Advanced Mathematical Programming

Formulations & Applications

Leo Liberti, CNRS LIX Ecole Polytechnique
`liberti@lix.polytechnique.fr`

INF580

Outline

What is *Mathematical Programming*?

- Formal declarative language for describing optimization problems
- As expressive as any imperative language
- Interpreter = solver
- Shifts focus from *algorithmics* to *modelling*

Syntax

- **A valid sentence:**

$$\left.\begin{array}{ll} \min & x_1 + 2x_2 - \log(x_1 x_2) \\ & x_1 x_2^2 \geq 1 \\ & 0 \leq x_1 \leq 1 \\ & x_2 \in \mathbb{N}. \end{array}\right\} \qquad [P]$$

- **An invalid one:**

$$\left.\begin{array}{ll} \min & \frac{\cdot}{x_2} + x_1 + + \sin\cos \\ & x_{x_2} \geq x_{x_1} \\ & \sum\limits_{i \leq x_1} x_i = 0 \\ & x_1 \neq x_2 \\ & x_1 < x_2. \end{array}\right\}$$

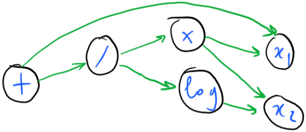
MINLP Formulation

Given functions $f, g_1, \ldots, g_m : \mathbb{Q}^n \to \mathbb{Q}$ and $Z \subseteq \{1, \ldots, n\}$

$$\left.\begin{array}{rrcl} \min & f(x) & & \\ \forall i \le m & g_i(x) & \le & 0 \\ \forall j \in Z & x_j & \in & \mathbb{Z} \end{array}\right\} \quad [P]$$

- $\phi(x) = 0 \quad \Leftrightarrow \quad (\phi(x) \le 0 \wedge -\phi(x) \le 0)$
- $L \le x \le U \quad \Leftrightarrow \quad (L - x \le 0 \wedge x - U \le 0)$
- f, g_i **represented by** ***expression DAGs***

$$x_1 + \frac{x_1 x_2}{\log(x_2)}$$



Class of all formulations P: $\mathbb{MP}$

Semantics

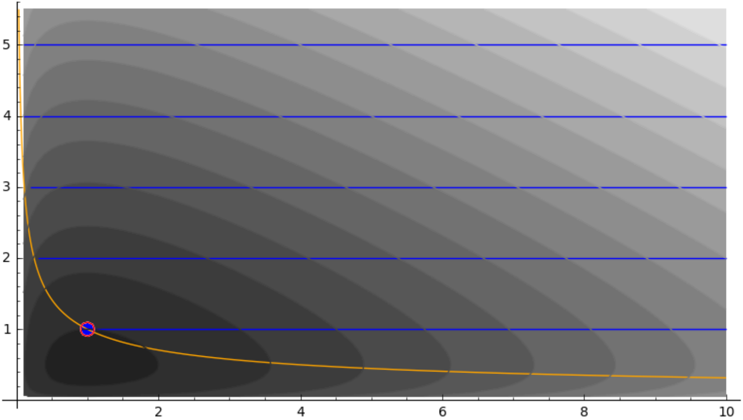
- **Given $P \in \mathbb{MP}$, there are three possibilities:**
  $[\![P]\!]$ exists, P is unbounded, P is infeasible
- ***P is feasible* iff $[\![P]\!]$ exists or is unbounded**
  otherwise it is infeasible
- ***P has an optimum* iff $[\![P]\!]$ exists**
  otherwise it is infeasible or unbounded

Are feasibility and optimality really different?

- **Feasibility prob. $g(x) \le 0$:**
  can be written as MP $\min\{0 \mid g(x) \le 0\}$
- **Bounded MP $\min\{f(x) \mid g(x) \le 0\}$:**
  bisection on f_0 in $f(x) \le f_0 \wedge g(x) \le 0$
- **Unbounded MP: not equivalent to feasibility**

Example

$$P \equiv \min\{x_1 + 2x_2 - \log(x_1 x_2) \mid x_1 x_2^2 \geq 1 \wedge 0 \leq x_1 \leq 1 \wedge x_2 \in \mathbb{N}\}$$



$[\![P]\!] = (\mathsf{opt}(P), \mathsf{val}(P))$ $\mathsf{opt}(P) = (1, 1)$ $\mathsf{val}(P) = 3$

Solvers (or "interpreters")

- Take formulation P as input
- Output $[\![P]\!]$ *and possibly other information*
- Trade-off between generality and efficiency

(i) Linear Programming (LP)
f, g_i linear, $Z = \varnothing$

(ii) Mixed-Integer LP (MILP)
f, g_i linear, $Z \neq \varnothing$

(iii) Nonlinear Programming (NLP)
some nonlinearity in f, g_i, $Z = \varnothing$
f, g_i convex: convex NLP (cNLP)

(iv) Mixed-Integer NLP (MINLP)
some nonlinearity in f, g_i, $Z \neq \varnothing$
f, g_i convex: convex MINLP (cMINLP)

- Each solver targets a given class

Some application fields

- Production industry
  planning, scheduling, allocation, ...
- Transportation & logistics
  facility location, routing, rostering, ...
- Service industry
  pricing, strategy, product placement, ...
- Energy industry (*all of the above*)
- Machine Learning & Artificial Intelligence
  clustering, approximation error minimization
- Biochemistry & medicine
  protein structure, blending, tomography, ...
- Mathematics
  Kissing number, packing of geometrical objects,...

Outline

Can we solve MPs?

- **"Solve MPs": is there an algorithm $\mathcal{D}$ s.t.:**

$$\forall P \in \mathbb{MP} \quad \mathcal{D}(P) = \begin{cases} \mathsf{infeasible} & P \text{ is infeasible} \\ \mathsf{unbounded} & P \text{ is unbounded} \\ [\![P]\!] & otherwise \end{cases}$$

- **I.e. does there exist a single, all-powerful solver?**

Formal systems (FS)

- A *formal system* consists of:
  - an *alphabet*
  - a *formal grammar*
    allowing the determination of *formulæ* and *sentences*
  - a set A of *axioms* (given sentences)
  - a set R of *inference rules*
    allowing the derivation of new sentences from old ones
- A *theory* T is the smallest set of sentences that is obtained by recursively applying R to A

Example: PA1

- **Theory:** 1st order sentences about $\mathbb{N}$
- **Alphabet:** $+, \times, \wedge, \vee, \forall, \exists, \neg, =, S(\cdot)$ and variable names
- **Peano's Axioms:**

  1. $\forall x\ (0 \neq S(x))$
  2. $\forall x, y\ (S(x) = S(y) \to x = y)$
  3. $\forall x\ (x + 0 = x)$
  4. $\forall x\ (x \times 0 = 0)$
  5. $\forall x, y\ (x + S(y) = S(x + y))$
  6. $\forall x, y\ (x \times S(y) = x \times y + x)$
  7. axiom schema over all $(k+1)$-ary ϕ: $\forall y = (y_1, \ldots, y_k)$ $(\phi(0, y) \wedge \forall x \phi(x, y) \to \phi(S(x), y)) \to \forall x \phi(x, y)$

- **Inference:** see `https://en.wikipedia.org/wiki/List_of_rules_of_inference` e.g. *modus ponens* $((P \to Q) \wedge P) \to Q)$
- Generates ring $(\mathbb{Z}, +, \times)$ and arithmetical proofs
  e.g. $\exists x \in \mathbb{N}^n\ \forall i\ (p_i(x) \leq 0)$ (polynomial MINLP feasibility)

Example: Reals

- **Theory:** **polynomial systems over** $\mathbb{R}$
- **Alphabet:** $+, \times, \wedge, \vee, \forall, \exists, =, <, \leq, 0, 1$,**variable names**
- **Axioms:** **field and order**
- **Inference:** **see**

  `https://en.wikipedia.org/wiki/List_of_rules_of_inference`

  e.g. *modus ponens* $((P \to Q) \wedge P) \to Q)$
- **Generates polynomial rings** $\mathbb{R}[X_1, \ldots, X_k]$ **(for all** k**)**

  e.g. $\exists x \in \mathbb{R}^n \ \forall i \ (p_i(x) \leq 0)$ **(polynomial NLP feasibility)**

The use of formal systems

Given a FS $\mathcal{F}$:

- A *decision problem* is a set P of sentences
  Decide if a given sentence f belongs to P
- Decidability in formal systems:
  $$P \equiv \textbf{provable sentences}$$
- *Proof of f*: finite sequence of sentences ending with f; sentences either axioms or derived from predecessors by inference rules
- PA1: decide if sentence f about $\mathbb{N}$ has a proof
  PA1 contains $\exists x \in \mathbb{Z}^n\ \forall i\ p_i(x) \leq 0$ **(poly p)**
- Reals: decide if sentence f about $\mathbb{R}$ has a proof
  Reals contains $\exists x \in \mathbb{Z}^n\ \forall i\ p_i(x) \leq 0$ **(poly p)**
- Formal study of MINLP/NLP feasibility

Decidability, computability, solvability

- Decidability: applies to decision problems
- Computability: applies to function evaluation
  - Is the function f, mapping i to the i-th prime integer, computable?
  - Is the function g, mapping Cantor's CH to 1 if provable in ZFC axiom system and to 0 otherwise, computable?
- Solvability: applies to other problems
  E.g. to optimization problems!

Completeness and decidability

- *Complete* **FS $\mathcal{F}$:**
  for $f \in \mathcal{F}$, either f or $\neg f$ is provable
  otherwise $\mathcal{F}$ is *incomplete*
- *Decidable* **FS $\mathcal{F}$:**
  $\exists$ algorithm $\mathcal{D}$ s.t.

$$\forall f \in \mathcal{F} \begin{cases} \mathcal{D}(f) = 1 & \textbf{iff } f \textbf{ is provable} \\ \mathcal{D}(f) = 0 & \textbf{iff } f \textbf{ is not provable} \end{cases}$$

  otherwise $\mathcal{F}$ is *undecidable*

Example: PA1

- **Gödel's 1st incompleteness theorem:**
  PA1 **is incomplete**
- **Turing's theorem:** PA1 **is undecidable**
- PA1 **is undecidable and incomplete**

Gödel's 1st incompleteness theorem

- $\mathcal{F}$: any FS extending PA1
- **Thm.** $\boxed{\mathcal{F} \text{ is either incomplete or inconsistent}}$
- ϕ: sentence "ϕ not provable in $\mathcal{F}$"
  denoted $\mathcal{F} \nvdash \phi$; it can be constructed in $\mathcal{F}$; hard part of thm.
  - *Assume $\mathcal{F}$ is complete:* either $\mathcal{F} \vdash \phi$ or $\mathcal{F} \vdash \neg\phi$
  - If $\mathcal{F} \vdash \phi$ then $\mathcal{F} \vdash (\mathcal{F} \nvdash \phi)$ i.e. $\mathcal{F} \nvdash \phi$, contradiction
  - If $\mathcal{F} \vdash \neg\phi$ then $\mathcal{F} \vdash \neg(\mathcal{F} \nvdash \phi)$ i.e. $\mathcal{F} \vdash (\mathcal{F} \vdash \phi)$
    this implies $\mathcal{F} \vdash \phi$, i.e. $\mathcal{F} \vdash \phi \wedge \neg\phi$, $\mathcal{F}$ inconsistent
  - *Assume $\mathcal{F}$ is inconsistent:* any sentence is provable, i.e. $\mathcal{F}$ complete

    details: $0 = 1$, hence $0 \vee \psi$ and $1 \vee \psi$, hence $(0 \wedge 1) \vee \psi$, i.e. ψ, and symmetrically for $\neg\psi$, for any ψ
- **WARNING:** $\mathcal{F} \nvdash \phi \not\equiv \mathcal{F} \vdash \neg\phi$

Turing's theorem

- **Turing Machine (TM):** *computation model*
  - infinite tape with cells storing finite alphabet letters
  - head reads/writes/skips i-th cell, moves left/right
  - states=program (e.g. if s write 0 & move left)
  - initial tape content: input, final tape content: output
  - final state $\perp$: termination; $\varnothing$ nonterm.
- TM dynamics can be written in PA1 statements
- Any PA1 sentence $p(x)$ can be represented by TM:

  `while(1) i=0; if` $p(x)$ `return YES; else i=i+1`

  only terminates if true; loops forever if false
- $\exists$ universal TM (UTM) representing *all* PA1 sentences
- TM termination $\Leftrightarrow$ decidability in PA1
- Halting Problem (HP):

  TM M & input x, is $M(x) = \perp$?
- HP is undecidable

Turing's theorem

- **enumerate all TMs:** $(M_i \mid i \in \mathbb{N})$
- **halting function** $H(i,x) = \begin{cases} 1 & M_i(x) = \bot \\ 0 & M_i(x) = \varnothing \end{cases}$
- **show** $H \neq F$ **for any computable** $F(i,x)$**:**
  - **let** $G(i) = \begin{cases} 0 & F(i,i) = 0 \\ 1 & \textbf{othw} \end{cases}$

    G is partial computable because F is computable
  - **let** M_y **be the TM computing** G
  - **consider** $H(y,y)$**:**
    - **if** $F(y,y) = 0$ **then** $G(y) = 0$
      so $M_y(y) = \bot$ **and** $H(y,y) = 1$
    - **if** $F(y,y) \neq 0$ **then** $G(y)$ **is undefined**
      so $M_y(y) = \varnothing$ **and** $H(y,y) = 0$
  - **so** $H(y,y) \neq F(y,y)$ **for all** y
- H **is uncomputable** $\Rightarrow$ PA1 **is undecidable**

Example: Reals

- Tarski's theorem: Reals is decidable
- **Algorithm:**
  constructs solution sets (YES) or derives contradictions(NO)
  $\Rightarrow$ **provides proofs or contradictions for all sentences!**
- $\Rightarrow$ Reals is complete
- Reals is decidable and complete

Tarski's theorem

- **Algorithm based on quantifier elimination**
- *Feasible sets of polynomial systems* $p(x) \leq 0$ *have finitely many connected components*
- **Each connected component recursively built of cylinders over points or intervals**
  extremities: pts., $\pm\infty$*, algebraic curves at previous recursion levels*
- **In some sense, generalization of Reals in** $\mathbb{R}^1$

Dense linear orders

Given a sentence ϕ in DLO

- Reduce to DNF $\exists x_i \; q_i(x)$ with $q_i = \bigwedge q_{ij}$
- Each q_{ij} has form $s = t$ or $s < t$ (s, t vars or consts)
  - s, t both constants:
    $s < t$, $s = t$ verified and replaced by 1 or 0
  - s, t the same variable x_i:
    $s < t$ replaced by 0, $s = t$ replaced by 1
  - if s is x_i and t is not:
    $s = t$ means "replace x_i by t" (eliminate x_i)
  - remaining case:
    q_i conj. of $s < x_i$ and $x_i < t$:
    replace by $s < t$ (eliminate x_i)
- q_i no longer depends on x_i, rewrite $\exists x_i \; q_i$ as q_i
- Repeat over vars. x_i, obtain real intervals or contradictions

Quantifier elimination!

Decidability and completeness

- PA1 is incomplete and undecidable
- Reals is complete and decidable
- Are there FS $\mathcal{F}$ that are:
  - incomplete and decidable?
  - complete and undecidable?

Incomplete and decidable (trivial)

- NoInference:
  Any FS with $<\infty$ axiom schemata and no inference rules
- **Only possible proofs: sequences of axioms**
- **Only provable sentences: axioms**
- **For any other sentence f: no proof of f or $\neg f$**
- **Trivial decision algorithm:**
  given f, output YES if f is a finite axiom sequence, NO otherwise
- NoInference **is decidable and incomplete**

Incomplete and decidable (nontrivial)

- ACF: Algebraically Closed Fields (e.g. $\mathbb{C}$)
  field axioms + "every polynomial splits" schema
- *ACF* ***decidable*** *by quantifier elimination*
- ACF_p: $\mathsf{ACF} \cup$ **AXIOM**$(\mathsf{C}_p \equiv [\sum_{j \leq p} 1 = 0])$ (p **prime**)
- $\forall p$ (prime) C_p independent of ACF $\Rightarrow$
  $\Rightarrow$ ***decidability as in ACF***
- $\exists$ *fields of every prime characteristic* p
  $\Rightarrow$ *each* ACF_p *satisfies* C_p *and negates* C_q *for* $q \neq p$
- In ACF, no proof of C_p nor $\neg\mathsf{C}_p$ possible
- Decision alg. $\mathcal{D}(\psi)$ for ACF:
  - if $\psi \equiv C_p$ or $\neg C_p$ for some prime p, return NO
  - else run quantifier elimination on ψ
- ACF is decidable and incomplete

if ACF axioms include $\neg C_p$ *for all* p*, then ACF complete*

Complete and undecidable (impossible)

- FS $\mathcal{F}$ complete:
  $\forall\psi \in \mathcal{F}\ \exists$ proof of ψ or $\neg\psi$
- Proofs are finite sequences of sentences
- Algorithm $\mathcal{D}$:
  1. iteratively generate all (countably many) proofs
     combine axioms w/inference rules and repeat
  2. for each new sentence τ, is $\tau \equiv \psi$ or $\tau \equiv \neg\psi$?
     Return 1 or 0 and break; else continue
- $\mathcal{D}$ terminates because $\mathcal{F}$ is complete
- If FS is complete, then it is decidable

The two meanings of *completeness*

- **WARNING!!!**
  "complete" is used in two different ways in logic
  1. Gödel's 1st incompleteness theorem
     FS $\mathcal{F}$ complete if ϕ or $\neg\phi$ provable $\forall\phi$
  2. A: sentences; R: inference rules
     A *complete* wrt R if $A \vDash \psi \Rightarrow A \vdash \psi$
     - $A \vDash \psi$: ψ is *logically valid*
       never false for any FS w/axioms A and infer. rules R
     - Gödel's completeness theorem: FOL is complete
- *Pay attention when reading literature/websites*

Undecidability & Incompleteness

- **[Nonexistence of a proof for f] $\not\equiv$ [Proof of $\neg f$]**
  If FS decidable & incomplete, decision alg. answers NO to f and $\neg f$ for f independent
- ***Information complexity*:**
  decision $=$ 1 bit, proof $=$ many bits
- **Undecidability and incompleteness are different!**

Is MP solvable?

- Hilbert's 10th problem: is there an algorithm for solving polynomial Diophantine equations?
- *Modern formulation*:
  are polynomial systems over $\mathbb{Z}$ solvable?
- [Matiyasevich 1970]: NO
  can use them to model UTM dynamics
- Let $p(\alpha, x) = 0$ be a Univ. Dioph. Eq. (UDE)
- $\min\{0 \mid p(\alpha, x) = 0\}$ is an undecidable (feasibility) MP
- $\min (p(\alpha, x))^2$ is an unsolvable (optimization) MP

Outline

Worst-case algorithmic complexity

- **Computational complexity theory:**
  worst-case time/space taken by an algorithm to complete
- **Algorithm $\mathcal{A}$**
  - e.g. to determine whether a graph $G = (V, E)$ is connected or not
  - input: G; size of input: $\nu = |V| + |E|$
- **How does the CPU time $\tau(\mathcal{A})$ used by $\mathcal{A}$ vary with ν?**
  - $\tau(\mathcal{A}) = O(\nu^k)$ for fixed k: polytime
  - $\tau(\mathcal{A}) = O(2^\nu)$: exponential
- **polytime $\leftrightarrow$ efficient**
- **exponential $\leftrightarrow$ inefficient**

Polytime algorithms are "efficient"

- Why are polynomials special?
- Many different variants of Turing Machines (TM)
- Polytime is *invariant* to all definitions of TM
- In practice, $O(\nu)$-$O(\nu^3)$ is an acceptable range covering most practically useful efficient algorithms
- Many exponential algorithms are also usable in practice for limited sizes

Instances and problems

- An input to an algorithm $\mathcal{A}$: *instance*
- Collection of all inputs for $\mathcal{A}$: *problem*
  consistent with "set of sentences" from decidability
- BUT:
  - A problem can be solved by different algorithms
  - There are problems which no algorithm can solve
- Given a problem P, what is the complexity of the *best algorithm* that solves P?

Complexity classes

- Focus on *decision problems*
- If $\exists$ polytime algorithm for P, then $P \in \mathbf{P}$
- If there is a polytime checkable *certificate* for all YES instances of P, then $P \in \mathbf{NP}$
- No-one knows whether $\mathbf{P} = \mathbf{NP}$ (we think not)
- NP includes problems for which we don't think a polytime algorithms exist

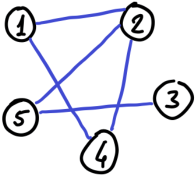
  e.g. k-CLIQUE, SUBSET-SUM, KNAPSACK, HAMILTONIAN CYCLE, SAT, ...

Subsection 1

Some combinatorial problems

k-CLIQUE

- **Instance:** $(G = (V, E), k)$
- **Problem:** determine whether G has a *clique* of size k



- 1-CLIQUE? YES (every graph is YES)
- 2-CLIQUE? YES (every non-empty graph is YES)
- 3-CLIQUE? YES (triangle $\{1, 2, 4\}$ is a certificate)
  certificate can be checked in $O(k) < O(n)$
- 4-CLIQUE? NO
  no polytime certificate unless $\mathrm{P} = \mathrm{NP}$

MP formulations for CLIQUE

Variables? Objective? Constraints?

MP formulations for CLIQUE

Variables? Objective? Constraints?

- *Pure feasibility problem*:

$$\left.\begin{array}{rrcl}\forall \{i,j\} \not\in E & x_i + x_j & \le & 1 \\ & \sum\limits_{i \in V} x_i & = & k \\ & x & \in & \{0,1\}^n \end{array}\right\}$$

MP formulations for CLIQUE

Variables? Objective? Constraints?

- *Pure feasibility problem*:

$$\left.\begin{array}{rrcl} \forall \{i,j\} \notin E & x_i + x_j & \le & 1 \\ & \sum\limits_{i \in V} x_i & = & k \\ & x & \in & \{0,1\}^n \end{array}\right\}$$

- MAX CLIQUE:

$$\left.\begin{array}{rrcl} \max & \sum\limits_{i \in V} x_i & & \\ \forall \{i,j\} \notin E & x_i + x_j & \le & 1 \\ & x & \in & \{0,1\}^n \end{array}\right\}$$

SUBSET-SUM

- **Instance: list** $a = (a_1, \ldots, a_n) \in \mathbb{N}^n$ **and** $b \in \mathbb{N}$
- **Problem: is there** $J \subseteq \{1, \ldots, n\}$ **such that** $\sum_{j \in J} a_j = b$**?**

- $a = (1, 1, 1, 4, 5), b = 3$: **YES** $J = \{1, 2, 3\}$

  all $b \in \{0, \ldots, 12\}$ *yield YES instances*
- $a = (3, 6, 9, 12), b = 20$: **NO**

MP formulations for SUBSET-SUM

Variables? Objective? Constraints?

MP formulations for SUBSET-SUM

Variables? Objective? Constraints?

- *Pure feasibility problem*:

$$\left.\begin{array}{rcl} \sum\limits_{j \le n} a_j x_j & = & b \\ x & \in & \{0,1\}^n \end{array}\right\}$$

KNAPSACK

- **Instance:** $c, w \in \mathbb{N}^n$, $K \in \mathbb{N}$
- **Problem:** find $J \subseteq \{1, \ldots, n\}$ s.t. $c(J) \leq K$ and $w(J)$ is maximum

---

- $c = (1, 2, 3), w = (3, 4, 5), K = 3$
  - $c(J) \leq K$ feasible for J in $\varnothing, \{j\}, \{1, 2\}$
  - $w(\varnothing) = 0, w(\{1, 2\}) = 3 + 4 = 7, w(\{j\}) \leq 5$ for $j \leq n$

    $\Rightarrow J_{\mathsf{max}} = \{1, 2\}$
- $K = 0$: infeasible

- natively expressed as an optimization problem
- *notation:* $c(J) = \sum_{j \in J} c_j$ *(similarly for* $w(J)$*)*

MP formulation for KNAPSACK

Variables? Objective? Constraints?

MP formulation for KNAPSACK

Variables? Objective? Constraints?

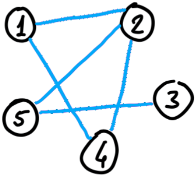
$$\left.\begin{array}{rcl} \max \displaystyle\sum_{j \le n} w_j x_j & & \\ \displaystyle\sum_{j \le n} c_j x_j & \le & K \\ x & \in & \{0,1\}^n \end{array}\right\}$$

Hamiltonian Cycle

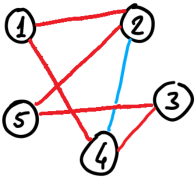
- **Instance:** $G = (V, E)$
- **Problem: does G have a *Hamiltonian cycle*?**

  cycle covering every $v \in V$ exactly once

NO



YES (cert. $1 \to 2 \to 5 \to 3 \to 4 \to 1$)



MP formulation for Hamiltonian Cycle

Variables? Objective? Constraints?

MP formulation for HAMILTONIAN CYCLE

Variables? Objective? Constraints?

$$\forall i \in V \quad \sum_{\substack{j \in V \\ \{i,j\} \in E}} x_{ij} = 1 \qquad (1)$$

$$\forall j \in V \quad \sum_{\substack{i \in V \\ \{i,j\} \in E}} x_{ij} = 1 \qquad (2)$$

$$\forall \varnothing \subsetneq S \subsetneq V \quad \sum_{\substack{i \in S, j \notin S \\ \{i,j\} \in E}} x_{ij} \geq 1 \qquad (3)$$

***WARNING*: second order statement!**

quantified over sets

other warning: need arcs not edges in (1)-(3)

Satisfiability (sat)

- **Instance: boolean logic sentence f in CNF**

$$\bigwedge_{i \leq m} \bigvee_{j \in C_i} \ell_j$$

where $\ell_j \in \{x_j, \bar{x}_j\}$ for $j \leq n$

- **Problem: is there $\phi : x \to \{0,1\}^n$ s.t. $\phi(f) = 1$?**

- $f \equiv (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2)$

  $x_1 = x_2 = 1, x_3 = 0$ **is a YES certificate**

- $f \equiv (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$

ϕ	$x = (1,1)$	$x = (0,0)$	$x = (1,0)$	$x = (0,1)$
false	C_2	C_1	C_3	C_4

MP formulation for SAT

Exercise

Subsection 2

NP-hardness

NP-Hardness

- Do *hard* problems exist? Depends on $\mathbf{P} \neq \mathbf{NP}$
- Next best thing: define *hardest problem in* NP
- A problem P is NP-*hard* if
  Every problem Q in NP can be solved in this way:
  1. given an instance q of Q transform it in polytime to an instance $\rho(q)$ of P s.t. q is YES iff $\rho(q)$ is YES
  2. run the best algorithm for P on $\rho(q)$, get answer $\alpha \in \{\text{YES}, \text{NO}\}$
  3. return α

  ρ is called a *polynomial reduction* from Q to P
- If P is in NP and is NP-hard, it is called NP-*complete*
- Every problem in NP reduces to SAT [Cook 1971]

Cook's theorem

Theorem 1: If a set S of strings is accepted by some nondeterministic Turing machine within polynomial time, then S is P-reducible to {DNF tautologies}.

