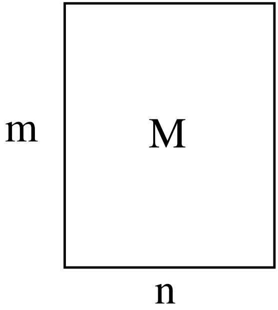
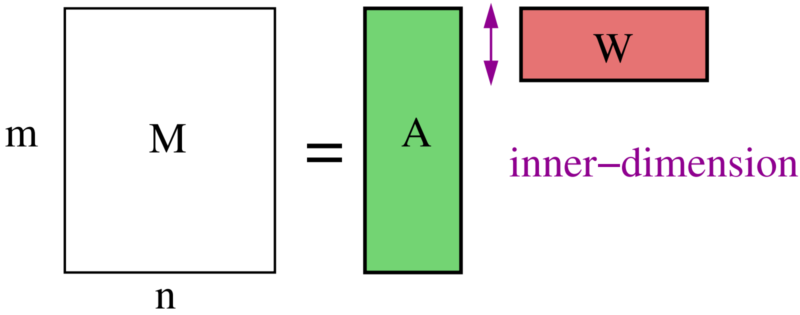
Computing a Nonnegative Matrix Factorization – Provably

Ankur Moitra, IAS

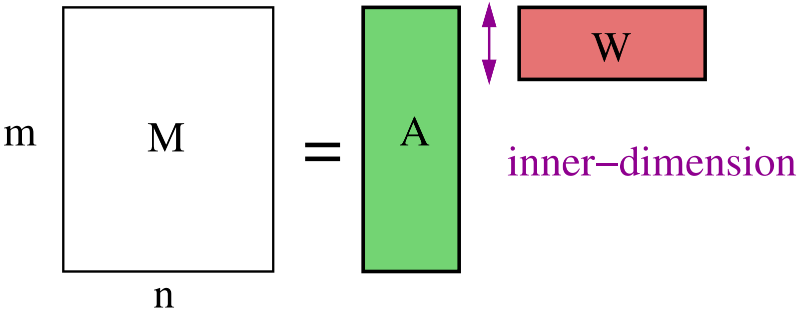
joint work with Sanjeev Arora, Rong Ge and Ravi Kannan

