Random projections in Linear Programming

Leo Liberti, Vu Khac Ky, Pierre-Louis Poirion
CNRS LIX Ecole Polytechnique, France

IWAASS 2016 — SVAN@impa.br

There is nothing stochastic about the problem

The randomness is in the method

The gist

- **Goal**: solving very large LPs

$$\min\{c^\top x \mid Ax = b \wedge x \geq 0\}$$

- **Trade-off**: approximate / wrong with low probability: OK

- **Means**: random map T projects cols of $Ax = b$, get

$$Ax = b \wedge x \geq 0 \;\Leftrightarrow\; TAx = Tb \wedge x \geq 0$$

  with high probability

- Solve

$$\min\{c^\top x \mid TAx = Tb \wedge x \geq 0\}$$

Plan

- Restricted Linear Membership
- Johnson-Lindenstrauss Lemma
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Solving large LPs

Restricted Linear Membership

Linear feasibility with constrained multipliers

Restricted Linear Membership (RLM)
Given vectors $A_1, \ldots, A_n, b \in \mathbb{R}^m$ and $X \subseteq \mathbb{R}^n$, is there $x \in X$ s.t.

$$b = \sum_{i \le n} x_i A_i \text{ ?}$$

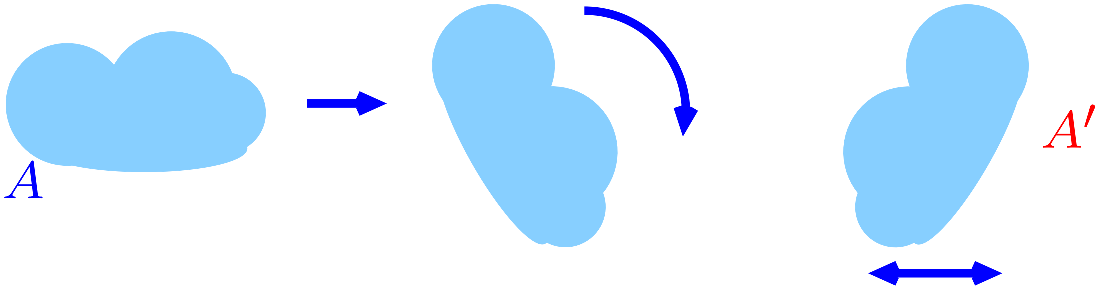
RLM$_X$ is a fundamental problem class, which subsumes:

- **Linear Feasibility Problem** (LFP) with $X = \mathbb{R}^n_+$
- **Integer Feasibility Problem** (IFP) with $X = \mathbb{Z}^n_+$
- Efficient solution of LFP/IFP yields sol. of LP/IP via bisection

The shape of a set of points

- **Lose dimensions but not too much accuracy**
  Given $A_1, \ldots, A_n \in \mathbb{R}^m$ find $k \ll m$ and points $A'_1, \ldots, A'_n \in \mathbb{R}^k$ s.t.
  A and A' "have almost the same shape"

- What is the shape of a set of points?



congruent sets have the same shape

- Approximate congruence: A, A' have **almost the same shape** if

$$\forall i < j \leq n \quad (1-\varepsilon)\|A_i - A_j\| \leq \|A'_i - A'_j\| \leq (1+\varepsilon)\|A_i - A_j\|$$

for some small $\varepsilon > 0$

Assume norms are all Euclidean

Losing dimensions in the RLM

Given $X \subseteq \mathbb{R}^n$ and $b, A_1, \ldots, A_n \in \mathbb{R}^m$, find $k \ll m$, $b', A'_1, \ldots, A'_n \in \mathbb{R}^k$ such that:

