Sampling, Matrices, Tensors

January 11, 2013

Set-up

- A an $m \times n$ matrix, real entries.

Set-up

- A an $m \times n$ matrix, real entries.
- Attach a probability p_j with j th column of A. The p_j sum to 1.

Set-up

- A an $m \times n$ matrix, real entries.
- Attach a probability p_j with j th column of A. The p_j sum to 1.
- In s i.i.d. trials, pick s columns of A with these probabilities.

Set-up

- A an $m \times n$ matrix, real entries.
- Attach a probability p_j with j th column of A. The p_j sum to 1.
- In s i.i.d. trials, pick s columns of A with these probabilities.
- Scale picked columns.

Set-up

- A an $m \times n$ matrix, real entries.
- Attach a probability p_j with j th column of A. The p_j sum to 1.
- In s i.i.d. trials, pick s columns of A with these probabilities.
- Scale picked columns.
- Form $m \times s$ matrix B of sampled, scaled columns.

Set-up

- A an $m \times n$ matrix, real entries.
- Attach a probability p_j with j th column of A. The p_j sum to 1.
- In s i.i.d. trials, pick s columns of A with these probabilities.
- Scale picked columns.
- Form $m \times s$ matrix B of sampled, scaled columns.
- Want $B_{m \times s} \approx A_{m \times n}$. Makes sense??

Set-up

- A an $m \times n$ matrix, real entries.
- Attach a probability p_j with j th column of A. The p_j sum to 1.
- In s i.i.d. trials, pick s columns of A with these probabilities.
- Scale picked columns.
- Form $m \times s$ matrix B of sampled, scaled columns.
- Want $B_{m\times s} \approx A_{m\times n}$. Makes sense??
- Try $BB^T \approx AA^T$. Both are $m \times m$!
  With correct scaling, can make it Unbiased:

$$E(BB^T) = AA^T.$$

Length Squared Sampling

- Large Matrix A. Sampling and scaling some columns to form B got us
$$E(BB^T) = AA^T.$$

Length Squared Sampling

- Large Matrix A. Sampling and scaling some columns to form B got us
$$E(BB^T) = AA^T.$$
- Minimize Variance. [Of a matrix?]

Length Squared Sampling

- Large Matrix A. Sampling and scaling some columns to form B got us
$$E(BB^T) = AA^T.$$
- Minimize Variance. [Of a matrix?]
- Frieze, K., Vempala Sampling probabilities proportional to SQUARED LENGTHS of columns minimize the variance.

Length Squared Sampling

- Large Matrix A. Sampling and scaling some columns to form B got us
$$E(BB^T) = AA^T.$$
- Minimize Variance. [Of a matrix?]
- Frieze, K., Vempala Sampling probabilities proportional to SQUARED LENGTHS of columns minimize the variance.
- Many applications of length-squared sampling:

Length Squared Sampling

- Large Matrix A. Sampling and scaling some columns to form B got us
$$E(BB^T) = AA^T.$$
- Minimize Variance. [Of a matrix?]
- Frieze, K., Vempala Sampling probabilities proportional to SQUARED LENGTHS of columns minimize the variance.
- Many applications of length-squared sampling:
  - Estimate of invariants of matrix.

Length Squared Sampling

- Large Matrix A. Sampling and scaling some columns to form B got us
$$E(BB^T) = AA^T.$$
- Minimize Variance. [Of a matrix?]
- Frieze, K., Vempala Sampling probabilities proportional to SQUARED LENGTHS of columns minimize the variance.
- Many applications of length-squared sampling:
  - Estimate of invariants of matrix.
  - Matrix Compression by sampling: Sample of **rows and columns** sufficient to approximate any matrix. Drineas, K., Mahoney
  - Approximate maximization of cubic and higher forms.

How many Samples do we need?

- We fix one measure of error, namely, relative spectral norm for this talk:

$$\frac{\text{Spectral Norm of } (AA^T - BB^T)}{\text{Spectral Norm of } AA^T}.$$

How many Samples do we need?