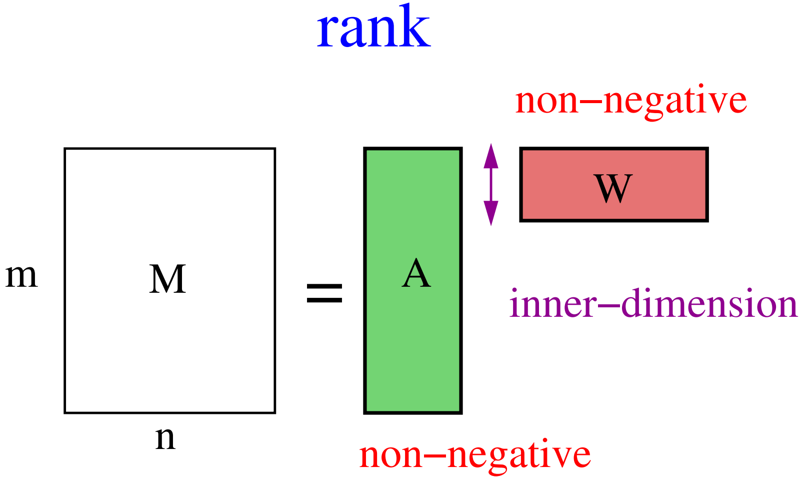
June 20, 2012

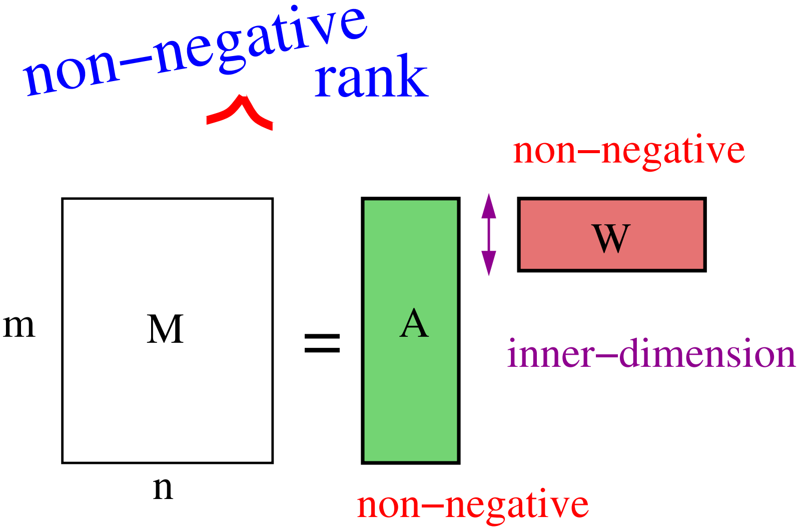

m
M
n


m
M
n
=
A
W
inner–dimension

rank

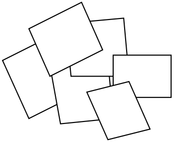



rank
non–negative
W
m
M
=
A
inner–dimension
n
non–negative

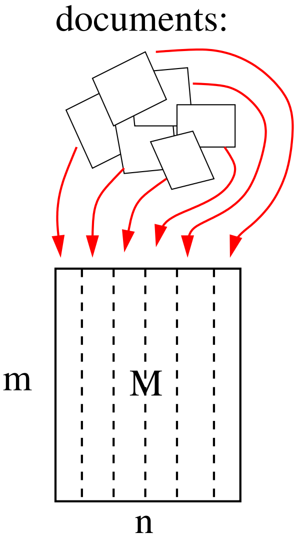

non–negative
rank
non–negative
m
M
=
A
W
inner–dimension
n
non–negative

Information Retrieval

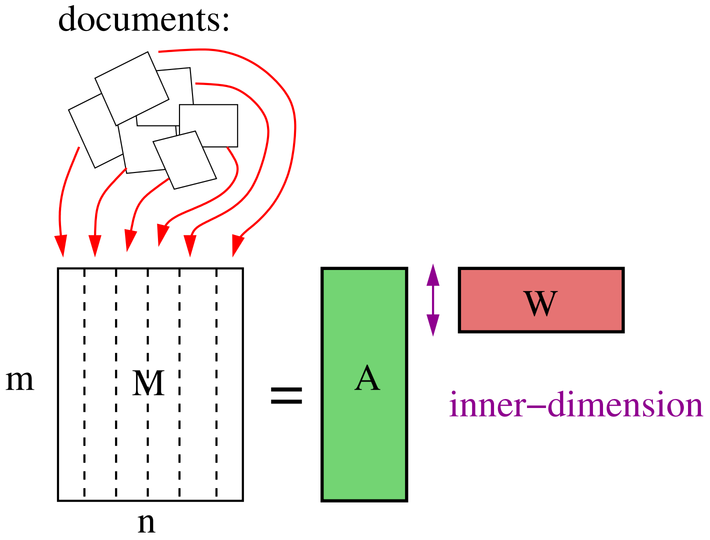
documents:



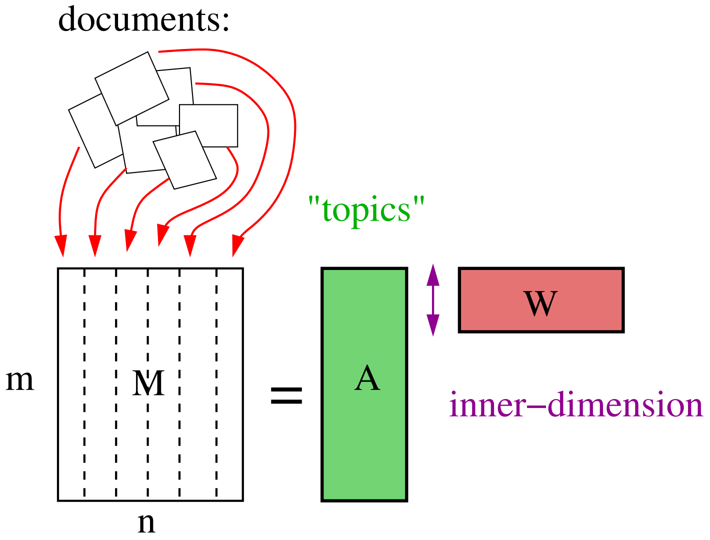
Information Retrieval



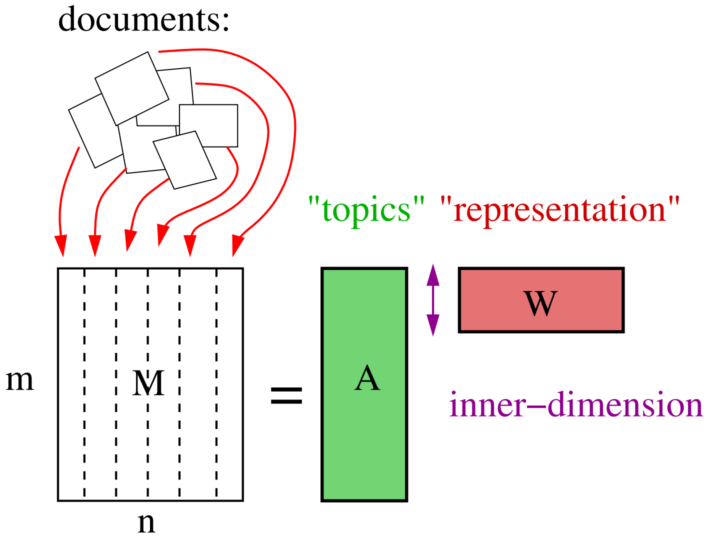
Information Retrieval



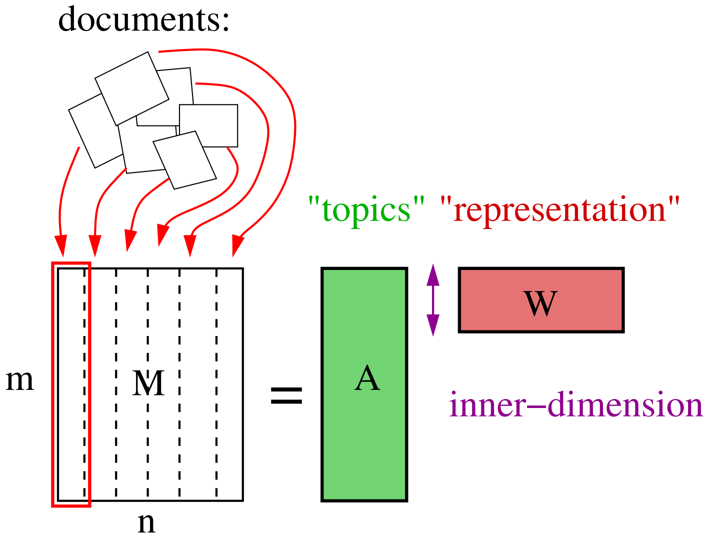
Information Retrieval



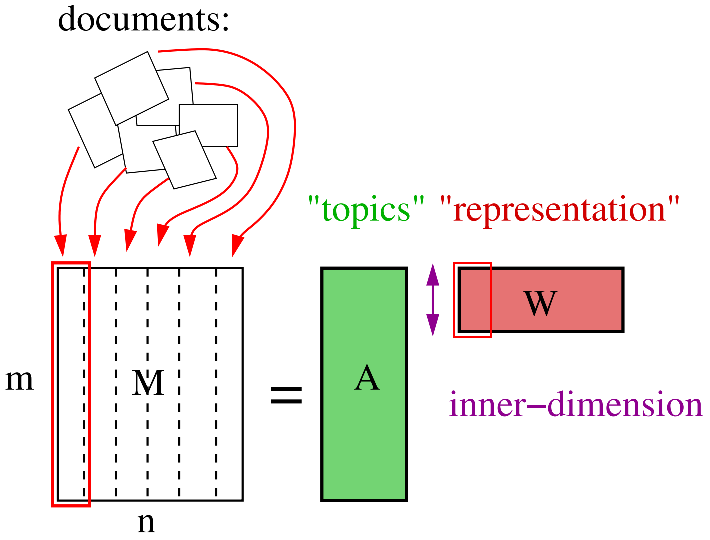
Information Retrieval



Information Retrieval



Information Retrieval



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]

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

Local Search: Given A, compute W, compute A, ...

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Hardness of NMF

Theorem (Vavasis)

NMF is NP-hard to compute

Hardness of NMF

Theorem (Vavasis)

NMF is NP-hard to compute

Hence it is unlikely that there is an exact algorithm that runs in time polynomial in n, m and r

Hardness of NMF

Theorem (Vavasis)

NMF is NP-hard to compute

Hence it is unlikely that there is an exact algorithm that runs in time polynomial in n, m and r

Question

Should we expect r to be large?

Hardness of NMF

Theorem (Vavasis)

NMF is NP-hard to compute

Hence it is unlikely that there is an exact algorithm that runs in time polynomial in n, m and r

Question

Should we expect r to be large?

What if you gave me a collection of 100 documents, and I told you there are 75 topics?

Hardness of NMF

Theorem (Vavasis)

NMF is NP-hard to compute

Hence it is unlikely that there is an exact algorithm that runs in time polynomial in n, m and r

Question

Should we expect r to be large?

What if you gave me a collection of 100 documents, and I told you there are 75 topics?

How quickly can we solve NMF if r is small?

The Worst-Case Complexity of NMF

Theorem (Arora, Ge, Moitra, Kannan)

There is an $(nm)^{O(r^2)}$ *time exact algorithm for NMF*

The Worst-Case Complexity of NMF

Theorem (Arora, Ge, Moitra, Kannan)

There is an $(nm)^{O(r^2)}$ time exact algorithm for NMF

Previously, the fastest (provable) algorithm for $r = 3$ ran in time exponential in n and m

The Worst-Case Complexity of NMF

Theorem (Arora, Ge, Moitra, Kannan)

There is an $(nm)^{O(r^2)}$ time exact algorithm for NMF

Previously, the fastest (provable) algorithm for $r = 3$ ran in time exponential in n and m

Can we improve the exponential dependence on r?

The Worst-Case Complexity of NMF

Theorem (Arora, Ge, Moitra, Kannan)

There is an $(nm)^{O(r^2)}$ time exact algorithm for NMF

Previously, the fastest (provable) algorithm for $r = 3$ ran in time exponential in n and m

Can we improve the exponential dependence on r?

Theorem (Arora, Ge, Moitra, Kannan)

An exact algorithm for NMF that runs in time $(nm)^{o(r)}$ would yield a sub-exponential time algorithm for 3-SAT

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Question

Can we give a theoretical explanation for why simple heuristics are so effective?

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Question

Can we give a theoretical explanation for why simple heuristics are so effective?

(What distinguishes a realistic input from an artificial one?)

Separability [Donoho, Stodden], Reinterpreted

- Each topic has an **anchor word**, and any document that contains this word is very likely to be (at least partially) about the corresponding topic

Separability [Donoho, Stodden], Reinterpreted

- Each topic has an **anchor word**, and any document that contains this word is very likely to be (at least partially) about the corresponding topic

  e.g. personal finance $\leftarrow$ 401k, baseball $\leftarrow$ outfield, ...

Separability [Donoho, Stodden], Reinterpreted

- Each topic has an **anchor word**, and any document that contains this word is very likely to be (at least partially) about the corresponding topic

  e.g. personal finance $\leftarrow$ $401k$, baseball $\leftarrow$ outfield, ...

- A document can contain no anchor words, but when one occurs it is a strong indicator

Separability [Donoho, Stodden], Reinterpreted

- Each topic has an **anchor word**, and any document that contains this word is very likely to be (at least partially) about the corresponding topic

  e.g. personal finance $\leftarrow$ 401k, baseball $\leftarrow$ outfield, ...

- A document can contain no anchor words, but when one occurs it is a strong indicator

Observation (Blei)

This condition is met by topics found on real data, say, by local search

Separability [Donoho, Stodden], Reinterpreted

- Each topic has an **anchor word**, and any document that contains this word is very likely to be (at least partially) about the corresponding topic

  e.g. personal finance $\leftarrow$ $401k$, baseball $\leftarrow$ outfield, ...

- A document can contain no anchor words, but when one occurs it is a strong indicator

Observation (Blei)

This condition is met by topics found on real data, say, by local search

Separability was introduced to understand when NMF is unique – Is it enough to make NMF easy?