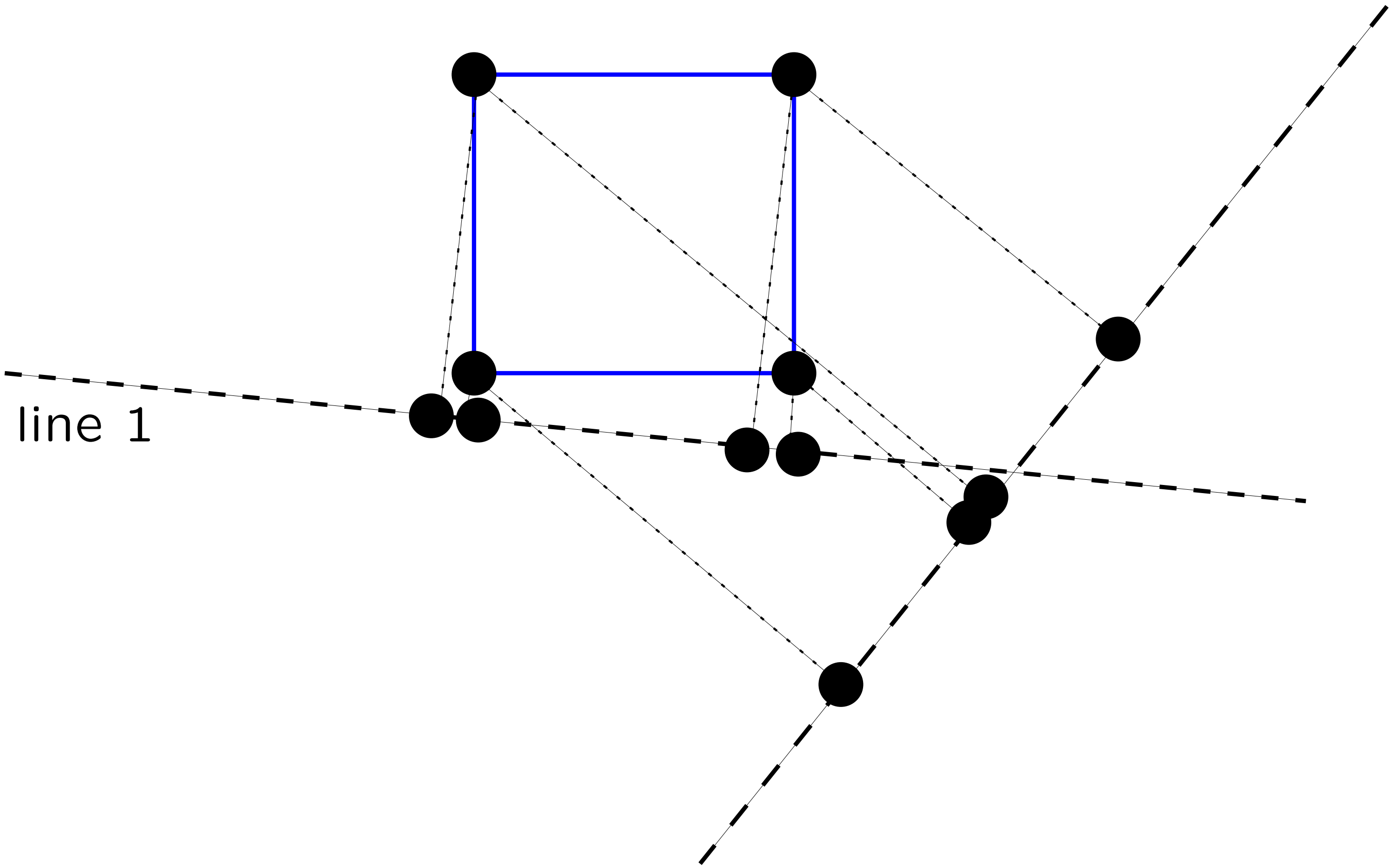
$$\underbrace{\exists x \in X \;\; b = \sum_{i \le n} x_i A_i}_{\text{high dimensional}} \quad \text{iff} \quad \underbrace{\exists x \in X \;\; b' = \sum_{i \le n} x_i A'_i}_{\text{low dimensional}}$$

with high probability

- If this is possible, then solve $\mathsf{RLM}_X(b', A')$
- Since $k \ll m$, solving $\mathsf{RLM}_X(b', A')$ should be faster
- $\mathsf{RLM}_X(b', A') = \mathsf{RLM}_X(b, A)$ with high probability

Losing dimensions = "projection"

In the plane, hopeless



In 3D: no better

The Johnson-Lindenstrauss Lemma

Johnson-Lindenstrauss Lemma

Thm.

Given $A \subseteq \mathbb{R}^m$ with $|A| = n$ and $\varepsilon > 0$ there is $k \sim O(\frac{1}{\varepsilon^2} \ln n)$ and a $k \times m$ matrix T s.t.

$$\forall x, y \in A \quad (1-\varepsilon)\|x-y\| \leq \|Tx - Ty\| \leq (1+\varepsilon)\|x-y\|$$

If $k \times m$ matrix T is sampled componentwise from $N(0, \frac{1}{\sqrt{k}})$, then A and TA have almost the same shape

Discrete approximations of $N(0, \frac{1}{\sqrt{k}})$ can also be used, e.g.

$$\mathbf{P}(T_{ij} = \tfrac{1}{\sqrt{k}}) = \mathbf{P}(T_{ij} = -\tfrac{1}{\sqrt{k}}) = \tfrac{1}{6},\ \mathbf{P}(T_{ij} = 0) = \tfrac{2}{3}$$

