Tensors and Optimization

Ravi Kannan

September 16, 2013

Reducing MAX-r-CSP to Tensor Optimization

- MAX-3-SAT: Clause: $(\bar{x}_i + x_j + x_k)$. Given a list of 3-clauses on n variables, find the assignment maximizing the number of clauses satisfied.

Reducing MAX-r-CSP to Tensor Optimization

- MAX-3-SAT: Clause: $(\bar{x}_i + x_j + x_k)$. Given a list of 3-clauses on n variables, find the assignment maximizing the number of clauses satisfied.
- Set up 8 $n \times n \times n$ tensors: $A^{(1)}, A^{(2)}, \ldots, A^{(8)}$, where

Reducing MAX-r-CSP to Tensor Optimization

- MAX-3-SAT: Clause: $(\bar{x}_i + x_j + x_k)$. Given a list of 3-clauses on n variables, find the assignment maximizing the number of clauses satisfied.
- Set up 8 $n \times n \times n$ tensors: $A^{(1)}, A^{(2)}, \ldots, A^{(8)}$, where
  - $A^{(1)}_{ijk}$ = number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (1, 1, 1)$

Reducing MAX-r-CSP to Tensor Optimization

- MAX-3-SAT: Clause: $(\bar{x}_i + x_j + x_k)$. Given a list of 3-clauses on n variables, find the assignment maximizing the number of clauses satisfied.
- Set up 8 $n \times n \times n$ tensors: $A^{(1)}, A^{(2)}, \ldots, A^{(8)}$, where
  - $A^{(1)}_{ijk}$ = number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (1, 1, 1)$
  - $A^{(2)}_{ijk}$ = number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (0, 1, 1)$......

Reducing MAX-r-CSP to Tensor Optimization

- MAX-3-SAT: Clause: $(\bar{x}_i + x_j + x_k)$. Given a list of 3-clauses on n variables, find the assignment maximizing the number of clauses satisfied.
- Set up 8 $n \times n \times n$ tensors: $A^{(1)}, A^{(2)}, \ldots, A^{(8)}$, where
  - $A^{(1)}_{ijk}$ = number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (1, 1, 1)$
  - $A^{(2)}_{ijk}$ = number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (0, 1, 1)$......
  - Each clause contributes to 7 of the 8 tensors.

Reducing MAX-r-CSP to Tensor Optimization

- MAX-3-SAT: Clause: $(\bar{x}_i + x_j + x_k)$. Given a list of 3-clauses on n variables, find the assignment maximizing the number of clauses satisfied.
- Set up 8 $n \times n \times n$ tensors: $A^{(1)}, A^{(2)}, \ldots, A^{(8)}$, where
  - $A^{(1)}_{ijk}$ = number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (1, 1, 1)$
  - $A^{(2)}_{ijk}$ = number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (0, 1, 1)$......
  - Each clause contributes to 7 of the 8 tensors.
- Maximize $\sum_{ijk} A^{(1)}_{ijk} x_i x_j x_k + \sum_{ijk} A^{(2)}_{ijk} (1 - x_i) x_j x_k + \cdots$.

Reducing MAX-r-CSP to Tensor Optimization

- MAX-3-SAT: Clause: $(\bar{x}_i + x_j + x_k)$. Given a list of 3-clauses on n variables, find the assignment maximizing the number of clauses satisfied.
- Set up 8 $n \times n \times n$ tensors: $A^{(1)}, A^{(2)}, \ldots, A^{(8)}$, where
  - $A^{(1)}_{ijk} =$ number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (1, 1, 1)$
  - $A^{(2)}_{ijk} =$ number of clauses satisfied by the assignment $(x_i, x_j, x_k) = (0, 1, 1)$......
  - Each clause contributes to 7 of the 8 tensors.
- Maximize $\sum_{ijk} A^{(1)}_{ijk} x_i x_j x_k + \sum_{ijk} A^{(2)}_{ijk} (1 - x_i) x_j x_k + \cdots$.
- Can be done for all MAX-r-CSP problems. (Get 2^r r-tensors. But for this talk, $r = 3$.) Goodbye MAX-CSP. Only Tensor Optimization.

