Approximating a tensor as a sum of rank-one components

Petros Drineas

Rensselaer Polytechnic Institute
Computer Science Department

To access my web page:

Google drineas

Research interests in my group

Algorithmic tools

(Randomized, Approximate) Matrix/tensor algorithms and – in particular – matrix/tensor decompositions.

Goal

Learn a model for the underlying "physical" system generating the data.

Matrices/tensors in data mining

<u>**Data are represented by matrices**</u>

Numerous modern datasets are in matrix form.

n features

m objects $\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix}$

We are given *m* objects and *n* features describing the objects.

A_{ij} shows the "importance" of feature *j* for object *i*.

Data are also represented by tensors.

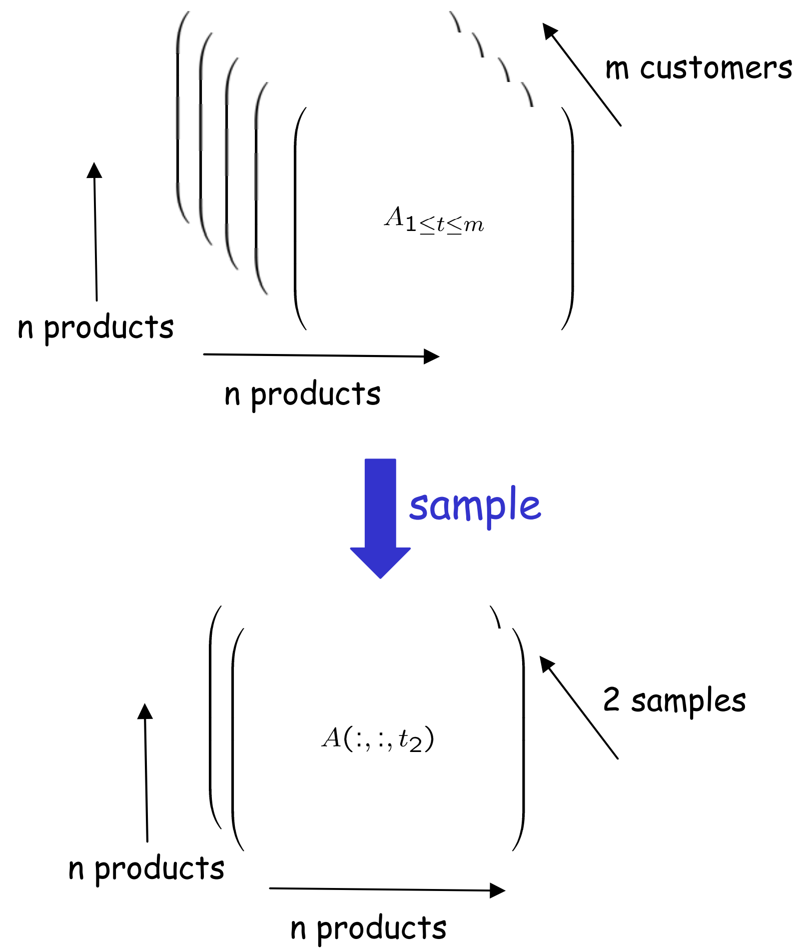
Linear algebra and numerical analysis provide the fundamental mathematical and algorithmic tools to deal with matrix and tensor computations.

The TensorCUR algorithm

Mahoney, Maggioni, & Drineas KDD '06, SIMAX '08, Drineas & Mahoney LAA '07

- Definition of Tensor-CUR decompositions
- Theory behind Tensor-CUR decompositions
- Applications of Tensor-CUR decompositions:

recommendation systems, hyperspectral image analysis



Theorem:

$$\|A - CU \times_\alpha R\|_F^2 \le \left\| A_{[\alpha]} - \left(A_{[\alpha]}\right)_{k_\alpha} \right\|_F^2 + \epsilon \|A\|_F^2$$

Unfold R along the α dimension and pre-multiply by CU

Best rank k_a approximation to $A_{[a]}$

Overview

- Preliminaries, notation, etc.
- Negative results
- Positive results
  - Existential result (full proof)
  - Algorithmic result (sketch of the algorithm)
- Open problems