- We fix one measure of error, namely, relative spectral norm for this talk:

$$\frac{\text{Spectral Norm of } (AA^T - BB^T)}{\text{Spectral Norm of } AA^T}.$$

- How many samples ($= s$, the number of columns of B) do we need to ensure that with high probability, the error is at most 0.01?

How many Samples do we need?

- We fix one measure of error, namely, relative spectral norm for this talk:
$$\frac{\text{Spectral Norm of } (AA^T - BB^T)}{\text{Spectral Norm of } AA^T}.$$
- How many samples ($= s$, the number of columns of B) do we need to ensure that with high probability, the error is at most 0.01?
- Let r =rank(A). [Actually, $r = ||A||_F^2/||A||^2$ which is at most rank will do.]

How many Samples do we need?

- We fix one measure of error, namely, relative spectral norm for this talk:
$$\frac{\text{Spectral Norm of } (AA^T - BB^T)}{\text{Spectral Norm of } AA^T}.$$
- How many samples ($= s$, the number of columns of B) do we need to ensure that with high probability, the error is at most 0.01?
- Let r =rank(A). [Actually, $r = ||A||_F^2/||A||^2$ which is at most rank will do.]
  - Original FKV: $s = r^3$ works.

How many Samples do we need?

- We fix one measure of error, namely, relative spectral norm for this talk:
$$\frac{\text{Spectral Norm of } (AA^T - BB^T)}{\text{Spectral Norm of } AA^T}.$$
- How many samples ($= s$, the number of columns of B) do we need to ensure that with high probability, the error is at most 0.01?
- Let r =rank(A). [Actually, $r = ||A||_F^2/||A||^2$ which is at most rank will do.]
    - Original FKV: $s = r^3$ works.
    - Drineas, K., Mahoney $s = r^2$ suffices.

How many Samples do we need?

- We fix one measure of error, namely, relative spectral norm for this talk:
$$\frac{\text{Spectral Norm of } (AA^T - BB^T)}{\text{Spectral Norm of } AA^T}.$$
- How many samples ($= s$, the number of columns of B) do we need to ensure that with high probability, the error is at most 0.01?
- Let r =rank(A). [Actually, $r = ||A||_F^2/||A||^2$ which is at most rank will do.]
  - Original FKV: $s = r^3$ works.
  - Drineas, K., Mahoney $s = r^2$ suffices.
  - Rudelson and Vershynin $s = r \log r$ suffices. Uses some nice ideas from Functional Analysis. (Decoupling). Simpler proof of main tool by Ahlswede and Winter in Information Theory.

Variance-covariance matrices

- v a vector valued random variable (with a probability distribution (or density) in $n-$ space.)

Variance-covariance matrices

- v a vector valued random variable (with a probability distribution (or density) in $n-$ space.)
  - Eg. 1: v is a random column of a fixed matrix.

Variance-covariance matrices

- v a vector valued random variable (with a probability distribution (or density) in $n-$ space.)
  - Eg. 1: v is a random column of a fixed matrix.
  - Eg. 2: v has general (non-spherical) Gaussian or other densities.

Variance-covariance matrices

- v a vector valued random variable (with a probability distribution (or density) in $n-$ space.)
  - Eg. 1: v is a random column of a fixed matrix.
  - Eg. 2: v has general (non-spherical) Gaussian or other densities.
- How many i.i.d. samples of v are sufficient to ensure **relative error** approximation to the variance-covariance matrix - Evv^T ?
  Want:
  Sample Variance-Covariance matrix $\approx_\varepsilon$ true Variance-Covariance matrix. ($M_1 \approx_\varepsilon M_2$ if $x^T M_1 x \approx_\varepsilon x^T M_2 x \ \forall x$.)