Planted Clique Problem

- Given $G(n, 1/2) + p$ (planted) clique, find clique.

Planted Clique Problem

- Given $G(n, 1/2) \ + \ p$ (planted) clique, find clique.
- If $p \geq \Omega(\sqrt{n})$, spectral methods work. Alon, Krivelevich, Sudakov. ($p \in O(n^{0.5-\varepsilon})$? Still open.)

$$A_{ijk} = \begin{cases} 1 & \text{if number of edges in } G \text{ among } (i,j),(j,k),(k,i) \text{ is odd} \\ -1 & \text{otherwise.} \end{cases}$$

Planted Clique Problem

- Given $G(n, 1/2) \; + \; p$ (planted) clique, find clique.
- If $p \geq \Omega(\sqrt{n})$, spectral methods work. Alon, Krivelevich, Sudakov. ($p \in O(n^{0.5-\varepsilon})$? Still open.)

$$A_{ijk} = \begin{cases} 1 & \text{if number of edges in } G \text{ among } (i,j),(j,k),(k,i) \text{ is odd} \\ -1 & \text{otherwise.} \end{cases}$$

- Frieze, K. Arg-Max$_{|x|=1} \sum_{ijk} A_{ijk} x_i x_j x_k$, gives the planted clique provided $p \in \Omega^*(n^{1/3})$.

Planted Clique Problem

- Given $G(n, 1/2) + p$ (planted) clique, find clique.
- If $p \geq \Omega(\sqrt{n})$, spectral methods work. Alon, Krivelevich, Sudakov. ($p \in O(n^{0.5-\varepsilon})$? Still open.)

$$A_{ijk} = \begin{cases} 1 & \text{if number of edges in } G \text{ among } (i,j),(j,k),(k,i) \text{ is odd} \\ -1 & \text{otherwise.} \end{cases}$$

- Frieze, K. Arg-Max$_{|x|=1} \sum_{ijk} A_{ijk} x_i x_j x_k$, gives the planted clique provided $p \in \Omega^*(n^{1/3})$.
- Brubaker, Vempala r-tensor problem gives planted clique provided $p \in \Omega^*(n^{1/r})$.

Planted Clique Problem

- Given $G(n, 1/2) \; + \; p$ (planted) clique, find clique.
- If $p \geq \Omega(\sqrt{n})$, spectral methods work. Alon, Krivelevich, Sudakov. ($p \in O(n^{0.5-\varepsilon})$? Still open.)

$$A_{ijk} = \begin{cases} 1 & \text{if number of edges in } G \text{ among } (i,j),(j,k),(k,i) \text{ is odd} \\ -1 & \text{otherwise.} \end{cases}$$

- Frieze, K. Arg-Max$_{|x|=1} \sum_{ijk} A_{ijk} x_i x_j x_k$, gives the planted clique provided $p \in \Omega^*(n^{1/3})$.
- Brubaker, Vempala r-tensor problem gives planted clique provided $p \in \Omega^*(n^{1/r})$.
- Planted Gaussian problem: A $n \times n$ i.i.d. $N(0,1)$ entries. B has i.i.d $N(\mu, 1)$ entries in (hidden) $p \times p$ sub-matrix and 0 o.w. Given $A + B$, find B. [Spectral methods for $p\mu \geq c\sqrt{n}$.]

Planted Clique Problem

- Given $G(n, 1/2) + p$ (planted) clique, find clique.
- If $p \geq \Omega(\sqrt{n})$, spectral methods work. Alon, Krivelevich, Sudakov. ($p \in O(n^{0.5-\varepsilon})$? Still open.)

$$A_{ijk} = \begin{cases} 1 & \text{if number of edges in } G \text{ among } (i,j),(j,k),(k,i) \text{ is odd} \\ -1 & \text{otherwise.} \end{cases}$$

- Frieze, K. Arg-Max$_{|x|=1} \sum_{ijk} A_{ijk} x_i x_j x_k$, gives the planted clique provided $p \in \Omega^*(n^{1/3})$.
- Brubaker, Vempala r-tensor problem gives planted clique provided $p \in \Omega^*(n^{1/r})$.
- Planted Gaussian problem: A $n \times n$ i.i.d. $N(0,1)$ entries. B has i.i.d $N(\mu, 1)$ entries in (hidden) $p \times p$ sub-matrix and 0 o.w. Given $A + B$, find B. [Spectral methods for $p\mu \geq c\sqrt{n}$.]
- Planted Dense sub-graph problems.