Boolean decision variables store TM dynamics

Proposition symbols:

$P^i_{s,t}$ for $1 \le i \le \ell$, $1 \le s,t \le T$. $P^i_{s,t}$ is true iff tape square number s at step t contains the symbol σ_i.

Q^i_t for $1 \le i \le r$, $1 \le t \le T$. Q^i_t is true iff at step t the machine is in state q_i.

$S_{s,t}$ for $1 \le s,t \le T$ is true iff at time t square number s is scanned by the tape head.

Definition of TM dynamics in CNF

B_t asserts that at time t one and only one square is scanned:

$$B_t = (S_{1,t} \vee S_{2,t} \vee \ldots \vee S_{T,t}) \,\&\, \Big[\mathop{\&}_{1 \le i < j \le T} (\neg S_{i,t} \vee \neg S_{j,t})\Big]$$

$G^t_{i,j}$ asserts that if at time t the machine is in state q_i scanning symbol σ_j, then at time $t+1$ the machine is in state q_k, where q_k is the state given by the transition function for M.

$$G^t_{i,j} = \mathop{\&}_{s=1}^{T} (\neg Q^i_t \vee \neg S_{s,t} \vee \neg P^j_{s,t} \vee Q^k_{t+1})$$

Description of a dynamical system using a declarative programming language (SAT) — what MP is all about!

Cook's theorem: sets and params

- Reduce nondeterministic polytime TM M to MILP
- Tuple $(Q, \Sigma, s, F, \delta)$:
  states, alphabet, initial, final, transition
- Transition relation δ: $(Q \setminus F \times \Sigma) \times (Q \times \Sigma \times \{-1, 1\})$
- M polytime: terminates in $p(n)$
  n *size of input,* $p(\cdot)$ *polynomial*
- *Index sets:*
  states Q, characters Σ, tape cells I, steps K
  $|K| = O(p(n))$, $|I| = 2|K|$
- *Parameters:*
  initial tape string $\tau_i =$ symbol $j \in \Sigma$ in cell i

Cook's theorem: decision vars

- $\forall i \in I, j \in \Sigma, k \in K$
  $t_{ijk} = 1$ ***iff tape cell*** i ***contains symbol*** j ***at step*** k
- $\forall i \in I, k \in K$
  $h_{ik} = 1$ ***iff head is at tape cell*** i ***at step*** k
- $\forall \ell \in Q, k \in K$
  $q_{\ell k} = 1$ ***iff*** M ***is in state*** ℓ ***at step*** k

Cook's theorem: constraints (informal)

1. *Initialization:*
   1.1 initial string τ on tape at step $k = 0$
   1.2 M in initial state s at step $k = 0$
   1.3 head initial position on cell $i = 0$ at $k = 0$
2. *Execution:*
   2.1 $\forall i, k$: cell i has exactly one symbol j at step k
   2.2 $\forall i, k$: if cell i changes symbol between step k and $k + 1$, head must be on cell i at step k
   2.3 $\forall k$: M is in exactly one state
   2.4 $\forall k, i, j \in \Sigma$: cell i and symbol j in state k lead to possible cells, symbol and states as given by δ
3. *Termination:*
   3.1 M reaches termination at some step $k \leq p(n)$

Cook's theorem: constraints (informal)

1. *Initialization:*

   1.1 $\forall i\ (t_{i,\tau_i,0} = 1)$
   1.2 $q_{s,0} = 1$
   1.3 $h_{0,0} = 1$

2. *Execution:*

   2.1 $\forall i, k\ (\sum_j t_{ijk} = 1)$
   2.2 $\forall i, j \neq j', k < p(n)\ (t_{ijk}\, t_{i,j',k+1} = h_{ik})$
   2.3 $\forall k\ \ \sum_i h_{ik} = 1$
   2.4 $\forall i, \ell, j, k$
   $$(h_{ik}\, q_{\ell k}\, t_{ijk} = \sum_{((\ell,j),(\ell',j',d))\in\delta} h_{i+d,k+1}\, q_{\ell',k+1}\, t_{i+d,j',k+1})$$

3. *Termination:*

   3.1 $\displaystyle\sum_{k, f\in F} q_{fk} = 1$

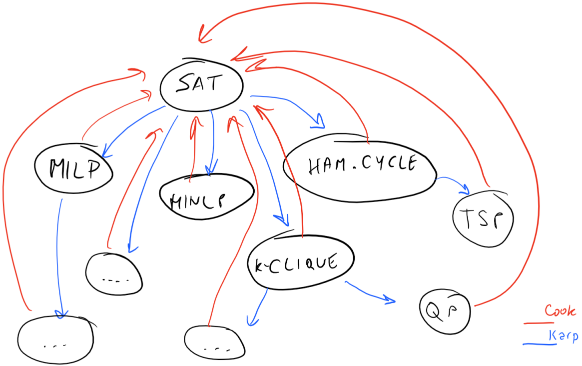
Cook's theorem: conclusion

- **Nonlinear constraints can be linearized:**
  $z = xy \wedge x, y \in \{0,1\} \wedge z \in [0,1] \quad \equiv$
  $z \leq x \wedge z \leq y \wedge z \geq x + y - 1 \wedge x \in \{0,1\} \wedge z \in [0,1]$
- **MILP is feasibility only**
- **MILP has polynomial size**
- $\Rightarrow$ ***MILP is* NP*-hard***

Reduction graph

After Cook's theorem

To prove NP-hardness of a new problem P, pick a known NP-hard problem Q that "looks similar enough" to P and find a polynomial reduction ρ from Q to P **[Karp 1972]**



Why it works: suppose P easier than Q, solve Q by calling $\rho \circ \mathrm{Alg}_P$, conclude Q as easy as P, contradiction

Example of polynomial reduction

- STABLE: given $G = (V, E)$ and $k \in \mathbb{N}$, does it contain a stable set of size k?
- We know k-CLIQUE is NP-complete, reduce from it
  - Given instance (G, k) of CLIQUE consider the *complement graph* (computable in polytime)

$$\bar{G} = (V, \bar{E} = \{\{i, j\} \mid i, j \in V \wedge \{i, j\} \notin E\})$$

  - **Thm.:** G has a clique of size k iff $\bar{G}$ has a stable set of size k
  - $\rho(G) = \bar{G}$ is a polynomial reduction from CLIQUE to STABLE
- $\Rightarrow$ STABLE is $\mathbb{NP}$-hard
- STABLE is also in NP
  $U \subseteq V$ is a stable set iff $E(G[U]) = \varnothing$ (polytime verification)
- $\Rightarrow$ STABLE is $\mathbb{NP}$-complete

MILP is NP-hard (from SAT)

- SAT is NP-hard by Cook's theorem, Reduce from SAT in CNF

$$\bigwedge_{i\le m}\bigvee_{j\in C_i}\ell_j$$

  where ℓ_j is either x_j or $\bar{x}_j \equiv \neg x_j$
- Polynomial reduction ρ

SAT	x_j	$\bar{x}_j$	$\vee$	$\wedge$
MILP	x_j	$1-x_j$	$+$	≥ 1

- E.g. ρ maps $(x_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3)$ to

$$\min\{0 \mid x_1 + x_2 \ge 1 \wedge x_3 - x_2 \ge 0 \wedge x \in \{0,1\}^3\}$$

- SAT is YES iff MILP is feasible

  (same solution, actually)

Complexity of Quadratic Programming

$$\left.\begin{array}{rrcl} \min & x^\top Q x & + & c^\top x \\ & Ax & \geq & b \end{array}\right\}$$

- Quadratic Programming = **QP**
- Quadratic objective, linear constraints, continuous variables
- Many applications (e.g. portfolio selection)
- If Q PSD then objective is convex, problem is in P
- If Q has at least one negative eigenvalue, NP-hard
- *Decision problem: "is the min. obj. fun. value ≤ 0?"*

QP is NP-hard

- By reduction from SAT, let σ be an instance
- $\hat{\rho}(\sigma, x) \geq 1$: linear constraints of SAT $\rightarrow$ MILP reduction
- Consider QP

$$\left.\begin{array}{ll} \min & f(x) = \sum_{j \leq n} x_j(1 - x_j) \\ & \hat{\rho}(\sigma, x) \geq 1 \\ & 0 \leq x \leq 1 \end{array}\right\} \quad (\dagger)$$

- **Claim:** σ is YES iff $\mathsf{val}(\dagger) = 0$
- *Proof:*
  - assume σ YES with soln. x^*, then $x^* \in \{0, 1\}^n$, hence $f(x^*) = 0$, since $f(x) \geq 0$ for all x, $\mathsf{val}(\dagger) = 0$
  - assume σ NO, suppose $\mathsf{val}(\dagger) = 0$, then $(\dagger)$ feasible with soln. x', since $f(x') = 0$ then $x' \in \{0, 1\}$, feasible in SAT hence σ is YES, contradiction

Box-constrained QP is NP-hard

- **Add surplus vars v to SAT→MILP constraints:**
  $\hat{\rho}(\sigma, x) - 1 - v = 0$
  (denote by $\forall i \le m \ (a_i^\top x - b_i - v_i = 0)$)
- **Now sum them on the objective**

$$\left.\begin{array}{ll} \min & \sum\limits_{j \le n} x_j(1 - x_j) + \sum\limits_{i \le m} (a_i^\top x - b_i - v_i)^2 \\ & 0 \le x \le 1, v \ge 0 \end{array}\right\}$$

- **Issue: v not bounded above**
- **Reduce from 3SAT, get ≤ 3 literals per clause**
  $\Rightarrow$ ***can consider*** $0 \le v \le 2$

cQKP is NP-hard

- CONTINUOUS QUADRATIC KNAPSACK PROBLEM (cQKP)

$$\left.\begin{array}{rrcl} \min & f(x) = x^\top Q x & + & c^\top x \\ & \sum\limits_{j \le n} a_j x_j & = & \gamma \\ & x & \in & [0,1]^n, \end{array}\right\}$$

- Reduction from SUBSET-SUM

  given list $a \in \mathbb{Q}^n$ and γ, is there $J \subseteq \{1,\ldots,n\}$ s.t. $\sum\limits_{j \in J} a_j = \gamma$?

  reduce to $f(x) = \sum_j x_j(1 - x_j)$

- σ is a YES instance of SUBSET-SUM
  - let $x^*_j = 1$ iff $j \in J$, $x^*_j = 0$ otherwise
  - feasible by construction
  - f is non-negative on $[0,1]^n$ and $f(x^*) = 0$: optimum
- σ is a NO instance of SUBSET-SUM
  - suppose $\mathsf{opt}(\text{cQKP}) = x^*$ s.t. $f(x^*) = 0$
  - then $x^* \in \{0,1\}^n$ because $f(x^*) = 0$
  - feasibility of $x^* \to \mathsf{supp}(x^*)$ solves σ, contradiction, hence $f(x^*) > 0$

QP on a simplex is NP-hard

$$\left.\begin{array}{rrcl} \min & f(x) = x^\top Q x & + & c^\top x \\ & \sum\limits_{j \le n} x_j & = & 1 \\ \forall j \le n & x_j & \ge & 0 \end{array}\right\}$$

- **Reduce MAX CLIQUE to subclass** $f(x) = -\sum\limits_{\{i,j\} \in E} x_i x_j$

  Motzkin-Straus formulation (MSF)

- **Theorem [Motzkin& Straus 1964]**

  Let C be the maximum clique of the instance $G = (V, E)$ of MAX CLIQUE

  $\exists x^* \in \mathsf{opt}\,(\mathbf{MSF}) \qquad f^* = f(x^*) = \frac{1}{2}\left(1 - \frac{1}{\omega(G)}\right)$

  $\forall j \in V \qquad x_j^* = \begin{cases} \frac{1}{\omega(G)} & \text{if } j \in C \\ 0 & \text{otherwise} \end{cases}$

Proof of the Motzkin-Straus theorem

$$x^* = \mathsf{opt}(\max_{\substack{x\in[0,1]^n \\ \sum_j x_j = 1}} \sum_{ij\in E} x_i x_j) \textbf{ s.t. } |C = \{j \in V \mid; x^*_j > 0\}| \textbf{ smallest } (\ddagger)$$

1. $\boxed{\textit{C is a clique}}$
   - Suppose $1, 2 \in C$ but $\{1, 2\} \not\in E[C]$, then $x^*_1, x^*_2 > 0$, can perturb by small $\epsilon \in [-x^*_1, x^*_2]$, get $x^\epsilon = (x^*_1 + \epsilon, x^*_2 - \epsilon, \ldots)$, feasible w.r.t. simplex and bounds
   - $\{1, 2\} \not\in E \Rightarrow x_1x_2$ does not appear in $f(x) \Rightarrow f(x^\epsilon)$ depends linearly on ϵ; by optimality of x^*, f achieves max for $\epsilon = 0$, in interior of its range $\Rightarrow f(\epsilon)$ constant
   - set $\epsilon = -x^*_1$ or $= x^*_2$ yields global optima with more zero components than x^*, against assumption ($\ddagger$), hence $\{1, 2\} \in E[C]$, by relabeling C is a clique

Proof of the Motzkin-Straus theorem

$$x^* = \mathsf{opt}(\max_{\substack{x\in[0,1]^n\\ \sum_j x_j=1}} \sum_{ij\in E} x_i x_j) \text{ s.t. } |C = \{j \in V \mid; x^*_j > 0\}| \text{ smallest } (\ddagger)$$

2. $\boxed{|C| = \omega(G)}$

- square simplex constraint $\sum_j x_j = 1$, get

$$\sum_{j\in V} x_j^2 + 2\sum_{i<j\in V} x_i x_j = 1$$

- by construction $x^*_j = 0$ for $j \notin C \Rightarrow$

$$\psi(x^*) = \sum_{j\in C}(x^*_j)^2 + 2\sum_{i<j\in C} x^*_j x^*_j = \sum_{j\in C}(x^*_j)^2 + 2f(x^*) = 1$$

- $\psi(x) = 1$ for all feasible x, so $f(x)$ achieves maximum when $\sum_{j\in C}(x^*_j)^2$ is minimum, i.e. $x^*_j = \frac{1}{|C|}$ for all $j \in C$
- again by simplex constraint

$$f(x^*) = 1 - \sum_{j\in C}(x^*_j)^2 = 1 - |C|\frac{1}{|C|^2} \le 1 - \frac{1}{\omega(G)}$$

so $f(x^*)$ attains maximum $1 - 1/\omega(G)$ when $|C| = \omega(G)$

Copositive programming

- **StQP:** $\min x^\top Qx : \sum_j x_j = 1 \wedge x \geq 0$
  NP-hard by Motzkin-Straus
- **Linearize:** $X = xx^\top$
- $A \bullet B = \mathsf{tr}(A^\top B)$
  write StQP objective as $\min Q \bullet X$
- **Let** $C = \{X \mid X = xx^\top \wedge x \geq 0\}$, $\mathcal{C} = \mathsf{conv}(C)$
- $\sum_j x_j = 1 \Leftrightarrow (\sum_j x_j)^2 = 1^2 \Leftrightarrow \mathbf{1} \bullet X = 1$
- StQP $\equiv \min Q \bullet X : \mathbf{1} \bullet X = 1 \wedge X \in \mathcal{C}$
- **Dual** $\equiv \max y : Q - y\mathbf{1} \in \mathcal{C}^*$
  $\mathcal{C}^* = \{A \mid \forall x \geq 0\ (x^\top Ax \geq 0)\}$ **(copositive cone)**
- $\Rightarrow$ **cNLP which is NP-hard!**

Two exercises

- Prove that *quartic polynomial optimization* is NP-hard; reduce from one of the combinatorial problems given during the course, and make sure that at least one monomial of degree four appears with non-zero coefficient in the MP formulation.
- As above, but for *cubic polynomial optimization*.

Portfolio optimization

You, a private investment banker, are seeing a customer. She tells you "I have 3,450,000$ I don't need in the next three years. Invest them in low-risk assets so I get at least 2.5% return per year."

Model the problem of determining the required portfolio. Missing data are part of the fun (and of real life).

[Hint: what are the decision variables, objective, constraints? What data are missing?]

Outline

Types of MP

Continuous variables:

- LP (linear functions)
- QP (quadratic obj. over affine sets)
- QCP (linear obj. over quadratically def'd sets)
- QCQP (quadr. obj. over quadr. sets)
- cNLP (convex sets, convex obj. fun.)
- SOCP (LP over 2nd ord. cone)
- SDP (LP over PSD cone)
- COP (LP over copositive cone)
- NLP (nonlinear functions)

Types of MP

Mixed-integer variables:

- IP (integer programming), MIP (mixed-integer programming)
- *extensions:* MILP, MIQ, MIQCP, MIQCQP, cMINLP, MINLP
- BLP (LP over $\{0,1\}^n$)
- BQP (QP over $\{0,1\}^n$)

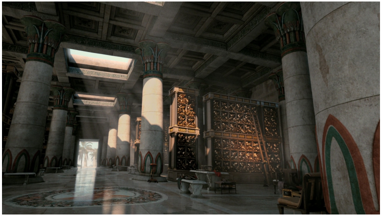
More "exotic" classes:

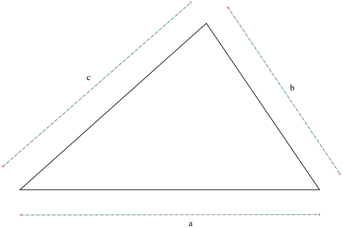
- MOP (multiple objective functions)
- BLevP (optimization constraints)
- SIP (semi-infinite programming)

Outline

A gem in Distance Geometry

- *Heron's theorem*
- Heron lived around year 0
- Hang out at Alexandria's library



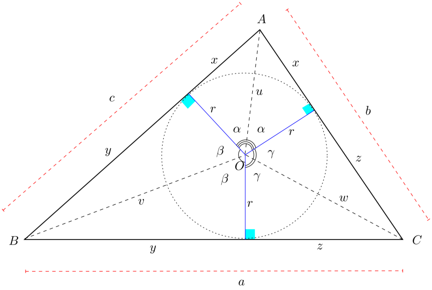


$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

- $A =$ **area of triangle**
- $s = \frac{1}{2}(a+b+c)$

Useful to measure areas of agricultural land

Heron's theorem: *Proof* [M. Edwards, high school student, 2007]



A. $2\alpha + 2\beta + 2\gamma = 2\pi \Rightarrow \alpha + \beta + \gamma = \pi$

$$\begin{aligned} r + ix &= ue^{i\alpha} \\ r + iy &= ve^{i\beta} \\ r + iz &= we^{i\gamma} \end{aligned}$$

$\Rightarrow (r+ix)(r+iy)(r+iz) = (uvw)e^{i(\alpha+\beta+\gamma)} = uvw\,e^{i\pi} = -uvw \in \mathbb{R}$

$\Rightarrow \mathrm{Im}((r+ix)(r+iy)(r+iz)) = 0$

$\Rightarrow r^2(x+y+z) = xyz \Rightarrow r = \sqrt{\frac{xyz}{x+y+z}}$

B. $s = \frac{1}{2}(a+b+c) = x + y + z$

$$\begin{aligned} s - a &= x + y + z - y - z = x \\ s - b &= x + y + z - x - z = y \\ s - c &= x + y + z - x - y = z \end{aligned}$$

$$\mathcal{A} = \frac{1}{2}(ra + rb + rc) = r\frac{a+b+c}{2} = rs = \sqrt{s(s-a)(s-b)(s-c)}$$

Subsection 1

The universal isometric embedding

Representing metric spaces in $\mathbb{R}^n$

- **Given metric space (X, d) with dist. matrix $D = (d_{ij})$,** ***embed X in a Euclidean space with same dist. matrix***
- **Consider i-th row $\delta_i = (d_{i1}, \ldots, d_{in})$ of D**
- **Embed $i \in X$ by vector $\delta_i \in \mathbb{R}^n$**
- **Define $f(X) = \{\delta_1, \ldots, \delta_n\}$, $f(d(i,j)) = \|f(i) - f(j)\|_\infty$**
- **Thm.: $(f(X), \ell_\infty)$ is a metric space with distance matrix D**
- **Practical issue:** ***embedding is high-dimensional ($\mathbb{R}^n$)***

[Kuratowski 1935]

Proof

- **Consider** $i, j \in X$ **with distance** $d(i,j) = d_{ij}$
- **Then**

$$f(d(i,j)) = \|\delta_i - \delta_j\|_\infty = \max_{k \leq n} |d_{ik} - d_{jk}| \leq \max_{k \leq n} |d_{ij}| = d_{ij}$$

ineq. $\leq$ *above from triangular inequalities in metric space:*

$d_{ik} \leq d_{ij} + d_{jk} \wedge d_{jk} \leq d_{ij} + d_{ik} \Rightarrow |d_{ik} - d_{jk}| \leq d_{ij}$

- $\max |d_{ik} - d_{jk}|$ **over** $k \leq n$ **is achieved when**

$$k \in \{i, j\} \Rightarrow f(d(i,j)) = d_{ij}$$

Subsection 2

Dimension reduction

Schoenberg's theorem

- [I. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espaces distanciés vectoriellement applicable sur l'espace de Hilbert"*, Ann. Math., 1935]
- **Question:** Given $n \times n$ symmetric matrix D, what are necessary and sufficient conditions s.t. D is a EDM[1] corresponding to n points $x_1, \ldots, x_n \in \mathbb{R}^K$ with K minimum?
- **Main theorem**:
  Thm.
  $D = (d_{ij})$ is an EDM iff $\frac{1}{2}(d_{1i}^2 + d_{1j}^2 - d_{ij}^2 \mid 2 \leq i, j \leq n)$ is PSD of rank K
- Gave rise to one of the most important results in data science: **Classic Multidimensional Scaling**

[1] *Euclidean Distance Matrix*

Gram in function of EDM

- $x = (x_1, \ldots, x_n) \subseteq \mathbb{R}^K$, written as $n \times K$ matrix
- matrix $G = xx^\top = (x_i \cdot x_j)$ is the *Gram matrix* of x
  Lemma: $G \succeq 0$ *and each* $M \succeq 0$ *is a Gram matrix of some* x
- A variant of Schoenberg's theorem
  Relation between EDMs and Gram matrices:

$$G = -\frac{1}{2} J D^2 J \qquad (\S)$$

- where $D^2 = (d_{ij}^2)$ and

$$J = I_n - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top = \begin{pmatrix} 1 - \frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & 1 - \frac{1}{n} & \cdots & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \cdots & 1 - \frac{1}{n} \end{pmatrix}$$

Multidimensional scaling (MDS)

- **Often get approximate EDMs $\tilde{D}$ from raw data**
  (dissimilarities, discrepancies, differences)
- $\tilde{G} = -\frac{1}{2} J \tilde{D}^2 J$ **is an approximate Gram matrix**
- Approximate Gram $\Rightarrow$ spectral decomposition $P \tilde{\Lambda} P^\top$ has $\tilde{\Lambda} \not\geq 0$
- **Let Λ closest PSD diagonal matrix to $\tilde{\Lambda}$:**
  zero the negative components of $\tilde{\Lambda}$
- $x = P\sqrt{\Lambda}$ **is an "approximate realization" of $\tilde{D}$**

Classic MDS: Main result

1. Prove lemma: matrix is Gram iff it is PSD
2. Prove Schoenberg's theorem: $G = -\frac{1}{2} J D^2 J$

Proof of lemma

- *Gram $\subseteq$ PSD*
  - x is an $n \times K$ real matrix
  - $G = xx^\top$ its Gram matrix
  - For each $y \in \mathbb{R}^n$ we have

    $$yGy^\top = y(xx^\top)y^\top = (yx)(x^\top y^\top) = (yx)(yx)^\top = \|yx\|_2^2 \geq 0$$

  - $\Rightarrow G \succeq 0$
- *PSD $\subseteq$ Gram*
  - Let $G \succeq 0$ be $n \times n$
  - *Spectral decomposition:* $G = P\Lambda P^\top$

    (P orthogonal, $\Lambda \geq 0$ diagonal)
  - $\Lambda \geq 0 \Rightarrow \sqrt{\Lambda}$ exists
  - $G = P\Lambda P^\top = (P\sqrt{\Lambda})(\sqrt{\Lambda}^\top P^\top) = (P\sqrt{\Lambda})(P\sqrt{\Lambda})^\top$
  - Let $x = P\sqrt{\Lambda}$, then G is the Gram matrix of x

Schoenberg's theorem proof (1/2)

- Assume zero centroid WLOG (can translate x as needed)
- Expand: $d_{ij}^2 = \|x_i - x_j\|_2^2 = (x_i - x_j)(x_i - x_j) = x_i x_i + x_j x_j - 2 x_i x_j \qquad (*)$
- Aim at "inverting" $(*)$ to express $x_i x_j$ in function of d_{ij}^2
- Sum $(*)$ over i: $\sum_i d_{ij}^2 = \sum_i x_i x_i + n x_j x_j - 2 x_j \sum_i x_i$ 0 by zero centroid
- Similarly for j and divide by n, get:

$$
\begin{aligned}
\frac{1}{n} \sum_{i \le n} d_{ij}^2 &= \frac{1}{n} \sum_{i \le n} x_i x_i + x_j x_j \quad (\dagger) \\
\frac{1}{n} \sum_{j \le n} d_{ij}^2 &= x_i x_i + \frac{1}{n} \sum_{j \le n} x_j x_j \quad (\ddagger)
\end{aligned}
$$

- Sum $(\dagger)$ over j, get:

$$\frac{1}{n} \sum_{i,j} d_{ij}^2 = n \frac{1}{n} \sum_i x_i x_i + \sum_j x_j x_j = 2 \sum_i x_i x_i$$

- Divide by n, get:

$$\frac{1}{n^2} \sum_{i,j} d_{ij}^2 = \frac{2}{n} \sum_i x_i x_i \quad (**)$$

Schoenberg's theorem proof (2/2)