Beyond Worst-Case Analysis

Theorem (Arora, Ge, Kannan, Moitra)

There is a polynomial time exact algorithm for NMF when the topic matrix A is separable

Beyond Worst-Case Analysis

Theorem (Arora, Ge, Kannan, Moitra)

There is a polynomial time exact algorithm for NMF when the topic matrix A is separable

What if documents do not contain many words compared to the dictionary? (e.g. we are given samples from M)

Beyond Worst-Case Analysis

Theorem (Arora, Ge, Kannan, Moitra)

There is a polynomial time exact algorithm for NMF when the topic matrix A is separable

What if documents do not contain many words compared to the dictionary? (e.g. we are given samples from M)

In fact, the above algorithm can be made robust to noise:

Theorem (Arora, Ge, Moitra)

There is a polynomial time algorithm for learning a separable topic matrix A in various probabilistic models - e.g. LDA, CTM

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Question

Can we give a theoretical explanation for why simple heuristics are so effective?

(What distinguishes a realistic input from an artificial one?)

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Question

Can we give a theoretical explanation for why simple heuristics are so effective?

(What distinguishes a realistic input from an artificial one?)

Is NMF Computable?

Is NMF Computable?

[Cohen, Rothblum]: Yes

Is NMF Computable?

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

Semi-algebraic sets: s polynomials, k variables, Boolean function B

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

Semi-algebraic sets: s polynomials, k variables, Boolean function B

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

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

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

Question

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

Naive bound: 3^s (all of $\{-1, 0, 1\}^s$),

Semi-algebraic sets: s polynomials, k variables, Boolean function B

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

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

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

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 $\{-1, 0, 1\}^s$), [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**

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?

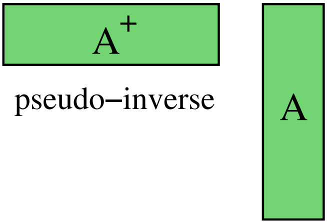
Reducing the Number of Variables

Reducing the Number of Variables

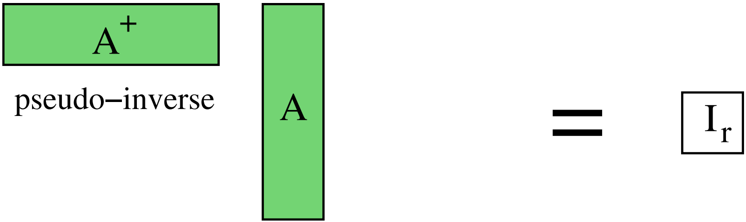
- Easy: If A has full rank, then $f(r) = 2r^2$