Tensor Optimization - What norms?

- Problem: Maximize $\sum_{ijk} A_{ijk} y_i y_j y_k$, where, there are some constraints of the form $y_i \in \{0, 1\}$ and $y_i = 1 - y_j$.

Tensor Optimization - What norms?

- Problem: Maximize $\sum_{ijk} A_{ijk} y_i y_j y_k$, where, there are some constraints of the form $y_i \in \{0, 1\}$ and $y_i = 1 - y_j$.
- Notation: $A(x, y, z) = \sum_{ijk} A_{ijk} x_i y_j z_k$.

Tensor Optimization - What norms?

- Problem: Maximize $\sum_{ijk} A_{ijk} y_i y_j y_k$, where, there are some constraints of the form $y_i \in \{0, 1\}$ and $y_i = 1 - y_j$.
- Notation: $A(x, y, z) = \sum_{ijk} A_{ijk} x_i y_j z_k$.
- Suppose we can approximate A by a "simpler to optimize" (low rank) tensor B so that

$$\text{Max}_{|x|=|y|=|z|=1} |A(x, y, z) - B(x, y, z)| = ||A - B|| \leq \Delta.$$

Tensor Optimization - What norms?

- Problem: Maximize $\sum_{ijk} A_{ijk} y_i y_j y_k$, where, there are some constraints of the form $y_i \in \{0, 1\}$ and $y_i = 1 - y_j$.
- Notation: $A(x, y, z) = \sum_{ijk} A_{ijk} x_i y_j z_k$.
- Suppose we can approximate A by a "simpler to optimize" (low rank) tensor B so that

$$\text{Max}_{|x|=|y|=|z|=1} |A(x, y, z) - B(x, y, z)| = ||A - B|| \leq \Delta.$$

- Then, solving the problem with B instead of A ensures error is at most $\Delta |x||y||z|$.

Tensor Optimization - What norms?

- Problem: Maximize $\sum_{ijk} A_{ijk} y_i y_j y_k$, where, there are some constraints of the form $y_i \in \{0, 1\}$ and $y_i = 1 - y_j$.
- Notation: $A(x, y, z) = \sum_{ijk} A_{ijk} x_i y_j z_k$.
- Suppose we can approximate A by a "simpler to optimize" (low rank) tensor B so that

$$\text{Max}_{|x|=|y|=|z|=1} |A(x, y, z) - B(x, y, z)| = ||A - B|| \leq \Delta.$$

- Then, solving the problem with B instead of A ensures error is at most $\Delta |x||y||z|$.
- Moral of this: Enough to ensue that A is well approximated by B in spectral norm.

Any tensor can be approximated

- Notation: $x \otimes y \otimes z$ is the tensor with entries $x_i y_j z_k$. It is a rank 1 tensor. $||A||_F^2 =$ sum of squares of all entries.

Any tensor can be approximated

- Notation: $x \otimes y \otimes z$ is the tensor with entries $x_i y_j z_k$. It is a rank 1 tensor. $||A||_F^2 =$ sum of squares of all entries.
- **Lemma** For any $r-$tensor A, there are $1/\varepsilon^2$ rank 1 tensors whose sum B satisfies

$$||A - B|| \leq \varepsilon ||A||_F.$$

Any tensor can be approximated

- Notation: $x \otimes y \otimes z$ is the tensor with entries $x_i y_j z_k$. It is a rank 1 tensor. $||A||_F^2 =$ sum of squares of all entries.
- **Lemma** For any $r-$tensor A, there are $1/\varepsilon^2$ rank 1 tensors whose sum B satisfies

$$||A - B|| \leq \varepsilon ||A||_F.$$

  - Proof: Start with $B = 0$. If Lemma not already satisfied, there are x, y, z such that $|(A - B)(x, y, z)| \geq \varepsilon ||A||_F$. Take $cx \otimes y \otimes z$ as the next rank 1 tensor to subtract... [Greedy. Imitation of SVD.]