Variance-covariance matrices

- v a vector valued random variable (with a probability distribution (or density) in $n-$ space.)
  - Eg. 1: v is a random column of a fixed matrix.
  - Eg. 2: v has general (non-spherical) Gaussian or other densities.
- How many i.i.d. samples of v are sufficient to ensure **relative error** approximation to the variance-covariance matrix - Evv^T ? Want:
  Sample Variance-Covariance matrix $\approx_\varepsilon$ true Variance-Covariance matrix. ($M_1 \approx_\varepsilon M_2$ if $x^T M_1 x \approx_\varepsilon x^T M_2 x \ \forall x$.)
- Question raised for log-concave densities by K., Lovász, Simonovits for computing volumes of convex sets. First improvement by Bourgain, then Rudelson to $O(n \log n)$ and most recently Srivatsava, Vershynin to $O(n)$. Relative error is important (and more difficult) for

Variance-covariance matrices

- v a vector valued random variable (with a probability distribution (or density) in $n-$ space.)
  - Eg. 1: v is a random column of a fixed matrix.
  - Eg. 2: v has general (non-spherical) Gaussian or other densities.
- How many i.i.d. samples of v are sufficient to ensure **relative error** approximation to the variance-covariance matrix - Evv^T ? Want:
  Sample Variance-Covariance matrix $\approx_\varepsilon$ true Variance-Covariance matrix. ($M_1 \approx_\varepsilon M_2$ if $x^T M_1 x \approx_\varepsilon x^T M_2 x \; \forall x$.)
- Question raised for log-concave densities by K., Lovász, Simonovits for computing volumes of convex sets. First improvement by Bourgain, then Rudelson to $O(n \log n)$ and most recently Srivatsava, Vershynin to $O(n)$. Relative error is important (and more difficult) for
  - Linear Regression when we are looking for x minimizing x^T Var-Covar x.

Variance-covariance matrices

- v a vector valued random variable (with a probability distribution (or density) in $n-$ space.)
  - Eg. 1: v is a random column of a fixed matrix.
  - Eg. 2: v has general (non-spherical) Gaussian or other densities.
- How many i.i.d. samples of v are sufficient to ensure **relative error** approximation to the variance-covariance matrix - Evv^T ? Want:

  Sample Variance-Covariance matrix $\approx_\varepsilon$ true Variance-Covariance matrix. ($M_1 \approx_\varepsilon M_2$ if $x^T M_1 x \approx_\varepsilon x^T M_2 x \; \forall x$.)
- Question raised for log-concave densities by K., Lovász, Simonovits for computing volumes of convex sets. First improvement by Bourgain, then Rudelson to $O(n \log n)$ and most recently Srivatsava, Vershynin to $O(n)$. Relative error is important (and more difficult) for
  - Linear Regression when we are looking for x minimizing x^T Var-Covar x.
  - Graph, Matrix Sparsification Spielman, Srivatsava, Batman, Teng.

Matrix-Valued Random Variables

- Last Slide: prove concentration for vv^T, v random vector. Rank 1.

Matrix-Valued Random Variables

- Last Slide: prove concentration for vv^T, v random vector. Rank 1.
- Generally, concentration for $||X||$, $X = X_1 + X_2 + \cdots + X_n$ independent $d \times d$ matrix-valued r.v.s's. with

$$0 \leq X_i \leq I.$$

Matrix-Valued Random Variables

- Last Slide: prove concentration for vv^T, v random vector. Rank 1.
- Generally, concentration for $||X||$, $X = X_1 + X_2 + \cdots + X_n$ independent $d \times d$ matrix-valued r.v.s's. with

$$0 \leq X_i \leq I.$$

- Traditional methods: Wigner ... Bound E Tr$(X_1 + X_2 + \cdots + X_n)^m$, m large even.