Easy Case: A has Full Column Rank

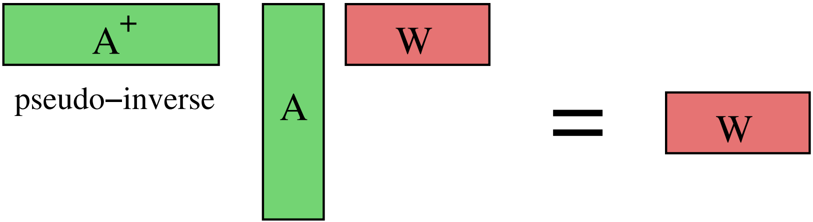
Easy Case: A has Full Column Rank



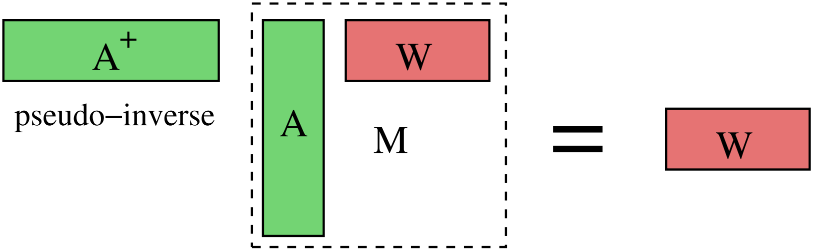
Easy Case: A has Full Column Rank



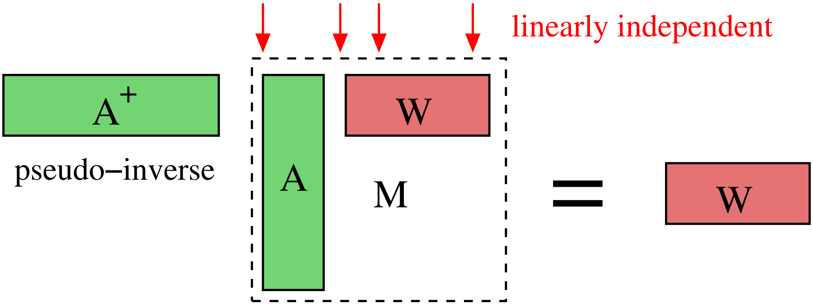
Easy Case: A has Full Column Rank



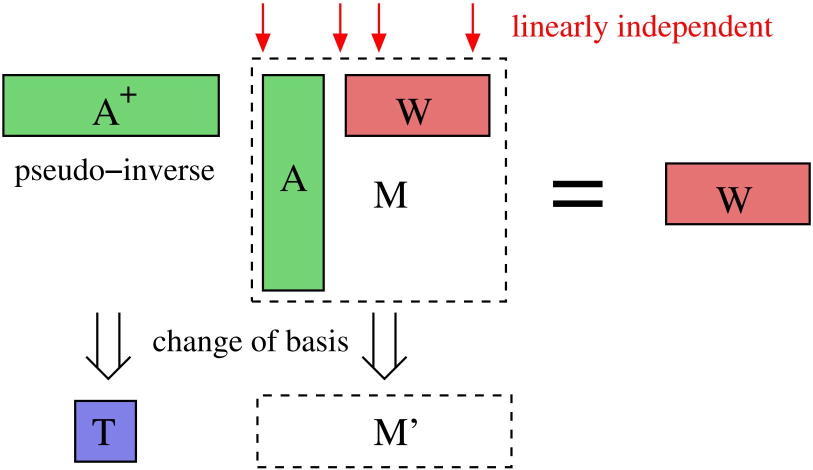
Easy Case: A has Full Column Rank



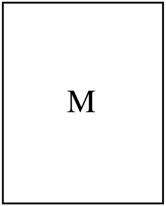
Easy Case: A has Full Column Rank



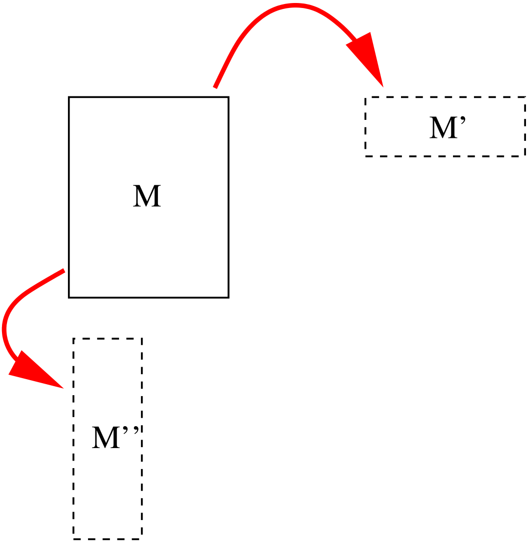
Easy Case: A has Full Column Rank



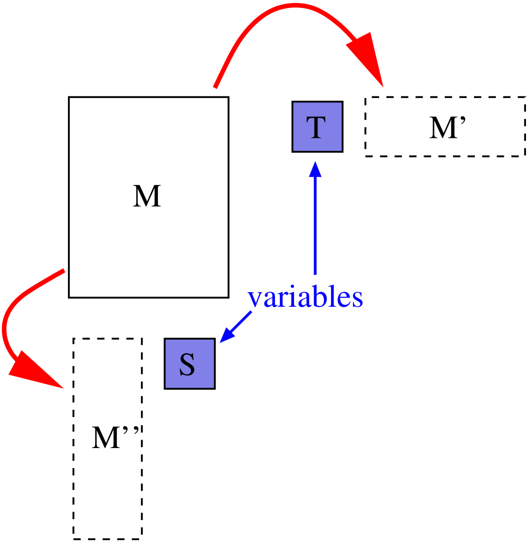
Putting it Together: $2r^2$ Variables



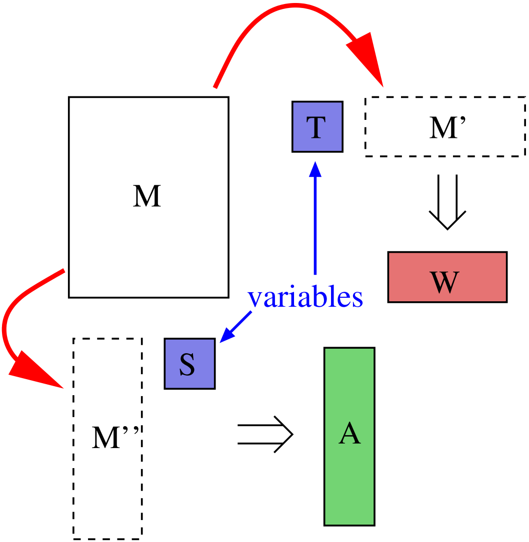
Putting it Together: $2r^2$ Variables



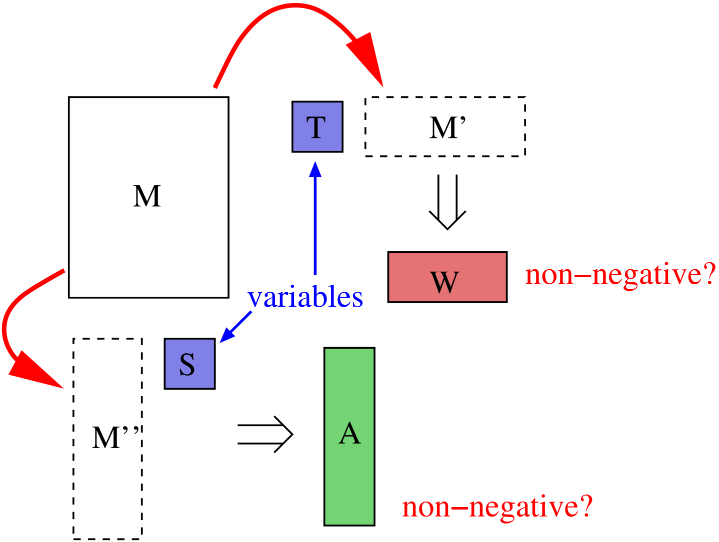
Putting it Together: $2r^2$ Variables



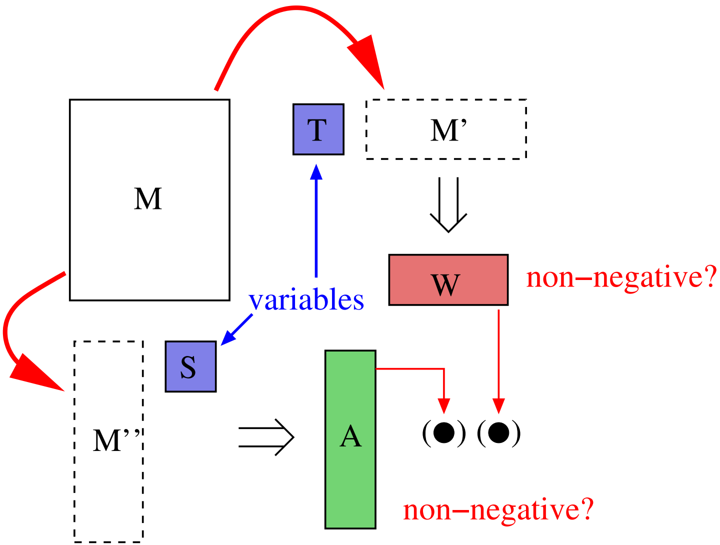
Putting it Together: $2r^2$ Variables



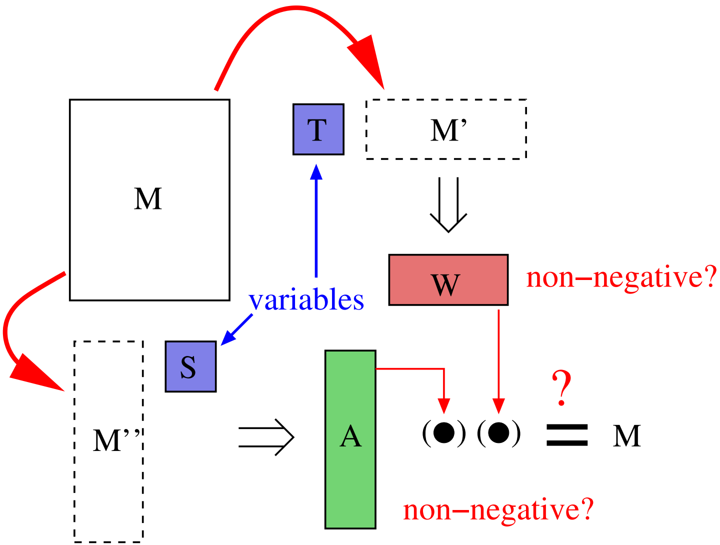
Putting it Together: $2r^2$ Variables



Putting it Together: $2r^2$ Variables



Putting it Together: $2r^2$ Variables



Reducing the Number of Variables

- Easy: If A has full rank, then $f(r) = 2r^2$

Reducing the Number of Variables

- Easy: If A has full rank, then $f(r) = 2r^2$
- [Arora, Ge, Kannan, Moitra]: In general, $f(r) = 2r^2 2^r$, which is constant for $r = O(1)$

Reducing the Number of Variables

- Easy: If A has full rank, then $f(r) = 2r^2$
- [Arora, Ge, Kannan, Moitra]: In general, $f(r) = 2r^2 2^r$, which is constant for $r = O(1)$
- [Moitra]: In general, $f(r) = 2r^2$ (using a normal form)

Reducing the Number of Variables