Any tensor can be approximated

- Notation: $x \otimes y \otimes z$ is the tensor with entries $x_i y_j z_k$. It is a rank 1 tensor. $||A||_F^2 =$ sum of squares of all entries.
- **Lemma** For any $r-$tensor A, there are $1/\varepsilon^2$ rank 1 tensors whose sum B satisfies

$$||A - B|| \leq \varepsilon ||A||_F.$$

  - Proof: Start with $B = 0$. If Lemma not already satisfied, there are x, y, z such that $|(A - B)(x, y, z)| \geq \varepsilon ||A||_F$. Take $cx \otimes y \otimes z$ as the next rank 1 tensor to subtract... [Greedy. Imitation of SVD.]
  - To make this constructive, need to find x, y, z. NP-hard : Hillar, Lim in "Most Tensor Problems are NP-Hard".

Any tensor can be approximated

- Notation: $x \otimes y \otimes z$ is the tensor with entries $x_i y_j z_k$. It is a rank 1 tensor. $||A||_F^2 =$ sum of squares of all entries.
- **Lemma** For any $r-$tensor A, there are $1/\varepsilon^2$ rank 1 tensors whose sum B satisfies

$$||A - B|| \leq \varepsilon ||A||_F.$$

  - Proof: Start with $B = 0$. If Lemma not already satisfied, there are x, y, z such that $|(A - B)(x, y, z)| \geq \varepsilon ||A||_F$. Take $cx \otimes y \otimes z$ as the next rank 1 tensor to subtract... [Greedy. Imitation of SVD.]
  - To make this constructive, need to find x, y, z. NP-hard : Hillar, Lim in "Most Tensor Problems are NP-Hard".
- **Theorem** dela Vega, K., Karpinski, Vempala For any $r-$tensor A, we can find in $O^*(n^{1/\varepsilon^2})$ time $4/\varepsilon^2$ rank 1 tensors whose sum B satisfies whp: $||A - B|| \leq \varepsilon ||A||_F$.

Any tensor can be approximated

- Notation: $x \otimes y \otimes z$ is the tensor with entries $x_i y_j z_k$. It is a rank 1 tensor. $||A||_F^2 =$ sum of squares of all entries.
- **Lemma** For any $r-$tensor A, there are $1/\varepsilon^2$ rank 1 tensors whose sum B satisfies
$$||A - B|| \leq \varepsilon ||A||_F.$$
  - Proof: Start with $B = 0$. If Lemma not already satisfied, there are x, y, z such that $|(A - B)(x, y, z)| \geq \varepsilon ||A||_F$. Take $cx \otimes y \otimes z$ as the next rank 1 tensor to subtract... [Greedy. Imitation of SVD.]
  - To make this constructive, need to find x, y, z. NP-hard : Hillar, Lim in "Most Tensor Problems are NP-Hard".
- **Theorem** dela Vega, K., Karpinski, Vempala For any $r-$tensor A, we can find in $O^*(n^{1/\varepsilon^2})$ time $4/\varepsilon^2$ rank 1 tensors whose sum B satisfies whp: $||A - B|| \leq \varepsilon ||A||_F$.
- No Free Lunch: Cannot put $|| \cdot ||_F$ in lhs or $|| \cdot ||$ on rhs.

Norm Maximization for tensors

Central Problem: Find x, y, z unit vectors to maximize $\sum_{ijk} A_{ijk} x_i y_j z_k$.

Norm Maximization for tensors

Central Problem: Find x, y, z unit vectors to maximize $\sum_{ijk} A_{ijk} x_i y_j z_k$.

1. If we knew the optimizing y, z, then the optimizing x is easy to find: it is just the vector $A(\cdot, y, z)$ (whose i th component is $A(e_i, y, z)$) scaled to length 1.

Norm Maximization for tensors

Central Problem: Find x, y, z unit vectors to maximize $\sum_{ijk} A_{ijk} x_i y_j z_k$.