Matrix-Valued Random Variables

- Last Slide: prove concentration for vv^T, v random vector. Rank 1.
- Generally, concentration for $||X||$, $X = X_1 + X_2 + \cdots + X_n$ independent $d \times d$ matrix-valued r.v.s's. with

$$0 \leq X_i \leq I.$$

- Traditional methods: Wigner ... Bound E $\text{Tr}(X_1 + X_2 + \cdots + X_n)^m$, m large even.
- Ahlswede and Winter A Chernoff bound using Bernstein method. Crucial: Golden-Thompson inequality.

Matrix-Valued Random Variables

- Last Slide: prove concentration for vv^T, v random vector. Rank 1.
- Generally, concentration for $||X||$, $X = X_1 + X_2 + \cdots + X_n$ independent $d \times d$ matrix-valued r.v.s's. with

$$0 \leq X_i \leq I.$$

- Traditional methods: Wigner ... Bound E Tr$(X_1 + X_2 + \cdots + X_n)^m$, m large even.
- Ahlswede and Winter A Chernoff bound using Bernstein method. Crucial: Golden-Thompson inequality.
  - **Theorem** X_i **i.i.d.**. $\Pr\left(X \notin (1-\varepsilon)EX, (1+\varepsilon)EX\right) \leq d\ e^{-\varepsilon^2 n}$, for $\varepsilon \leq 1$.

Matrix-Valued Random Variables

- Last Slide: prove concentration for vv^T, v random vector. Rank 1.
- Generally, concentration for $||X||$, $X = X_1 + X_2 + \cdots + X_n$ independent $d \times d$ matrix-valued r.v.s's. with

$$0 \leq X_i \leq I.$$

- Traditional methods: Wigner ... Bound E $\text{Tr}(X_1 + X_2 + \cdots + X_n)^m$, m large even.
- Ahlswede and Winter A Chernoff bound using Bernstein method. Crucial: Golden-Thompson inequality.
  - **Theorem** X_i **i.i.d.**. $\Pr\left(X \notin (1-\varepsilon)EX, (1+\varepsilon)EX\right) \leq d\ e^{-\varepsilon^2 n}$, for $\varepsilon \leq 1$.
- Tropp Independence suffices; don't need i.i.d. [Lieb's inequality instead of Golden-Thompson.]

Matrix-Valued Random Variables

- Last Slide: prove concentration for vv^T, v random vector. Rank 1.
- Generally, concentration for $||X||$, $X = X_1 + X_2 + \cdots + X_n$ independent $d \times d$ matrix-valued r.v.s's. with

$$0 \leq X_i \leq I.$$

- Traditional methods: Wigner ... Bound E Tr$(X_1 + X_2 + \cdots + X_n)^m$, m large even.
- Ahlswede and Winter A Chernoff bound using Bernstein method. Crucial: Golden-Thompson inequality.
  - **Theorem** X_i **i.i.d.**. $\Pr\left(X \notin (1-\varepsilon)EX, (1+\varepsilon)EX\right) \leq d\ e^{-\varepsilon^2 n}$, for $\varepsilon \leq 1$.
- Tropp Independence suffices; don't need i.i.d. [Lieb's inequality instead of Golden-Thompson.]
- Open: Prove such concentration for negatively correlated (but not independent) X_i.

Matrix Sparsification

- $n \times m$ matrix A. [Think of $m >> n$.] [Each column is a record in a database.]

Matrix Sparsification

- $n \times m$ matrix A. [Think of $m >> n$.] [Each column is a record in a database.]
- Sample s columns of A (with a probability distribution of your choice) to get matrix B so that for **every** x:

$$x^T(AA^T)x \approx_\varepsilon x^T(BB^T)x \quad \equiv \quad |x^TA| \approx_\varepsilon |x^TB|.$$

Matrix Sparsification

- $n \times m$ matrix A. [Think of $m >> n$.] [Each column is a record in a database.]
- Sample s columns of A (with a probability distribution of your choice) to get matrix B so that for **every** x:

$$x^T(AA^T)x \approx_\varepsilon x^T(BB^T)x \quad \equiv \quad |x^TA| \approx_\varepsilon |x^TB|.$$

- What probability distribution and what s ?