- Easy: If A has full rank, then $f(r) = 2r^2$
- [Arora, Ge, Kannan, Moitra]: In general, $f(r) = 2r^2 2^r$, which is constant for $r = O(1)$
- [Moitra]: In general, $f(r) = 2r^2$ (using a normal form)

Corollary

There is an $(nm)^{O(r^2)}$ time algorithm for NMF

Reducing the Number of Variables

- Easy: If A has full rank, then $f(r) = 2r^2$
- [Arora, Ge, Kannan, Moitra]: In general, $f(r) = 2r^2 2^r$, which is constant for $r = O(1)$
- [Moitra]: In general, $f(r) = 2r^2$ (using a normal form)

Corollary

There is an $(nm)^{O(r^2)}$ time algorithm for NMF

In fact, **any** $(nm)^{o(r)}$ time algorithm would yield a $2^{o(n)}$ time algorithm for 3-SAT

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Question

Can we give a theoretical explanation for why simple heuristics are so effective?

(What distinguishes a realistic input from an artificial one?)

Local Search: Given A, compute W, compute A, ...

- Known to fail on worst-case inputs (stuck in local minima)
- Highly sensitive to cost function, regularization, update procedure

Question (theoretical)

Is there an algorithm that (provably) works on all inputs?

Question

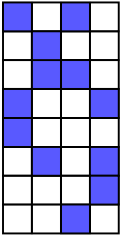
Can we give a theoretical explanation for why simple heuristics are so effective?

(What distinguishes a realistic input from an artificial one?)

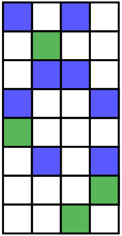
Separable Instances

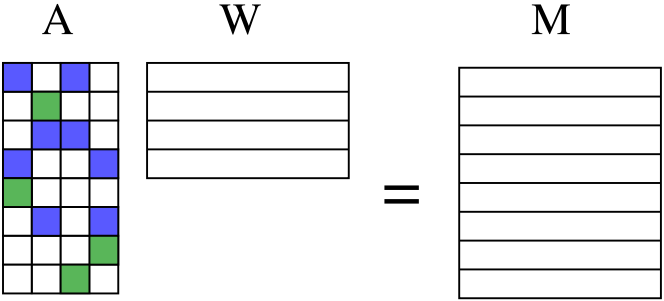
Recall: For each topic, there is some (anchor) word that only appears in this topic