(This makes T sparser)

Sampling to desired accuracy

- *(one can show that)* Distortion has low probability:

$$\forall x, y \in A \quad \mathbf{P}(\|Tx - Ty\| \leq (1-\varepsilon)\|x-y\|) \;\leq\; \frac{1}{n^2}$$

$$\forall x, y \in A \quad \mathbf{P}(\|Tx - Ty\| \geq (1+\varepsilon)\|x-y\|) \;\leq\; \frac{1}{n^2}$$

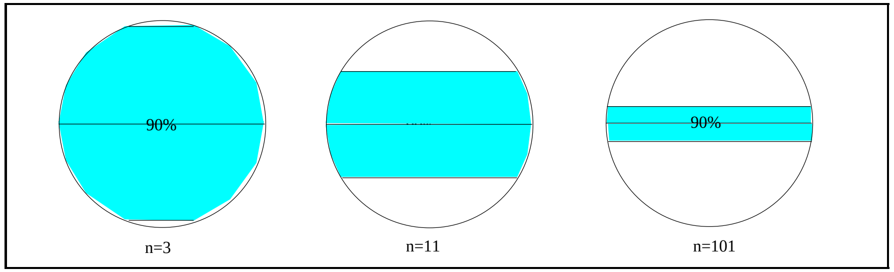
- Probability $\exists$ pair $x, y \in A$ distorting Euclidean distance:
  union bound over $\binom{n}{2}$ pairs

$$\mathbf{P}(\neg(A \text{ and } TA \text{ have almost the same shape})) \;\leq\; \binom{n}{2}\frac{2}{n^2} = 1 - \frac{1}{n}$$

$$\mathbf{P}(A \text{ and } TA \text{ have almost the same shape}) \;\geq\; \frac{1}{n}$$

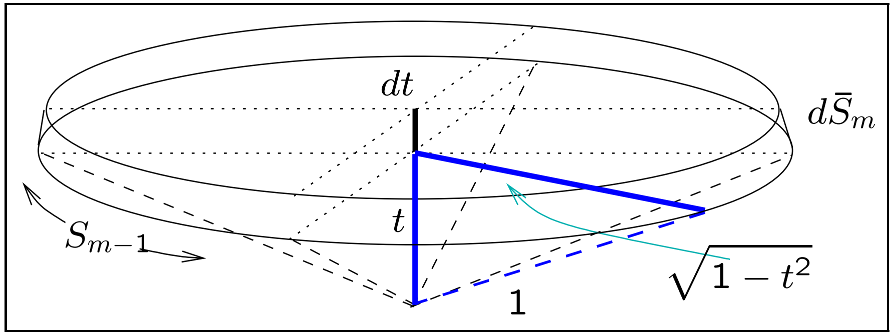
$\Rightarrow$ re-sampling T gives JLL with arbitrarily high probability