1. If we knew the optimizing y, z, then the optimizing x is easy to find: it is just the vector $A(\cdot, y, z)$ (whose i th component is $A(e_i, y, z)$) scaled to length 1.
2. Now, $A(e_i, y, z) = \sum_{j,k} A_{ijk} y_j z_k$. The sum can be estimated by having just a few terms. But, an important question is: how do we make sure the variance is not too high, since the entries can have disparate values ?

Norm Maximization for tensors

Central Problem: Find x, y, z unit vectors to maximize $\sum_{ijk} A_{ijk} x_i y_j z_k$.

1. If we knew the optimizing y, z, then the optimizing x is easy to find: it is just the vector $A(\cdot, y, z)$ (whose i th component is $A(e_i, y, z)$) scaled to length 1.
2. Now, $A(e_i, y, z) = \sum_{j,k} A_{ijk} y_j z_k$. The sum can be estimated by having just a few terms. But, an important question is: how do we make sure the variance is not too high, since the entries can have disparate values ?
3. Length squared sampling works ! [Stated here without proof.]

Norm Maximization for tensors

Central Problem: Find x, y, z unit vectors to maximize $\sum_{ijk} A_{ijk} x_i y_j z_k$.

1. If we knew the optimizing y, z, then the optimizing x is easy to find: it is just the vector $A(\cdot, y, z)$ (whose i th component is $A(e_i, y, z)$) scaled to length 1.
2. Now, $A(e_i, y, z) = \sum_{j,k} A_{ijk} y_j z_k$. The sum can be estimated by having just a few terms. But, an important question is: how do we make sure the variance is not too high, since the entries can have disparate values ?
3. Length squared sampling works ! [Stated here without proof.]
4. This gives us many candidate x 's. How do we check which one is good ? For each x, recursively solve the matrix problem (SVD!) to determine its value !

Norm Maximization - Some Detail

- Estimate $\sum_{jk} A_{ijk} y_j z_k$ for all i. (Really for all y, z.)

Norm Maximization - Some Detail

- Estimate $\sum_{jk} A_{ijk} y_j z_k$ for all i. (Really for all y, z.)
- Pick a set S of $O(1)$ pairs (j, k) in i.i.d. trials, in each with probabilities: $\frac{\sum_i A_{jk}^2}{||A||_F^2}$.

Norm Maximization - Some Detail

- Estimate $\sum_{jk} A_{ijk} y_j z_k$ for all i. (Really for all y, z.)
- Pick a set S of $O(1)$ pairs (j, k) in i.i.d. trials, in each with probabilities: $\frac{\sum_i A_{jk}^2}{||A||_F^2}$.
- For each $(j, k) \in S$, enumerate all possible values of y_j, z_k (in discrete steps). [Only POLY$^{O(1)}$ $=$POLY many sets of values.]

Norm Maximization - Some Detail

- Estimate $\sum_{jk} A_{ijk} y_j z_k$ for all i. (Really for all y, z.)
- Pick a set S of $O(1)$ pairs (j, k) in i.i.d. trials, in each with probabilities: $\frac{\sum_i A_{jk}^2}{||A||_F^2}$.
- For each $(j, k) \in S$, enumerate all possible values of y_j, z_k (in discrete steps). [Only POLY$^{O(1)}$ =POLY many sets of values.]
- Treat $\sum_{(j,k)\in S} A_{ijk} y_j z_k$ as an estimate of $\sum_{\text{all}(j,k)} A_{ijk} y_j z_k$.

For what CSP's is this good?

- First, 2-CSP: MAX-2-SAT. Or MAX-CUT. n number of variables or vertices and m number of clauses or edges. A has $||A||_F^2 = m$.

For what CSP's is this good?

- First, 2-CSP: MAX-2-SAT. Or MAX-CUT. n number of variables or vertices and m number of clauses or edges. A has $||A||_F^2 = m$.
- Error = $||A - B||\ |x|\ |1 - x| \leq \varepsilon ||A||_F \sqrt{n}\sqrt{n} \leq \varepsilon \sqrt{m}n$.

For what CSP's is this good?