- Rearrange (∗), (†), (‡) as follows:

$$
\begin{aligned}
2x_i x_j &= x_i x_i + x_j x_j - d_{ij}^2 &\quad (4)\\
x_i x_i &= \frac{1}{n}\sum_j d_{ij}^2 - \frac{1}{n}\sum_j x_j x_j &\quad (5)\\
x_j x_j &= \frac{1}{n}\sum_i d_{ij}^2 - \frac{1}{n}\sum_i x_i x_i &\quad (6)
\end{aligned}
$$

- Replace LHS of Eq. (5)-(6) in RHS of Eq. (4), get

$$2x_i x_j = \frac{1}{n}\sum_k d_{ik}^2 + \frac{1}{n}\sum_k d_{kj}^2 - d_{ij}^2 - \frac{2}{n}\sum_k x_k x_k$$

- By (∗∗) replace $\frac{2}{n}\sum_i x_i x_i$ with $\frac{1}{n^2}\sum_{i,j} d_{ij}^2$, get

$$2x_i x_j = \frac{1}{n}\sum_k (d_{ik}^2 + d_{kj}^2) - d_{ij}^2 - \frac{1}{n^2}\sum_{h,k} d_{hk}^2 \quad (\S)$$

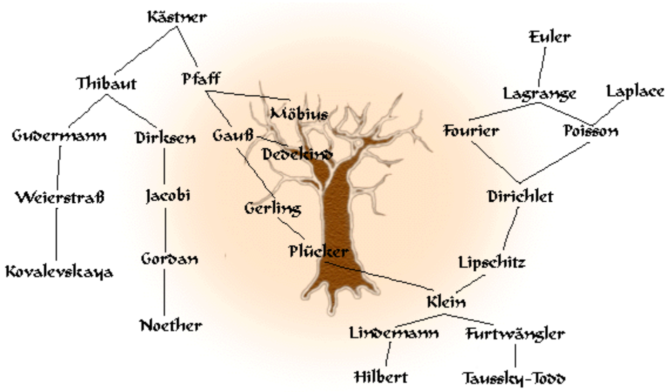
which expresses $x_i x_j$ in function of D

Principal Component Analysis (PCA)

- Given an approximate distance matrix D
- find $x = \mathbf{MDS}(D)$
- However, you want $x = P\sqrt{\Lambda}$ in K dimensions
  but $\mathsf{rank}(\Lambda) > K$
- Only keep K largest components of Λ
  zero the rest
- Get realization in desired space

Example 1/3

Mathematical genealogy skeleton



Example 2/3

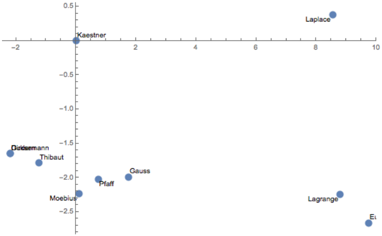
A partial view

	Euler	Thibaut	Pfaff	Lagrange	Laplace	Möbius	Gudermann	Dirksen	Gauss
Kästner	10	1	1	9	8	2	2	2	2
Euler		11	9	1	3	10	12	12	8
Thibaut			2	10	10	3	1	1	3
Pfaff				8	8	1	3	3	1
Lagrange					2	9	11	11	7
Laplace						9	11	11	7
Möbius							4	4	2
Gudermann								2	4
Dirksen									4

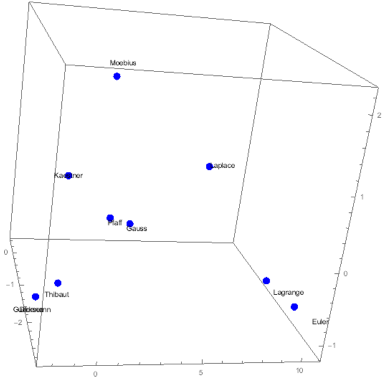
$$D = \begin{pmatrix} 0 & 10 & 1 & 1 & 9 & 8 & 2 & 2 & 2 & 2 \\ 10 & 0 & 11 & 9 & 1 & 3 & 10 & 12 & 12 & 8 \\ 1 & 11 & 0 & 2 & 10 & 10 & 3 & 1 & 1 & 3 \\ 1 & 9 & 2 & 0 & 8 & 8 & 1 & 3 & 3 & 1 \\ 9 & 1 & 10 & 8 & 0 & 2 & 9 & 11 & 11 & 7 \\ 8 & 3 & 10 & 8 & 2 & 0 & 9 & 11 & 11 & 7 \\ 2 & 10 & 3 & 1 & 9 & 9 & 0 & 4 & 4 & 2 \\ 2 & 12 & 1 & 3 & 11 & 11 & 4 & 0 & 2 & 4 \\ 2 & 12 & 1 & 3 & 11 & 11 & 4 & 2 & 0 & 4 \\ 2 & 8 & 3 & 1 & 7 & 7 & 2 & 4 & 4 & 0 \end{pmatrix}$$

Example 3/3

In 2D



In 3D



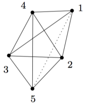
Subsection 3

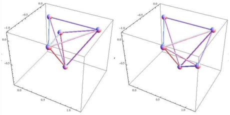
Distance geometry problem

The Distance Geometry Problem (DGP)

Given $K \in \mathbb{N}$ ***and*** $G = (V, E, d)$ ***with*** $d : E \to \mathbb{R}_+$***,***
find $x : V \to \mathbb{R}^K$ ***s.t.***

$$\forall \{i, j\} \in E \quad \|x_i - x_j\|_2^2 = d_{ij}^2$$

Given a weighted graph 

, draw it so edges are drawn as segments with lengths = weights 

Some applications

- **clock synchronization** ($K = 1$)
- **sensor network localization** ($K = 2$)
- **molecular structure from distance data** ($K = 3$)
- **autonomous underwater vehicles** ($K = 3$)
- **distance matrix completion** (whatever K)

Clock synchronization

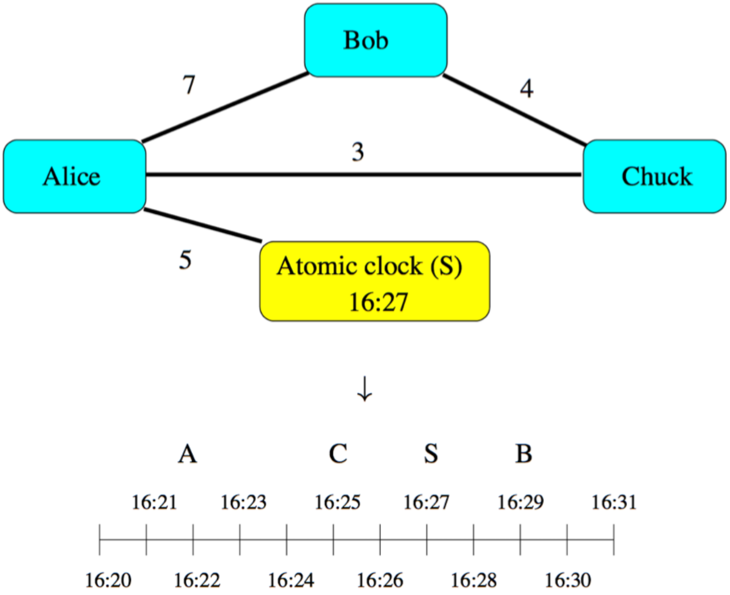
From [Singer, *Appl. Comput. Harmon. Anal.* 2011]

Determine a set of unknown timestamps from partial measurements of their time differences

- $K = 1$
- V: timestamps
- $\{u, v\} \in E$ if known time difference between u, v
- d: values of the time differences

Used in time synchronization of distributed networks

Clock synchronization



Sensor network localization

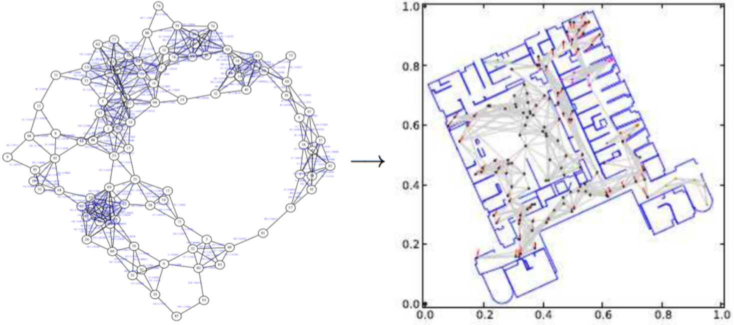
From [Yemini, *Proc. CDSN*, 1978]

The positioning problem arises when it is necessary to locate a set of geographically distributed objects using measurements of the distances between some object pairs

- $K = 2$
- V: (mobile) sensors
- $\{u, v\} \in E$ iff distance between u, v is measured
- d: distance values

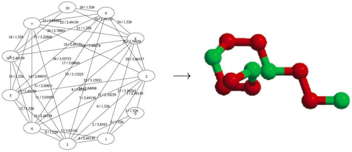
Used whenever GPS not viable (e.g. underwater)
$d_{uv} \propto$ battery consumption in P2P communication betw. u, v

Sensor network localization



Molecular structure from distance data

From [Liberti et al., *SIAM Rev.*, 2014]



- $K = 3$
- V: atoms
- $\{u, v\} \in E$ iff distance between u, v is known
- d: distance values

Used whenever X-ray crystallography does not apply (e.g. liquid)
Covalent bond lengths and angles known precisely
Distances $\lessapprox 5.5$ measured approximately by NMR

Complexity

- **DGP$_1$ with $d : E \to \mathbb{Q}_+$ is in NP**
  - if instance YES $\exists$ realization $x \in \mathbb{R}^{n\times 1}$
  - if some component $x_i \notin \mathbb{Q}$ translate x so $x_i \in \mathbb{Q}$
  - consider some other x_j
  - let $\ell = |\text{sh. path } p : i \to j| = \sum_{\{u,v\}\in p} d_{uv} \in \mathbb{Q}$
  - then $x_j = x_i \pm \ell \to x_j \in \mathbb{Q}$
  - $\Rightarrow$ verification of

$$\forall \{i,j\} \in E \quad |x_i - x_j| = d_{ij}$$

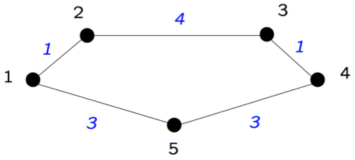
    in polytime
- **DGP$_K$ may not be in NP for $K > 1$**
  don't know how to verify $\|x_i - x_j\|_2 = d_{ij}$ for $x \notin \mathbb{Q}^{nK}$

Hardness

Partition *is* NP*-hard*

Given $a = (a_1, \ldots, a_n) \in \mathbb{N}^n$, $\exists\, I \subseteq \{1, \ldots, n\}$ **s.t.** $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$ **?**

- Reduce Partition to DGP$_1$
- $a \longrightarrow$ **cycle** C
  $V(C) = \{1, \ldots, n\}, E(C) = \{\{1, 2\}, \ldots, \{n, 1\}\}$
- **For** $i < n$ **let** $d_{i,i+1} = a_i$
  $d_{n,n+1} = d_{n1} = a_n$
- *E.g. for* $a = (1, 4, 1, 3, 3)$*, get cycle graph:*



[Saxe, 1979]

PARTITION is YES $\Rightarrow$ DGP$_1$ is YES

- **Given:** $I \subset \{1, \ldots, n\}$ **s.t.** $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$
- **Construct: realization** x **of** C **in** $\mathbb{R}$
  1. $x_1 = 0$ `// start`
  2. **induction step: suppose** x_i **known**
     if $i \in I$
         let $x_{i+1} = x_i + d_{i,i+1}$ `// go right`
     else
         let $x_{i+1} = x_i - d_{i,i+1}$ `// go left`
- **Correctness proof: by the same induction**
  but careful when $i = n$***: have to show*** $x_{n+1} = x_1$

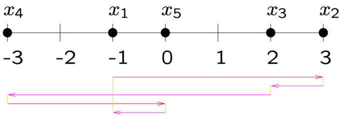
Partition is YES ⇒ DGP$_1$ is YES

$$\begin{aligned}
(1) = \sum_{i\in I}(x_{i+1}-x_i) &= \sum_{i\in I} d_{i,i+1} = \\
= \sum_{i\in I} a_i &= \sum_{i\notin I} a_i = \\
= \sum_{i\notin I} d_{i,i+1} &= \sum_{i\notin I}(x_i - x_{i+1}) = (2)
\end{aligned}$$

$$\begin{aligned}
(1) = (2) \Rightarrow \sum_{i\in I}(x_{i+1}-x_i) = \sum_{i\notin I}(x_i - x_{i+1}) \Rightarrow \sum_{i\le n}(x_{i+1}-x_i) &= 0 \\
\Rightarrow (x_{n+1}-x_n) + (x_n - x_{n-1}) + \cdots + (x_3 - x_2) + (x_2 - x_1) &= 0 \\
\Rightarrow x_{n+1} &= x_1
\end{aligned}$$

PARTITION is NO $\Rightarrow$ DGP$_1$ is NO

- By contradiction: suppose DGP$_1$ is YES, x realization of C
- $F = \{\{u,v\} \in E(C) \mid x_u \le x_v\}$,
  $E(C) \smallsetminus F = \{\{u,v\} \in E(C) \mid x_u > x_v\}$
- Trace $x_1, \ldots, x_n$: follow edges in F ($\rightarrow$) and in $E(C) \smallsetminus F$ ($\leftarrow$)



$$\boxed{\begin{aligned} \sum_{\{u,v\}\in F} (x_v - x_u) &= \sum_{\{u,v\}\not\in F} (x_u - x_v) \\ \sum_{\{u,v\}\in F} |x_u - x_v| &= \sum_{\{u,v\}\not\in F} |x_u - x_v| \\ \sum_{\{u,v\}\in F} d_{uv} &= \sum_{\{u,v\}\not\in F} d_{uv} \end{aligned}}$$

- Let $J = \{i < n \mid \{i, i+1\} \in F\} \cup \{n \mid \{n, 1\} \in F\}$

$$\Rightarrow \quad \sum_{i \in J} a_i = \sum_{i \not\in J} a_i$$

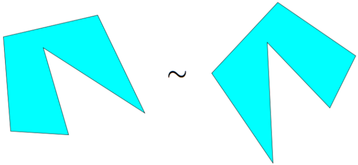
- So J solves Partition instance, contradiction
- $\Rightarrow$ DGP is NP-hard, DGP$_1$ is NP-complete

Number of solutions

- (G, K): DGP instance
- $\tilde{X} \subseteq \mathbb{R}^{Kn}$: set of solutions
- *Congruence*: composition of translations, rotations, reflections
- $C =$ set of congruences in $\mathbb{R}^K$
- $x \sim y$ means $\exists \rho \in C \ (y = \rho x)$:
  distances in x are preserved in y through ρ
- $\Rightarrow$ if $|\tilde{X}| > 0$, $|\tilde{X}| = 2^{\aleph_0}$

Number of solutions modulo congruences

- Congruence is an *equivalence relation* $\sim$ on $\tilde{X}$ (reflexive, symmetric, transitive)



- Partitions $\tilde{X}$ into *equivalence classes*
- $X = \tilde{X}/\sim$: sets of representatives of equivalence classes
- **Focus on $|X|$ rather than $|\tilde{X}|$**

Rigidity, flexibility and $|X|$

- **infeasible** $\Leftrightarrow |X| = 0$
- **rigid graph** $\Leftrightarrow |X| < \aleph_0$
- **globally rigid graph** $\Leftrightarrow |X| = 1$
- **flexible graph** $\Leftrightarrow |X| = 2^{\aleph_0}$
- $|X| = \aleph_0$**: impossible by Milnor's theorem**

Milnor's theorem implies $|X| \neq \aleph_0$

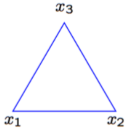
- System S of polynomial equations of degree 2

$$\forall i \leq m \quad p_i(x_1, \ldots, x_{nK}) = 0$$

- Let X be the set of $x \in \mathbb{R}^{nK}$ satisfying S
- **Number of connected components of X is $O(3^{nK})$** [Milnor 1964]
- Assume $|X|$ is countable; then G cannot be flexible
  $\Rightarrow$ *each incongruent rlz is in a separate component*
  $\Rightarrow$ by Milnor's theorem, there's finitely many of them

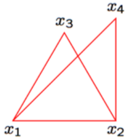
Examples

$V^1 = \{1,2,3\}$
$E^1 = \{\{u,v\} \mid u < v\}$
$d^1 = 1$



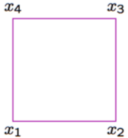
ρ congruence in $\mathbb{R}^2$
$\Rightarrow \rho x$ valid realization
$|X| = 1$

$V^2 = V^1 \cup \{4\}$
$E^2 = E^1 \cup \{\{1,4\},\{2,4\}\}$
$d^2 = 1 \wedge d_{14} = \sqrt{2}$



ρ reflects x_4 wrt $\overline{x_1, x_2}$
$\Rightarrow \rho x$ valid realization
$|X| = 2$ (,)

$V^3 = V^2$
$E^3 = \{\{u,u+1\} | u \leq 3\} \cup \{1,4\}$
$d^1 = 1$



ρ rotates $\overline{x_2 x_3}$, $\overline{x_1 x_4}$ by θ
$\Rightarrow \rho x$ valid realization
$|X|$ is uncountable
(, , , , ...)

Subsection 4

Distance geometry in MP

DGP formulations and methods

- System of equations
- Unconstrained global optimization (GO)
- Constrained global optimization
- SDP relaxations and their properties
- Diagonal dominance
- Concentration of measure in SDP
- Isomap for DGP

System of quadratic equations

$$\forall \{u,v\} \in E \quad \|x_u - x_v\|^2 = d_{uv}^2 \tag{7}$$

Computationally: useless
reformulate using slacks:

$$\min_{x,s} \Big\{ \sum_{\{u,v\}\in E} s_{uv}^2 \;\Big|\; \forall \{u,v\} \in E \quad \|x_u - x_v\|^2 = d_{uv}^2 + s_{uv} \Big\} \tag{8}$$

Unconstrained Global Optimization

$$\min_x \sum_{\{u,v\}\in E} (\|x_u - x_v\|^2 - d_{uv}^2)^2 \qquad \textbf{(9)}$$

$\boxed{\textbf{Globally optimal obj. fun. value of (9) is 0 iff } x \textbf{ solves (7)}}$

Computational experiments in [Liberti et al., 2006]:

- **GO solvers from 10 years ago**
- **randomly generated protein data:** ≤ 50 **atoms**
- **cubic crystallographic grids:** ≤ 64 **atoms**

Constrained global optimization

- $\min_x \sum\limits_{\{u,v\}\in E} |\|x_u - x_v\|^2 - d_{uv}^2|$ exactly reformulates (7)
- Relax objective f to concave part, remove constant term, rewrite $\min -f$ as $\max f$
- Reformulate convex part of obj. fun. to convex constraints
- *Exact reformulation*

$$\left.\begin{array}{rl} \max_x & \sum\limits_{\{u,v\}\in E} \|x_u - x_v\|^2 \\ \forall \{u,v\} \in E & \|x_u - x_v\|^2 \le d_{uv}^2 \end{array}\right\} \qquad (10)$$

Theorem (Activity)

At a glob. opt. x^* of a YES instance, all constraints of (10) are active

Linearization

$$\Rightarrow \quad \forall\{i,j\} \in E \quad \|x_i\|_2^2 + \|x_j\|_2^2 - 2x_i \cdot x_j = d_{ij}^2$$

$$\Rightarrow \left\{ \begin{array}{rcl} \forall\{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} & = & d_{ij}^2 \\ X & = & x\,x^\top \end{array} \right.$$

Relaxation

$$\begin{array}{rrcl} & X & = & x\,x^\top \\ \Rightarrow & X - x\,x^\top & = & 0 \end{array}$$

$$\begin{array}{rrcl} \text{(relax)} & \Rightarrow \quad X - x\,x^\top & \succeq & 0 \end{array}$$

$$\mathsf{Schur}(X,x) = \begin{pmatrix} I_K & x^\top \\ x & X \end{pmatrix} \succeq 0$$

If x does not appear elsewhere $\Rightarrow$ get rid of it (e.g. choose $x = 0$):

replace $\mathsf{Schur}(X,x) \succeq 0$ ***by*** $X \succeq 0$

SDP relaxation

$$\begin{array}{rrcl} & \min F \bullet X & & \\ \forall \{i,j\} \in E & X_{ii} + X_{jj} - 2X_{ij} & = & d_{ij}^2 \\ & X & \succeq & 0 \end{array}$$

How do we choose F?

$F \bullet X = \mathsf{Tr}(F^\top X)$

Some possible objective functions

- **For protein conformation:**

$$\min \sum_{\{i,j\}\in E} (X_{ii} + X_{jj} - 2X_{ij})$$

with $=$ changed to $\geq$ in constraints *(or* $\max$ *and* $\leq$*)*
"push-and-pull" the realization

- [Ye, 2003], application to wireless sensors localization

$$\min \mathsf{Tr}(X)$$

$\mathsf{Tr}(X) = \mathsf{Tr}(P^{-1}\Lambda P) = \mathsf{Tr}(P^{-1}P\Lambda) = \mathsf{Tr}(\Lambda) = \sum_i \lambda_i$
$\Rightarrow$ *hope to minimize rank*

- How about "just random"?

How do you choose?

for want of some better criterion...

TEST!

- Download protein files from Protein Data Bank (PDB)

  they contain atom realizations
- Mimick a Nuclear Magnetic Resonance experiment

  Keep only pairwise distances < 5.5
- Try and reconstruct the protein shape from those weighted graphs
- Quality evaluation of results:
  - $\textbf{LDE}(x) = \max\limits_{\{i,j\}\in E} |\, \|x_i - x_j\| - d_{ij} \,|$
  - $\textbf{MDE}(x) = \frac{1}{|E|} \sum\limits_{\{i,j\}\in E} |\, \|x_i - x_j\| - d_{ij} \,|$

Objective function tests

SDP solved with Mosek

$SDP + PCA$

Instance			LDE			MDE			CPU		
Name	$\|V\|$	$\|E\|$	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*
`C0700odd.1`	15	39	**3.31**	4.57	4.44	**1.92**	2.52	2.50	0.13	**0.07**	0.08
`C0700odd.C`	36	242	10.61	**4.85**	**4.85**	3.02	3.02	3.02	0.69	**0.43**	0.44
`C0700.odd.G`	36	308	**4.57**	4.77	4.77	**2.41**	2.84	2.84	0.86	**0.54**	**0.54**
`C0150alter.1`	37	335	**4.66**	4.88	4.86	**2.52**	3.00	3.00	0.97	0.59	**0.58**
`C0080create.1`	60	681	7.17	**4.86**	**4.86**	**3.08**	3.19	3.19	2.48	**1.46**	**1.46**
`tiny`	37	335	**4.66**	4.88	4.88	**2.52**	3.00	3.00	0.97	**0.60**	**0.60**
`1guu-1`	150	959	10.20	**4.93**	**4.93**	3.43	3.43	3.43	9.23	**5.68**	5.70

$SDP + PCA + NLP$

Instance			LDE			MDE			CPU		
Name	$\|V\|$	$\|E\|$	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*
`1b03`	89	456	0.00	0.00	0.00	0.00	0.00	0.00	8.69	**6.28**	9.91
`1crn`	138	846	0.81	0.81	0.81	0.07	0.07	0.07	33.33	**31.32**	44.48
`1guu-1`	150	959	0.97	4.93	**0.92**	0.10	3.43	**0.08**	56.45	**7.89**	65.33

Choice

- *Ye* very fast but often imprecise
- *Random* good but nondeterministic
- ***Push-and-Pull***: can relax $X_{ii} + X_{jj} - 2X_{ij} = d_{ij}^2$ to $X_{ii} + X_{jj} - 2X_{ij} \geq d_{ij}^2$
  easier to satisfy feasibility, useful later on
- **Heuristic:** add $+\eta \mathsf{Tr}(X)$ to objective, with $\eta \ll 1$
  might help minimize solution rank
- $\min \sum_{\{i,j\} \in E} (X_{ii} + X_{jj} - 2X_{ij}) + \eta \mathsf{Tr}(X)$

When SDP solvers hit their size limit

- **SDP solver: technological bottleneck**
- How can we best use an LP solver?
- Diagonally Dominant (DD) matrices are PSD
- Not *vice versa*: inner approximate PSD cone $Y \succeq 0$
- *Idea by A.A. Ahmadi [Ahmadi & Hall 2015]*

You won't see this in TD, Octave+YALMIP is very slow, interface bottleneck

Diagonally dominant matrices

$n \times n$ **matrix** X **is DD if**

$$\forall i \le n \quad X_{ii} \ge \sum_{j \ne i} |X_{ij}|.$$

E.g. $\begin{pmatrix} 1 & 0.1 & -0.2 & 0 & 0.04 & 0 \\ 0.1 & 1 & -0.05 & 0.1 & 0 & 0 \\ -0.2 & -0.05 & 1 & 0.1 & 0.01 & 0 \\ 0 & 0.1 & 0.1 & 1 & 0.2 & 0.3 \\ 0.04 & 0 & 0.01 & 0.2 & 1 & -0.3 \\ 0 & 0 & 0 & 0.3 & -0.3 & 1 \end{pmatrix}$

DD Linearization

$$\forall i \le n \quad X_{ii} \ge \sum_{j \ne i} |X_{ij}| \qquad (*)$$

- introduce "sandwiching" variable T
- write $|X|$ as T
- add constraints $-T \le X \le T$
- by $\ge$ constraint sense, write $(*)$ as

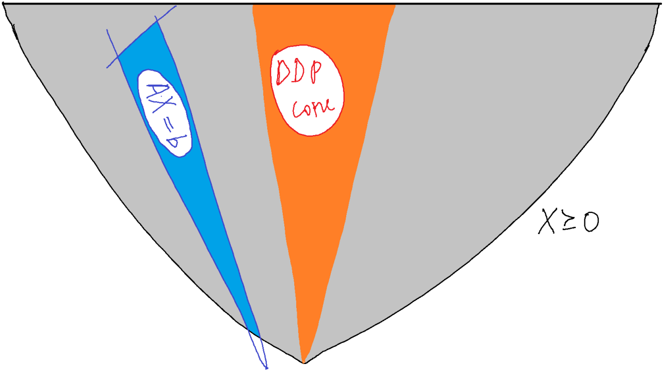
$$X_{ii} \ge \sum_{j \ne i} T_{ij}$$

DD Programming (DDP)

$$\left.\begin{array}{rcl} \forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} & = & d_{ij}^2 \\ X & \textbf{is} & \textbf{DD} \end{array}\right\}$$

$$\Rightarrow \left\{\begin{array}{lrcl} \forall \{i,j\} \in E & X_{ii} + X_{jj} - 2X_{ij} & = & d_{ij}^2 \\ \quad \forall i \le n & \sum\limits_{\substack{j \le n \\ j \ne i}} T_{ij} & \le & X_{ii} \\ & -T \le X & \le & T \end{array}\right.$$

The issue with inner approximations



DDP could be infeasible!