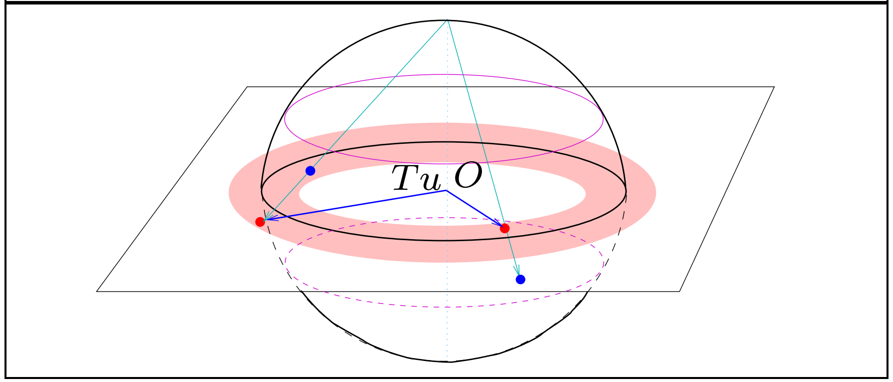
Sketch of a JLL proof by pictures

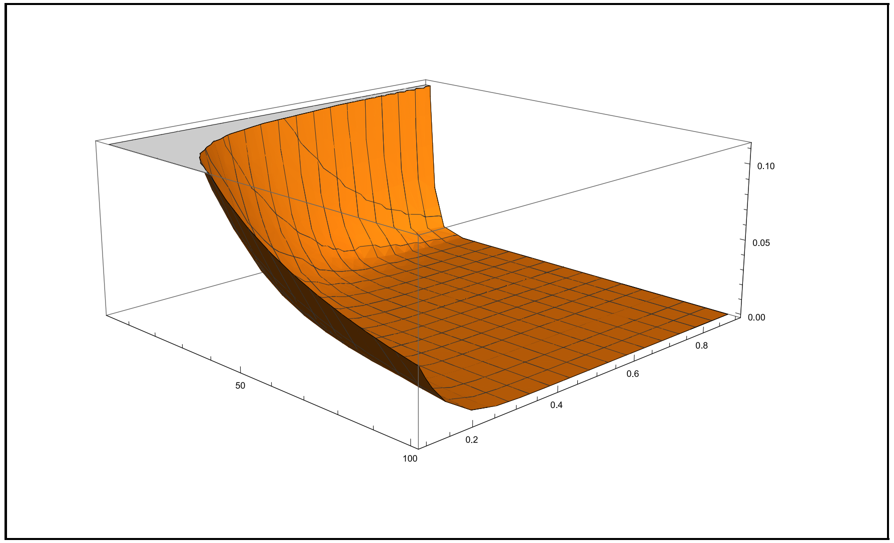


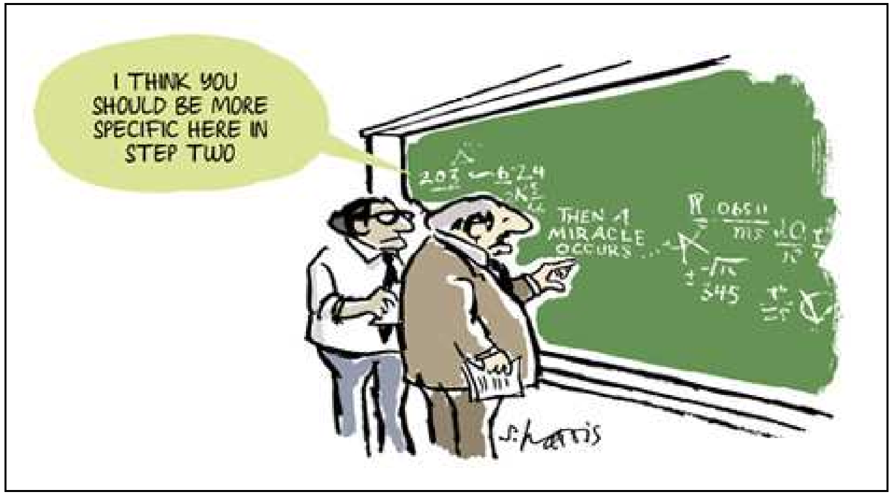
Thm.

Let T be a $k \times m$ rectangular matrix with each component sampled from $N(0, \frac{1}{\sqrt{k}})$, and $u \in \mathbb{R}^m$ s.t. $\|u\| = 1$. Then $\mathsf{E}(\|Tu\|^2) = 1$









In practice

- Empirical estimation of $\mathcal{C}$ in $k = \mathcal{C}\varepsilon^2 \ln n$: $\mathcal{C} \approx 1.8$
  [Venkatasubramanian & Wang 2011]
- Empirically, sample T very few times (e.g. once will do!)
  on average $\|Tx - Ty\| \approx \|x - y\|$*, and distortion decreases exponentially with* n

We only need a logarithmic number of dimensions
in function of the number of points

Surprising fact:

k *is independent of the original number of dimensions* m

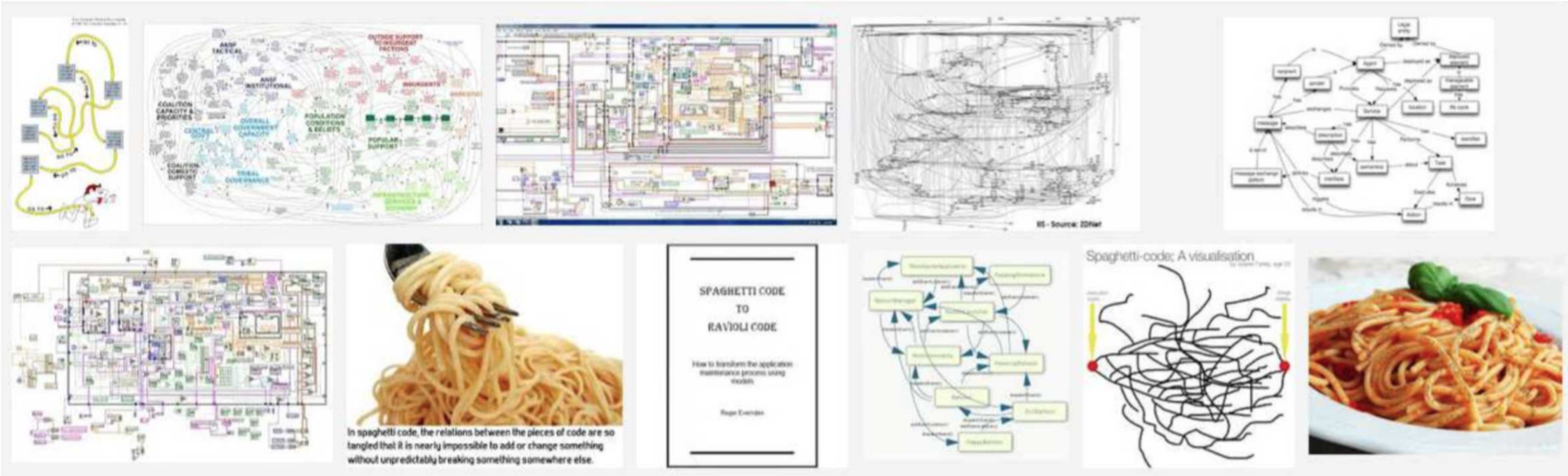
Typical applications of JLL

Problems involving Euclidean distances only

- Euclidean clustering
  k-means, k-nearest neighbors

- Linear regression
  $\min_x \|Ax - b\|_2$ where A is $m \times n$ with $m \gg n$

Google Images




```
VHimg = Map[Flatten[ImageData[#]] &, Himg];
```

```
VHcl = Timing[ClusteringComponents[VHimg, 3, 1]]

{0.405908, {1, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3}}
```

```
Get["Projection.m"];
VKimg = JohnsonLindenstrauss[VHimg, 0.1];
VKcl = Timing[ClusteringComponents[VKimg, 3, 1]]

{0.002232, {1, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3}}
```

Projecting feasibility

Projecting infeasibility (easy cases)

Thm.

$T : \mathbb{R}^m \to \mathbb{R}^k$ a JLL random projection, $b, A_1, \ldots, A_n \in \mathbb{R}^m$ a RLM$_X$ instance. For any given vector $x \in X$, we have:

(i) If $b = \sum_{i=1}^{n} x_i A_i$ then $Tb = \sum_{i=1}^{n} x_i T A_i$

(ii) If $b \neq \sum_{i=1}^{n} x_i A_i$ then $\mathbf{P}\left(Tb \neq \sum_{i=1}^{n} x_i T A_i\right) \geq 1 - 2e^{-\mathcal{C}k}$

(iii) If $b \neq \sum_{i=1}^{n} y_i A_i$ for all $y \in X \subseteq \mathbb{R}^n$, where $|X|$ is finite, then

$$\mathbf{P}\left(\forall y \in X \;\; Tb \neq \sum_{i=1}^{n} y_i T A_i\right) \geq 1 - 2|X|e^{-\mathcal{C}k}$$

for some constant $\mathcal{C} > 0$ (independent of n, k).

[VPL, arXiv:1507.00990v1/math.OC]

Proof (ii)

Cor.

$\forall \varepsilon \in (0,1)$ and $z \in \mathbb{R}^m$, there is a constant $\mathcal{C}$ such that

$$\mathbf{P}((1-\varepsilon)\|z\| \le \|Tz\| \le (1+\varepsilon)\|z\|) \ge 1 - 2e^{-\mathcal{C}\varepsilon^2 k}$$

Proof

By the JLL

Lemma

If $z \neq 0$, there is a constant $\mathcal{C}$ such that $\mathbf{P}(Tz \neq 0) \ge 1 - 2e^{-\mathcal{C}k}$

Proof

Consider events $\mathscr{A} : Tz \neq 0$ and $\mathscr{B} : (1-\varepsilon)\|z\| \le \|Tz\| \le (1+\varepsilon)\|z\|$
$\Rightarrow \mathscr{A}^{\mathsf{c}} \cap \mathscr{B} = \varnothing$, othw $Tz = 0 \Rightarrow (1-\varepsilon)\|z\| \le \|Tz\| = 0 \Rightarrow z = 0$, contradiction
$\Rightarrow \mathscr{B} \subseteq \mathscr{A} \Rightarrow \mathbf{P}(\mathscr{A}) \ge \mathbf{P}(\mathscr{B}) \ge 1 - e^{-\mathcal{C}\varepsilon^2 k}$ by Corollary
Holds $\forall \varepsilon \in (0,1)$ hence result