Approximating a tensor

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Notation

A is an order-r tensor (e.g., a tensor with r modes)

A rank-one component is an outer product of r vectors:

$$\mathcal{C}^{(i)} = u^{(1)} \otimes u^{(2)} \ldots \otimes u^{(r)}$$

A rank-one component has the same dimensions as A, and

$$\mathcal{C}^{(i)}_{j1,j2,\ldots,jr} = u^{(1)}_{j1} \cdot u^{(2)}_{j2} \ldots \cdot u^{(r)}_{jr}$$

Approximating a tensor

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Notation

A is an order-r tensor (e.g., a tensor with r modes)

A rank-one component is an outer product of r vectors:

$$\mathcal{C}^{(i)} = u^{(1)} \otimes u^{(2)} \ldots \otimes u^{(r)}$$

We will measure the error:

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_{F,2}$$

Approximating a tensor

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Notation

A is an order-r tensor (e.g., a tensor with r modes)

We will measure the error: $\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_{F,2}$

Frobenius norm: $\|\mathcal{A}\|_F^2 = \sum_{j_1, j_2, \ldots, j_r} \mathcal{A}_{j_1, j_2, \ldots, j_r}^2$

Spectral norm: $\|\mathcal{A}\|_2 = \max_{x^{(1)}, \ldots, x^{(r)}} \frac{\mathcal{A}\left(x^{(1)}, \ldots, x^{(r)}\right)}{\left\|x^{(1)}\right\|_2 \cdots \left\|x^{(r)}\right\|_2}$

Approximating a tensor

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Notation

A is an order-r tensor (e.g., a tensor with r modes)

We will measure the error: $\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_{F,2}$

Frobenius norm: $\|\mathcal{A}\|_F^2 = \sum_{j_1, j_2, \ldots, j_r} \mathcal{A}_{j_1, j_2, \ldots, j_r}^2$

Spectral norm: $\|\mathcal{A}\|_2 = \max_{x^{(1)}, \ldots, x^{(r)}} \frac{\mathcal{A}\left(x^{(1)}, \ldots, x^{(r)}\right)}{\left\|x^{(1)}\right\|_2 \cdots \left\|x^{(r)}\right\|_2}$

$\mathcal{A}\left(x^{(1)}, \ldots, x^{(r)}\right) = \sum_{j_1, \ldots, j_r} \mathcal{A}_{j_1, j_2, \ldots, j_r} x_{j_1}^{(1)} x_{j_2}^{(2)} \cdots x_{j_r}^{(r)}$

Approximating a tensor

Fundamental Question

Given a tensor $\mathcal{A}$ and an integer k find k rank-one tensors such that their sum is as "close" to $\mathcal{A}$ as possible.

Notation

$\mathcal{A}$ is an order-r tensor (e.g., a tensor with r modes)

We will measure the error: $\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_{F,2}$

Frobenius norm: $\|\mathcal{A}\|_F^2 = \sum_{j_1, j_2, \ldots, j_r} \mathcal{A}_{j_1, j_2, \ldots, j_r}^2$

Spectral norm: $\|\mathcal{A}\|_2 = \max_{x^{(1)}, \ldots, x^{(r)}} \frac{\mathcal{A}\left(x^{(1)}, \ldots, x^{(r)}\right)}{\left\|x^{(1)}\right\|_2 \cdots \left\|x^{(r)}\right\|_2}$

Equivalent to the corresponding matrix norms for r=2

Approximating a tensor: **negative results**

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Negative results

(***A*** is an order-***r*** tensor)

1. For ***r=3***, computing the minimal ***k*** such that ***A*** is exactly equal to the sum of rank-one components is NP-hard [Hastad '89, '90]

Approximating a tensor: **negative results**

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Negative results

(***A*** is an order-***r*** tensor)

1. For ***r=3***, computing the minimal ***k*** such that ***A*** is exactly equal to the sum of rank-one components is NP-hard [Hastad '89, '90]
2. For ***r=3***, identifying ***k*** rank-one components such that the Frobenius norm error of the approximation is minimized might not even have a solution (L.-H. Lim '04)

