An Almost Optimal Algorithm for Computing Nonnegative Rank

Ankur Moitra
Institute for Advanced Study

January 8, 2013
inner−dimension

m M = m M

inner−dimension
\text{rank} \quad m \quad M \quad = \quad A \quad \text{inner-dimension} \quad W
rank

non-negative

inner-dimension

non-negative

m × n = m inner-dimension
M

= A

W
\( A = \text{inner-dimension} = \text{rank} = \text{non-negative} \)

\[ m \quad \begin{array}{c} \text{M} \end{array} \quad = \quad \begin{array}{c} \text{A} \\ \text{non-negative} \\ \text{inner-dimension} \end{array} \quad = \quad \begin{array}{c} \text{W} \end{array} \quad = \quad \text{non-negative} \]
Applications

- **Statistics and Machine Learning:**
  - extract *latent* relationships in data
  - image segmentation, text classification, information retrieval, collaborative filtering, ...
  
  [Lee, Seung], [Xu et al], [Hofmann], [Kumar et al], [Kleinberg, Sandler]

- **Combinatorics:**
  - extended formulation, log-rank conjecture
  
  [Yannakakis], [Lovász, Saks]

- **Physical Modeling:**
  - interaction of components is additive
  - visual recognition, environmetrics
Applications

- **Statistics and Machine Learning:**
  - extract *latent* relationships in data
  - image segmentation, text classification, information retrieval, collaborative filtering, ...
    - [Lee, Seung], [Xu et al], [Hofmann], [Kumar et al], [Kleinberg, Sandler]

- **Combinatorics:**
  - extended formulation, log-rank conjecture
    - [Yannakakis], [Lovász, Saks]
Applications

- **Statistics and Machine Learning:**
  - extract latent relationships in data
  - image segmentation, text classification, information retrieval, collaborative filtering, ...
  - [Lee, Seung], [Xu et al], [Hofmann], [Kumar et al], [Kleinberg, Sandler]

- **Combinatorics:**
  - extended formulation, log-rank conjecture
    - [Yannakakis], [Lovász, Saks]

- **Physical Modeling:**
  - interaction of components is additive
  - visual recognition, environmetrics
The Complexity of Nonnegative Rank

Vavasis: It is $\text{NP}$-hard to compute the nonnegative rank.

Cohen and Rothblum: The nonnegative rank can be computed in time $O(n^m r + m^r)$.

Arora, Ge, Kannan and Moitra: The nonnegative rank can be computed in time $O(n^m r)$ where $f(r) = O(2^{\sqrt{r}})$ and any algorithm that runs in time $O(n^m o(r))$ would yield a sub exponential time algorithm for 3-SAT.

Theorem: The nonnegative rank can be computed in time $O(n^m r^2)$.

...these algorithms are about an algebraic question, about how to best encode nonnegative rank as a system of polynomial inequalities
[Vavasis]: It is $NP$-hard to compute the nonnegative rank.
The Complexity of Nonnegative Rank

[Vavasis]: It is \( NP \)-hard to compute the nonnegative rank.

[Cohen and Rothblum]: The nonnegative rank can be computed in time \( (nm)^{O(nr+mr)} \).

Theorem: The nonnegative rank can be computed in time \( (nm)^{O(r^2)} \).
The Complexity of Nonnegative Rank

[Vavasis]: It is $\text{NP}$-hard to compute the nonnegative rank.

[Cohen and Rothblum]: The nonnegative rank can be computed in time $(nm)^{O(nr+mr)}$.

[Arora, Ge, Kannan and Moitra]: The nonnegative rank can be computed in time $(nm)^{f(r)}$ where $f(r) = O(2^r)$ and any algorithm that runs in time $(nm)^{o(r)}$ would yield a sub exponential time algorithm for 3-SAT.
[Vavasis]: It is $NP$-hard to compute the nonnegative rank.

[Cohen and Rothblum]: The nonnegative rank can be computed in time $(nm)^{O(nr+mr)}$.

[Arora, Ge, Kannan and Moitra]: The nonnegative rank can be computed in time $(nm)^{f(r)}$ where $f(r) = O(2^r)$ and any algorithm that runs in time $(nm)^{o(r)}$ would yield a sub exponential time algorithm for 3-SAT.

Theorem

The nonnegative rank can be computed in time $(nm)^{O(r^2)}$. 
The Complexity of Nonnegative Rank

[Vavasis]: It is $NP$-hard to compute the nonnegative rank.

[Cohen and Rothblum]: The nonnegative rank can be computed in time $(nm)^{O(nr+mr)}$.