Exploit push-and-pull

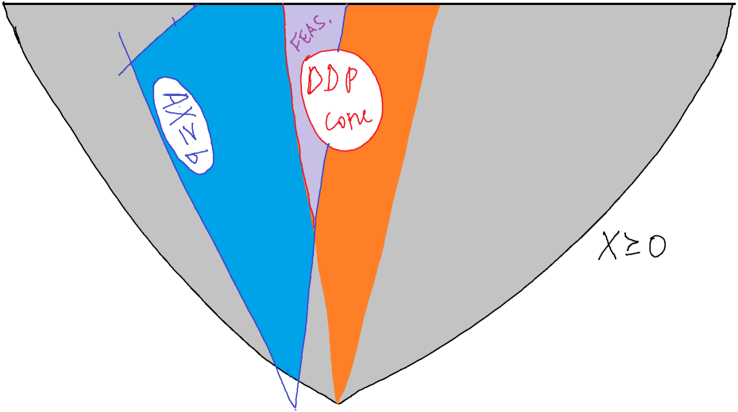
- Enlarge the feasible region
- From

$$\forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} = d_{ij}^2$$

- Use "push" objective $\min \sum_{ij \in E} X_{ii} + X_{jj} - 2X_{ij}$
- Relax to

$$\forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} \geq d_{ij}^2$$

Hope to achieve LP feasibility



DDP formulation for the DGP

$$\left.\begin{array}{rrcl}
\min & \sum\limits_{\{i,j\}\in E}(X_{ii}+X_{jj}-2X_{ij}) & & \\
\forall\{i,j\}\in E & X_{ii}+X_{jj}-2X_{ij} & \ge & d_{ij}^2 \\
\forall i\le n & \sum\limits_{\substack{j\le n\\ j\ne i}} T_{ij} & \le & X_{ii} \\
 & -T\le X & \le & T \\
 & T & \ge & 0
\end{array}\right\}$$

SDP vs. DDP: tests

Using "push-and-pull" objective in SDP
SDP solved with Mosek, DDP with CPLEX

SDP + PCA

	SDP			DDP		
Instance	*LDE*	*MDE*	*CPU modl/soln*	*LDE*	*MDE*	*CPU modl/soln*
`C0700odd.1`	0.79	0.34	0.06/**0.12**	**0.38**	**0.30**	0.15/0.15
`C0700.odd.G`	2.38	0.89	0.57/1.16	**1.86**	**0.58**	1.11/**0.95**
`C0150alter.1`	**1.48**	**0.45**	0.73/1.33	1.54	0.55	1.23/**1.04**
`C0080create.1`	2.49	0.82	1.63/7.86	**0.98**	**0.67**	3.39/**4.07**
`1guu-1`	**0.50**	**0.15**	6.67/684.89	1.00	0.85	37.74/**153.17**

Subsection 5

DGP cones

Cones

- Set C is a *cone* if:

$$\forall A, B \in C,\ \alpha, \beta \geq 0 \quad \alpha A + \beta B \in C$$

- If C is a cone, the *dual cone* is

$$C^* = \{y \mid \forall x \in C\ \langle x, y \rangle \geq 0\}$$

- If $C \subset \mathbb{S}_n$ (set $n \times n$ symmetric matrices)

$$C^* = \{Y \mid \forall X \in C\ (Y \bullet X \geq 0)\}$$

- A $n \times n$ matrix cone C is *finitely generated* by $\mathcal{X} \subset \mathbb{R}^n$ if

$$\forall X \in C\ \exists \delta \in \mathbb{R}_+^{|\mathcal{X}|} \quad X = \sum_{x \in \mathcal{X}} \delta_x x x^\top$$

- $\mathbb{PSD}$, $\mathbb{DD}$ *are cones (prove it)*

Representations of $\mathbb{DD}$

- **Consider $E_{ii}, E_{ij}^{+}, E_{ij}^{-}$ in $\mathbb{S}_n$**
  Define $\mathcal{E}_0 = \{E_{ii} \mid i \le n\}, \mathcal{E}_1 = \{E_{ij}^{\pm} \mid i < j\}, \mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_1$
- $E_{ii} = \mathsf{diag}(0, \ldots, 0, 1_i, 0, \ldots, 0)$
- E_{ij}^{+} has minor $\begin{pmatrix} 1_{ii} & 1_{ij} \\ 1_{ji} & 1_{jj} \end{pmatrix}$, 0 elsewhere
- E_{ij}^{-} has minor $\begin{pmatrix} 1_{ii} & -1_{ij} \\ -1_{ji} & 1_{jj} \end{pmatrix}$, 0 elsewhere
- **Thm. $\mathbb{DD}$ = cone generated by $\mathcal{E}$** [Barker & Carlson 1975]
  Pf. *Rays in $\mathcal{E}$ are extreme, all DD matrices generated by $\mathcal{E}$*
- **Cor. $\mathbb{DD}$ finitely gen. by**
  $\mathcal{X}_{\mathbb{DD}} = \{e_i \mid i \le n\} \cup \{e_i \pm e_j \mid j < \ell \le n\}$
  Pf. *Write* $E_{ii} = e_i e_i^{\top}, E_{ij}^{\pm} = (e_i \pm e_j)(e_i \pm e_j)^{\top}$, *where* e_i *is the* i*-th std basis element of* $\mathbb{R}^n$

Finitely generated dual cone theorem

Thm. If C finitely gen. by $\mathcal{X}$, then

$$C^* = \{Y \mid \forall x \in \mathcal{X}\ (Y \bullet xx^\top \geq 0)\}$$

- ($\Rightarrow$) Let Y s.t. $\forall x \in \mathcal{X}\ (Y \bullet xx^\top \geq 0)$
  - $\forall X \in C$, $X = \sum_{x \in \mathcal{X}} \delta_x xx^\top$ (by fin. gen.)
  - hence $Y \bullet X = \sum_x \delta_x Y \bullet xx^\top \geq 0$ (by hyp.)
  - whence $Y \in C^*$
- ($\Leftarrow$) Suppose $Z \in C^* \smallsetminus \{Y \mid \forall x \in \mathcal{X}\ (Y \bullet xx^\top \geq 0)\}$
  - then $\exists \mathcal{X}' \subset \mathcal{X}$ s.t. $\forall x \in \mathcal{X}'\ (Z \bullet xx^\top < 0)$ (by hyp.)
  - consider any $Y = \sum_{x \in \mathcal{X}'} \delta_x xx^\top \in C$ with $\delta \geq 0$
  - then $Z \bullet Y = \sum_{x \in \mathcal{X}'} \delta_x Z \bullet xx^\top < 0$ so $Z \notin C^*$
  - contradiction $\Rightarrow C^* = \{Y \mid \forall x \in \mathcal{X}\ (Y \bullet xx^\top \geq 0)\}$

Dual cone constraints

- **Remark:** $X \bullet vv^\top = v^\top X v$
- Use finitely generated dual cone theorem
- Decision variable matrix X
- **Constraints:**

$$\forall v \in \mathcal{X} \quad v^\top X v \geq 0$$

- If $|\mathcal{X}|$ polysized, get compact formulation

  otherwise use column generation
- $|\mathcal{X}_{\mathbb{DD}}| = |\mathcal{E}| = O(n^2)$

Dual cone DDP formulation for DGP

$$\left.\begin{array}{rrcl} \min & \sum\limits_{\{i,j\}\in E}(X_{ii}+X_{jj}-2X_{ij}) & & \\ \forall\{i,j\}\in E & X_{ii}+X_{jj}-2X_{ij} & = & d_{ij}^2 \\ \forall v\in\mathcal{X}_{\mathbb{DD}} & v^\top X v & \geq & 0 \end{array}\right\}$$

- $v^\top X v \geq 0$ **for** $v \in \mathcal{X}_{\mathbb{DD}}$ **equivalent to:**

$$\begin{array}{rrcl} \forall i\le n & X_{ii} & \geq & 0 \\ \forall\{i,j\}\notin E & X_{ii}+X_{jj}-2X_{ij} & \geq & 0 \\ \forall i<j & X_{ii}+X_{jj}+2X_{ij} & \geq & 0 \end{array}$$

Properties

- SDP relaxation of original problem
- **Thm.** *Dual cone DDP is a relaxation of SDP*
  Pf. *If* $X \succeq 0$*, then* $\forall v \in \mathbb{R}^n \; v^\top X v \geq 0$ *by defn., and* $\mathcal{X}_{\mathbb{DD}} \subset \mathbb{R}^n$
- Yields extremely tight obj fun bounds
- Solutions have large negative rank, unfortunately
  retrieving feasible solutions is difficult

Subsection 6

Barvinok's Naive Algorithm

Concentration of measure

From [Barvinok, 1997]

> *The value of a "well behaved" function at a random point of a "big" probability space X is "very close" to the mean value of the function.*

and

> *In a sense, measure concentration can be considered as an extension of the law of large numbers.*

Concentration of measure

Given Lipschitz function $f : X \to \mathbb{R}$ s.t.

$$\forall x, y \in X \quad |f(x) - f(y)| \leq L\|x - y\|_2$$

for some $L \geq 0$, there is *concentration of measure* **if $\exists$ constants c, C s.t.**

$$\forall \varepsilon > 0 \quad \mathsf{P}_x(|f(x) - \mathsf{E}(f)| > \varepsilon) \leq c\, e^{-C\varepsilon^2/L^2}$$

$\equiv$ *"discrepancy from mean is unlikely"*

Barvinok's theorem

Consider:

- for each $k \leq m$, manifolds $\mathcal{X}_k = \{x \in \mathbb{R}^n \mid x^\top Q^k x = a_k\}$
- a feasibility problem $x \in \bigcap\limits_{k \leq m} \mathcal{X}_k$
- its SDP relaxation $\forall x \leq m \ (Q^k \bullet X = a_k)$ with soln. $\bar{X}$

Let $\boxed{T = \mathsf{factor}(\bar{X})}$, $\boxed{y \sim \mathcal{N}^n(0,1) \text{ and } x' = Ty}$

Then $\exists c$ **and** $n_0 \in \mathbb{N}$ **s.t. if** $n \geq n_0$,

$$\mathsf{Prob}\left(\forall k \leq m \ \mathsf{dist}(x', \mathcal{X}_k) \leq c\sqrt{\|\bar{X}\|_2 \ln n}\right) \geq 0.9.$$

IDEA: since x' is "close" to each $\mathcal{X}_k$, try local descent!

Application to the DGP

- $\forall \{i,j\} \in E \quad \mathcal{X}_{ij} = \{x \mid \|x_i - x_j\|_2^2 = d_{ij}^2\}$
- **DGP can be written as** $\bigcap\limits_{\{i,j\} \in E} \mathcal{X}_{ij}$
- **SDP relaxation** $X_{ii} + X_{jj} - 2X_{ij} = d_{ij}^2 \wedge X \succeq 0$ **with soln.** $\bar{X}$

- **Difference with Barvinok:** $x \in \mathbb{R}^{Kn}$, $\mathbf{rk}(\bar{X}) \leq K$
- **IDEA: sample** $y \sim \mathcal{N}^{nK}(0, \frac{1}{\sqrt{K}})$
- **Thm. Barvinok's theorem works in rank** K

The heuristic

1. **Solve SDP relaxation of DGP, get soln. $\bar{X}$**
   use DDP+LP if SDP+IPM too slow

2. **a.** $T = \mathsf{factor}(\bar{X})$
   b. $y \sim \mathcal{N}^{nK}(0, \frac{1}{\sqrt{K}})$
   c. $x' = Ty$

3. **Use x' as starting point for a local NLP solver on formulation**

$$\min_x \sum_{\{i,j\} \in E} \left(\|x_i - x_j\|^2 - d_{ij}^2 \right)^2$$

and return improved solution x

SDP+Barvinok vs. DDP+Barvinok

	SDP			DDP		
Instance	*LDE*	*MDE*	*CPU*	*LDE*	*MDE*	*CPU*
`C0700odd.1`	**0.00**	**0.00**	**0.63**	0.00	0.00	1.49
`C0700.odd.G`	**0.00**	**0.00**	**21.67**	0.42	0.01	30.51
`C0150alter.1`	**0.00**	**0.00**	**29.30**	0.00	0.00	34.13
`C0080create.1`	**0.00**	**0.00**	**139.52**	0.00	0.00	141.49
`1b03`	**0.18**	0.01	132.16	0.38	0.05	**101.04**
`1crn`	0.78	**0.02**	800.67	**0.76**	0.04	**522.60**
`1guu-1`	**0.79**	**0.01**	1900.48	0.90	0.04	**667.03**

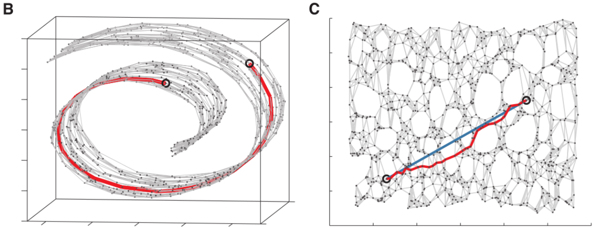
Most of the CPU time taken by local NLP solver

Subsection 7

Isomap for the DGP

Isomap for DG

1. Let D' be the (square) weighted adjacency matrix of G
2. Complete D' to *approximate* sqEDM $\tilde{D}$
3. Perform PCA on $\tilde{D}$ given K dimensions
   (a) Let $\tilde{B} = -(1/2)J\tilde{D}J$, where $J = I - (1/n)\mathbf{1}\mathbf{1}^\top$
   (b) Find eigenval/vects Λ, P so $\tilde{B} = P^\top \Lambda P$
   (c) Keep $\leq K$ largest nonneg. eigenv. of Λ to get $\tilde{\Lambda}$
   (d) Let $\tilde{x} = P^\top \sqrt{\tilde{\Lambda}}$



Vary Step 2 to generate Isomap heuristics

Variants for Step 2

A. Floyd-Warshall all-shortest-paths algorithm on G
(classic Isomap)

B. Find a spanning tree (SPT) of G and compute a random realization in $\bar{x} \in \mathbb{R}^K$, use its sqEDM

C. Solve a push-and-pull SDP relaxation to find a realization $\bar{x} \in \mathbb{R}^n$, use its sqEDM

D. Solve an SDP relaxation with Barvinok objective to find $\bar{x} \in \mathbb{R}^r$ (with $r \leq \lfloor(\sqrt{8|E|+1}-1)/2\rfloor$), use its sqEDM
haven't really talked about this, sorry

Post-processing: *Use $\tilde{x}$ as starting point for local NLP solver*

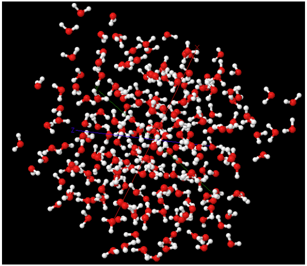
Results

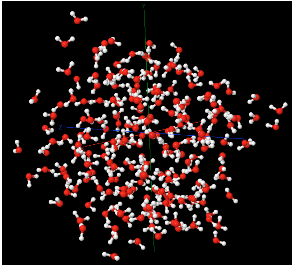
Comparison with `dgsol` [Moré, Wu 1997]

			A		B	C	D		A		B	C	D		A		B	C	D	
Instance			mde						lde						CPU					
Name	n	$\lvert E\rvert$	Isomap	IsoNLP	SPT	SDP	Barvinok	DGSol	Isomap	IsoNLP	SPT	SDP	Barvinok	DGSol	Isomap	IsoNLP	SPT	SDP	Barvinok	DGSol
C0700odd.1	15	39	0.585	0.001	0.190	0.068	**0.000**	0.135	0.989	0.004	0.896	0.389	**0.001**	0.634	**0.002**	1.456	1.589	0.906	1.305	1.747
C0700odd.2	15	39	0.599	**0.000**	0.187	0.086	**0.000**	0.128	0.985	**0.002**	0.956	0.389	0.009	1.000	**0.003**	1.376	1.226	1.002	1.063	0.887
C0700odd.3	15	39	0.599	**0.000**	0.060	0.086	**0.000**	0.128	0.985	**0.002**	0.326	0.389	0.009	1.000	**0.003**	1.259	1.256	0.861	1.167	0.877
C0700odd.4	15	39	0.599	**0.000**	0.283	0.086	0.001	0.128	0.985	**0.002**	2.449	0.389	0.008	1.000	**0.003**	1.347	1.222	0.976	1.063	1.033
C0700odd.5	15	39	0.599	**0.000**	0.225	0.086	0.000	0.128	0.985	**0.002**	0.867	0.389	0.007	1.000	**0.003**	1.284	1.157	0.987	1.100	0.700
C0700odd.6	15	39	0.599	**0.000**	0.283	0.086	0.000	0.128	0.985	**0.002**	1.520	0.389	**0.002**	1.000	**0.002**	1.372	1.196	0.998	1.305	0.909
C0700odd.7	15	39	0.585	0.001	0.080	0.068	**0.000**	0.135	0.989	0.004	0.361	0.389	**0.001**	0.634	**0.003**	1.469	1.322	0.894	1.093	1.719
C0700odd.8	15	39	0.585	0.001	0.056	0.068	**0.000**	0.135	0.989	0.004	0.275	0.389	**0.003**	0.634	**0.003**	1.408	1.306	0.692	1.079	1.744
C0700odd.9	15	39	0.585	0.001	0.057	0.068	**0.000**	0.135	0.989	0.004	0.301	0.389	**0.002**	0.634	**0.002**	1.430	1.172	0.791	1.093	1.745
C0700odd.A	15	39	0.585	0.001	0.043	0.068	**0.000**	0.135	0.989	**0.004**	0.316	0.389	**0.004**	0.634	**0.002**	1.294	1.269	0.722	1.220	1.523
C0700odd.B	15	39	0.585	0.001	0.151	0.068	**0.000**	0.135	0.989	**0.004**	1.022	0.389	**0.004**	0.634	**0.002**	1.297	1.279	0.871	1.111	1.747
C0700odd.C	15	39	0.835	**0.022**	0.033	0.039	0.031	0.025	1.012	**0.147**	0.393	0.211	0.294	0.167	**0.004**	6.803	6.369	7.371	7.030	7.000
C0700odd.D	36	242	0.835	**0.022**	0.041	0.039	0.042	0.025	1.012	**0.147**	0.423	0.211	0.268	0.167	**0.006**	6.806	6.575	7.422	7.603	7.095
C0700odd.E	36	242	0.835	**0.022**	0.064	0.039	0.031	0.025	1.012	**0.147**	0.894	0.211	0.260	0.167	**0.006**	6.911	6.638	7.365	6.979	7.008
C0700odd.F	36	242	0.599	**0.000**	0.047	0.086	**0.000**	0.128	0.985	**0.002**	0.308	0.389	0.005	1.000	**0.002**	1.299	1.310	1.008	1.100	1.040
C0150alter.1	37	335	0.786	0.058	0.066	0.014	0.015	**0.010**	0.992	0.571	0.693	0.256	0.285	**0.253**	**0.004**	9.492	9.456	10.276	10.120	9.272
C0080create.1	60	681	0.887	0.053	0.083	**0.024**	**0.024**	0.054	1.967	0.949	0.789	**0.511**	0.516	0.718	**0.012**	18.835	19.720	21.247	20.906	19.962
C0080create.2	60	681	0.887	0.053	0.047	**0.024**	**0.024**	0.054	1.967	0.949	0.585	**0.511**	0.512	0.718	**0.008**	18.791	20.009	21.728	20.885	19.740
C0020pdb	107	999	0.939	0.110	0.119	**0.059**	0.060	0.103	1.242	1.113	1.349	1.082	1.138	**0.798**	**0.035**	29.024	27.772	35.273	35.486	32.479
1guu	150	955	0.986	0.068	0.069	**0.057**	**0.057**	0.061	0.999	0.854	0.830	**0.735**	0.751	0.768	**0.048**	30.869	28.784	41.488	41.852	37.848
1guu-1	150	959	0.986	0.061	0.063	0.058	**0.057**	0.060	1.000	**0.711**	0.855	0.805	0.829	0.778	**0.053**	31.322	31.442	42.308	41.590	37.218
1guu-4000	150	968	0.974	0.081	0.080	0.072	**0.065**	0.079	1.000	0.901	**0.728**	0.760	0.961	0.826	**0.050**	30.352	29.856	42.330	39.832	42.015
C0030pkl	198	3247	0.961	0.112	0.160	**0.076**	0.077	0.137	**1.197**	1.354	2.230	1.995	2.054	1.401	**0.091**	105.175	104.775	149.192	146.360	111.859
1PPT	302	3102	0.984	**0.121**	0.129	0.128	0.129	0.123	**1.000**	1.519	1.219	1.944	1.956	1.224	**0.356**	112.448	110.345	185.815	187.182	118.681
100d	488	5741	0.987	0.146	0.146	0.155	0.157	**0.137**	**1.000**	1.577	1.397	1.764	1.749	1.358	**0.828**	229.809	213.136	659.638	659.280	233.115
GeoMean			0.74	**0.00**	0.09	0.06	**0.00**	0.08	1.07	**0.04**	0.73	0.50	0.06	0.66	**0.01**	6.30	6.04	5.93	6.63	6.30
Avg			0.76	0.04	0.11	0.07	**0.03**	0.10	1.09	**0.44**	0.88	0.63	0.47	0.77	**0.06**	26.12	25.21	49.69	49.55	27.96
StDev			0.17	0.05	0.07	**0.03**	0.04	0.04	**0.27**	0.55	0.57	0.52	0.65	0.34	**0.18**	51.69	48.82	135.08	134.97	53.26

Large instances

Instance			mde		lde		CPU	
Name	$\|V\|$	$\|E\|$	IsoNLP	dgsol	IsoNLP	dgsol	IsoNLP	dgsol
water	648	11939	**0.005**	0.15	**0.557**	0.81	26.98	**15.16**
3al1	678	17417	0.036	**0.007**	0.884	**0.810**	**170.91**	210.25
1hpv	1629	18512	**0.074**	0.078	0.936	**0.932**	374.01	**60.28**
il2	2084	45251	**0.012**	0.035	**0.910**	0.932	465.10	**139.77**
1tii	5684	69800	0.078	**0.077**	0.950	**0.897**	7400.48	**454.375**





Subsection 8

Concluding remarks

Summary of difficulties

- **Quadratic nonconvex too difficult?**
- *Solve SDP relaxation*
- **SDP relaxation too large?**
- *Solve DDP approximation*
- **Get $n \times n$ matrix solution, need $K \times n$!**

Rank reduction methods

- Multidimensional Scaling (MDS)
- Principal Component Analysis (PCA)
- Barvinok's naive algorithm (BNA)
- Isomap

Can also use them for dimensionality reduction!
n vectors in $\mathbb{R}^n \longrightarrow \mathbb{R}^K$

Outline

Job offers

Optimisation / Operations Senior Manager
VINCI Airports
Rueil-Malmaison, Île-de-France, France
...for the delivery of the various **optimization** projects... to the success of each **optimization** project...

Pricing Data Scientist/Actuary - Price **Optimization** Specialist(H-F)
AXA Global Direct
Région de Paris, France
...**optimization**. The senior price **optimization**... **Optimization** and Innovation team, and will be part...

Growth Data scientist - Product Features Team
Deezer
Paris, FR
OverviewPress play on your next adventure! Music... to join the Product Performance & **Optimization** team... www.deezer.com

Analystes et Consultants - Banque -Optimisation des opérations financières...
Accenture
Région de Paris, France
Nous recherchons des analystes jeunes diplômés et des consultants H/F désireux de travailler sur des problématiques d'optimisation des opérations bancaires (optimisation des modèles opérationnels et des processus) en France et au Benelux. Les postes sont à pourvoir en CDI, sur base d'un rattachement...

Electronic Health Record (EHR) Coordinator (Remote)
Aledade, Inc. - Bethesda, MD
Must have previous implementation or optimization experience with ambulatory EHRs and practice management software, preferably with expertise in Greenway,...

Operations Research Scientist
Ford Motor Company - ★★★★☆ 2,381 reviews - Dearborn, MI
Strong knowledge of optimization techniques (e.g. Develop optimization frameworks to support models related to mobility solution, routing problem, pricing and...

IS&T Controller
Alstom
Saint-Ouen, FR
The Railway industry today is characterized... reviews, software deployment **optimization**, running... jobsearch.alstom.com

Fares Specialist / Spécialiste Optimisation des Tarifs Aériens
Egencia, an Expedia company
Courbevoie - FR
EgenciaChaque année, Egencia accompagne des milliers de sociétés réparties dans plus de 60 pays à mieux gérer leurs programmes de voyage. Nous proposons des solutions modernes et des services d'exception à des millions de voyageurs, de la planification à la finalisation de leur voyage. Nous répondons...

Automotive HMI Software Experts or Software Engineers
Elektrobit (EB)
Paris Area, France
Elektrobit Automotive offers in Paris interesting... performances and **optimization** area, and/or software...

Deployment Engineer, Professional Services, Google Cloud
Google
Paris, France
Note: By applying to this position your... migration, network **optimization**, security best...

Operations Research Scientist
Marriott International, Inc - ★★★★☆ 4,694 reviews - Bethesda, MD 20817
Analyzes data and builds optimization,. Programming models and familiarity with optimization software (CPLEX, Gurobi)....

Research Scientist - AWS New Artificial Intelligence Team!
A Research Scientist - AWS New Artificial Intelligence Team! eviews - Palo Alto, CA
We are pioneers in areas such as recommendation engines, product search, eCommerce fraud detection, and large-scale optimization of fulfillment center...