- First, 2-CSP: MAX-2-SAT. Or MAX-CUT. n number of variables or vertices and m number of clauses or edges. A has $||A||_F^2 = m$.
- Error = $||A - B||\ |x|\ |1 - x| \leq \varepsilon ||A||_F \sqrt{n}\sqrt{n} \leq \varepsilon \sqrt{m}n$.
- But all MAX-CSP problems can be easily solved with error at most $O(m)$.

For what CSP's is this good?

- First, 2-CSP: MAX-2-SAT. Or MAX-CUT. n number of variables or vertices and m number of clauses or edges. A has $||A||_F^2 = m$.
- Error = $||A - B||\ |x|\ |1 - x| \leq \varepsilon ||A||_F \sqrt{n}\sqrt{n} \leq \varepsilon \sqrt{m} n$.
- But all MAX-CSP problems can be easily solved with error at most $O(m)$.
- So, no use unless $m \in \Omega(n^2)$. Dense. Similar argument for higher r.

Generalizing Metrics, Dense problems

- Scaling A: Let D_i be the sum of the i th row. [Degree if A is the adjacency matrix.]

Generalizing Metrics, Dense problems

- Scaling A: Let D_i be the sum of the i th row. [Degree if A is the adjacency matrix.]
- In many situations, a natural scaling of A is given by $B_{ij} = \frac{A_{ij}}{\sqrt{D_i D_j}}$.

Generalizing Metrics, Dense problems

- Scaling A: Let D_i be the sum of the i th row. [Degree if A is the adjacency matrix.]
- In many situations, a natural scaling of A is given by $B_{ij} = \frac{A_{ij}}{\sqrt{D_i D_j}}$.
- Define $\bar{D} = \sum_i D_i / n$. Our scaling $B_{ij} = \frac{A_{ij}}{\sqrt{(D_i + \bar{D})(D_j + \bar{D})}}$.

Generalizing Metrics, Dense problems

- Scaling A: Let D_i be the sum of the i th row. [Degree if A is the adjacency matrix.]
- In many situations, a natural scaling of A is given by $B_{ij} = \frac{A_{ij}}{\sqrt{D_i D_j}}$.
- Define $\bar{D} = \sum_i D_i/n$. Our scaling $B_{ij} = \frac{A_{ij}}{\sqrt{(D_i+\bar{D})(D_j+\bar{D})}}$.
- A is **core-dense** if $||B||_F \in O(1)$.

Generalizing Metrics, Dense problems

- Scaling A: Let D_i be the sum of the i th row. [Degree if A is the adjacency matrix.]
- In many situations, a natural scaling of A is given by $B_{ij} = \frac{A_{ij}}{\sqrt{D_i D_j}}$.
- Define $\bar{D} = \sum_i D_i / n$. Our scaling $B_{ij} = \frac{A_{ij}}{\sqrt{(D_i + \bar{D})(D_j + \bar{D})}}$.
- A is **core-dense** if $||B||_F \in O(1)$.
- Dense matrices, Metrics (triangle inequality), powers of metrics - all are core-dense !

Generalizing Metrics, Dense problems

- Scaling A: Let D_i be the sum of the i th row. [Degree if A is the adjacency matrix.]
- In many situations, a natural scaling of A is given by $B_{ij} = \frac{A_{ij}}{\sqrt{D_i D_j}}$.
- Define $\bar{D} = \sum_i D_i/n$. Our scaling $B_{ij} = \frac{A_{ij}}{\sqrt{(D_i + \bar{D})(D_j + \bar{D})}}$.
- A is **core-dense** if $||B||_F \in O(1)$.
- Dense matrices, Metrics (triangle inequality), powers of metrics - all are core-dense !
- Theorem PTAS's for all core-dense MAX-r-CSP's.

Moment Tensors

- $x_1, x_2, \ldots, x_n$ (dependent) r.v.s. with $Ex_i = 0$.

Moment Tensors

- $x_1, x_2, \ldots, x_n$ (dependent) r.v.s. with $Ex_i = 0$.
- $A_{ij} = E(x_i x_j)$ - Variance-Covariance matrix.