Now it suffices to apply the Lemma to $Ax - b$

Remarks

- (i) and (ii): checking certificates
  given x, with high probability $b = \sum_i x_i A_i \Leftrightarrow Tb = \sum_i x_i T A_i$

- (iii) RLM_X whenever $|X|$ is polynomially bounded
  e.g. knapsack set $\{x \in \{0,1\}^n \mid \sum_{i \leq n} \alpha_i x_i \leq d\}$ for a fixed d with $\alpha > 0$

- (iii) hints that LFP case is more complicated
  as $X = \mathbb{R}^n_+$ is *not* polynomially bounded

Separating hyperplanes

When $|X|$ is large, project separating hyperplanes instead

- **Convex $C \subseteq \mathbb{R}^m$, $x \notin C$: then $\exists$ hyperplane c separating x, C**
- In particular, true if $C = \text{cone}(A_1, \ldots, A_n)$ for $A \subseteq \mathbb{R}^m$
- **We aim to show $x \in C \Leftrightarrow Tx \in TC$ with high probability**
- As above, if $x \in C$ then $Tx \in TC$ by linearity of T
  real issue is proving the converse

Projecting the separation

Thm.

Given $c, b, A_1, \ldots, A_n \in \mathbb{R}^m$ of unit norm s.t. $b \notin \text{cone}\{A_1, \ldots, A_n\}$ pointed, $\varepsilon > 0$, $c \in \mathbb{R}^m$ s.t. $c^\top b < -\varepsilon$, $c^\top A_i \geq \varepsilon$ $(i \leq n)$, and T a random projector:

$$\mathbf{P}\left[Tb \notin \text{cone}\{TA_1, \ldots, TA_n\}\right] \geq 1 - 4(n+1)e^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k}$$

for some constant $\mathcal{C}$.

Proof

Let $\mathscr{A}$ be the event that T approximately preserves $\|c - \chi\|^2$ and $\|c + \chi\|^2$ for all $\chi \in \{b, A_1, \ldots, A_n\}$. Since $\mathscr{A}$ consists of $2(n+1)$ events, by the JLL Corollary (squared version) and the union bound, we get

$$\mathbf{P}(\mathscr{A}) \geq 1 - 4(n+1)e^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k}$$

Now consider $\chi = b$

$$\begin{aligned}
\langle Tc, Tb \rangle &= \frac{1}{4}(\|T(c+b)\|^2 - \|T(c-b)\|^2) \\
\text{by JLL} \quad &\leq \frac{1}{4}(\|c+b\|^2 - \|c-b\|^2) + \frac{\varepsilon}{4}(\|c+b\|^2 + \|c-b\|^2) \\
&= c^\top b + \varepsilon < 0
\end{aligned}$$

and similarly $\langle Tc, TA_i \rangle \geq 0$

[VPL, arXiv:1507.00990v1/math.OC]

Is this useful?

Previous results look like:

orig. LFP infeasible $\Rightarrow$ $\mathbf{P}$(proj. LFP infeasible) $\geq 1 - p(n)e^{-\mathcal{C}r(\varepsilon)k}$

where p, r two polynomials

- Pick a suitable $\delta > 0$
- Choose $k \sim O(\frac{1}{\mathcal{C}r(\varepsilon)}(\ln p(n) + \ln \frac{1}{\delta}))$ so that RHS $\geq 1 - \delta$
- Preserve infeasibility with probability $\geq 1 - \delta$
- Useful for $m \leq n$ large enough that $k \ll m$

Remarks

- **Applicable to LFP**
- Probability depends on ε (the larger the better)
- Largest ε given by LP
  max$\{\varepsilon \geq 0 \mid c^\top b \leq -\varepsilon \wedge \forall i \leq n \, (c^\top A_i \geq \varepsilon)\}$
- **If** cone$(A_1, \ldots, A_n)$ **is almost non-pointed,** ε **can be very small**

Projecting minimum distances to a cone

- **Thm.**: minimum distance to a cone is approximately preserved

  (proof: technical)

- **This result also works with non-pointed cones**

  Trade-off: need larger k, m, n

- We appear to be all set for LFPs

The feasibility projection theorem

Established so far:

Thm.