Matrix Sparsification

- $n \times m$ matrix A. [Think of $m >> n$.] [Each column is a record in a database.]
- Sample s columns of A (with a probability distribution of your choice) to get matrix B so that for **every** x:

$$x^T(AA^T)x \approx_\varepsilon x^T(BB^T)x \quad \equiv \quad |x^TA| \approx_\varepsilon |x^TB|.$$

- What probability distribution and what s ?
- Length-squared sampling only gives us $\big||x^TA| - |x^TB|\big| \leq 0.01||A||$. Bad for x with small $|x^TA|$.

Matrix Sparsification

- $n \times m$ matrix A. [Think of $m >> n$.] [Each column is a record in a database.]
- Sample s columns of A (with a probability distribution of your choice) to get matrix B so that for **every** x:

$$x^T(AA^T)x \approx_\varepsilon x^T(BB^T)x \quad \equiv \quad |x^TA| \approx_\varepsilon |x^TB|.$$

- What probability distribution and what s ?
- Length-squared sampling only gives us $\big||x^TA| - |x^TB|\big| \leq 0.01||A||$. Bad for x with small $|x^TA|$.
- Do length-squared sampling on (basically) $A^{-1}A$ (!!??!!) - Isometry, equally good for all x ! Spielman, Srivatisava, Batsman; Drineas, Mahoney, Muthukrishnan

Matrix Sparsification

- $n \times m$ matrix A. [Think of $m >> n$.] [Each column is a record in a database.]
- Sample s columns of A (with a probability distribution of your choice) to get matrix B so that for **every** x:

$$x^T(AA^T)x \approx_\varepsilon x^T(BB^T)x \quad \equiv \quad |x^TA| \approx_\varepsilon |x^TB|.$$

- What probability distribution and what s ?
- Length-squared sampling only gives us $\big||x^TA| - |x^TB|\big| \leq 0.01||A||$. Bad for x with small $|x^TA|$.
- Do length-squared sampling on (basically) $A^{-1}A$ (!!??!!) - Isometry, equally good for all x ! Spielman, Srivatisava, Batsman; Drineas, Mahoney, Muthukrishnan
- $s = O^*(n)$ will do (whatever m is). Implies:

Matrix Sparsification

- $n \times m$ matrix A. [Think of $m >> n$.] [Each column is a record in a database.]
- Sample s columns of A (with a probability distribution of your choice) to get matrix B so that for **every** x:

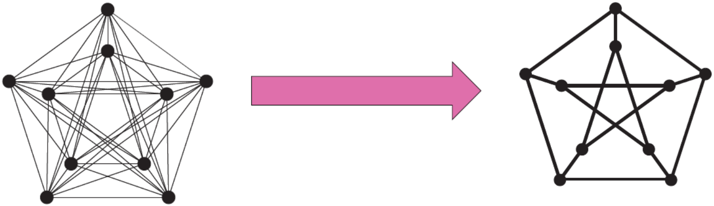
$$x^T(AA^T)x \approx_\varepsilon x^T(BB^T)x \quad \equiv \quad |x^TA| \approx_\varepsilon |x^TB|.$$

- What probability distribution and what s ?
- Length-squared sampling only gives us $\big||x^TA| - |x^TB|\big| \leq 0.01||A||$. Bad for x with small $|x^TA|$.
- Do length-squared sampling on (basically) $A^{-1}A$ (!!??!!) - Isometry, equally good for all x ! Spielman, Srivatisava, Batsman; Drineas, Mahoney, Muthukrishnan
- $s = O^*(n)$ will do (whatever m is). Implies:
- Theorem For any $n \times m$ matrix A, there is a subset B of $O(n)$ (scaled) columns of A such that for every x,

$$|x^TA| \approx_{0.01} |x^TB|.$$

Graph Spasification - a special case of Matrix Sparsification

Sample edges to represent every cut size to **relative error.** Then find sparsest cut in sampled graph.