[Arora, Ge, Kannan and Moitra]: The nonnegative rank can be computed in time $(nm)^{f(r)}$ where $f(r) = O(2^r)$ and any algorithm that runs in time $(nm)^{o(r)}$ would yield a sub exponential time algorithm for 3-SAT.

**Theorem**

The nonnegative rank can be computed in time $(nm)^{O(r^2)}$.

...these algorithms are about an algebraic question, about how to best encode nonnegative rank as a systems of polynomial inequalities
Is NMF Computable?

[Cohen, Rothblum]: Yes

DETOUR

Variables: entries in $A$ and $W$ ($nr + mr$ total)

Constraints: $A, W \geq 0$ and $AW = M$ (degree two)

Running time for a solver is exponential in the number of variables

Question

What is the smallest formulation, measured in the number of variables? Can we use only $f(r)$ variables?
Is NMF Computable?

[Cohen, Rothblum]: Yes
Is NMF Computable?

[Cohen, Rothblum]: Yes (DETOUR)
Semi-algebraic sets: $s$ polynomials, $k$ variables, Boolean function $B$

$$S = \{ x_1, x_2 \ldots x_k | B(\text{sgn}(f_1), \text{sgn}(f_2), \ldots \text{sgn}(f_s)) = \text{"true"} \}$$
Semi-algebraic sets: $s$ polynomials, $k$ variables, Boolean function $B$

$$S = \{ x_1, x_2 \ldots x_k | B(\text{sgn}(f_1), \text{sgn}(f_2), \ldots \text{sgn}(f_s)) = \text{"true"} \}$$

**Question**

*How many sign patterns arise (as $x_1, x_2, \ldots x_k$ range over $\mathbb{R}^k$)?*
Semi-algebraic sets: $s$ polynomials, $k$ variables, Boolean function $B$

$$S = \{ x_1, x_2 \ldots x_k | B(\text{sgn}(f_1), \text{sgn}(f_2), \ldots \text{sgn}(f_s)) = \text{"true"} \}$$

**Question**

*How many sign patterns arise (as $x_1, x_2, \ldots x_k$ range over $\mathbb{R}^k$)?*

Naive bound: $3^s$ (all of $\{-1, 0, 1\}^s$),
Semi-algebraic sets: $s$ polynomials, $k$ variables, Boolean function $B$

$$S = \{x_1, x_2...x_k \mid B(\text{sgn}(f_1), \text{sgn}(f_2), ...\text{sgn}(f_s)) = "true" \}$$

**Question**

How many sign patterns arise (as $x_1, x_2, ...x_k$ range over $\mathbb{R}^k$)?

Naive bound: $3^s$ (all of $\{-1, 0, 1\}^s$), [Milnor, Warren]: at most $(ds)^k$, where $d$ is the maximum degree
Semi-algebraic sets: \( s \) polynomials, \( k \) variables, Boolean function \( B \)

\[
S = \{ x_1, x_2 \ldots x_k \mid B(\text{sgn}(f_1), \text{sgn}(f_2), \ldots \text{sgn}(f_s)) = "true" \}
\]

**Question**

*How many sign patterns arise (as \( x_1, x_2, \ldots x_k \) range over \( \mathbb{R}^k \))?*

Naive bound: \( 3^s \) (all of \( \{-1, 0, 1\}^s \)), [Milnor, Warren]: at most \( (ds)^k \), where \( d \) is the maximum degree

In fact, best known algorithms (e.g. [Renegar]) for finding a point in \( S \) run in \( (ds)^{O(k)} \) time
Is NMF Computable?

[Cohen, Rothblum]: Yes (DETOUR)
Is NMF Computable?

[Cohen, Rothblum]: Yes (DETOUR)

- Variables: entries in $A$ and $W$ ($nr + mr$ total)
Is NMF Computable?

[Cohen, Rothblum]: Yes (**DETOUR**)

- Variables: entries in $A$ and $W$ ($nr + mr$ total)
- Constraints: $A, W \geq 0$ and $AW = M$ (degree two)
Is NMF Computable?

[Cohen, Rothblum]: Yes (DETOUR)

- Variables: entries in $A$ and $W$ ($nr + mr$ total)
- Constraints: $A, W \geq 0$ and $AW = M$ (degree two)

Running time for a solver is exponential in the number of variables.
[Cohen, Rothblum]: Yes (DETOUR)

- Variables: entries in $A$ and $W$ ($nr + mr$ total)
- Constraints: $A, W \geq 0$ and $AW = M$ (degree two)

Running time for a solver is exponential in the number of variables.

**Question**

*What is the smallest formulation, measured in the number of variables?*
Is NMF Computable?