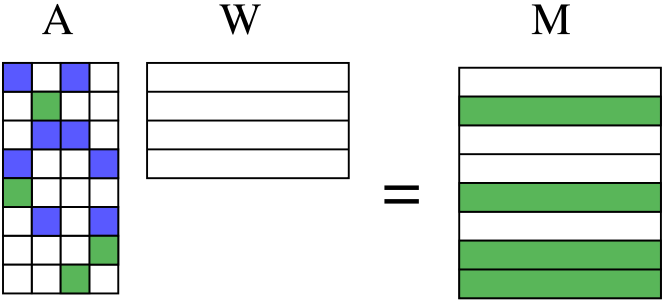
A



A




A
W
=
M


A
W
=
M

Separable Instances

Recall: For each topic, there is some (anchor) word that only appears in this topic

Separable Instances

Recall: For each topic, there is some (anchor) word that only appears in this topic

Observation

Rows of W appear as (scaled) rows of M

Separable Instances

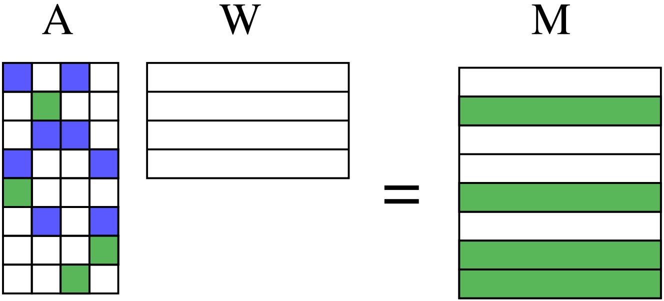
Recall: For each topic, there is some (anchor) word that only appears in this topic

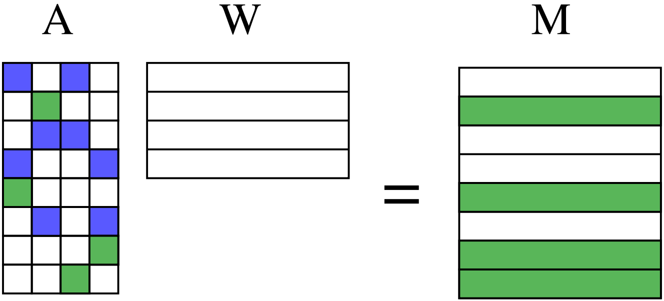
Observation

Rows of W appear as (scaled) rows of M

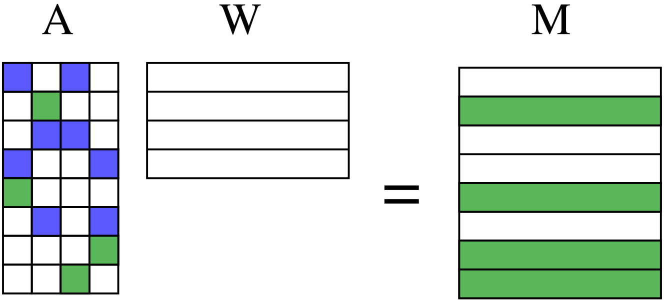
Question

How can we identify anchor words?