Indeed, for graphs, sampling probabilities proportional to electrical resistances work and make sparsification possible in nearly linear time. No such fast algorithm is known for general matrix sparsification.

Maximizing Cubic and higher forms

- Given $m \times n \times p$ array A_{ijk}, find $||A|| = \text{Max}_{|x|=|y|=|z|=1} A(x,y,z) = \sum_{ijk} A_{ijk} x_i y_j z_k$. All we say here applies higher forms, A_{ijkl}, etc..

Maximizing Cubic and higher forms

- Given $m \times n \times p$ array A_{ijk}, find
  $||A|| = \text{Max}_{|x|=|y|=|z|=1} A(x,y,z) = \sum_{ijk} A_{ijk} x_i y_j z_k$.
  All we say here applies higher forms, A_{ijkl}, etc..
- No clean, nice theory, algorithms as for matrices. In fact, exact maximization is computationally hard for quartic and higher forms.

Maximizing Cubic and higher forms

- Given $m \times n \times p$ array A_{ijk}, find
$||A|| = \text{Max}_{|x|=|y|=|z|=1} A(x,y,z) = \sum_{ijk} A_{ijk} x_i y_j z_k$.
All we say here applies higher forms, A_{ijkl}, etc..
- No clean, nice theory, algorithms as for matrices. In fact, exact maximization is computationally hard for quartic and higher forms.
- **Theorem** Using length squared sampling, we can find (in polynomial time) a x, y, z such that with high probability

$$A(x,y,z) \geq ||A|| - 0.01||A||_F,$$

where, $||A||_F^2$ is the sum of squares of all entries of A. [Alas, we cannot replace $||\cdot||$ on the left by $||\cdot||_F$ or vice varsa.] de la Vega, Karpinski, K., Vempala

Maximizing cubic forms

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

Maximizing cubic forms

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

- 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.

Maximizing cubic forms

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

- 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.
- Now, $A(e_i, y, z) = \sum_{j,k,l} A_{i,j,k} y_j z_k$.
  - The sum can be estimated by having just a few terms, namely, y_j, z_k values for a few j, k.

Maximizing cubic forms

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

- 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.
- Now, $A(e_i, y, z) = \sum_{j,k,l} A_{i,j,k} y_j z_k$.
  - The sum can be estimated by having just a few terms, namely, y_j, z_k values for a few j, k.
  - Of course don't know these values, but FEW $\implies$ we can enumerate all possibilities.

Maximizing cubic forms

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

- 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.
- Now, $A(e_i, y, z) = \sum_{j,k,l} A_{i,j,k} y_j z_k$.
  - The sum can be estimated by having just a few terms, namely, y_j, z_k values for a few j, k.
  - Of course don't know these values, but FEW $\implies$ we can enumerate all possibilities.
  - How do we make sure the variance is not too high, since the entries can have disparate values ?

Maximizing cubic forms

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

- 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.
- Now, $A(e_i, y, z) = \sum_{j,k,l} A_{i,j,k} y_j z_k$.
  - The sum can be estimated by having just a few terms, namely, y_j, z_k values for a few j, k.
  - Of course don't know these values, but FEW $\implies$ we can enumerate all possibilities.
  - How do we make sure the variance is not too high, since the entries can have disparate values ?
  - Length squared sampling works ! [Stated here without proof.]

Maximizing cubic forms

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

- 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.
- Now, $A(e_i, y, z) = \sum_{j,k,l} A_{i,j,k} y_j z_k$.
  - The sum can be estimated by having just a few terms, namely, y_j, z_k values for a few j, k.
  - Of course don't know these values, but FEW $\implies$ we can enumerate all possibilities.
  - How do we make sure the variance is not too high, since the entries can have disparate values ?
  - Length squared sampling works ! [Stated here without proof.]
- This gives us many candidate x 's. How do we check which one is good ? For each x, form the matrix $A(x)$. Solve the quadratic form maximization for the matrix to find best y, z. Take the best candidate x.