Given $\delta > 0$, $\exists$ sufficiently large $m \leq n$ such that:

for any LFP input A, b where A is $m \times n$

we can sample a random $k \times m$ matrix T with $k \ll m$ and

$$\mathbf{P}(\text{orig. LFP feasible} \iff \text{proj. LFP feasible}) \geq 1 - \delta$$

Projecting optimality

Notation

- $P \equiv \min\{cx \mid Ax = b \wedge x \geq 0\}$ *(original problem)*

- $TP \equiv \min\{cx \mid TAx = Tb \wedge x \geq 0\}$ *(projected problem)*

- $v(P) =$ optimal objective function value of P

- $v(TP) =$ optimal objective function value of TP

The optimality projection theorem

- Assume feas(P) is bounded
- Assume all optima of P satisfy $\sum_j x_j \leq \theta$ for some given $\theta > 0$

  (prevents cones from being "too flat")

Thm.

Given $\delta > 0$,

$$v(P) - \delta \leq v(TP) \leq v(P) \qquad (*)$$

holds with arbitrarily high probability (w.a.h.p.)

in fact $(*)$ holds with prob. $1 - 4ne^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k}$ where $\varepsilon = \delta/(2(\theta+1)\eta)$ and $\eta = O(\|y\|_2)$ where y is a dual optimal solution of P having minimum norm

The easy part: $v(TP) \leq v(P)$

- Constraints of P: $Ax = b \ \wedge \ x \geq 0$
- Constraints of TP: $TAx = Tb \ \wedge \ x \geq 0$
- $\Rightarrow$ constraints of TP are lin. comb. of constraints of P
- $\Rightarrow$ any solution of P is feasible in TP
  (btw, the converse holds almost never)
- P and TP have the same objective function
- $\Rightarrow$ TP is a **relaxation** of P $\Rightarrow v(TP) \leq v(P)$

The hard part (sketch)

- Eq. (1) equivalent to P for $\delta = 0$

$$\left.\begin{array}{rcl} cx + s & = & v(P) - \delta \\ Ax & = & b \\ (x,s) & \geq & 0 \end{array}\right\} \qquad (1)$$

  Note: for $\delta > 0$, Eq. (1) is infeasible

- *"By feasibility projection theorem"*,

$$\left.\begin{array}{rcl} cx + s & = & v(P) - \delta \\ TAx & = & Tb \\ (x,s) & \geq & 0 \end{array}\right\}$$

  is infeasible w.a.h.p. for $\delta > 0$

- Hence $cx < v(P) - \delta \wedge TAx = Tb \wedge x \geq 0$ infeasible w.a.h.p.

- $\Rightarrow cx \geq v(P) - \delta$ holds w.a.h.p. for $x \in \text{feas}(TP)$

- $\Rightarrow v(P) - \delta \leq v(TP)$

Solution retrieval

Projected solutions are infeasible in P

- $Ax = b \Rightarrow TAx = Tb$ by linearity, however
- Thm.
  For $x \geq 0$ s.t. $TAx = Tb$, $Ax = b$ with probability 0
- Can't get solution for original LFP using projected LFP!

Solution retrieval from optimal basis

- Primal $\min\{c^\top x \mid Ax = b \wedge x \geq 0\} \Rightarrow$ dual $\max\{b^\top y \mid A^\top y \leq c\}$
- Let $x' = \text{sol}(TP)$ and $y' = \text{sol}(\text{dual}(TP))$
- $\Rightarrow (TA)^\top y' = (A^\top T^\top)y' = A^\top(T^\top y') \leq c$
- $\Rightarrow T^\top y'$ is a solution of dual(P)
- $\Rightarrow$ we can compute an optimal basis J for P
- $A_J x_J = b$ not square, but $(A_J^\top A_J)x_J = A_J^\top b$ is
- Solve it, get x_J, pad with zeros to obtain a solution x^* of P

Solving large LPs

Some results on uniform dense LFP

- Matrix product TA takes too long
  ($\exists$ advanced technology we're not exploiting yet)

- **Infeasible instances** *(sizes from 1000×1500 to 2000×2400)*:

Uniform	ϵ	$k \approx$	CPU saving	accuracy
$(-1, 1)$	0.1	$0.5m$	30%	50%
$(-1, 1)$	0.15	$0.25m$	92%	0%
$(-1, 1)$	0.2	$0.12m$	99.2%	0%
$(0, 1)$	0.1	$0.5m$	10%	100%
$(0, 1)$	0.15	$0.25m$	90%	100%
$(0, 1)$	0.2	$0.12m$	97%	100%

- **Feasible instances**:
  - similar CPU savings
  - obviously 100% accuracy

A single random dense LP instance

- A: 5000 $\times$ 5500 binary matrix with 0.8 density
- x: basic solution with nonzeros sampled uniformly in $\{1,\ldots,9\}$
- $b = Ax$
- $c \in \{0,1\}^n$ picked to make x optimal, i.e. $c_j = 0$ iff $x_j > 0$
- $\Rightarrow$ Optimal objective function value: zero
- Solver: CPLEX
- Subsolver for TP: `barrier` *(helps dense)*
- Subsolver for P: `auto`