[Cohen, Rothblum]: Yes (DETOUR)

- Variables: entries in $A$ and $W$ ($nr + mr$ total)
- Constraints: $A, W \geq 0$ and $AW = M$ (degree two)

Running time for a solver is exponential in the number of variables.

**Question**

*What is the smallest formulation, measured in the number of variables? Can we use only $f(r)$ variables?*
Is NMF Computable?

[Cohen, Rothblum]: Yes (DETOUR)

- Variables: entries in $A$ and $W$ ($nr + mr$ total)
- Constraints: $A, W \geq 0$ and $AW = M$ (degree two)

Running time for a solver is exponential in the number of variables.

**Question**

*What is the smallest formulation, measured in the number of variables? Can we use only $f(r)$ variables? $O(r^2)$ variables?*
Easy Case: $A$ has Full Column Rank (AGKM)
Easy Case: $A$ has Full Column Rank (AGKM)

\[
\begin{array}{c}
A^+ \\
pseudo-inverse
\end{array}
\hspace{1cm}
A
\]
Easy Case: $A$ has Full Column Rank (AGKM)

\[ A^+ \quad A = \begin{bmatrix} \text{pseudo−inverse} \\ A \end{bmatrix} = \begin{bmatrix} I_r \end{bmatrix} \]
Easy Case: $A$ has Full Column Rank (AGKM)

\[
A^+ = \begin{bmatrix} A \end{bmatrix}^{-1} + W
\]
Easy Case: $A$ has Full Column Rank (AGKM)

$A^+$

pseudo-inverse

$A$

$M$

$W$

$W$
Easy Case: $A$ has Full Column Rank (AGKM)

$A^+ = \text{pseudo-inverse}$

$A = \text{linearly independent}$

$M + W = \text{linearly independent}$
Easy Case: $A$ has Full Column Rank (AGKM)

$A^+$: pseudo-inverse

$A$: linearly independent

$M'$: change of basis

$M$: pseudo-inverse

$W$: linearly independent
Putting it Together: $2r^2$ Variables
Putting it Together: $2r^2$ Variables
Putting it Together: $2r^2$ Variables
Putting it Together: $2r^2$ Variables
Putting it Together: $2r^2$ Variables

Variables

non-negative?

non-negative?
Putting it Together: $2r^2$ Variables

$M$ \quad variables \quad M' \quad non-negative?

$S$ \quad M'' \quad \Rightarrow \quad A \quad (\bullet) (\bullet) \quad non-negative?
Putting it Together: $2r^2$ Variables

$$T \quad M'$$

variables

$$W \quad \text{non-negative?}$$

$$A \quad (\bullet) (\bullet) \quad M$$

non-negative?

$$M'$$

$$\Rightarrow$$

$$S$$

$$M''$$

$$M$$
Most interesting case in e.g. extended formulations:

\[ \#\text{facets}\{\text{conv}(A)\} \gg \#\text{vertices}\{\text{conv}(A)\} \]

which can only happen if (say) \( \text{rank}(A) = r/2 \)
Most interesting case in e.g. extended formulations:

\[ \#\text{facets}\{\text{conv}(A)\} \gg \#\text{vertices}\{\text{conv}(A)\} \]

which can only happen if (say) \( \text{rank}(A) = r/2 \)

**Question**

*Can we still find the rows of \( W \) from many linear transformations of rows of \( M \)?*
\[ T_1 = \left( A_1 A_2 A_3 \right)^+ \]
\[ T_1 = (A_1 A_2 A_3)^+ \]
\[ T_2 = (A_2 A_3 A_4)^+ \]
\[ \begin{align*}
T_1 &= (A_1A_2A_3)^+ \\
T_2 &= (A_2A_3A_4)^+
\end{align*} \]
\[
A_3 A_2 = (A_1 A_2 A_3)^+ \\
A_3 = (A_2 A_3 A_4)^+
\]
Most interesting case in e.g. extended formulations:

\[
\#\text{facets}\{\text{conv}(A)\} \gg \#\text{vertices}\{\text{conv}(A)\}
\]

which can only happen if (say) \( \text{rank}(A) = r/2 \)

**Question**

*Can we still find the rows of \( W \) from many linear transformations of rows of \( M \)?*
Most interesting case in e.g. extended formulations:

\[ \#\text{facets}\{\text{conv}(A)\} \gg \#\text{vertices}\{\text{conv}(A)\} \]

which can only happen if (say) \( \text{rank}(A) = r/2 \)

**Question**

*Can we still find the rows of \( W \) from many linear transformations of rows of \( M \)?*

This could require exponentially many \( (2^r) \) linear transformations (e.g. covering the cross polytope by simplices)
Most interesting case in e.g. extended formulations:

\[ \#\text{facets}\{\text{conv}(A)\} \gg \#\text{vertices}\{\text{conv}(A)\} \]

which can only happen if (say) \( \text{rank}(A) = r/2 \)

**Question**

*Can we still find the rows of \( W \) from *many* linear transformations of rows of \( M \)?*

This could require exponentially many \( (2^r) \) linear transformations (e.g. covering the cross polytope by simplices)

**Key**

*These linear transformations can be defined using a common set of \( r^2 \) variables!*
Goal

*Encode nonnegative rank as a semi-algebraic set with $2r^2$ variables*

We give a new normal form for nonnegative matrix factorization (that uses exponentially many $r \times r$ unknown linear transformations).