Combinatorial Application of Low-rank approximations

- Szemeredi's Regularity Lemma:

Combinatorial Application of Low-rank approximations

- Szemeredi's Regularity Lemma:
  - Graph G on n vertices ($n \to \infty$).

- Szemeredi's Regularity Lemma:
  - Graph G on n vertices ($n \to \infty$).
  - Can partition the vertex set into $O(1)$ parts so that the edge sets between most pairs behave as if they were thrown in at random with the correct density.

Combinatorial Application of Low-rank approximations

- Szemeredi's Regularity Lemma:
  - Graph G on n vertices ($n \to \infty$).
  - Can partition the vertex set into $O(1)$ parts so that the edge sets between most pairs behave as if they were thrown in at random with the correct density.
- Beautiful Theorem with many applications including van der Warden conjecture.

Combinatorial Application of Low-rank approximations

- Szemeredi's Regularity Lemma:
  - Graph G on n vertices ($n \to \infty$).
  - Can partition the vertex set into $O(1)$ parts so that the edge sets between most pairs behave as if they were thrown in at random with the correct density.
- Beautiful Theorem with many applications including van der Warden conjecture.
- Gowers The number of parts has to be at least a tower of height $1/\varepsilon^{20}$ in error parameter ε.

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.
- Density d_{ij} between part V_i and V_j is the fraction of number of edges between V_i, V_j.

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.
- Density d_{ij} between part V_i and V_j is the fraction of number of edges between V_i, V_j.
- Think of edges between a vertex in V_i and one in V_j being thrown in at random with probability d_{ij}.

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.
- Density d_{ij} between part V_i and V_j is the fraction of number of edges between V_i, V_j.
- Think of edges between a vertex in V_i and one in V_j being thrown in at random with probability d_{ij}.
- Partition is "weakly" ε regular if for any subsets S, T of vertices we have

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.
- Density d_{ij} between part V_i and V_j is the fraction of number of edges between V_i, V_j.
- Think of edges between a vertex in V_i and one in V_j being thrown in at random with probability d_{ij}.
- Partition is "weakly" ε regular if for any subsets S, T of vertices we have
  - Number of edges between S and T = E(of that number) $\pm \varepsilon n^2$.

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.
- Density d_{ij} between part V_i and V_j is the fraction of number of edges between V_i, V_j.
- Think of edges between a vertex in V_i and one in V_j being thrown in at random with probability d_{ij}.
- Partition is "weakly" ε regular if for any subsets S, T of vertices we have
  - Number of edges between S and T = E(of that number) $\pm \varepsilon n^2$.
- Frieze, K. There is a weakly ε regular partition with $2^{1/\varepsilon^2}$ parts.

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.
- Density d_{ij} between part V_i and V_j is the fraction of number of edges between V_i, V_j.
- Think of edges between a vertex in V_i and one in V_j being thrown in at random with probability d_{ij}.
- Partition is "weakly" ε regular if for any subsets S, T of vertices we have
  - Number of edges between S and T = E(of that number) $\pm \varepsilon n^2$.
- Frieze, K. There is a weakly ε regular partition with $2^{1/\varepsilon^2}$ parts.
- Such a partition can be found in poly time.

Weak Regularity Lemma

- Vertex set V of a graph partitioned into $V_1, V_2, \ldots, V_k$.
- Density d_{ij} between part V_i and V_j is the fraction of number of edges between V_i, V_j.
- Think of edges between a vertex in V_i and one in V_j being thrown in at random with probability d_{ij}.
- Partition is "weakly" ε regular if for any subsets S, T of vertices we have
  - Number of edges between S and T = E(of that number) $\pm\varepsilon n^2$.
- Frieze, K. There is a weakly ε regular partition with $2^{1/\varepsilon^2}$ parts.
- Such a partition can be found in poly time.
- But why state this in this talk?