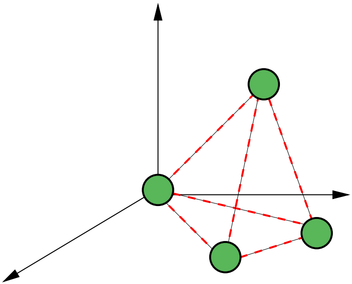

A
W
=
M

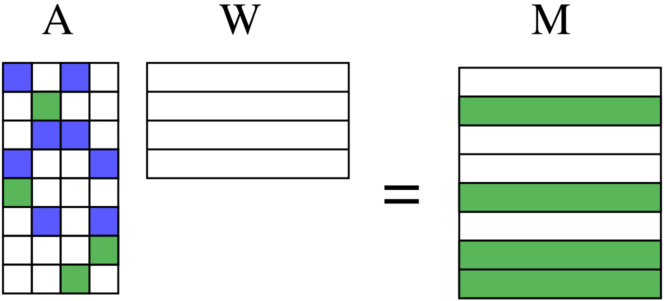


brute force: n^r

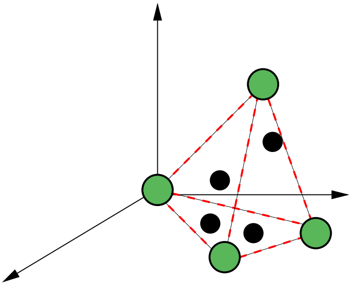


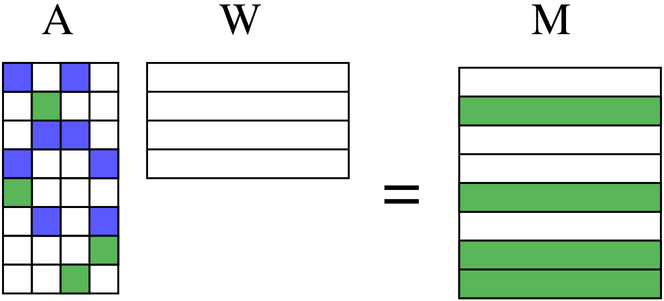
brute force: n^r



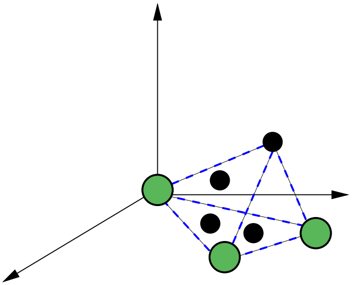


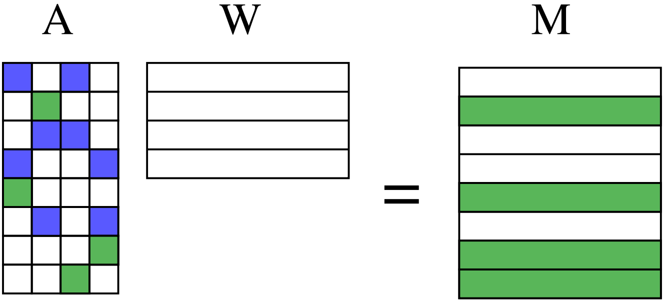
brute force: n^r



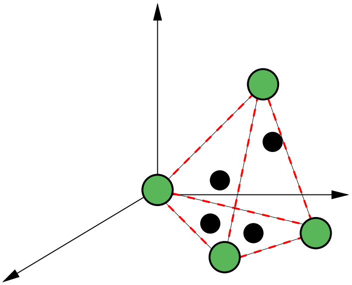


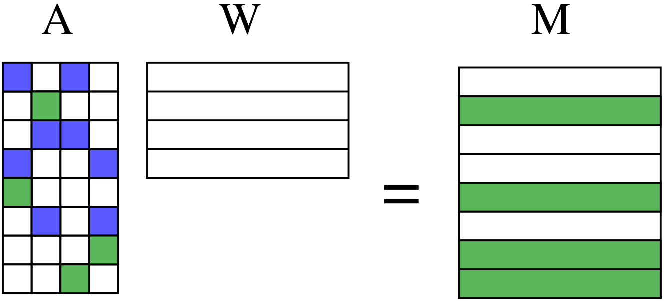
brute force: n^r



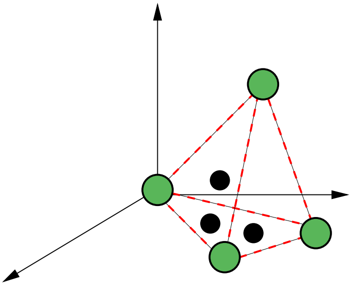


brute force: n^r





brute force: n^r



Separable Instances

Recall: For each topic, there is some (anchor) word that only appears in this topic

Observation

Rows of W appear as (scaled) rows of M

Question

How can we identify anchor words?

Separable Instances

Recall: For each topic, there is some (anchor) word that only appears in this topic

Observation

Rows of W appear as (scaled) rows of M

Question

How can we identify anchor words?

Removing a row from M strictly changes the convex hull iff it is an anchor word

Separable Instances

Recall: For each topic, there is some (anchor) word that only appears in this topic

Observation

Rows of W appear as (scaled) rows of M

Question

How can we identify anchor words?

Removing a row from M strictly changes the convex hull iff it is an anchor word

Hence we can identify all the anchor words via linear programming

Separable Instances

Recall: For each topic, there is some (anchor) word that only appears in this topic

Observation

Rows of W appear as (scaled) rows of M

Question

How can we identify anchor words?

Removing a row from M strictly changes the convex hull iff it is an anchor word

Hence we can identify all the anchor words via linear programming (can be made robust to noise)

Topic Models

- Topic matrix A, generate W stochastically

Topic Models

- Topic matrix A, generate W stochastically
- For each document (column in $M = AW$) sample N words

Topic Models

- Topic matrix A, generate W stochastically
- For each document (column in $M = AW$) sample N words
- e.g. Latent Dirichlet Allocation (LDA), Correlated Topic Model (CTM), Pachinko Allocation Model (PAM), ...

Topic Models

- Topic matrix A, generate W stochastically
- For each document (column in $M = AW$) sample N words
- e.g. Latent Dirichlet Allocation (LDA), Correlated Topic Model (CTM), Pachinko Allocation Model (PAM), ...

Question

Can we estimate A, given random samples from M?

Topic Models

- Topic matrix A, generate W stochastically
- For each document (column in $M = AW$) sample N words
- e.g. Latent Dirichlet Allocation (LDA), Correlated Topic Model (CTM), Pachinko Allocation Model (PAM), ...

Question

Can we estimate A, given random samples from M?

Yes! [Arora, Ge, Moitra] we give a provable algorithm based on (noise-tolerant) NMF

Advertisement: Sanjeev will talk about this here in July

Concluding Remarks

This is just part of a broader agenda:

Question

When is machine learning **provably** *easy?*

Concluding Remarks

This is just part of a broader agenda:

Question

When is machine learning **provably** *easy?*

Often, theoretical models for learning are too hard or focus on mistake bounds (e.g. PAC)

Concluding Remarks

This is just part of a broader agenda:

Question

When is machine learning **provably** *easy?*

Often, theoretical models for learning are too hard or focus on mistake bounds (e.g. PAC)

Question

Will this involve a better understanding of real data?

Concluding Remarks

This is just part of a broader agenda:

Question

When is machine learning **provably** *easy?*

Often, theoretical models for learning are too hard or focus on mistake bounds (e.g. PAC)

Question

Will this involve a better understanding of real data? A better understanding of popular algorithms in machine learning?

Concluding Remarks

This is just part of a broader agenda:

Question

When is machine learning **provably** *easy?*

Often, theoretical models for learning are too hard or focus on mistake bounds (e.g. PAC)

Question

Will this involve a better understanding of real data? A better understanding of popular algorithms in machine learning? Both?

Concluding Remarks

This is just part of a broader agenda:

Question

When is machine learning **provably** *easy?*

Often, theoretical models for learning are too hard or focus on mistake bounds (e.g. PAC)

Question

Will this involve a better understanding of real data? A better understanding of popular algorithms in machine learning? Both?

Some interesting problems worth further investigation: Topic Models, Independent Component Analysis, Graphical Models, Deep Learning

Thanks!