A single random dense LP instance

JLL			P		TP		
ε	$O()$	k	$v(P)$	CPU	$v(TP)$	CPU	infeas/rhs
0.05	1.2	4134	0.0	**697.1**	0.0	923.6	0.001%
0.1	1.2	1033	0.0	697.1	0.0	**120.1**	0.016%
0.1	2.0	1722	0.0	697.1	0.0	**207.8**	0.010%
0.1	2.5	2153	0.0	697.1	0.0	**286.8**	0.007%

- infeas/rhs $= \frac{1}{1b} \sum_{i \le m} |A_i x^* - b_i|$
- **"Best" configuration**: $\varepsilon = 0.1$ *and* $O()$ *const* $= 2.5$

Some "meh" results from `netlib`

Instance				P		TP		
name	m	n	k	$v(P)$	CPU	$v(TP)$	CPU	infeas/rhs
`bnl2`	2324	4487	2102	1811.24	**0.06**	1073.85	21.1	0.10%
`d2q06c`	2171	5832	2167	122784	**0.35**	122784	68.2	0.00%
`fit2p`	3000	13526	2378	68464	**0.29**	54372	20.4	0.61%
`greenbea`	2392	5599	2157	-7.25e+7	**0.27**	-7.32e+7	157.6	0.00%
`greenbeb`	2392	5599	2157	-4.30e+6	**0.14**	-4.34e+6	35.5	0.00%
`maros-r7`	3136	9409	2287	1.49e+6	**0.59**	1.05e+6	52.6	0.21%

All other instances were too small ($k > m$)

Conclusion, current work & dreams

- **Where we're at**
  not ready to beat CPLEX on sparse LPs
  but heading in the right direction!

- **What we want to try next**
  random projections *directly* on dual LP
  allows explicit feasibility & optimality guarantees

- **What we dream of**
  projecting *both* row and column sizes at the same time

Some references

- K. Vu, P.-L. Poirion, L. Liberti, *Using the Johnson-Lindenstrauss lemma in linear and integer programming*, arXiv report 1507.00990v1/math.OC, 2015
- K. Vu, P.-L. Poirion, L. Liberti, *Gaussian random projections for Euclidean membership problems*, arXiv report 1509.00630v1/math.OC, 2015

- B. Prabhakar, A. Kulkarni, *Dimensionality reduction of affine variational inequalities using random projections*, arXiv report 1408.4551v2/math.OC, 2014
- W. Johnson and J. Lindenstrauss, *Extensions of Lipschitz mappings into a Hilbert space*, in Contemporary Mathematics, Vol. 26, 189-206, 1984
- S. Dasgupta and A. Gupta, *An elementary proof of a theorem of Johnson and Lindenstrauss*, Random Structures and Algorithms, **22**(1):60-65, 2003
- S. Venkatasubramanian, Q. Wang, *The Johnson-Lindenstrauss transform: an empirical study*, in Proceedings of ALENEX, 2011
- J. Nash, *C^1 Isometric embeddings*, in Annals of Mathematics, **60**(3):383-396, 1954

A few results on fully dense random (infeasible) IFP

		Uniform			**Exponential**			**Gamma**		
m	n	Acc.	Orig.	Proj.	Acc.	Orig.	Proj.	Acc.	Orig.	Proj.
500	800	100%	20.32s	**4.15**s	100%	**4.69**s	10.56s	100%	**4.25**s	8.11s
600	800	100%	26.41s	**4.22**s	100%	**6.08**s	10.45s	100%	**5.96**s	8.27s
700	800	100%	38.68s	**4.19**s	100%	**8.25**s	10.67s	100%	**7.93**s	10.28s
600	1000	100%	51.20s	**7.84**s	100%	10.31s	**8.47**s	100%	8.78s	**6.90**s
700	1000	100%	73.73s	**7.86**s	100%	12.56s	**10.91**s	100%	9.29s	**8.43**s
800	1000	100%	117.8s	**8.74**s	100%	14.11s	**10.71**s	100%	12.29s	**7.58**s
900	1000	100%	130.1s	**7.50**s	100%	15.58s	**10.75**s	100%	15.06s	**7.65**s
1000	1500	100%	275.8s	**8.84**s	100%	38.57s	**22.62**s	100%	35.70s	**8.74**s