An example

Under the responsibility of the Commercial Director, the Optimisation / Operations Senior Manager will have the responsibility to optimise and develop operational aspects for VINCI Airports current and future portfolio of airports. They will also be responsible for driving forward and managing key optimisation projects that assist the Commercial Director in delivering the objectives of the Technical Services Agreements activities of VINCI Airports. The Optimisation Manager will support the Commercial Director in the development and implementation of plans, strategies and reporting processes. As part of the exercise of its function, the Optimisation Manager will undertake the following: Identification and development of cross asset synergies with a specific focus on the operations and processing functions of the airport. Definition and implementation of the Optimisation Strategy in line with the objectives of the various technical services agreements, the strategy of the individual airports and the Group. This function will include: Participation in the definition of airport strategy. Definition of this airport strategy into the Optimisation Strategy. Regularly evaluate the impact of the Optimisation Strategy. Ensure accurate implementation of this strategy at all airports. Management of the various technical services agreements with our airports by developing specific technical competences from the Head Office level. Oversee the management and definition of all optimisation projects. Identification, overview and management of the project managers responsible for the delivery of the various optimization projects at each asset. Construction of good relationships with the key stakeholders, in order to contribute to the success of each optimization project. Development and implementation of the Group optimisation plan. Definition of economic and quality of service criteria, in order to define performance goals. Evaluation of the performance of the Group operations in terms of processing efficiency, service levels, passenger convenience and harmonization of the non-aeronautical activities. Monitoring the strategies, trends and best practices of the airport industry and other reference industries in terms of the applicability to the optimization plan. Study of the needs and preferences of the passengers, through a continuous process of marketing research at all of the airports within the VINCI Airports portfolio. Development of benchmarking studies in order to evaluate the trends, in international airports or in the local market. Development and participation in the expansion or refurbishment projects of the airports, to assure a correct configuration and positioning of the operational and commercial area that can allow the optimization of the revenues and operational efficiency. Support the Director Business Development through the analysis and opportunity assessment of areas of optimization for all target assets in all bids and the preparation and implementation of the strategic plan once the assets are acquired. Maintain up to date knowledge of market trends and key initiatives related to the operational and commercial aspects of international airports [...]

...and blah blah blah: IS THIS APPROPRIATE FOR MY CV?

Try Natural Language Processing

- *Automated summary*
- *Relation Extraction*
- *Named Entity Recognition* **(NER)**
- *Keywords*

Automated summary

`./summarize.py job01.txt`

```
They will also be responsible for driving forward and managing key
optimisation projects that assist the Commercial Director in delivering
the objectives of the Technical Services Agreements activities of VINCI
Airports. The Optimisation Manager will support the Commercial Director
in the development and implementation of plans, strategies and reporting
processes. Identification and development of cross asset synergies with
a specific focus on the operations and processing functions of the airport.
Construction of good relationships with the key stakeholders, in order to
contribute to the success of each optimization project. Definition of
economic and quality of service criteria, in order to define performance
goals. Evaluation of the performance of the Group operations in terms of
processing efficiency, service levels, passenger convenience and
harmonization of the non-aeronautical activities. Development of
benchmarking studies in order to evaluate the trends, in international
airports or in the local market. Maintain up to date knowledge of market
trends and key initiatives related to the operational and commercial
aspects of international airports. You have a diverse range of
experiences working at or with airports across various disciplines such
as operations, ground handling, commercial, etc. Demonstrated high
level conceptual thinking, creativity and analytical skills.
```

Does it help? hard to say

Relation Extraction

```
./relextr-mitie.py job01.txt
```

```
======= RELATIONS =======
Optimisation Strategy [ INCLUDES_EVENT ] VINCI Airports
Self [ INCLUDES_EVENT ] Head Office
Head Office [ INFLUENCED_BY ] Self
Head Office [ INTERRED_HERE ] Self
VINCI Airports [ INTERRED_HERE ] Optimisation Strategy
Head Office [ INVENTIONS ] Self
Optimisation Strategy [ LOCATIONS ] VINCI Airports
Self [ LOCATIONS ] Head Office
Self [ ORGANIZATIONS_WITH_THIS_SCOPE ] Head Office
Self [ PEOPLE_INVOLVED ] Head Office
Self [ PLACE_OF_DEATH ] Head Office
Head Office [ RELIGION ] Self
VINCI Airports [ RELIGION ] Optimisation Strategy
```

Does it help? **hardly**

Named Entity Recognition

```
./ner-mitie.py job01.txt
```

```
==== NAMED ENTITIES =====
English MISC
French MISC
Head Office ORGANIZATION
Optimisation / Operations ORGANIZATION
Optimisation Strategy ORGANIZATION
Self PERSON
Technical Services Agreements MISC
VINCI Airports ORGANIZATION
```

Does it help? ...maybe

For a document D, let $\mathsf{NER}(D)$ = named entity words

Subsection 1

Clustering on graphs

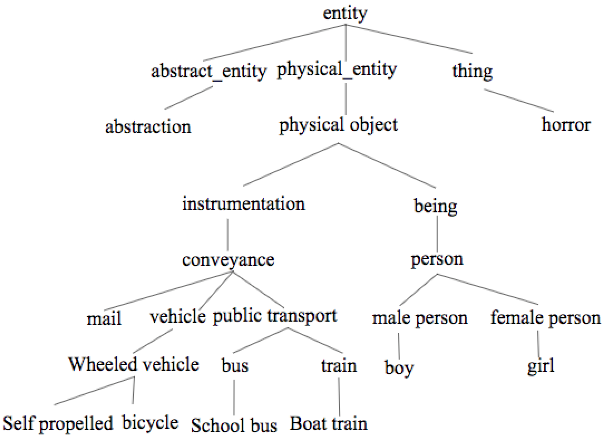
Exploit NER to infer relations

1. Recognize named entities from all documents
2. Use them to compute distances among documents
3. Use modularity clustering

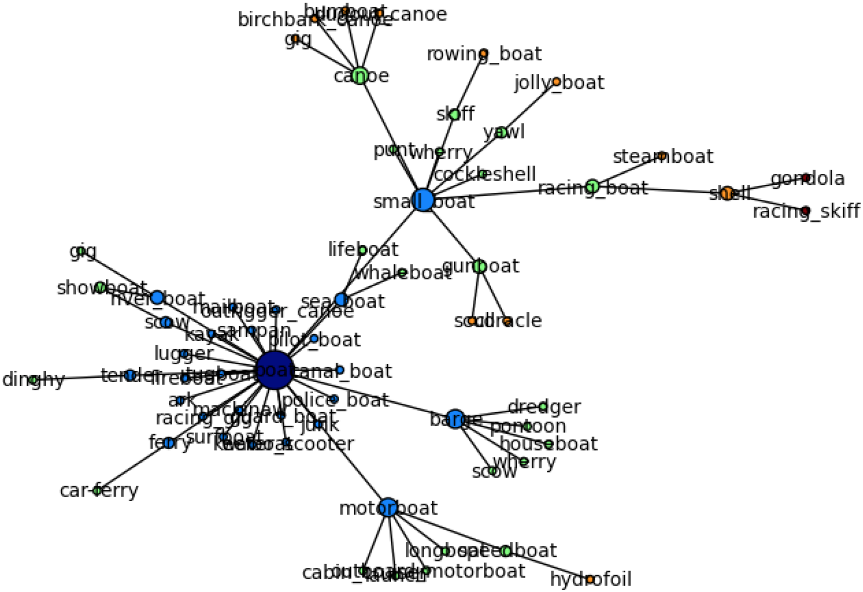
The named entities

1. *Operations Head Airports Office VINCI Technical Self French / Strategy Agreements English Services Optimisation*
2. *Europe and P&C Work Optimization Head He/she of Price Global PhDs Direct Asia Earnix AGD AXA Innovation Coordinate International English*
3. *Scientist Product Analyze Java Features & Statistics Science PHP Pig/Hive/Spark Optimization Crunch/analyze Team Press Performance Deezer Data Computer*
4. *Lean6Sigma Lean-type Office Banking Paris CDI France RPA Middle Accenture English Front Benelux*
5. *Partners Management Monitor BC Provide Support Sites Regions Mtiers Program Performance market develop Finance & IS&T Saint-Ouen Region Control Followings VP Sourcing external Corporate Sector and Alstom Tax Directors Strategic Committee*
6. *Customer Specialist Expedia Service Interact Paris Travel Airline French France Management Egencia English Fares with Company Inc*
7. *Paris Integration France Automation Automotive French . Linux/Genivi HMI UI Software EB Architecture Elektrobit technologies GUIDE Engineers German Technology SW well-structured Experts Tools*
8. *Product Google Managers Python JavaScript AWS JSON BigQuery Java Platform Engineering HTML MySQL Services Professional Googles Ruby Cloud OAuth*
9. *EHR Aledades Provide Wellness Perform ACO Visits EHR-system-specific Coordinator Aledade Medicare Greenway Allscripts*
10. *Global Java EXCEL Research Statistics Mathematics Analyze Smart Teradata & Python Company GDIA Ford Visa SPARK Data Applied Science Work C++ R Unix/Linux Physics Microsoft Operations Monte JAVA Mobility Insight Analytics Engineering Computer Motor SQL Operation Carlo PowerPoint*
11. *Management Java CANDIDATE Application Statistics Gurobi Provides Provider Mathematics Service Maintains Deliver SM&G SAS/HPF SAS Data Science Economics Marriott PROFILE Providers OR Engineering Computer SQL Education*
12. *Alto Statistics Java Sunnyvale Research ML Learning Science Operational Machine Amazon Computer C++ Palo Internet R Seattle*
13. *LLamasoft Work Fortune Chain Supply C# Top Guru What Impactful Team LLamasofts Makes Gartner Gain*
14. *Worldwide Customer Java Mosel Service Python Energy Familiarity CPLEX Research Partnering Amazon R SQL CS Operations*
15. *Operations Science Research Engineering Computer Systems or Build*
16. *Statistics Italy Broad Coins France Australia Python Amazon Germany SAS Appstore Spain Economics Experience R Research US Scientist UK SQL Japan Economist*
17. *Competency Statistics Knowledge Employer communication Research Machine EEO United ORMA Way OFCCP Corporation Mining & C# Python Visual Studio Opportunity Excellent Modeling Data Jacksonville Arena Talent Skills Science Florida Life Equal AnyLogic Facebook CSX Oracle The Strategy Vision Operations Industrial Stream of States Analytics Engineering Computer Framework Technology*
18. *Java Asia Research Safety in Europe Activities North Company WestRocks Sustainability America Masters WRK C++ Norcross Optimization GA ILOG South NYSE Operations AMPL CPLEX Identify Participate OPL WestRock*
19. *Management Federal Administration System NAS Development JMP Traffic Aviation FAA Advanced McLean Center CAASD Flow Air Tableau Oracle MITRE TFM Airspace National SQL Campus*
20. *Abilities & Skills 9001-Quality S Management ISO GED*
21. *Statistics Group RDBMS Research Mathematics Teradata ORSA Greenplum Java SAS U.S. Solution Time Oracle Military Strategy Physics Linear/Non-Linear Operations both Industrial Series Econometrics Engineering Clarity Regression*

Word similarity: WordNet



WordNet: hyponyms of “boat”



Wu-Palmer word similarity

Semantic WordNet distance between words w_1, w_2

$$\mathsf{wup}(w_1, w_2) = \frac{2\,\mathsf{depth}(\mathsf{lcs}(w_1, w_2))}{\mathsf{len}(\mathsf{shortest\_path}(w_1, w_2)) + 2\,\mathsf{depth}(\mathsf{lcs}(w_1, w_2))}$$

- lcs: *lowest common subsumer*

  earliest common word in paths from both words to WordNet root

- depth: *length of path from root to word*

Example: wup(dog, boat)?

```
depth( whole ) = 4
18 -> dog -> canine -> carnivore -> placental -> mammal -> vertebrate
   -> chordate -> animal -> organism -> living_thing -> whole -> artifact
   -> instrumentality -> conveyance -> vehicle -> craft -> vessel -> boat

13 -> dog -> domestic_animal -> animal -> organism -> living_thing
   -> whole -> artifact -> instrumentality -> conveyance -> vehicle
   -> craft -> vessel -> boat
```

$$\mathsf{wup}(\mathbf{dog}, \mathbf{boat}) = 8/21 = 0.380952380952$$

Extensions of Wu-Palmer similarity

- **to lists of words H, L:**

$$\mathsf{wup}(H, L) = \frac{1}{|H|\,|L|} \sum_{v \in H} \sum_{w \in L} \mathsf{wup}(v, w)$$

- **to pairs of documents D_1, D_2:**

$$\mathsf{wup}(D_1, D_2) = \mathsf{wup}(\mathsf{NER}(D_1), \mathsf{NER}(D_2))$$

- wup **and its extensions are always in** $[0, 1]$

The similarity matrix

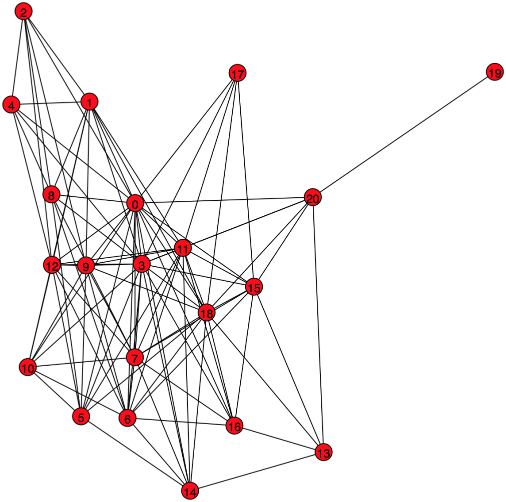
$$
\begin{pmatrix}
1.00 & 0.63 & 0.51 & 0.51 & 0.66 & 0.45 & 0.46 & 0.47 & 0.72 & 0.58 & 0.54 & 0.50 & 0.72 & 0.38 & 0.49 & 0.47 & 0.47 & 0.44 & 0.54 & 0.31 & 0.44 \\
0.63 & 1.00 & 0.45 & 0.45 & 0.54 & 0.40 & 0.42 & 0.42 & 0.57 & 0.49 & 0.46 & 0.45 & 0.59 & 0.35 & 0.43 & 0.42 & 0.42 & 0.41 & 0.47 & 0.32 & 0.40 \\
0.51 & 0.45 & 1.00 & 0.40 & 0.53 & 0.35 & 0.37 & 0.37 & 0.58 & 0.47 & 0.43 & 0.40 & 0.59 & 0.28 & 0.39 & 0.37 & 0.38 & 0.35 & 0.43 & 0.24 & 0.35 \\
0.51 & 0.45 & 0.40 & 1.00 & 0.63 & 0.45 & 0.46 & 0.46 & 0.67 & 0.56 & 0.52 & 0.49 & 0.68 & 0.38 & 0.48 & 0.47 & 0.47 & 0.45 & 0.53 & 0.33 & 0.44 \\
0.66 & 0.54 & 0.53 & 0.63 & 1.00 & 0.34 & 0.35 & 0.35 & 0.49 & 0.42 & 0.39 & 0.37 & 0.50 & 0.29 & 0.36 & 0.35 & 0.35 & 0.34 & 0.40 & 0.26 & 0.34 \\
0.45 & 0.40 & 0.35 & 0.45 & 0.34 & 1.00 & 0.42 & 0.43 & 0.66 & 0.54 & 0.49 & 0.45 & 0.67 & 0.34 & 0.44 & 0.43 & 0.43 & 0.40 & 0.49 & 0.28 & 0.40 \\
0.46 & 0.42 & 0.37 & 0.46 & 0.35 & 0.42 & 1.00 & 0.44 & 0.66 & 0.54 & 0.49 & 0.47 & 0.67 & 0.34 & 0.45 & 0.45 & 0.44 & 0.42 & 0.50 & 0.28 & 0.40 \\
0.47 & 0.42 & 0.37 & 0.46 & 0.35 & 0.43 & 0.44 & 1.00 & 0.67 & 0.55 & 0.51 & 0.48 & 0.68 & 0.36 & 0.47 & 0.45 & 0.45 & 0.43 & 0.51 & 0.30 & 0.42 \\
0.72 & 0.57 & 0.58 & 0.67 & 0.49 & 0.66 & 0.66 & 0.67 & 1.00 & 0.33 & 0.31 & 0.29 & 0.40 & 0.23 & 0.28 & 0.27 & 0.28 & 0.26 & 0.31 & 0.21 & 0.26 \\
0.58 & 0.49 & 0.47 & 0.56 & 0.42 & 0.54 & 0.54 & 0.55 & 0.33 & 1.00 & 0.46 & 0.43 & 0.59 & 0.34 & 0.42 & 0.41 & 0.41 & 0.39 & 0.46 & 0.31 & 0.39 \\
0.54 & 0.46 & 0.43 & 0.52 & 0.39 & 0.49 & 0.49 & 0.51 & 0.31 & 0.46 & 1.00 & 0.39 & 0.56 & 0.29 & 0.38 & 0.36 & 0.36 & 0.34 & 0.41 & 0.24 & 0.35 \\
0.50 & 0.45 & 0.40 & 0.49 & 0.37 & 0.45 & 0.47 & 0.48 & 0.29 & 0.43 & 0.39 & 1.00 & 0.70 & 0.40 & 0.50 & 0.49 & 0.48 & 0.46 & 0.54 & 0.35 & 0.46 \\
0.72 & 0.59 & 0.59 & 0.68 & 0.50 & 0.67 & 0.67 & 0.68 & 0.40 & 0.59 & 0.56 & 0.70 & 1.00 & 0.23 & 0.29 & 0.29 & 0.29 & 0.28 & 0.33 & 0.20 & 0.27 \\
0.38 & 0.35 & 0.28 & 0.38 & 0.29 & 0.34 & 0.34 & 0.36 & 0.23 & 0.34 & 0.29 & 0.40 & 0.23 & 1.00 & 0.48 & 0.45 & 0.46 & 0.42 & 0.52 & 0.30 & 0.43 \\
0.49 & 0.43 & 0.39 & 0.48 & 0.36 & 0.44 & 0.45 & 0.47 & 0.28 & 0.42 & 0.38 & 0.50 & 0.29 & 0.48 & 1.00 & 0.39 & 0.39 & 0.36 & 0.45 & 0.26 & 0.37 \\
0.47 & 0.42 & 0.37 & 0.47 & 0.35 & 0.43 & 0.45 & 0.45 & 0.27 & 0.41 & 0.36 & 0.49 & 0.29 & 0.45 & 0.39 & 1.00 & 0.48 & 0.46 & 0.54 & 0.33 & 0.44 \\
0.47 & 0.42 & 0.38 & 0.47 & 0.35 & 0.43 & 0.44 & 0.45 & 0.28 & 0.41 & 0.36 & 0.48 & 0.29 & 0.46 & 0.39 & 0.48 & 1.00 & 0.43 & 0.51 & 0.32 & 0.43 \\
0.44 & 0.41 & 0.35 & 0.45 & 0.34 & 0.40 & 0.42 & 0.43 & 0.26 & 0.39 & 0.34 & 0.46 & 0.28 & 0.42 & 0.36 & 0.46 & 0.43 & 1.00 & 0.53 & 0.31 & 0.43 \\
0.54 & 0.47 & 0.43 & 0.53 & 0.40 & 0.49 & 0.50 & 0.51 & 0.31 & 0.46 & 0.41 & 0.54 & 0.33 & 0.52 & 0.45 & 0.54 & 0.51 & 0.53 & 1.00 & 0.36 & 0.46 \\
0.31 & 0.32 & 0.24 & 0.33 & 0.26 & 0.28 & 0.28 & 0.30 & 0.21 & 0.31 & 0.24 & 0.35 & 0.20 & 0.30 & 0.26 & 0.33 & 0.32 & 0.31 & 0.36 & 1.00 & 0.47 \\
0.44 & 0.40 & 0.35 & 0.44 & 0.34 & 0.40 & 0.40 & 0.42 & 0.26 & 0.39 & 0.35 & 0.46 & 0.27 & 0.43 & 0.37 & 0.44 & 0.43 & 0.43 & 0.46 & 0.47 & 1.00
\end{pmatrix}
$$

The similarity matrix

Too uniform! Try zeroing values below median

$$
\begin{pmatrix}
1.00 & 0.63 & 0.51 & 0.51 & 0.66 & 0.45 & 0.46 & 0.47 & 0.72 & 0.58 & 0.54 & 0.50 & 0.72 & 0.00 & 0.49 & 0.47 & 0.47 & 0.44 & 0.54 & 0.00 & 0.44 \\
0.63 & 1.00 & 0.45 & 0.45 & 0.54 & 0.00 & 0.00 & 0.00 & 0.57 & 0.49 & 0.46 & 0.45 & 0.59 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.47 & 0.00 & 0.00 \\
0.51 & 0.45 & 1.00 & 0.00 & 0.53 & 0.00 & 0.00 & 0.00 & 0.58 & 0.47 & 0.00 & 0.00 & 0.59 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.51 & 0.45 & 0.00 & 1.00 & 0.63 & 0.45 & 0.46 & 0.46 & 0.67 & 0.56 & 0.52 & 0.49 & 0.68 & 0.00 & 0.48 & 0.47 & 0.47 & 0.45 & 0.53 & 0.00 & 0.44 \\
0.66 & 0.54 & 0.53 & 0.63 & 1.00 & 0.00 & 0.00 & 0.00 & 0.49 & 0.00 & 0.00 & 0.00 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.45 & 0.00 & 0.00 & 0.45 & 0.00 & 1.00 & 0.00 & 0.00 & 0.66 & 0.54 & 0.49 & 0.45 & 0.67 & 0.00 & 0.44 & 0.00 & 0.00 & 0.00 & 0.49 & 0.00 & 0.00 \\
0.46 & 0.00 & 0.00 & 0.46 & 0.00 & 0.00 & 1.00 & 0.44 & 0.66 & 0.54 & 0.49 & 0.47 & 0.67 & 0.00 & 0.45 & 0.45 & 0.44 & 0.00 & 0.50 & 0.00 & 0.00 \\
0.47 & 0.00 & 0.00 & 0.46 & 0.00 & 0.00 & 0.44 & 1.00 & 0.67 & 0.55 & 0.51 & 0.48 & 0.68 & 0.00 & 0.47 & 0.45 & 0.45 & 0.00 & 0.51 & 0.00 & 0.00 \\
0.72 & 0.57 & 0.58 & 0.67 & 0.49 & 0.66 & 0.66 & 0.67 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.58 & 0.49 & 0.47 & 0.56 & 0.00 & 0.54 & 0.54 & 0.55 & 0.00 & 1.00 & 0.46 & 0.43 & 0.59 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.46 & 0.00 & 0.00 \\
0.54 & 0.46 & 0.00 & 0.52 & 0.00 & 0.49 & 0.49 & 0.51 & 0.00 & 0.46 & 1.00 & 0.00 & 0.56 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.50 & 0.45 & 0.00 & 0.49 & 0.00 & 0.45 & 0.47 & 0.48 & 0.00 & 0.43 & 0.00 & 1.00 & 0.70 & 0.00 & 0.50 & 0.49 & 0.48 & 0.46 & 0.54 & 0.00 & 0.46 \\
0.72 & 0.59 & 0.59 & 0.68 & 0.50 & 0.67 & 0.67 & 0.68 & 0.00 & 0.59 & 0.56 & 0.70 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.48 & 0.45 & 0.46 & 0.00 & 0.52 & 0.00 & 0.43 \\
0.49 & 0.00 & 0.00 & 0.48 & 0.00 & 0.44 & 0.45 & 0.47 & 0.00 & 0.00 & 0.00 & 0.50 & 0.00 & 0.48 & 1.00 & 0.00 & 0.00 & 0.00 & 0.45 & 0.00 & 0.00 \\
0.47 & 0.00 & 0.00 & 0.47 & 0.00 & 0.00 & 0.45 & 0.45 & 0.00 & 0.00 & 0.00 & 0.49 & 0.00 & 0.45 & 0.00 & 1.00 & 0.48 & 0.46 & 0.54 & 0.00 & 0.44 \\
0.47 & 0.00 & 0.00 & 0.47 & 0.00 & 0.00 & 0.44 & 0.45 & 0.00 & 0.00 & 0.00 & 0.48 & 0.00 & 0.46 & 0.00 & 0.48 & 1.00 & 0.00 & 0.51 & 0.00 & 0.00 \\
0.44 & 0.00 & 0.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.46 & 0.00 & 0.00 & 0.00 & 0.46 & 0.00 & 1.00 & 0.53 & 0.00 & 0.00 \\
0.54 & 0.47 & 0.00 & 0.53 & 0.00 & 0.49 & 0.50 & 0.51 & 0.00 & 0.46 & 0.00 & 0.54 & 0.00 & 0.52 & 0.45 & 0.54 & 0.51 & 0.53 & 1.00 & 0.00 & 0.46 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.47 \\
0.44 & 0.00 & 0.00 & 0.44 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.46 & 0.00 & 0.43 & 0.00 & 0.44 & 0.00 & 0.00 & 0.46 & 0.47 & 1.00
\end{pmatrix}
$$

The graph



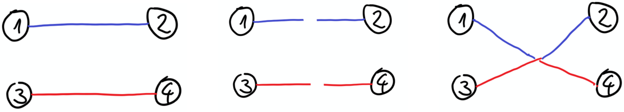
$G = (V, E)$, weighted adjacency matrix A

A is like B with zeroed low components

Modularity clustering

"Modularity is the fraction of the edges that fall within a cluster minus the expected fraction if edges were distributed at random."

- *"at random"* = random graphs over same degree sequence
- *degree sequence* = $(k_1, \ldots, k_n)$ where $k_i = |N(i)|$
- *"expected"* = all possible "half-edge" recombinations



- expected edges between u, v: $k_u k_v/(2m)$ where $m = |E|$
- $\mathsf{mod}(u, v) = (A_{uv} - k_u k_v/(2m))$
- $\mathsf{mod}(G) = \sum_{\{u,v\} \in E} \mathsf{mod}(u, v) x_{uv}$

  $x_{uv} = 1$ if u, v in the same cluster and 0 otherwise
- *"Natural extension" to weighted graphs:* $k_u = \sum_v A_{uv}$, $m = \sum_{uv} A_{uv}$

Use modularity to define clustering

- What is the "best clustering"?
- *Maximize discrepancy between actual and expected*
  "as far away as possible from average"

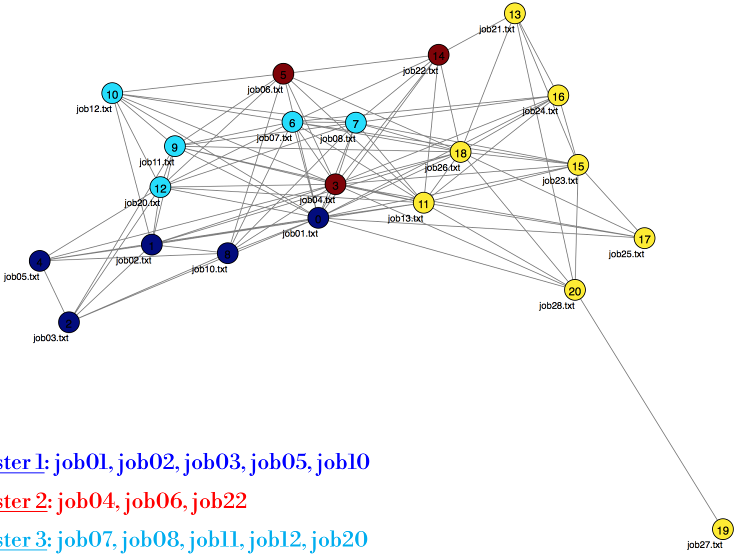
$$\left.\begin{array}{rl} \max & \sum_{\{u,v\}\in E} \mathsf{mod}(u,v)x_{uv} \\ \forall u\in V, v\in V & x_{uv}\in\{0,1\} \end{array}\right\}$$

- Issue: optimum could be intransitive
- Idea: *treat clusters as cliques (even if zero weight)*
  then clique partitioning constraints for transitivity

$$\begin{array}{rrcl} \forall i<j<k & x_{ij}+x_{jk}-x_{ik} & \leq & 1 \\ \forall i<j<k & x_{ij}-x_{jk}+x_{ik} & \leq & 1 \\ \forall i<j<k & -x_{ij}+x_{jk}+x_{ik} & \leq & 1 \end{array}$$

if $i,j\in C$ *and* $j,k\in C$ *then* $i,k\in C$

The resulting clustering



cluster 1: job01, job02, job03, job05, job10

cluster 2: job04, job06, job22

cluster 3: job07, job08, job11, job12, job20

cluster 4: job13, job21, job23, job24, job25, job26, job27, job28

Is it good?