We use Cramer's Rule to write all of these linear transformations using a common set of $r^2$ variables.

These transformations recover the factorization $A, W$, and we can check that it is a valid nonnegative matrix factorization.

**Theorem**

The nonnegative rank can be computed in time $O(r^2)$. 
Goal

*Encode nonnegative rank as a semi-algebraic set with $2r^2$ variables*

(i.e. the set is non-empty iff $\text{rank}^+(M) \leq r$)
Goal

Encode nonnegative rank as a semi-algebraic set with $2r^2$ variables

(i.e. the set is non-empty iff $\text{rank}^+(M) \leq r$)

- We give a new **normal form** for nonnegative matrix factorization (that uses exponentially many $r \times r$ unknown linear transformations)
Goal

Encode nonnegative rank as a semi-algebraic set with $2r^2$ variables

(i.e. the set is non-empty iff $\text{rank}^+(M) \leq r$)

- We give a new normal form for nonnegative matrix factorization (that uses exponentially many $r \times r$ unknown linear transformations)
- We use Cramer’s Rule to write all of these linear transformations using a common set of $r^2$ variables

Theorem

The nonnegative rank can be computed in time $O(r^2)$. 
Goal

*Encode nonnegative rank as a semi-algebraic set with $2r^2$ variables* (i.e. the set is non-empty iff $\text{rank}^+(M) \leq r$)

- We give a new **normal form** for nonnegative matrix factorization (that uses exponentially many $r \times r$ unknown linear transformations)

- We use **Cramer’s Rule** to write all of these linear transformations using a common set of $r^2$ variables

- These transformations recover the factorization $A, W$, and we can check that it is a valid nonnegative matrix factorization

**Theorem**

The nonnegative rank can be computed in time $O(r^2)$. 
Goal

*Encode nonnegative rank as a semi-algebraic set with $2r^2$ variables* (i.e. the set is non-empty iff $\text{rank}^+(M) \leq r$)

- We give a new **normal form** for nonnegative matrix factorization (that uses exponentially many $r \times r$ unknown linear transformations)
- We use **Cramer’s Rule** to write all of these linear transformations using a common set of $r^2$ variables
- These transformations recover the factorization $A, W$, and we can check that it is a valid nonnegative matrix factorization

**Theorem**

*The nonnegative rank can be computed in time $(nm)^{O(r^2)}$.*
Concluding Remarks

This algorithm is based on answering a purely algebraic question: How many variables do we need in a semi-algebraic set to encode nonnegative rank?
Concluding Remarks

This algorithm is based on answering a purely algebraic question: How many variables do we need in a semi-algebraic set to encode nonnegative rank?

**Question**

*Are there other examples of a better understanding of the expressive power of semi-algebraic sets can lead to a new algorithm?*
Concluding Remarks

This algorithm is based on answering a purely algebraic question: How many variables do we need in a semi-algebraic set to encode nonnegative rank?

Question

Are there other examples of a better understanding of the expressive power of semi-algebraic sets can lead to a new algorithm?

Similar: recent work [Anandkumar et al] on topic modeling is based on understanding how many moments are needed to find the parameters.
This algorithm is based on answering a purely algebraic question: How many variables do we need in a semi-algebraic set to encode nonnegative rank?

Question

Are there other examples of a better understanding of the expressive power of semi-algebraic sets can lead to a new algorithm?

Similar: recent work [Anandkumar et al] on topic modeling is based on understanding how many moments are needed to find the parameters.

Observation

The number of variables plays an analogous role to VC-dimension
Concluding Remarks

This algorithm is based on answering a purely algebraic question: How many variables do we need in a semi-algebraic set to encode nonnegative rank?

Question

Are there other examples of a better understanding of the expressive power of semi-algebraic sets can lead to a new algorithm?

Similar: recent work [Anandkumar et al] on topic modeling is based on understanding how many moments are needed to find the parameters.

Observation

The number of variables plays an analogous role to VC-dimension

Is there an elementary proof of the Milnor-Warren bound?
Any Questions?
Thanks!