Approximating a tensor: **negative results**

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Negative results

(***A*** is an order-***r*** tensor)

1. For ***r=3***, computing the minimal ***k*** such that ***A*** is exactly equal to the sum of rank-one components is NP-hard [Hastad '89, '90]
2. For ***r=3***, identifying ***k*** rank-one components such that the Frobenius norm error of the approximation is minimized might not even have a solution (L.-H. Lim '04)
3. For ***r=3***, identifying ***k*** rank-one components such that the Frobenius norm error of the approximation is minimized (assuming such components exist) is NP-hard.

Approximating a tensor: **positive results!**

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Positive results ! **Both from a paper of Kannan et al in STOC '05.**

(A is an order-***r*** tensor)

1. **(Existence)** For any tensor ***A***, and any $\varepsilon > 0$, there exist at most $k=1/\varepsilon^2$ rank-one tensors such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

Approximating a tensor: **positive results!**

Fundamental Question

Given a tensor ***A*** and an integer ***k*** find ***k*** rank-one tensors such that their sum is as "close" to ***A*** as possible.

Positive results ! **Both from a paper of Kannan et al in STOC '05.**

(A is an order-***r*** tensor)

2. **(Algorithmic)** For any tensor ***A***, and any ***ε > 0***, we can find at most ***k=4/ε²*** rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \|\mathcal{A}\|_F$$

Time: $(n/\epsilon)^{O(1/\epsilon^4)}, \mathcal{A} \in \mathbb{R}^{n \times n \times \ldots \times n}$

The matrix case…

Matrix result

For any matrix ***A***, and any ***ε > 0***, we can find at most ***k=1/ε²*** rank-one matrices such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

The matrix case…

Matrix result

For any matrix A, and any $\varepsilon > 0$, we can find at most $k=1/\varepsilon^2$ rank-one matrices such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

To prove this, simply recall that the best rank k approximation to a matrix A is given by A_k (as computed by the SVD). But, by setting $k=1/\varepsilon^2$

$$\begin{aligned} \|A - A_k\|_2 &= \sigma_{k+1}(A) \\ &\leq \frac{1}{\sqrt{k+1}} \|A\|_F \\ &\leq \epsilon \|A\|_F \end{aligned}$$

The matrix case…

Matrix result

For any matrix ***A***, and any ***ε > 0***, we can find at most ***k=1/ε²*** rank-one matrices such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

From an **existential perspective**, the result is the same for matrices ***and*** higher order tensors.

From an **algorithmic perspective**, in the matrix case, the algorithm is ***(i)*** more efficient, ***(ii)*** returns fewer rank one components, and ***(iii)*** there is no failure probability.

Existential result: the proof

1. **(Existence)** For any tensor ***A***, and any ***ε > 0***, there exist at most ***k=1/ε²*** rank-one tensors such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

Proof

If $\|\mathcal{A}\|_2 \leq \epsilon \|\mathcal{A}\|_F$ then we are done. Otherwise, by the definition of the spectral norm of the tensor,

$$\exists \underbrace{x^{(1)} \ldots x^{(r)}}_{\text{w.l.o.g., unit norm}}, \text{such that } \mathcal{A}\left(x^{(1)}, \ldots, x^{(r)}\right) \geq \epsilon \|\mathcal{A}\|_F$$

$$\sum_{j_1, \ldots, j_r} \mathcal{A}_{j_1, j_2, \ldots, j_r} x_{j_1}^{(1)} x_{j_2}^{(2)} \cdots x_{j_r}^{(r)}$$

Existential result: the proof

1. **(Existence)** For any tensor ***A***, and any ***ε > 0***, there exist at most ***k=1/ε²*** rank-one tensors such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

Proof

Consider the tensor:

$$\mathcal{B} = \mathcal{A} - \underbrace{\left(\mathcal{A} \left(x^{(1)}, \ldots, x^{(r)} \right) \right)}_{\text{scalar}} x^{(1)} \otimes x^{(2)} \ldots \otimes x^{(r)}$$

