.pdf
- "the modern view on what can be gained by higher-order methods"

7. Nesterov, Yurii. (2020) Inexact accelerated high-order proximal-point methods

- https://dial.uclouvain.be/pr/boreal/object/boreal:227219
- Bi-level Unconstrained Minimization framework, pth-order proximal point operation

8. Cartis et al. (2018) Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints

- https://arxiv.org/abs/1811.01220
- ARqp framework, e-approximate q order necessary minimizers for p order problems. Upper bounding the complexity of the tensor step in the previous paper.

Papers 9 and 10: Neural networks and visual data

9. **Novikov et al. (2015) Tensorizing Neural Networks**
   - `https://papers.nips.cc/paper/5787-tensorizing-neural-networks`
   - Tensors in deep neural networks

10. **Liu et al. (2012) Tensor completion for estimating missing values in visual data**
    - `https://www.cs.rochester.edu/u/jliu/paper/Ji-ICCV09.pdf`

Signup sheet

Winter term 1 2020 - Tensor basics and applications		Every Wednesday at 1:00 PM Online
Date	Presenter	Topic
Sep 30	Betty	Motivation
Oct 7		Tensor basics: Notation, operations, etc. (Kolda and Bader 2009)
Oct 14		Tensor basics: Complexity (arXiv ID: 0911.1393)
Oct 21	Bahare	Tensor factorization: Knowledge graphs (Kazemi and Poole 2018)
Oct 28		Latent models: Gaussian mixture models (arXiv ID: 1206.5766)
Nov 4		Latent models: Topic models (arXiv ID: 1204.6703)
Nov 11		Higher order methods: Estimate sequence methods (Baes 2009)
Nov 18		Higher order methods: Bi-level unconstrained minimization (Nesterov 2020)
Nov 25		Higher order methods: ARqp framework (arXiv ID: 1811.01220)
Dec 2		Other appl: Tensors in deep neural networks (arXiv ID: 1509.06569)
Dec 9		Other appl: Tensor completion in visual data (Liu et al. 2013)

Acknowledgements

- I used the first few lectures from **Elina Robeva's MATH 605D Fall 2020 Tensor decompositions and their applications**

  https://sites.google.com/view/ubc-math-605d/class-overview

- The collaborative filtering example is taken from Mark Schmidt's CPSC 340 slides https://www.cs.ubc.ca/~schmidtm/Courses/340-F19/L30.pdf

- The Strassen's algorithm example is taken from Landsberg's book *Tensors: Geometry and Applications*

References

- Landsberg, J.M. *Tensors: Geometry and Applications*
  The introduction chapter is available here.
- Robeva, Elina. MATH605D Fall 2020 Tensor decomposition and their applications
  `https://sites.google.com/view/ubc-math-605d/class-overview`
- Strang, G. *Linear Algebra and Learning From Data*

Thank you

Bonus

"The workhorse of scientific computation is matrix multiplication."

– *J.M. Landsberg, Tensors: Geometry and Applications*

It's been shown that $R(M_{2,2,2})$ is exactly 7 but not much else is known about $R(M_{n,n,n})$ except that

- $R(M_{3,3,3})$ is somewhere between 19 and 23.
- Best asymptotic lower bound: $\frac{5}{2}n^2 - 3n \leq R(M_{n,n,n})$