Combinatorial Rank 1 matrices and Regularity

- A *cut matrix* is of the form $\alpha v \otimes u$, where, α is a real number and u, v are 0-1 vectors.

Combinatorial Rank 1 matrices and Regularity

- A *cut matrix* is of the form $\alpha v \otimes u$, where, α is a real number and u, v are 0-1 vectors.
- (Easy) Any matrix can be approximated by a sum of a small number of cut matrices. Specifically, at most $1/\varepsilon^2$ cut matrices, so that the error in "cut norm" is at most $\varepsilon||A||_F$.

Combinatorial Rank 1 matrices and Regularity

- A *cut matrix* is of the form $\alpha v \otimes u$, where, α is a real number and u, v are 0-1 vectors.
- (Easy) Any matrix can be approximated by a sum of a small number of cut matrices. Specifically, at most $1/\varepsilon^2$ cut matrices, so that the error in "cut norm" is at most $\varepsilon||A||_F$.
- Cut Norm: Max. absolute value of the sum of entries in a rectangle (any subset of rows $\times$ any subset of columns)

Combinatorial Rank 1 matrices and Regularity

- A *cut matrix* is of the form $\alpha v \otimes u$, where, α is a real number and u, v are 0-1 vectors.
- (Easy) Any matrix can be approximated by a sum of a small number of cut matrices. Specifically, at most $1/\varepsilon^2$ cut matrices, so that the error in "cut norm" is at most $\varepsilon||A||_F$.
- Cut Norm: Max. absolute value of the sum of entries in a rectangle (any subset of rows $\times$ any subset of columns)
- Hard: Such an approximation can be found.

Combinatorial Rank 1 matrices and Regularity

- A *cut matrix* is of the form $\alpha v \otimes u$, where, α is a real number and u, v are 0-1 vectors.
- (Easy) Any matrix can be approximated by a sum of a small number of cut matrices. Specifically, at most $1/\varepsilon^2$ cut matrices, so that the error in “cut norm” is at most $\varepsilon ||A||_F$.
- Cut Norm: Max. absolute value of the sum of entries in a rectangle (any subset of rows $\times$ any subset of columns)
- Hard: Such an approximation can be found.
- Easy: Such an approximation gives a weakly regular partition.

Combinatorial Rank 1 matrices and Regularity

- A *cut matrix* is of the form $\alpha v \otimes u$, where, α is a real number and u, v are 0-1 vectors.
- (Easy) Any matrix can be approximated by a sum of a small number of cut matrices. Specifically, at most $1/\varepsilon^2$ cut matrices, so that the error in "cut norm" is at most $\varepsilon ||A||_F$.
- Cut Norm: Max. absolute value of the sum of entries in a rectangle (any subset of rows $\times$ any subset of columns)
- Hard: Such an approximation can be found.
- Easy: Such an approximation gives a weakly regular partition.
- Weak regularity partition not sufficient for many purely structural results. (Otherwise would contradict lower bounds for van der Warden problem). It suffices for algorithmic applications.

Combinatorial Rank 1 matrices and Regularity

- A *cut matrix* is of the form $\alpha v \otimes u$, where, α is a real number and u, v are 0-1 vectors.
- (Easy) Any matrix can be approximated by a sum of a small number of cut matrices. Specifically, at most $1/\varepsilon^2$ cut matrices, so that the error in "cut norm" is at most $\varepsilon ||A||_F$.
- Cut Norm: Max. absolute value of the sum of entries in a rectangle (any subset of rows $\times$ any subset of columns)
- Hard: Such an approximation can be found.
- Easy: Such an approximation gives a weakly regular partition.
- Weak regularity partition not sufficient for many purely structural results. (Otherwise would contradict lower bounds for van der Warden problem). It suffices for algorithmic applications.
- Extends to higher dimensional arrays (tensors).