We can prove (easily) that:

$$\|\mathcal{B}\|_F^2 = \|\mathcal{A}\|_F^2 - \left(\mathcal{A} \left(x^{(1)}, \ldots, x^{(r)} \right) \right)^2$$

Existential result: the proof

1. **(Existence)** For any tensor ***A***, and any ***ε > 0***, there exist at most ***k=1/ε²*** rank-one tensors such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \, \|\mathcal{A}\|_F$$

Proof

Now combine:

$$\left.\begin{array}{c} \|\mathcal{B}\|_F^2 = \|\mathcal{A}\|_F^2 - \left(\mathcal{A}\left(x^{(1)}, \ldots, x^{(r)}\right)\right)^2 \\ \mathcal{A}\left(x^{(1)}, \ldots, x^{(r)}\right) \geq \epsilon \|\mathcal{A}\|_F \end{array}\right\} \quad \|\mathcal{B}\|_F^2 \leq \left(1 - \epsilon^2\right) \|\mathcal{A}\|_F^2$$

Existential result: the proof

1. **(Existence)** For any tensor ***A***, and any ***ε > 0***, there exist at most ***k=1/ε²*** rank-one tensors such that

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

Proof

We now iterate this process using B instead of A. Since at every step we reduce the Frobenius norm of A, this process will eventually terminate.

The number of steps is at most ***k=1/ε²***, thus leading to ***k*** rank-one tensors.

Algorithmic result: outline

2. **(Algorithmic)** For any tensor ***A***, and any ***ε > 0***, we can find at most ***k=4/ε²*** rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

Time: $(n/\epsilon)^{O(1/\epsilon^4)}$, $\mathcal{A} \in \mathbb{R}^{n \times n \times \ldots \times n}$

Ideas:

For simplicity, focus on order-3 tensors. The only part of the existential proof that is not constructive, is how to identify unit vectors ***x***, ***y***, and ***z*** such that

$$\mathcal{A}(x, y, z) = \sum_{j_1, j_2, j_3} \mathcal{A}_{j_1, j_2, j_3} x_{j_1} y_{j_2} z_{j_3}$$

is maximized.

Algorithmic result: outline (cont'd)

2. **(Algorithmic)** For any tensor ***A***, and any ***ε > 0***, we can find at most ***k=4/ε²*** rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \|\mathcal{A}\|_F$$

Good news!

If ***x*** and ***y*** are known, then in order to maximize

$$\mathcal{A}(x, y, z) = \sum_{j_1, j_2, j_3} \mathcal{A}_{j_1, j_2, j_3} x_{j_1} y_{j_2} z_{j_3}$$

over all unit vectors ***x, y,*** and ***z,*** we can set ***z*** be the (normalized) vector whose j_3 entry is:

$$z_{j_3} = \sum_{j_1, j_2} A_{j_1 j_2 j_3} x_{j_1} y_{j_2} \quad \text{for all } j_3$$

Algorithmic result: outline (cont'd)

2. **(Algorithmic)** For any tensor ***A***, and any $\varepsilon > 0$, we can find at most $k=4/\varepsilon^2$ rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

Approximating ***z***…

Instead of computing the entries of ***z***, we approximate them by sub-sampling:

We draw a set ***S*** of random tuples (j_1, j_2) – we need roughly $1/\varepsilon^2$ such tuples – and we approximate the entries of ***z*** by using the tuples in ***S*** only!

$$z_{j_3} = \sum_{j_1, j_2} A_{j_1 j_2 j_3} x_{j_1} y_{j_2} \approx \sum_{(j_1, j_2) \in S} A_{j_1 j_2 j_3} x_{j_1} y_{j_2}$$

Algorithmic result: outline (cont'd)

2. **(Algorithmic)** For any tensor $\mathcal{A}$, and any $\varepsilon > 0$, we can find at most $k=4/\varepsilon^2$ rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \|\mathcal{A}\|_F$$

Weighted sampling…

Weighted sampling is used in order to pick the tuples (j_1, j_2).