Vinci	Accenture	Elektrobit	Amazon 1-3
Axa	Expedia	Google	CSX
Deezer	fragment1	Ford	Westrock
Alstom		Marriott	Mitre
Aledade		Llamasoft	Clarity
			fragment2

- ? – *named entities rarely appear in WordNet*
- Desirable property: *chooses number of clusters*

Subsection 2

Clustering in Euclidean spaces

Clustering vectors

Most frequent words $\boxed{w}$ *over collection* $\boxed{C}$ *of documents* $\boxed{d}$

`./keywords.py`

```
global environment customers strategic processes teams sql job industry use
java developing project process engineering field models opportunity drive
results statistical based operational performance using mathematical computer
new technical highly market company science role dynamic background products
level methods design looking modeling manage learning service customer
effectively technology requirements build mathematics problems plan services
time scientist implementation large analytical techniques lead available plus
technologies sas provide machine product functions organization algorithms
position model order identify activities innovation key appropriate different
complex best decision simulation strategy meet client assist quantitative
finance commercial language mining travel chain amazon pricing practices
cloud supply
```

$$
\begin{aligned}
\mathsf{tfidf}_C(w,d) &= \frac{|(t \in d \mid t = w)|\,|C|}{|\{h \in C \mid w \in h\}|} \\
\mathsf{keyword}_C(i,d) &= \textit{word } w \textit{ having } i^{th} \textit{ best } \mathsf{tfidf}_C(w,d)\textit{value} \\
\mathsf{vec}_C^m(d) &= (\mathsf{tfidf}_C(\mathsf{keyword}_C(i,d),d) \mid i \leq m)
\end{aligned}
$$

Transforms documents to vectors

Minimum sum-of-squares clustering

- MSSC, a.k.a. the k*-means problem*
- Given points $p_1, \ldots, p_n \in \mathbb{R}^m$, find clusters $C_1, \ldots, C_k$

$$\min \sum_{j \le k} \sum_{i \in C_j} \|p_i - \mathsf{centroid}(C_j)\|_2^2$$

where $\mathsf{centroid}(C_j) = \frac{1}{|C_j|} \sum_{i \in C_j} p_i$

- k-means alg.: given initial clustering $C_1, \ldots, C_k$

1: $\forall j \le k$ compute $y_j = \mathsf{centroid}(C_j)$
2: $\forall i \le n, j \le k$ if y_j is the closest centr. to p_i let $x_{ij} = 1$ else 0
3: $\forall j \le k$ update $C_j \leftarrow \{p_i \mid x_{ij} = 1 \land i \le n\}$
4: repeat until stability

k-means with $k = 2$

Vinci	AXA
Deezer	Alstom
Accenture	Elektrobit
Expedia	Ford
Google	Marriott
Aledade	Amazon 1-3
Llamasoft	CSX
	WestRock
	MITRE
	Clarity
	fragments 1-2

k-means with $k = 2$: *another run*

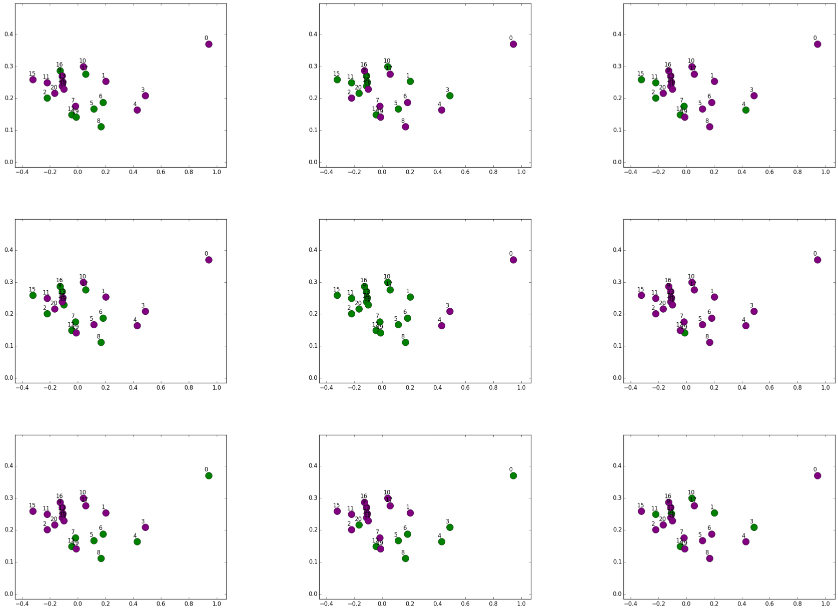
Deezer	Vinci
Elektrobit	AXA
Google	Accenture
Aledade	Alstom
	Expedia
	Ford
	Marriott
	Llamasoft
	Amazon 1-3
	CSX
	WestRock
	MITRE
	Clarity
	fragments 1-2

k-means with $k = 2$: *third run!*

AXA	Vinci
Deezer	Accenture
Expedia	Alstom
Ford	Elektrobit
Marriott	Google
Llamasoft	Aledade
Amazon 1-3	
CSX	
WestRock	
MITRE	
Clarity	
fragments 1-2	

A fickle algorithm

We can't trust k-means: why?



Subsection 3

Distance resolution limit

Nearest Neighbours

k-NEAREST NEIGHBOURS (k-NN).

Given:

- $k \in \mathbb{N}$
- *a distance function* $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$
- *a set* $\mathcal{X} \subset \mathbb{R}^n$
- *a point* $z \in \mathbb{R}^n \setminus \mathcal{X}$,

find the subset $\mathcal{Y} \subset \mathcal{X}$ *such that:*

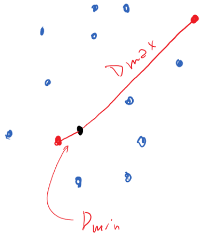
(a) $|\mathcal{Y}| = k$

(b) $\forall y \in \mathcal{Y}, x \in \mathcal{X} \quad (d(z, y) \leq d(z, x))$

- **basic problem in data science**
- pattern recognition, computational geometry, machine learning, data compression, robotics, recommender systems, information retrieval, natural language processing and more
- Example: Used in Step 2 of k-means:
  assign points to closest centroid

[Cover & Hart 1967]

With random variables



- Consider 1-NN
- Let $\ell = |\mathcal{X}|$
- Distance function family $\{d^m : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+\}_m$
- For each m:
  - random variable Z^m with some distribution over $\mathbb{R}^n$
  - for $i \leq \ell$, random variable X_i^m with some distrib. over $\mathbb{R}^n$
  - X_i^m iid w.r.t. i, Z^m independent of all X_i^m
  - $D_{\mathbf{min}}^m = \min_{i \leq \ell} d^m(Z^m, X_i^m)$
  - $D_{\mathbf{max}}^m = \max_{i \leq \ell} d^m(Z^m, X_i^m)$

Distance Instability Theorem

- Let $p > 0$ be a constant
- If

$$\exists i \leq \ell \quad (d^m(Z^m, X_i^m))^p \textit{ converges as } m \to \infty$$

then, for any $\varepsilon > 0$,

closest and furthest point are at about the same distance

Note "$\exists i$" suffices since $\forall m$ we have X_i^m iid w.r.t. i

[Beyer et al. 1999]

Distance Instability Theorem

- Let $p > 0$ be a constant
- If

$$\exists i \leq \ell \quad \lim_{m\to\infty} \mathsf{Var}((d^m(Z^m, X_i^m))^p) = 0$$

then, for any $\varepsilon > 0$,

$$\lim_{m\to\infty} \mathbb{P}(D^m_{\mathbf{max}} \leq (1+\varepsilon) D^m_{\mathbf{min}}) = 1$$

Note "$\exists i$" suffices since $\forall m$ we have X_i^m iid w.r.t. i

[Beyer et al. 1999]

Preliminary results

- ***Lemma***. $\{B^m\}_m$ **seq. of rnd. vars with finite variance and** $\lim_{m\to\infty} \mathbb{E}(B^m) = b \wedge \lim_{m\to\infty} \mathsf{Var}(B^m) = 0$; **then**

$$\forall \varepsilon > 0 \quad \lim_{m\to\infty} \mathbb{P}(\|B^m - b\| \leq \varepsilon) = 1$$

$\boxed{\textbf{denoted } B^m \to_{\mathbb{P}} b}$

- ***Slutsky's theorem***. $\{B^m\}_m$ **seq. of rnd. vars and** g **a continuous function; if** $B^m \to_{\mathbb{P}} b$ **and** $g(b)$ **exists, then** $g(B^m) \to_{\mathbb{P}} g(b)$

- ***Corollary***. **If** $\{A^m\}_m, \{B^m\}_m$ **seq. of rnd. vars. s.t.** $A^m \to_{\mathbb{P}} a$ **and** $B^m \to_{\mathbb{P}} b \neq 0$ **then** $\{\frac{A^m}{B^m}\}_m \to_{\mathbb{P}} \frac{a}{b}$

Proof

1. $\mu_m = \mathbb{E}((d^m(Z^m, X_i^m))^p)$ **independent of** i
   (since all X_i^m iid)
2. $V_m = \frac{(d^m(Z^m, X_i^m))^p}{\mu_m} \to_{\mathbb{P}} 1$**:**
   - $\mathbb{E}(V_m) = 1$ (rnd. var. over mean) $\Rightarrow \lim_m \mathbb{E}(V_m) = 1$
   - **Hypothesis of thm.** $\Rightarrow \lim_m \mathsf{Var}(V_m) = 0$
   - *Lemma* $\Rightarrow V_m \to_{\mathbb{P}} 1$
3. $\mathbf{D}^m = ((d^m(Z^m, X_i^m))^p \mid i \leq \ell) \to_{\mathbb{P}} \mathbf{1}$ (by iid)
4. *Slutsky's thm.* $\Rightarrow \min(\mathbf{D}^m) \to_{\mathbb{P}} \min(\mathbf{1}) = 1$
   simy for max
5. *Corollary* $\Rightarrow \frac{\max(\mathbf{D}^m)}{\min(\mathbf{D}^m)} \to_{\mathbb{P}} 1$
6. $\frac{D^m_{\mathbf{max}}}{D^m_{\mathbf{min}}} = \frac{\mu_m \max(\mathbf{D}^m)}{\mu_m \min(\mathbf{D}^m)} \to_{\mathbb{P}} 1$
7. **Result follows** (defn. of $\to_{\mathbb{P}}$ and $D^m_{\mathbf{max}} \geq D^m_{\mathbf{min}}$)

When it applies

- iid random variables from any distribution
- Particular forms of correlation
  e.g. $U_i \sim \mathsf{Uniform}(0, \sqrt{i})$, $X_1 = U_1$, $X_i = U_i + (X_{i-1}/2)$ for $i > 1$
- Variance tending to zero
  e.g. $X_i \sim \mathsf{N}(0, 1/i)$
- Discrete uniform distribution on m-dimensional hypercube
  for both data and query
- Computational experiments with k-means:
  instability already with $n > 15$

…and when it doesn't

- Complete linear dependence on all distributions
  can be reduced to NN in 1D
- Exact and approximate matching
  query point = (or ≈) data point
- Query point in a well-separated cluster in data
- Implicitly low dimensionality
  project; but NN must be stable in lower dim.

Subsection 4

MP formulations

MP formulation

$$\left.\begin{array}{rrcl}
\min\limits_{x,y,s} & \sum\limits_{i\le n}\sum\limits_{j\le k}\|p_i - y_j\|_2^2\, x_{ij} & & \\
\forall j\le k & \frac{1}{s_j}\sum\limits_{i\le n} p_i x_{ij} & = & y_j \\
\forall i\le n & \sum\limits_{j\le k} x_{ij} & = & 1 \\
\forall j\le k & \sum\limits_{i\le n} x_{ij} & = & s_j \\
\forall j\le k & y_j & \in & \mathbb{R}^m \\
 & x & \in & \{0,1\}^{nk} \\
 & s & \in & \mathbb{N}^k
\end{array}\right\} \quad (\textbf{MSSC})$$

MINLP: nonconvex terms; continuous, binary and integer variables

Reformulation

The (MSSC) formulation has the same optima as:

$$\left.\begin{array}{rrcl}
\min\limits_{x,y,P} & \sum\limits_{i\le n}\sum\limits_{j\le k} P_{ij}\,x_{ij} & & \\
\forall i\le n, j\le k & \|p_i-y_j\|_2^2 & \le & P_{ij} \\
\forall j\le k & \sum\limits_{i\le n} p_i x_{ij} & = & \sum\limits_{i\le n} y_j x_{ij} \\
\forall i\le n & \sum\limits_{j\le k} x_{ij} & = & 1 \\
\forall j\le k & y_j & \in & ([\min\limits_{i\le n} p_{ih}, \max\limits_{i\le n} p_{ih}] \mid h\le k) \\
 & x & \in & \{0,1\}^{nk} \\
 & P & \in & [0, P^U]^{nk}
\end{array}\right\}$$

- **The only nonconvexities are** *products of binary by continuous* *__bounded__* *variables*

Products of binary and continuous vars.

- Suppose term xy appears in a formulation
- Assume $x \in \{0,1\}$ and $\boxed{y \in [0,1]}$ is <u>bounded</u>
- means "either $z = 0$ or $z = y$"
- *Replace xy by a new variable z*
- Adjoin the following constraints:

$$\begin{array}{rcl} & z & \in [0,1] \\ y-(1-x) \le & z & \le y+(1-x) \\ -x \le & z & \le x \end{array}$$

- $\Rightarrow$ Everything's linear now!

[Fortet 1959]

Products of binary and continuous vars.

- **Suppose term xy appears in a formulation**
- **Assume $x \in \{0,1\}$ and $\boxed{y \in [y^L, y^U]}$ is <u>bounded</u>**
- **means "either $z = 0$ or $z = y$"**
- ***Replace xy by a new variable z***
- **Adjoin the following constraints:**

$$\begin{array}{rcl} & z \in & [\min(y^L, 0), \max(y^U, 0)] \\ y - (1-x)\max(|y^L|, |y^U|) \leq & z & \leq y + (1-x)\max(|y^L|, |y^U|) \\ -x\max(|y^L|, |y^U|) \leq & z & \leq x\max(|y^L|, |y^U|) \end{array}$$

- **$\Rightarrow$ Everything's linear now!**

[L. et al. 2009]

MSSC is a convex MINLP

$$\begin{array}{rrcl}
\min\limits_{x,y,P,\chi,\xi} & \sum\limits_{i\le n}\sum\limits_{j\le k} & \chi_{ij} & \\
\forall i\le n, j\le k & 0\le & \chi_{ij} & \le P_{ij} \\
\forall i\le n, j\le k & P_{ij}-(1-x_{ij})P^U\le & \chi_{ij} & \le x_{ij}P^U \\
\forall i\le n, j\le k & \|p_i-y_j\|_2^2 & \le & P_{ij} \qquad \boxed{\Leftarrow \quad \textbf{convex}} \\
\forall j\le k & \sum\limits_{i\le n} p_i x_{ij} & = & \sum\limits_{i\le n}\xi_{ij} \\
\forall i\le n, j\le k & y_j-(1-x_{ij})\max(|y^L|,|y^U|)\le & \xi_{ij} & \le y_j+(1-x_{ij})\max(|y^L|,|y^U|) \\
\forall i\le n, j\le k & -x_{ij}\max(|y^L|,|y^U|)\le & \xi_{ij} & \le x_{ij}\max(|y^L|,|y^U|) \\
\forall i\le n & \sum\limits_{j\le k} x_{ij} & = & 1 \\
\forall j\le k & y_j & \in & [y^L,y^U] \\
 & x & \in & \{0,1\}^{nk} \\
 & P & \in & [0,P^U]^{nk} \\
 & \chi & \in & [0,P^U]^{nk} \\
\forall i\le n, j\le k & \xi_{ij} & \in & [\min(y^L,0),\max(y^U,0)]
\end{array}$$

y_j, ξ_{ij}, y^L, y^U **are vectors in** $\mathbb{R}^m$

How to solve it

- cMINLP is NP-hard
- Algorithms:
  - *Outer Approximation* (OA)
  - *Branch-and-Bound* (BB)
- Best (open source) solver: Bonmin
- Another good (commercial) solver: KNitro
- *With* $k = 2$*, unfortunately...*

```
Cbc0010I After 8300 nodes, 3546 on tree, 14.864345 best solution,
  best possible 6.1855969 (32142.17 seconds)
```

- *Interesting feature: the bound*

guarantees we can't to better than *bound*

all BB algorithms provide it

Bonmin's first solution

Alstom	Vinci
Elektrobit	AXA
Ford	Deezer
Llamasoft	Accenture
Amazon 2	Expedia
CSX	Google
MITRE	Aledade
Clarity	Marriott
fragment 2	Amazon 1 & 3
	WestRock
	fragment 1

Couple of things left to try

- **Approximate ℓ_2 by ℓ_1 norm**
  ℓ_1 is a linearizable norm
- **Randomly project the data**
  lose dimensions but keep approximate shape

Linearizing convexity

- **Replace $\|p_i - y_j\|_2^2$ by $\|p_i - y_j\|_1$**
- **Warning**: optima will change
  but still within "clustering by distance" principle

$$\forall i \le n, j \le k \quad \|p_i - y_j\|_1 = \sum_{a \le d} |p_{ia} - y_{ja}|$$

- **Replace each $|\cdot|$ term by new vars. $Q_{ija} \in [0, P^U]$**
  Adjust P^U in terms of $\|\cdot\|_1$
- **Adjoin constraints**

$$\begin{array}{rrcl} \forall i \le n, j \le k & \sum_{a \le d} Q_{ija} & \le & P_{ij} \\ \forall i \le n, j \le k, a \le d & -Q_{ija} & \le & p_{ia} - y_{ja} \le Q_{ija} \end{array}$$

- **Obtain a MILP**
  Most advanced MILP solver: CPLEX

CPLEX's first solution

objective 112.24, bound 39.92, in 44.74s

AXA	Vinci
Deezer	Accenture
Ford	Alstom
Marriott	Expedia
Amazon 1-3	Elektrobit
Llamasoft	Google
CSX	Aledade
WestRok	
MITRE	
Clarity	
fragments 1-2	

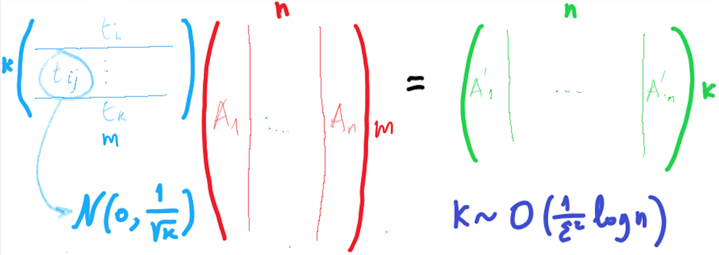
Interrupted after 281s with bound 59.68

Subsection 5

Clustering in high dimensions

The magic of random projections

- *"Mathematics of big data"*
- In a nutshell



- *Clustering on* $\boxed{A'}$ *rather than* A
- Approximate results with arbitrarily high probability (wahp)

[Johnson & Lindenstrauss, 1984]

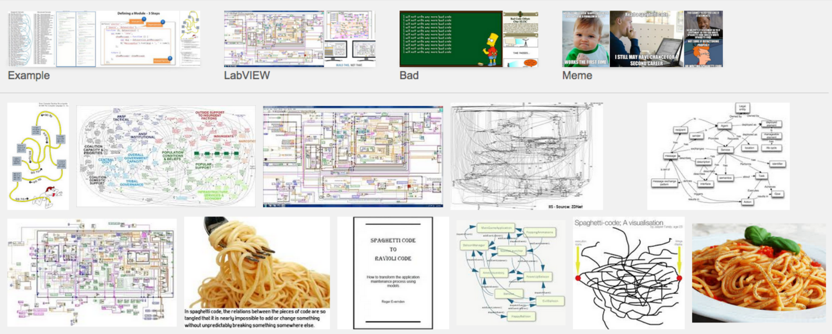
The magic of random projections

- *"Mathematics of big data"*
- **In a nutshell**
  1. **Given pts.** $A_i, \ldots, A_n \in \mathbb{R}^m$ **with** m **large and** $\varepsilon \in (0,1)$
  2. **Pick "appropriate"** $k \approx O(\frac{1}{\varepsilon^2} \ln n)$
  3. **Sample** $k \times d$ **matrix** T (*each comp. i.i.d.* $\mathcal{N}(0, \frac{1}{\sqrt{k}})$)
  4. **Consider** ***projected*** **points** $A'_i = TA_i \in \mathbb{R}^k$ **for** $i \leq n$
  5. **With prob**$\to 1$ **exponentially fast as** $k \to \infty$

$$\forall i,j \leq n \quad (1-\varepsilon)\|A_i - A_j\|_2 \leq \|A'_i - A'_j\|_2 \leq (1+\varepsilon)\|A_i - A_j\|_2$$

[Johnson & Lindenstrauss, 1984]

Clustering Google images

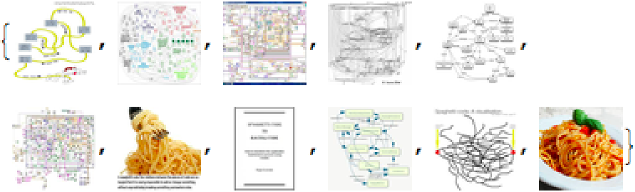


[L. & Lavor, *in press*]

k-means without random projections

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

Out[28]=



```
VHcl = Timing[ClusteringComponents[VHimg, 3, 1]]
Out[29]= {0.405908, {1, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3}}
```

Too slow!

k-means with random projections

```
Get["Projection.m"];
VKimg = JohnsonLindenstrauss[VHimg, 0.1];
VKcl = Timing[ClusteringComponents[VKimg, 3, 1]]
Out[34]= {0.002232, {1, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3}}
```

From 0.405s CPU time to 0.00232s

Same clustering

Works on the MSSC MP formulation too!

$$\left.\begin{array}{rrcl}
\min\limits_{x,y,s} & \sum\limits_{i\le n}\sum\limits_{j\le d} \|Tp_i - Ty_j\|_2^2\, x_{ij} & & \\
\forall j \le d & \frac{1}{s_j}\sum\limits_{i\le n} Tp_i x_{ij} & = & Ty_j \\
\forall i \le n & \sum\limits_{j\le d} x_{ij} & = & 1 \\
\forall j \le d & \sum\limits_{i\le n} x_{ij} & = & s_j \\
\forall j \le d & y_j & \in & \mathbb{R}^m \\
 & x & \in & \{0,1\}^{nd} \\
 & s & \in & \mathbb{N}^d
\end{array}\right\}$$

where T ***is a*** $k \times m$ ***random projector***

replace Ty **by** y'

Works on the MSSC MP formulation too!

$$\left.\begin{array}{rrcl}
\min\limits_{x,y',s} & \sum\limits_{i\le n}\sum\limits_{j\le d}\|Tp_i - y'_j\|_2^2\, x_{ij} & & \\
\forall j\le d & \frac{1}{s_j}\sum\limits_{i\le n} Tp_i x_{ij} & = & y'_j \\
\forall i\le n & \sum\limits_{j\le d} x_{ij} & = & 1 \\
\forall j\le d & \sum\limits_{i\le n} x_{ij} & = & s_j \\
\forall j\le d & y'_j & \in & \mathbb{R}^k \\
 & x & \in & \{0,1\}^{nd} \\
 & s & \in & \mathbb{N}^d
\end{array}\right\} \quad (\mathbf{MSSC'})$$

- ***where*** $k = O(\frac{1}{\varepsilon^2}\ln n)$
- **less data, $|y'| < |y| \Rightarrow$ get solutions faster**
- **Yields smaller cMINLP**

Bonmin on randomly proj. data

objective 5.07, bound 0.48, stopped at 180s

Deezer	Vinci
Ford	AXA
Amazon 1-3	Accenture
CSX	Alstom
MITRE	Expedia
fragment 1	Elektrobit
	Google
	Aledade
	Marriott
	Llamasoft
	WestRock
	Clarity
	fragment 2

CPLEX on randomly proj. data

...although it doesn't make much sense for $\|\cdot\|_1$ norm...

objective 53.19, bound 20.68, stopped at 180s

Vinci	AXA
Deezer	Accenture
Expedia	Alstom
Google	Elektrobit
Aledade	Marriott
Ford	Llamasoft
Amazon 1-3	WestRock
CSX	MITRE
Clarity	fragment 1-2

Many clusterings

- We obtained many different clusterings
- Is there any common sense?
- How do we compare them?
- Can we extract useful information from the comparison?
- How many clusters should we look for? Is $k = 2$ OK?
- *Did we just turn the issue of "I have too many data" into "I have too many solutions"?*

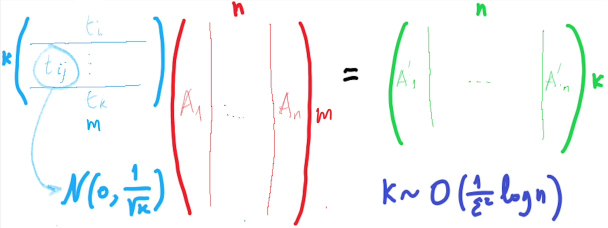
Outline

The gist

- Let A, b be very large, consider LP

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

- T short & fat normally sampled



- Then

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

with high probability

Linear feasibility

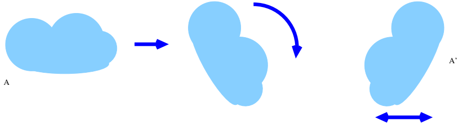
Restricted Linear Membership (RLM$_X$)
Given $A_1, \ldots, A_n, b \in \mathbb{R}^m$ **and** $X \subseteq \mathbb{R}^n$, $\exists\,?\, x \in X$ **s.t.**

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

- **Linear Feasibility Problem (LFP) with** $X = \mathbb{R}^n_+$
- **Integer Feasibility Problem (IFP) with** $X = \mathbb{Z}^n_+$

The shape of a set of points

- **Lose dimensions but not too much accuracy**
  Given $A_1, \dots, A_n \in \mathbb{R}^m$ find $k \ll m$ and points $A'_1, \dots, A'_n \in \mathbb{R}^k$ 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 $\Leftrightarrow$ distortion:
  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

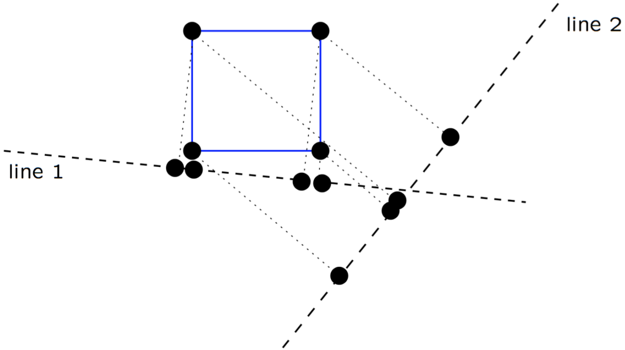
Given $X \subseteq \mathbb{R}^n$ **and** $b, A_1, \ldots, A_n \in \mathbb{R}^m$, **find** $k \ll m$, $b', A'_1, \ldots, A'_n \in \mathbb{R}^k$ **such that:**

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

with high probability

Losing dimensions = "projection"

In the plane, hopeless



In 3D: no better

Johnson-Lindenstrauss Lemma

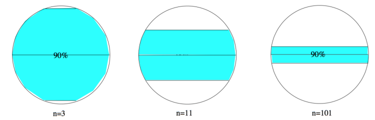
Thm.