Moment Tensors

- $x_1, x_2, \ldots, x_n$ (dependent) r.v.s. with $Ex_i = 0$.
- $A_{ij} = E(x_i x_j)$ - Variance-Covariance matrix.
- $A_{ijk} = E(x_i x_j x_k)$ - third moments tensor. So, $E((u \cdot x)^3) = A(u, u, u)$.

Moment Tensors

- $x_1, x_2, \ldots, x_n$ (dependent) r.v.s. with $Ex_i = 0$.
- $A_{ij} = E(x_i x_j)$ - Variance-Covariance matrix.
- $A_{ijk} = E(x_i x_j x_k)$ - third moments tensor. So, $E((u \cdot x)^3) = A(u, u, u)$.
- Frieze, Jerrum, K.,: If $E(x_i) = 0$ and x_i are 4-way independent and R is a orthonormal transformation, the **local** maxima of $F(u) = E[(u^T Rx)^4]$ over $|u| = 1$ are precisely the rows of R^{-1} corresponding to i with $E(x_i^4) > 3$. Yields an algorithm for ICA. **Moral** Some tensors are nice and we can do the maximization.

Moment Tensors

- $x_1, x_2, \ldots, x_n$ (dependent) r.v.s. with $Ex_i = 0$.
- $A_{ij} = E(x_i x_j)$ - Variance-Covariance matrix.
- $A_{ijk} = E(x_i x_j x_k)$ - third moments tensor. So, $E((u \cdot x)^3) = A(u, u, u)$.
- Frieze, Jerrum, K.,: If $E(x_i) = 0$ and x_i are 4-way independent and R is a orthonormal transformation, the **local** maxima of $F(u) = E[(u^T Rx)^4]$ over $|u| = 1$ are precisely the rows of R^{-1} corresponding to i with $E(x_i^4) > 3$. Yields an algorithm for ICA. **Moral** Some tensors are nice and we can do the maximization.
- Ananathkumar, Hsu, Kakade Third moment tensor used for Topic Modeling.

Epilogue

- These results can also be achieved (for a narrower class) by Sherali-Adams schemes. Yoshida, Zhou (2013).

Epilogue

- These results can also be achieved (for a narrower class) by Sherali-Adams schemes. Yoshida, Zhou (2013).
- Low-rank Approximation of polynomials : A $r-$tensor A represents a $r-$homogeneous polynomial : $\sum_{ijk} A_{ijk} x_i x_j x_k$. A rank 1 tensor $a \otimes b \otimes c$ represents a product of 3 linear polynomials - $(a \cdot x)(b \cdot x)(c \cdot x)$.

Epilogue

- These results can also be achieved (for a narrower class) by Sherali-Adams schemes. Yoshida, Zhou (2013).
- Low-rank Approximation of polynomials : A $r-$tensor A represents a $r-$homogeneous polynomial : $\sum_{ijk} A_{ijk}x_ix_jx_k$. A rank 1 tensor $a \otimes b \otimes c$ represents a product of 3 linear polynomials - $(a \cdot x)(b \cdot x)(c \cdot x)$.
- Any homogeneous polynomial can be approximated by the sum of a small number of products of linear polynomials.... Stronger Results Schrijver.

Epilogue

- These results can also be achieved (for a narrower class) by Sherali-Adams schemes. Yoshida, Zhou (2013).
- Low-rank Approximation of polynomials : A $r-$tensor A represents a $r-$homogeneous polynomial : $\sum_{ijk} A_{ijk} x_i x_j x_k$. A rank 1 tensor $a \otimes b \otimes c$ represents a product of 3 linear polynomials - $(a \cdot x)(b \cdot x)(c \cdot x)$.
- Any homogeneous polynomial can be approximated by the sum of a small number of products of linear polynomials.... Stronger Results Schrijver.
- OPEN: Use Tensors for other Optimization Problems. Suppose we can find spectral norm of 3-tensors to within a factor of $1 + \varepsilon$ for any constant $\varepsilon > 0$. [Not ruled out by NP-harness proofs.] Can one beat the best approximation factor for say Max-Cut obtained by SDP (a quadratic method) ?