More specifically,

$$\mathbf{Pr}\left[(j_1, j_2) \in S\right] = \frac{\sum_{j_3} \mathcal{A}^2_{j_1 j_2 j_3}}{\|\mathcal{A}\|_F^2}$$

Algorithmic result: outline (cont'd)

2. **(Algorithmic)** For any tensor $\mathcal{A}$, and any $\varepsilon > 0$, we can find at most $k=4/\varepsilon^2$ rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \|\mathcal{A}\|_F$$

Exhaustive search in a discretized interval…

We only need values for x_{j1} and y_{j2} in the set S.

We will exhaustively try "all" possible values (by placing a fine grid on the interval [-1,1]).

This leads to a number of trials that is exponential in $|S|$.

$$z_{j_3} = \sum_{j_1,j_2} A_{j_1 j_2 j_3} x_{j_1} y_{j_2} \approx \sum_{(j_1,j_2)\in S} A_{j_1 j_2 j_3} x_{j_1} y_{j_2}$$

Algorithmic result: outline (cont'd)

2. **(Algorithmic)** For any tensor ***A***, and any $\varepsilon > 0$, we can find at most $k=4/\varepsilon^2$ rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \, \|\mathcal{A}\|_F$$

Recursively figure out ***x*** and ***y***…

Each one of the possible values for x_{j1} and y_{j2} for (j_1, j_2) *in S*, leads to a possible vector ***z***.

We treat that vector as the true ***z***, and we try to figure out ***x*** and ***y*** recursively!

This is a smaller problem..

Algorithmic result: outline (cont'd)

2. **(Algorithmic)** For any tensor ***A***, and any ***ε > 0***, we can find at most ***k=4/ε²*** rank-one tensors such that with probability at least .75

$$\left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{C}^{(i)} \right\|_2 \leq \epsilon \left\| \mathcal{A} \right\|_F$$

Done!

Return the best ***x, y,*** and ***z***.

The running time is dominated by the cardinality of ***S***, which is not too bad assuming that ***ε*** is a constant…

The algorithm can also be generalized to higher order tensors.

Approximating Max- *r* -CSP problems

Max- *r* -CSP (=Max-SNP)

- The goal of the Kannan et al paper was to design PTAS (polynomial-time approximation schemes) for a large class of Max- *r* -CSP problems.
- Max- *r* -CSP problems are constraint satisfaction problems with *n* boolean variables and *m* constraints: each constraint is the ***logical OR*** of exactly *r* variables.
- **The goal is to maximize the number of satisfied constraints.**
- Max- *r* -CSP problems model a large number of problems, including Max-Cut, Bisection, Max-k-SAT, 3-coloring, Dense-k-subgraph, etc.
- Interestingly, tensors may be used to model Max- *r* -CSP as an optimization problem, and tensor decompositions help reduce its "dimensionality".

Approximating Max- *r* -CSP problems

Max- *r* -CSP (=Max-SNP)

- The goal of the Kannan et al paper was to design PTAS (polynomial-time approximation schemes) for a large class of Max- *r* -CSP problems.

- Max- *r* -CSP problems are constraint satisfaction problems with *n* boolean variables and *m* constraints: each constraint is the ***logical OR*** of exactly *r* variables.

- **The goal is to maximize the number of satisfied constraints.**

- Max- *r* -CSP problems model a large number of problems, including Max-Cut, Bisection, Max-k-SAT, 3-coloring, Dense-k-subgraph, etc.

- Interestingly, tensors may be used to model Max- *r* -CSP as an optimization problem, and tensor decompositions help reduce its "dimensionality".

See also:

Arora, Karger & Karpinski '95, Frieze & Kannan '96, Goldreich, Goldwasser & Ron '96, Alon, Vega, Kannan, & Karpinski '02, '03, Drineas, Kannan, & Mahoney, '05, '07.

Open problems

- **Similar error bounds for other norm combinations?**
  - What about Frobenius on both sides, or spectral on both sides?
  - Existential and/or algorithmic results are interesting.
  - Is it possible to get constant (or any) factor approximations in the case where the optimal solution exists?
- **Improved algorithmic results**
  - The exponential dependency on ε is totally impractical.
  - Provable algorithms would be preferable…