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.**

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

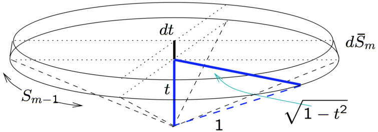
If $k \times m$ **matrix** T **is sampled componentwise from** $N(0, \frac{1}{\sqrt{k}})$**, then** A **and** TA **have almost the same shape**

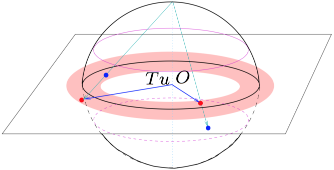
Sketch of a JLL proof by pictures

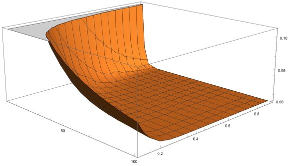


Thm.

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 $\mathsf{E}(\|Tu\|^2) = 1$







Mean invariance

Thm.

- T **a** $k \times m$ **rnd proj matrix, samples from** $N(0, 1/\sqrt{k})$
- $u \in \mathbb{R}^m$, $\|u\| = 1$
- $\Rightarrow \|Tu\| = \|u\| = 1$

Pf.

- $Tu = (y_1, \ldots, y_k) \Rightarrow \forall i \leq k \; y_i = \sum_{j \leq m} T_{ij} u_j$
- $\mathbb{E}(y_i) = \sum_j \mathbb{E}(T_{ij}) u_j = \sum_j 0 u_j = 0$
- $\mathbb{V}(y_i) = \sum_j \mathbb{V}(T_{ij}) u_j^2 = \sum_j \frac{1}{k} u_j^2 = \frac{1}{k} \|u\|^2$ **(**$\forall i$**)**
- $\frac{1}{k} \|u\|^2 = \mathbb{V}(y_i) = \mathbb{E}(y_i^2 - (\mathbb{E}(y_i))^2) = \mathbb{E}(y_i^2)$ **(**$\forall i$**)**
- $\mathbb{E}(\|Tu\|^2) = \mathbb{E}(\|y\|^2) = \mathbb{E}(\sum_i y_i^2) = \sum_i \mathbb{E}(y_i^2) = \frac{1}{k} \sum_{i \leq k} \|u\|^2 = \|u\|^2$

Sampling to desired accuracy

- Distortion has low probability:

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

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

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

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

In practice

- **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

Subsection 1

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** $\mathbf{RLM}_X$ **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**).**

[arXiv:1507.00990v1/math.OC]

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 $C = \mathsf{cone}(A_1, \ldots, A_n)$ for $A \subseteq \mathbb{R}^m$
- **We can 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
  Difficult part is proving the converse

We can also project point-to-cone distances

Projecting the separation

Thm.

Given $c, b, A_1, \ldots, A_n \in \mathbb{R}^m$ **of unit norm s.t.** $b \notin \mathsf{cone}\{A_1, \ldots, A_n\}$ **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:**

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

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

[arXiv:1507.00990v1/math.OC]

The feasibility projection theorem

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**

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

Subsection 2

Projecting optimality

Notation

- $P \equiv \mathbf{min}\{cx \mid Ax = b \wedge x \geq 0\}$ *(original problem)*
- $TP \equiv \mathbf{min}\{cx \mid TAx = Tb \wedge x \geq 0\}$ *(projected problem)*
- $v(P) =$ **optimal objective function value of** P
- $v(TP) =$ **optimal objective function value of** TP

The optimality projection theorem

- Assume $\mathsf{feas}(P)$ is bounded
- Assume all optima of P satisfy $\sum_j x_j \le \theta$ for some given $\theta > 0$
  (prevents cones from being "too flat")

Thm.

Given $\delta > 0$,

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

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

Show $v(TP) \leq v(P)$:

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

$$\left.\begin{array}{rcl} cx & \leq & v(P) - \delta \\ Ax & = & b \\ x & \geq & 0 \end{array}\right\} \qquad (11)$$

Note: for $\delta > 0$, Eq. (11) is infeasible

- By feasibility projection theorem,

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

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 $x \in \mathsf{feas}(TP)$
- $\Rightarrow v(P) - \delta \leq v(TP)$

Subsection 3

Solution retrieval

Projected solutions are infeasible in P

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

Solution retrieval by duality

- Primal $\min\{c^\top x \mid Ax = b \wedge x \ge 0\} \Rightarrow$ dual $\max\{b^\top y \mid A^\top y \le c\}$
- Let $x' = \mathsf{sol}(TP)$ and $y' = \mathsf{sol}(\mathsf{dual}(TP))$
- $\Rightarrow (TA)^\top y' = (A^\top T^\top)y' = A^\top(T^\top y') \le c$
- $\Rightarrow T^\top y'$ is a solution of $\mathsf{dual}(P)$
- $\Rightarrow$ **we can compute an optimal basis J for P**
- Solve $A_J x_J = b$, **get x_J,** obtain a solution x^* of P
- *Won't work in practice: errors in computing J*

Solution retrieval by pseudoinverse

- H: optimal basis of TP
  we can trust that — given by solver
- $|H| = k \Rightarrow A_H$ is $m \times k$ (tall and slim)
- **Pseudoinverse**: solve $k \times k$ system $A_H^\top A_H x_H = A_H^\top b$
- let $x = (x_H, 0)$
- *Can prove small feasibility error wahp*
- **ISSUE**: may be slightly infeasible
  empirically: $x \ngeq 0$ *but* $x^- \to 0$ *as* $k \to \infty$

Subsection 4

Quantile regression

Regression

- **multivariate random var.** X
  function $y = f(X)$
  sample $\{(a_i, b_i) \in \mathbb{R}^p \times \mathbb{R} \mid i \leq m\}$

- **sample mean:**

$$\hat{\mu} = \operatorname*{arg\,min}_{\mu \in \mathbb{R}} \sum_{i \leq m} (b_i - \mu)^2$$

- **sample mean conditional to** $X = A = (a_{ij})$**:**

$$\hat{\nu} = \operatorname*{arg\,min}_{\nu \in \mathbb{R}^p} \sum_{i \leq m} (b_i - \nu a_i)^2$$

Quantile regression

- *sample median:*

$$\begin{aligned}\hat{\xi} &= \arg\min_{\xi \in \mathbb{R}} \sum_{i \leq m} |b_i - \xi| \\ &= \arg\min_{\xi \in \mathbb{R}} \sum_{i \leq m} \left(\frac{1}{2} \max(b_i - \xi, 0) - \frac{1}{2} \min(b_i - \xi, 0) \right)\end{aligned}$$

- *sample τ-quantile:*

$$\hat{\xi} = \arg\min_{\xi \in \mathbb{R}} \sum_{i \leq m} (\tau \max(b_i - \xi, 0) - (1 - \tau) \min(b_i - \xi, 0))$$

- *sample τ-quantile conditional to $X = A = (a_{ij})$:*

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \sum_{i \leq m} (\tau \max(b_i - \beta a_i, 0) - (1 - \tau) \min(b_i - \beta a_i, 0))$$

Linear Programming formulation

$$\left.\begin{array}{lrcl} \textbf{min} & \tau u^+ + (1-\tau)u^- & & \\ & A(\beta^+ - \beta^-) + u^+ - u^- & = & b \\ & \beta, u & \geq & 0 \end{array}\right\}$$

- **Parameters:** A is $m \times p$, $b \in \mathbb{R}^m$, $\tau \in \mathbb{R}$
- **Decision variables:** $\beta^+, \beta^- \in \mathbb{R}^p$, $u^+, u^- \in \mathbb{R}^m$
- **LP constraint matrix is** $m \times (2p + 2m)$
  density: $p/(p+m)$ **— can be high**

Large datasets

- Russia Longitudinal Monitoring Survey *household data (`hh1995f`)*
  - $m = 3783, p = 855$
  - $A =$ `hf1995f`, $b = \log \text{avg}(A)$
  - 18.5% dense
  - poorly scaled data, CPLEX yields infeasible (!!!) after around 70s CPU
  - `quantreg` in R fails
- 14596 RGB photos on my HD, scaled to 90×90
  - $m = 14596, p = 24300$
  - each row of A is an image vector, $b = \sum A$
  - 62.4% dense
  - CPLEX killed by OS after ≈30min (presumably for lack of RAM) on 16GB

Results on large datasets

Instance			Projection				Original		
τ	m	p	k	opt	CPU	feas	opt	CPU	qnt err
					`hh1995f`				
0.25	3783	856	411	0.00	8.53	0.038%	71.34	17.05	0.16
0.50				0.00	8.44	0.035%	89.17	15.25	0.05
0.75				0.00	8.46	0.041%	65.37	31.67	3.91
					`jpegs`				
0.25	14596	24300	506	0.00	231.83	0.51%	0.00	3.69E+5	0.04
0.50				0.00	227.54	0.51%	0.00	3.67E+5	0.05
0.75				0.00	228.57	0.51%	0.00	3.68E+5	0.05
					`random`				
0.25	1500	100	363	0.25	2.38	0.01%	1.06	6.00	0.00
0.50				0.40	2.51	0.01%	1.34	6.01	0.00
0.75				0.25	2.57	0.01%	1.05	5.64	0.00
0.25	2000	200	377	0.35	4.29	0.01%	2.37	21.40	0.00
0.50				0.55	4.37	0.01%	3.10	23.02	0.00
0.75				0.35	4.24	0.01%	2.42	21.99	0.00

$$\text{feas} = 100\frac{\|Ax-b\|_2}{\|b\|_1/m}$$

$$\text{qnt err} = \frac{\|\text{qnt} - \text{proj. qnt}\|_2}{\#\,\text{cols}}$$

IPM with *no simplex crossover*:
solution w/o opt. guarantee
cannot trust results
simplex method won't work
due to ill-scaling and size

Subsection 5

Sparsity and ℓ_1 minimization

Coding problem 1

- **Need to send sparse vector $y \in \mathbb{R}^n$ with $n \gg 1$**

1. **Sample full rank $k \times n$ matrix A with $k \ll n$**

   preliminary: both parties know A

2. **Encode $b = Ay \in \mathbb{R}^k$**
3. **Send b**

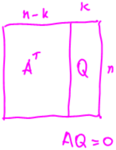
- **Decode by finding sparsest x s.t. $Ax = b$**

Coding problem 2

- Need to send a sequence $w \in \mathbb{R}^k$
- Encoding $n \times k$ matrix Q, with $n \gg k$, send $z = Qw \in \mathbb{R}^n$

  preliminary: both parties know Q
- (Low) prob. e of error: $e\, n$ comp. of z sent wrong

  they can be totally off
- *Receive (wrong) vector $\bar{z} = z + x$ where x is sparse*
- Can we recover z?
  - Choose $k \times n$ matrix A s.t. $AQ = 0$
  - Let $b = A\bar{z} = A(z + x) = A(Qw + x) = AQw + Ax = Ax$
  - *Suppose we can find sparsest x' s.t. $Ax' = b$*
  - $\Rightarrow$ we can recover $z' = \bar{z} - x'$



- Recover $w' = (Q^\top Q)^{-1} Q^\top z'$

  Likelihood of getting small $\|w - w'\|$?

Sparsest solution of a linear system

- **Problem** $\min\{\|x\|_0 \mid Ax = b\}$ **is NP-hard**

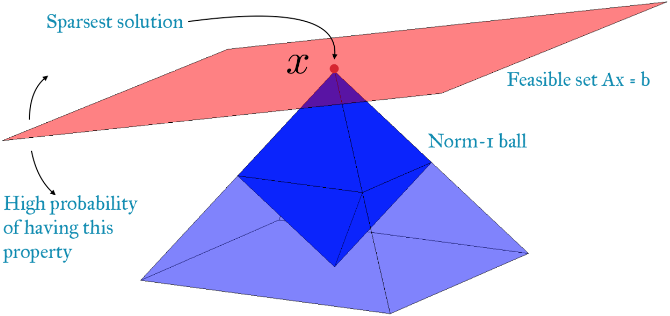
  Reduction from Exact Cover by 3-Sets [Garey&Johnson 1979, A6[MP5]]
- **Relax to** $\min\{\|x\|_1 \mid Ax = b\}$
- **Reformulate to LP:**

$$\left.\begin{array}{rrcl} \min & \sum_{j\le n} & s_j & \\ \forall j \le n & -s_j \le & x_j & \le s_j \\ & Ax & = & b \end{array}\right\} \qquad (\dagger)$$

- **Empirical observation: can often find optimum**

  Too often for this to be a coincidence
- **Theoretical justification by Candès, Tao, Donoho**
  "Mathematics of sparsity", "Compressed sensing"

Graphical intuition 1

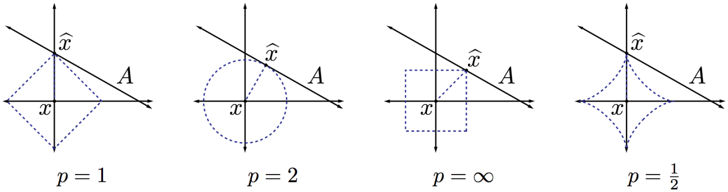


- **Wouldn't work with ℓ_2, ℓ_∞ norms**

  $Ax = b$ ***flat at poles*** — **"zero probability of sparse solution"**

Warning: *this is not a proof, and there are cases not explained by this drawing* **[Candès 2014]**

Graphical intuition 2



- $\hat{x}$ **such that** $A\hat{x} = b$ **approximates** x **in** ℓ_p **norms**
- $p = 1$ **only convex case zeroing some components**

From [Davenport et al., 2012]; again, this is not a proof!

Not for the faint-hearted

1. Hand, Voroninski:
   arxiv.org/pdf/1611.03935v1.pdf
2. Candès and Tao:
   statweb.stanford.edu/~candes/papers/DecodingLP.pdf
3. Candès:
   statweb.stanford.edu/~candes/papers/ICM2014.pdf
4. Davenport *et al.*:
   statweb.stanford.edu/~markad/publications/
   ddek-chapter1-2011.pdf
5. Lustig *et al.*:
   people.eecs.berkeley.edu/~mlustig/CS/CSMRI.pdf
6. *and many more* (look for "compressed sensing")

Finding orthogonal A, Q

- **[Matousek, Gärtner 2007]:**
  - sample A componentwise from $N(0,1)$
  - *approximately preserves Euclidean distances by JLL*
  - then "find Q s.t. $QA = 0$"
  - *in practice, Gaussian elim. on underdet. system $AQ = 0$*
- **Faster:**
  - sample $n \times n$ matrix from uniform distribution
  - *full rank with probability 1*
  - find eigenvectors (orthonormal basis)
  - *random rotation of standard basis: used in JLL proof*
  - Q: first k eigenvectors, A: last $n-k$ eigenvectors
  - *$AQ = 0$ by construction!*

From message to recovery

Procedure:

1. **message:** ***character string*** s
2. $w = \mathsf{bin}(\mathsf{char2asc(s)})$
3. **send** $z = Qw$**, receive** $\bar{z} = z + x$**, let** $b = A\bar{z}$
   $\delta =$ ***sparsity of*** x***,*** Q ***is*** $n \times k$ ***full rank with*** $n \gg k$
4. **use (†) to find sparsest** x' **satisfying** $Ax = b$
5. $z' = \bar{z} - x'$
6. $w' = \mathsf{cap}(\mathsf{round}((Q^\top Q)^{-1} Q^\top z'), [0, 1])$
7. $s' = \mathsf{asc2char}(\mathsf{bytechunk}(w'))$
8. **evaluate** $s_{\mathsf{err}} = \|s - s'\|$

Parameter choice [Matousek]:

- $\delta = 0.08$
- $n = 4k$

Improvements

- **Reduce CPU time spent on LP**
- $n = 4k$ **redundancy for** $\delta = 0.08$ **error seems excessive**

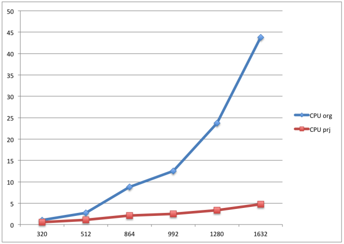
LP size reduction

- $Ax = b$ **is an** $(n-k) \times n$ **system**
- $n-k$ **"relatively close" to** n
- **Exploit JLL to project columns!**

Computational results

k	n	δ	ϵ	α	$s_{\mathsf{err}}^{\mathsf{org}}$	$s_{\mathsf{err}}^{\mathsf{prj}}$	CPU$^{\mathsf{org}}$	CPU$^{\mathsf{prj}}$
80	320	0.08	0.20	0.02	0	0	1.05	0.56
128	512	0.08	0.20	0.02	0	0	2.72	1.10
216	864	0.08	0.20	0.02	0	0	8.83	2.12
248	992	0.08	0.20	0.02	0	0	12.53	2.53
320	1280	0.08	0.20	0.02	0	0	23.70	3.35
408	1632	0.08	0.20	0.02	0	0	43.80	4.75

- $k = |s|, n = 4k, \delta = 0.08, \epsilon = 0.2$
- $\alpha =$ Achlioptas density
  $\mathsf{P}(T_{ij} = -1) = \mathsf{P}(T_{ij} = 1) = \frac{\alpha}{2}$
  $\mathsf{P}(T_{ij} = 0) = 1 - \alpha$
- $s_{\mathsf{err}} =$ number of different characters
- CPU: seconds of elapsed time
- *1 sampling of A, Q, T*



Sentence: *Conticuere omnes intentique ora tenebant, inde toro* [...]

Reducing redundancy in n

- *How about taking $n = (1+\delta)k$?*
- $n - k \approx \delta k$ is very small
- Makes $Ax = b$ very short and fat
- Prevents compressed sensing from working correctly
- Need $n - k \approx k$, $n \approx k$ and $AQ = 0$: impossible
- Relax to $AQ \approx 0$?

The JLL again

Aim $A^\top, Q$ of size $n \times k$ with $AQ \approx 0$

- **JLL Corollary:**
  $\exists O(e^d)$ ***approximately orthogonal vectors in*** $\mathbb{R}^d$
- **Algorithm:**
  1. $d = O(\ln n)$
  2. T **sampled componentwise from** $N(0, \frac{1}{\sqrt{d}})$ **(as in JLL)**
  3. **cols of** $T\, I_n$ **are** $n = O(e^d)$ **almost orthog. vect. in** $\mathbb{R}^d$
  4. **Pf.:** *JLL approximately preserves distances and scalar products*

<u>Concentration of measure:</u> *accuracy increases with* d

Strategy

- **Aim at $k \times n$ A and $n \times k$ Q s.t. $AQ \approx 0$**
  with $n = (1 + \delta')k$ and δ' "small" (say $\delta' < 1$)
- $\Rightarrow$ $2k$ **approxim. orthog. vectors in $\mathbb{R}^n$ with $n < 2k$**
- **JLL: errors too large for such "small" sizes**
- **Note we only need $AQ = 0$:**
  accept non-orthogonality in rows of A & cols of Q

LP for almost orthogonality

- **Sample Q and compute A using an LP**
  WLOG: we could sample A and compute Q
- $\max \sum_{\substack{i \leq k \\ j \leq n}} \mathsf{Uniform}(-1,1) A_{ij}$
- **subject to $AQ = 0$ and $A \in [-1,1]$**
- **for $k = 328$ and $n = 590$ (i.e. $\delta' = 0.8$):**
  - **error: $\sum A_i Q^j = O(10^{-10})$**
  - **rank: full**
  - **CPU: 688s (*meh*)**
- **for $k = 328$ and $n = 492$ (i.e. $\delta' = 0.5$): *the same***
- **for $k = 328$ and $n = 426$ (i.e. $\delta' = 0.3$): *CPU 470s***
- **Reduce CPU time by solving k LPs deciding A_i (for $i \leq k$)**

Computational results

k	n	δ'	$s_{\text{err}}^{\text{org}}$	$s_{\text{err}}^{\text{prj}}$	CPU^{org}	CPU^{prj}
328	426	0.3	182	15	2.45	1.87
328	426	0.3	154	0	2.20	1.49
328	459	0.4	0	1	4.47	2.45
328	459	0.4	5	17	2.86	1.46
328	492	0.5	60	0	4.53	1.18
328	492	0.5	34	0	5.38	1.18
328	590	0.8	14	0	8.30	1.41
328	590	0.8	107	4	6.76	1.43

- CPU for computing A, Q not counted: *precomputation is possible*
- Approximate beats precise!

Conclusion

- If s is short, set $\delta' = \delta$ and use compressed sensing (CS)
- If s is longer, try increasing δ' and use CS
- If s is very long, use *JLL-projected CS*
- Can use approximately orthogonal A, Q too

Conticuere omnes, intentique ora tenebant.
Inde toro pater Aeneas sic orsus ab alto:
Infandum, regina, iubes renovare dolorem.
Troianas ut opes et lamentabile regnum eruerint Danai
Quaequae ipse miserrima vidi et quorum pars magna fui.

[Virgil, *Aeneid*, Cantus II]

$k = 1896, n = 2465$

$\delta' = 0.3$: min s.t. CS is accurate

method	*error*	*CPU*
CS	**0**	**29.67s**
JLL-CS	**2**	**17.13s**

These results are consistent over 3 samplings

Technique applies to all sparse retrieval problems

Outline

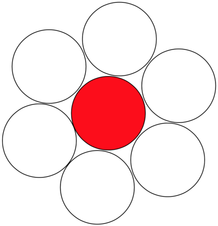
Definition

- *Optimization version.* Given $K \in \mathbb{N}$, determine the maximum number $\mathsf{kn}(K)$ of unit spheres that can be placed adjacent to a central unit sphere so their interiors do not overlap
- *Decision version.* Given $n, K \in \mathbb{N}$, is $\mathsf{kn}(K) \leq n$?
  in other words, determine whether n unit spheres can be placed adjacent to a central unit sphere so that their interiors do not overlap

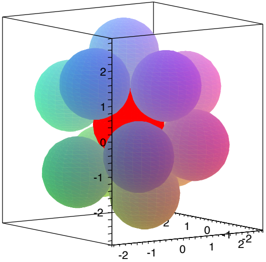
Funny story: *Newton and Gregory went down the pub...*

Some examples

$n = 6, K = 2$



$n = 12, K = 3$



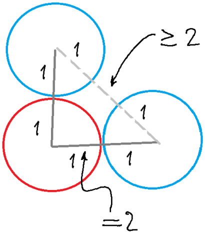
more dimensions

n	τ (lattice)	τ (nonlattice)
0	0	
1	2	
2	6	
3	12	
4	24	
5	40	
6	72	
7	126	
8	240	
9	272	(306)*
10	336	(500)*
11	438	(582)*
12	756	(840)*
13	918	(1130)*
14	1422	(1582)*
15	2340	
16	4320	
17	5346	
18	7398	
19	10668	
20	17400	
21	27720	
22	49896	

Radius formulation

Given $n, K \in \mathbb{N}$**, determine whether there exist** n **vectors** $x_1, \ldots, x_n \in \mathbb{R}^K$ **such that:**

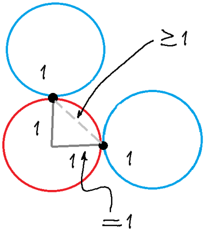
$$\begin{array}{rrcl} \forall i \le n & \|x_i\|_2^2 & = & 4 \\ \forall i < j \le n & \|x_i - x_j\|_2^2 & \ge & 4 \end{array}$$



Contact point formulation

Given $n, K \in \mathbb{N}$, determine whether there exist n vectors $x_1, \ldots, x_n \in \mathbb{R}^K$ such that:

$$\begin{array}{rrcl} \forall i \le n & \|x_i\|_2^2 & = & 1 \\ \forall i < j \le n & \|x_i - x_j\|_2^2 & \ge & 1 \end{array}$$



Spherical codes

- $S^{K-1} \subset \mathbb{R}^K$ **unit sphere centered at origin**
- *K-dimensional spherical z-code*:
  - (finite) subset $\mathcal{C} \subset S^{K-1}$
  - $\forall x \neq y \in \mathcal{C} \qquad x \cdot y \leq z$
- **non-overlapping interiors:**

$$\begin{aligned} \forall i < j \quad \|x_i - x_j\|_2^2 &\geq 1 \\ \Leftrightarrow \quad \|x_i\|_2^2 + \|x_j\|_2^2 - 2\, x_i \cdot x_j &\geq 1 \\ \Leftrightarrow \quad 1 + 1 - 2\, x_i \cdot x_j &\geq 1 \\ \Leftrightarrow \quad 2\, x_i \cdot x_j &\leq 1 \\ \Leftrightarrow \quad x_i \cdot x_j &\leq \frac{1}{2} = \cos\left(\frac{\pi}{3}\right) = z \end{aligned}$$

Subsection 1

Lower bounds

Lower bounds

- Construct spherical $\frac{1}{2}$-code $\mathcal{C}$ with $|\mathcal{C}|$ large
- Nonconvex NLP formulations
- SDP relaxations
- Combination of the two techniques

MINLP formulation

Maculan, Michelon, Smith 1995

Parameters:

- K: space dimension
- n: upper bound to $\mathsf{kn}(K)$

Variables:

- $x_i \in \mathbb{R}^K$: center of i-th vector
- $\alpha_i = 1$ iff vector i in configuration

$$\left.\begin{array}{rrcl}
\max & \sum\limits_{i=1}^{n} \alpha_i & & \\
\forall i \le n & \|x_i\|^2 & = & \alpha_i \\
\forall i < j \le n & \|x_i - x_j\|^2 & \ge & \alpha_i \alpha_j \\
\forall i \le n & x_i & \in & [-1,1]^K \\
\forall i \le n & \alpha_i & \in & \{0,1\}
\end{array}\right\}$$

Reformulating the binary products

- **Additional variables:** $\beta_{ij} = 1$ **iff vectors** i, j **in configuration**
- **Linearize** $\alpha_i \alpha_j$ **by** β_{ij}
- **Add constraints:**

$$\begin{array}{rrcl}
\forall i < j \leq n & \beta_{ij} & \leq & \alpha_i \\
\forall i < j \leq n & \beta_{ij} & \leq & \alpha_j \\
\forall i < j \leq n & \beta_{ij} & \geq & \alpha_i + \alpha_j - 1
\end{array}$$

Computational experiments

AMPL and Baron

- **Certifying YES**
  - $n = 6, K = 2$: **OK, 0.60s**
  - $n = 12, K = 3$: **OK, 0.07s**
  - $n = 24, K = 4$: **FAIL, CPU time limit (100s)**
- **Certifying NO**
  - $n = 7, K = 2$: **FAIL, CPU time limit (100s)**
  - $n = 13, K = 3$: **FAIL, CPU time limit (100s)**
  - $n = 25, K = 4$: **FAIL, CPU time limit (100s)**

Almost useless

Modelling the decision problem

$$\left.\begin{array}{rrcl}
\max\limits_{x,\alpha} & \alpha & & \\
\forall i \le n & \|x_i\|^2 & = & 1 \\
\forall i < j \le n & \|x_i - x_j\|^2 & \ge & \alpha \\
\forall i \le n & x_i & \in & [-1,1]^K \\
 & \alpha & \ge & 0
\end{array}\right\}$$

- **Feasible solution** (x^*, α^*)
- *KNP instance is YES iff* $\alpha^* \ge 1$

[Kucherenko, Belotti, Liberti, Maculan, *Discr. Appl. Math.* 2007]

Computational experiments

AMPL and Baron

- Certifying YES
  - $n = 6, K = 2$: FAIL, CPU time limit (100s)
  - $n = 12, K = 3$: FAIL, CPU time limit (100s)
  - $n = 24, K = 4$: FAIL, CPU time limit (100s)
- Certifying NO
  - $n = 7, K = 2$: FAIL, CPU time limit (100s)
  - $n = 13, K = 3$: FAIL, CPU time limit (100s)
  - $n = 25, K = 4$: FAIL, CPU time limit (100s)

Apparently even more useless

But more informative (arccos α = min. angular sep)

Certifying YES by $\alpha \geq 1$

- $n = 6, K = 2$: OK, 0.06s
- $n = 12, K = 3$: OK, 0.05s
- $n = 24, K = 4$: OK, 1.48s
- $n = 40, K = 5$: FAIL, CPU time limit (100s)

What about polar coordinates?

- $\forall i \le n \quad x_i = (x_{i1}, \ldots, x_{iK}) \mapsto (\vartheta_{i1}, \ldots, \vartheta_{i,K-1})$
- **Formulation**

$$\begin{array}{rrcl}
(\dagger) \quad \forall k \le K & \rho \sin \vartheta_{i,k-1} \prod_{h=k}^{K-1} \cos \vartheta_{ih} & = & x_{ik} \\
(\ddagger) \qquad \forall i < j \le n & \|x_i - x_j\|_2^2 & \ge & \rho^2 \\
\forall i \le n, k \le K & (\sin(\vartheta_{ik}))^2 + (\cos(\vartheta_{ik}))^2 & = & 1 \\
\textit{(optional)} & \rho & = & 1
\end{array}$$

- **Only need to decide $s_{ik} = \sin \vartheta_{ik}$ and $c_{ik} = \cos \vartheta_{ik}$**
- **Replace x in ($\ddagger$) using ($\dagger$): get polyprog in s, c**
- *Numerically more challenging to solve (polydeg $2K$)*
- **OPEN QUESTION: useful for bounds?**

Subsection 2

Upper bounds from SDP?

SDP relaxation of Euclidean distances

- **Linearization of scalar products**

$$\forall i,j \le n \qquad x_i \cdot x_j \longrightarrow X_{ij}$$

  where X is an $n \times n$ symmetric matrix
- $\|x_i\|_2^2 = x_i \cdot x_i = X_{ii}$
- $\|x_i - x_j\|_2^2 = \|x_i\|_2^2 + \|x_j\|_2^2 - 2x_i \cdot x_j = X_{ii} + X_{jj} - 2X_{ij}$
- $X = xx^\top \Rightarrow X - xx^\top = 0$ **makes linearization exact**
- **Relaxation:**

$$X - xx^\top \succeq 0 \Rightarrow \mathsf{Schur}(X,x) = \begin{pmatrix} I_K & x^\top \\ x & X \end{pmatrix} \succeq 0$$

SDP relaxation of binary constraints

- $\forall i \le n \qquad \alpha_i \in \{0,1\} \Leftrightarrow \alpha_i^2 = \alpha_i$
- **Let A be an $n \times n$ symmetric matrix**
- **Linearize $\alpha_i \alpha_j$ by A_{ij} (hence α_i^2 by A_{ii})**
- **$A = \alpha\alpha^\top$ makes linearization exact**
- **Relaxation:** $\mathsf{Schur}(A, \alpha) \succeq 0$

SDP relaxation of [MMS95]

$$\left.\begin{array}{rrcl}
\max & \sum_{i=1}^{n} \alpha_i & & \\
\forall i \le n & X_{ii} & = & \alpha_i \\
\forall i < j \le n & X_{ii} + X_{jj} - 2X_{ij} & \ge & A_{ij} \\
\forall i \le n & A_{ii} & = & \alpha_i \\
\forall i < j \le n & A_{ij} & \le & \alpha_j \\
\forall i < j \le n & A_{ij} & \le & \alpha_i \\
\forall i < j \le n & A_{ij} & \ge & \alpha_i + \alpha_j - 1 \\
 & \mathsf{Schur}(X, x) & \succeq & 0 \\
 & \mathsf{Schur}(A, \alpha) & \succeq & 0 \\
\forall i \le n & x_i & \in & [-1, 1]^K \\
 & \alpha & \in & [0, 1]^n \\
 & X & \in & [-1, 1]^{n^2} \\
 & A & \in & [0, 1]^{n^2}
\end{array}\right\}$$

Computational experiments

- Python, PICOS and Mosek
  or Octave and SDPT3
- bound always equal to n
- prominent failure :-(
- **Why?**
  - *can combine inequalities to remove A from SDP*

$$\begin{aligned} \forall i < j \; X_{ii} + X_{jj} - 2X_{ij} &\geq A_{ij} \geq \alpha_i + \alpha_i - 1 \\ \Rightarrow X_{ii} + X_{jj} - 2X_{ij} &\geq \alpha_i + \alpha_i - 1 \end{aligned}$$

  (then eliminate all constraints in A)
  - *integrality of α completely lost*

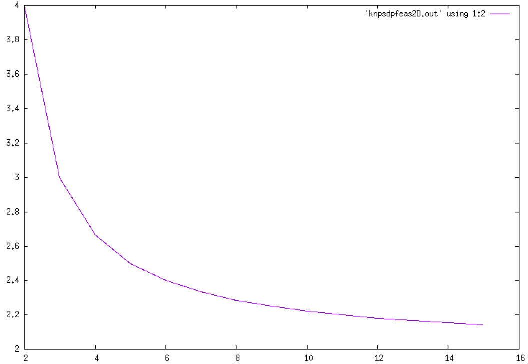
SDP relaxation of [KBLM07]

$$
\left.
\begin{array}{rrcl}
\max & \alpha & & \\
\forall i \le n & X_{ii} & = & 1 \\
\forall i < j \le n & X_{ii} + X_{jj} - 2X_{ij} & \ge & \alpha \\
 & X & \in & [-1,1]^{n^2} \\
 & X & \succeq & 0 \\
 & \alpha & \ge & 0
\end{array}
\right\}
$$

Computational experiments

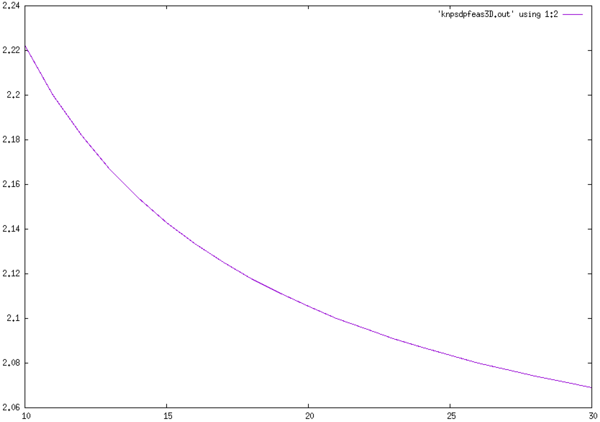
With $K = 2$

n	α^*
2	4.00
3	3.00
4	2.66
5	2.50
6	2.40
7	2.33
8	2.28
9	2.25
10	2.22
11	2.20
12	2.18
13	2.16
14	2.15
15	2.14



Computational experiments

With $K = 3$



Always $\longrightarrow 2?$

An SDP-based heuristic?

1. $X^* \in \mathbb{R}^{n^2}$: SDP relaxation solution of [KBLM07]
2. Perform PCA, get $\bar{x} \in \mathbb{R}^{nK}$
3. Local NLP solver on [KBLM07] with starting point $\bar{x}$

However...

The *Uselessness Theorem*

Thm.

1. The SDP relaxation of [KBLM07] is useless
2. In fact, it is *extremely* useless

1. Part 1: Uselessness
   - *Independent of K:*
     no useful bounds in function of K
2. Part 2: Extreme uselessness
   (a) *For all n, the bound is $\frac{2n}{n-1}$*
   (b) $\exists$ *opt. X^* with eigenvalues* $0, \frac{n}{n-1}, \ldots, \frac{n}{n-1}$

By 2(b), applying MDS/PCA makes no sense

Proof of extreme uselessness

Strategy:

- Pull a simple matrix solution out of a hat
- Write primal and dual SDP of [KBLM07]
- Show it is feasible in both
- Hence it is optimal
- Analyse solution:
  - all $n-1$ positive eigenvalues are equal
  - its objective function value is $2n/(n-1)$

Primal SDP

$\boxed{\forall 1 \leq i \leq j \leq n \textbf{ let } B_{ij} = (1_{ij}) \textbf{ and } 0 \textbf{ elsewhere}}$

quantifier	*natural form*	*standard form*	*dual var*
	$\max \alpha$	$\max \alpha$	
$\forall i \leq n$	$X_{ii} = 1$	$E_{ii} \bullet X = 1$	u_i
$\forall i < j \leq n$	$X_{ii} + X_{jj} - 2X_{ij} \geq \alpha$	$A_{ij} \bullet X + \alpha \leq 0$	w_{ij}
		$A_{ij} = -E_{ii} - E_{jj} + E_{ij} + E_{ji}$	
$\forall i < j \leq n$	$X_{ij} \leq 1$	$(E_{ij} + E_{ji}) \bullet X \leq 2$	y_{ij}
$\forall i < j \leq n$	$X_{ij} \geq -1$	$(-E_{ij} - E_{ji}) \bullet X \leq 2$	z_{ij}
	$X \succeq 0$	$X \succeq 0$	
	$\alpha \geq 0$	$\alpha \geq 0$	

Dual SDP

$$\min \sum_i u_i + 2\sum_{i<j}(y_{ij} + z_{ij})$$

$$\begin{aligned} \sum_i u_i E_{ii} + \sum_{i<j} \big((y_{ij} - z_{ij})(E_{ij} - E_{ji}) + w_{ij} A_{ij}\big) &\succeq 0 \\ \sum_{i<j} w_{ij} &\geq 1 \\ w, y, z \geq 0 & \end{aligned}$$

Simplify $|v| = y + z, v = y - z$:

$$\min \sum_i u_i + 2\sum_{i<j}|v_{ij}|$$

$$\begin{aligned} \sum_i u_i E_{ii} + \sum_{i<j} \big(v_{ij}(E_{ij} - E_{ji}) + w_{ij} A_{ij}\big) &\succeq 0 \\ \sum_{i<j} w_{ij} &\geq 1 \\ w, v \geq 0 & \end{aligned}$$

Pulling a solution out of a hat

$$
\begin{aligned}
\alpha^* &= \frac{2n}{n-1} \\
X^* &= \frac{n}{n-1} I_n - \frac{1}{n-1} \mathbf{1}_n \\
u^* &= \frac{2}{n-1} \\
w^* &= \frac{1}{n(n-1)} \\
v^* &= 0
\end{aligned}
$$

where $\mathbf{1}_n =$ *all-one* $n \times n$ *matrix*

Solution verification

- **linear constraints:** ***by inspection***
- $X \succeq 0$**:** ***eigenvalues of*** X^* ***are*** $0, \frac{n}{n-1}, \ldots, \frac{n}{n-1}$
- $\sum_i u_i E_{ii} + \sum_{i<j}(v_{ij}(E_{ij} - E_{ji}) + w_{ij} A_{ij}) \succeq 0$**:**

$$
\begin{aligned}
& \sum_i u_i^* E_{ii} + \sum_{i<j} w_{ij}^* A_{ij} \\
= \quad & \frac{2}{n-1} \sum_i E_{ii} + \frac{1}{n(n-1)} \sum_{i<j} A_{ij} \\
= \quad & \frac{2}{n-1} I_n + \frac{1}{n(n-1)} \big(-(n-1) I_n + (\mathbf{1}_n - I_n) \big) \\
= \quad & \frac{1}{n(n-1)} \mathbf{1}_n \succeq 0
\end{aligned}
$$

Corollary

$$\lim_{n\to\infty} \mathsf{v}(n, \textbf{[KBLM07]}) = \lim_{n\to\infty} \frac{2n}{n-1} = 2$$

as observed in computational experiments

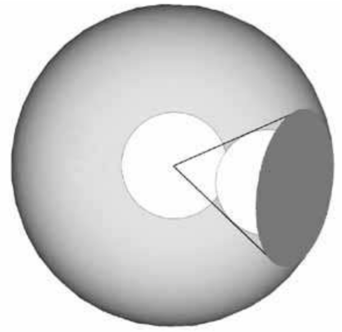
Subsection 3

Gregory's upper bound

Surface upper bound

Gregory 1694, Szpiro 2003

Consider a $\mathsf{kn}(3)$ configuration inscribed into a super-sphere of radius 3. Imagine a lamp at the centre of the central sphere that casts shadows of the surrounding balls onto the inside surface of the super-sphere. Each shadow has a surface area of 7.6; the total surface of the super-ball is 113.1. So $\frac{113.1}{7.6} = 14.9$ is an upper bound to $\mathsf{kn}(3)$.



At end of XVII century, yielded Newton/Gregory dispute

Subsection 4

Delsarte's upper bound

Pair distribution on sphere surface

- Spherical z-code $\mathcal{C}$ has $x_i \cdot x_j \leq z$ $(i < j \leq n = |\mathcal{C}|)$

$$\forall t \in [-1,1] \quad \sigma_t = \frac{1}{n}\Big|\{(i,j) \mid i,j \leq n \wedge x_i \cdot x_j = t\}\Big|$$

- t-**code: let** $\sigma_t = 0$ **for** $t \in (1/2, 1)$
- $|\mathcal{C}| = n < \infty$**: only finitely many** $\sigma_t \neq 0$

$$\begin{aligned}
\int_{[-1,1]} \sigma_t dt = \sum_{t \in [-1,1]} \sigma_t = \frac{1}{n}|\textbf{all pairs}| = \frac{n^2}{n} &= n \\
\sigma_1 = \frac{1}{n} n &= 1 \\
\forall t \in (1/2, 1) \quad \sigma_t &= 0 \\
\forall t \in [-1,1] \quad \sigma_t &\geq 0 \\
|\{\sigma_t > 0 \mid t \in [-1,1]\}| &< \infty
\end{aligned}$$

Growing Delsarte's LP

- **Decision variables:** σ_t, **for** $t \in [-1, 1]$
- **Objective function:**

$$\max |\mathcal{C}| = \max n \quad = \quad \max_{\sigma} \sum_{t\in[-1,1]} \sigma_t$$

$$= \sigma_1 + \max_{\sigma} \sum_{t\in[-1,1/2]} \sigma_t \quad = \quad 1 + \max_{\sigma} \sum_{t\in[-1,1/2]} \sigma_t$$

Note n not a parameter in this formulation

- **Constraints:**

$$\forall t \in [-1, 1/2] \quad \sigma_t \geq 0$$

- **LP unbounded!** — **need more constraints**

Gegenbauer cuts

- **Look for function family $\mathscr{F} : [-1,1] \to \mathbb{R}$ s.t.**

$$\forall \phi \in \mathscr{F} \qquad \sum_{t \in [-1,1/2]} \phi(t)\sigma_t \geq 0$$

- **Most popular $\mathscr{F}$:** *Gegenbauer polynomials G_h^K*
- Special case $G_h^K = P_h^{\gamma,\gamma}$ of *Jacobi polynomials* (where $\gamma = (K-2)/2$)

$$P_h^{\alpha,\beta} = \frac{1}{2^h} \sum_{i=0}^{h} \binom{h+\alpha}{i} \binom{h+\beta}{h-1} (t+1)^i (t-1)^{h-i}$$

- **Matlab knows them:** $G_h^K(t) = \mathtt{gegenbauerC}(h, (K-2)/2, t)$
- **Octave knows them:** $G_h^K(t) = \mathtt{gsl\_sf\_gegenpoly\_n}(h, \frac{K-2}{2}, t)$

  need command `pkg load gsl` *before function call*
- **They encode dependence on K**

Delsarte's LP

- **Primal:**

$$\left.\begin{array}{rrcl} 1+\max & \displaystyle\sum_{t\in[-1,\frac{1}{2}]} \sigma_t & & \\ \forall h\in H & \displaystyle\sum_{t\in[-1,\frac{1}{2}]} G_h^K(t)\sigma_t & \geq & -G_h^K(1) \\ \forall t\in[-1,\frac{1}{2}] & \sigma_t & \geq & 0. \end{array}\right\} \textbf{[DelP]}$$

- **Dual:**

$$\left.\begin{array}{rrcl} 1+\min & \displaystyle\sum_{h\in H}(-G_h^K(1))d_h & & \\ \forall t\in[-1,\frac{1}{2}] & \displaystyle\sum_{h\in H} G_h^K(t)d_h & \geq & 1 \\ \forall h\in H & d_h & \leq & 0. \end{array}\right\} \textbf{[DelD]}$$

Delsarte's theorem

- [Delsarte et al., 1977]

Theorem

Let $d_0 > 0$ *and* $F : [-1, 1] \to \mathbb{R}$ *such that:*

(i) $\exists H \subseteq (\mathbb{N} \cup \{0\})$ *and* $d \in \mathbb{R}_+^{|H|} \geq 0$
s.t. $F(t) = \sum_{h \in H} d_h G_h^K(t)$

(ii) $\forall t \in [-1, z]\ F(t) \leq 0$

Then $kn(K) \leq \frac{F(1)}{d_0}$

- *Proof based on properties of Gegenbauer polynomials*
- **Best upper bound:** $\min F(1)/d_0 \Rightarrow \min_{d_0=1} F(1) \Rightarrow$ **[DelD]**
- **[DelD] "models" Delsarte's theorem**

Delsarte's normalized LP ($G_h^K(1) = 1$)

- **Primal:**

$$\left.\begin{array}{rrcl} 1 + \max & \displaystyle\sum_{t \in [-1, \frac{1}{2}]} \sigma_t & & \\ \forall h \in H & \displaystyle\sum_{t \in [-1, \frac{1}{2}]} G_h^K(t) \sigma_t & \geq & -1 \\ \forall t \in [-1, \frac{1}{2}] & \sigma_t & \geq & 0 \end{array}\right\} \textbf{[DelP]}$$

- **Dual:**

$$\left.\begin{array}{rrcl} 1 + \min & \displaystyle\sum_{h \in H} (-1) d_h & & \\ \forall t \in [-1, \frac{1}{2}] & \displaystyle\sum_{h \in H} G_h^K(t) d_h & \geq & 1 \\ \forall h \in H & d_h & \leq & 0 \end{array}\right\} \textbf{[DelD]}$$

- $d_0 = 1 \Rightarrow$ *remove* 0 *from* H

Focus on normalized [DelD]

Rewrite $-d_h$ ***as*** d_h**:**

$$\left.\begin{array}{rrcl} 1 + \min & \sum_{h\in H} d_h & & \\ \forall t \in [-1, \frac{1}{2}] & \sum_{h\in H} G_h^K(t) d_h & \leq & -1 \\ \forall h \in H & d_h & \geq & 0 \end{array}\right\} \textbf{[DelD]}$$

Issue: ***semi-infinite LP*** **(SILP) (how do we solve it?)**

Approximate SILP solution

- *Only keep finitely many constraints*
- **Discretize $[-1,1]$ with a finite $T \subset [-1,1]$**
- *Obtain <u>relaxation</u>* $[\mathbf{DelD}]_T$*:*

$$\mathsf{val}([\mathbf{DelD}]_T) \leq \mathsf{val}([\mathbf{DelD}])$$

- **Risk:** $\mathsf{val}([\mathbf{DelD}]_T) < \min F(1)/d_0$
  not a valid bound to $\mathsf{kn}(K)$
- **Happens if soln. of $[\mathbf{DelD}]_T$ infeasible in $[\mathbf{DelD}]$**
  i.e. infeasible w.r.t. some of the ∞ly many removed constraints

SILP feasibility

- **Given SILP** $\bar{S} \equiv \min\{c^\top x \mid \forall i \in \bar{I}\ a_i^\top x \leq b_i\}$
- **Relax to LP** $S \equiv \min\{c^\top x \mid \forall i \in I\ a_i^\top x \leq b_i\}$**, where** $I \subsetneq \bar{I}$
- **Solve** S**, get solution** x^*
- **Let** $\epsilon = \max\{a_i^\top x^* - b_i \mid i \in \bar{I}\}$
  $\boxed{\textit{continuous optimization w.r.t. single var. } i}$
- **If** $\epsilon \leq 0$ **then** x^* **feasible in** $\bar{S}$
  $\Rightarrow \mathsf{val}(\bar{S}) \leq c^\top x^*$
- **If** $\epsilon > 0$ **refine** S **and repeat**
- **Apply to [DelD]**$_T$**, get solution** d^* **feasible in [DelD]**

[DelD] feasibility

1. **Choose discretization T of $[-1, 1/2]$**
2. **Solve**

$$\left.\begin{array}{rrcl} 1+\min & \sum_{h\in H} d_h & & \\ \forall t\in T & \sum_{h\in H} G_h^K(t) d_h & \le & -1 \\ \forall h\in H & d_h & \ge & 0 \end{array}\right\} \textbf{[DelD]}_T$$

   get solution d^*

3. **Solve** $\epsilon = \max\{1 + \sum_{h\in H} G_h^K(t) d_h \mid t\in[-1,1/2]\}$
4. **If $\epsilon \le 0$ then d^* feasible in [DelD]**

$$\boxed{\Rightarrow \mathsf{kn}(K) \le 1 + \textstyle\sum_{h\in H} d_h^*}$$

5. *Else refine T and repeat from Step 2*

Subsection 5

Pfender's upper bound

Pfender's upper bound theorem

Thm.

Let $\mathcal{C}_z = \{x_i \in \mathbb{S}^{K-1} \mid i \leq n \wedge \forall j \neq i \; (x_i \cdot x_j \leq z)\}$**;** $c_0 > 0$**;** $f : [-1,1] \to \mathbb{R}$ **s.t.:**
(i) $\sum_{i,j \leq n} f(x_i \cdot x_j) \geq 0$ **(ii)** $f(t) + c_0 \leq 0$ **for** $t \in [-1, z]$ **(iii)** $f(1) + c_0 \leq 1$

Then $n \leq \frac{1}{c_0}$

([Pfender 2006])

Let $g(t) = f(t) + c_0$

$$
\begin{aligned}
n^2 c_0 &\leq n^2 c_0 + \sum_{i,j \leq n} f(x_i \cdot x_j) && \textbf{by (i)} \\
&= \sum_{i,j \leq n} (f(x_i \cdot x_j) + c_0) = \sum_{i,j \leq n} g(x_i \cdot x_j) \\
&\leq \sum_{i \leq n} g(x_i \cdot x_i) && \textbf{since } g(t) \leq 0 \textbf{ for } t \leq z \textbf{ and } x_i \in \mathcal{C}_z \textbf{ for } i \leq n \\
&= n g(1) && \textbf{since } \|x_i\|_2 = 1 \textbf{ for } i \leq n \\
&\leq n && \textbf{since } g(1) \leq 1.
\end{aligned}
$$

Pfender's LP

- Condition (i) of Theorem valid for conic combinations of suitable functions $\mathcal{F}$:

$$f(t) = \sum_{h \in H} c_h f_h(t) \quad \text{for some } c_h \geq 0,$$

e.g. $\mathcal{F}$ = Gegenbauer polynomials (again)

- Get SILP

$$\left.\begin{array}{rrcll}
\max\limits_{c \in \mathbb{R}^{|H|}} & c_0 & & & \text{(minimize } 1/c_0 \geq n\text{)} \\
\forall\, t \in [-1, z] & \sum\limits_{h \in H} c_h G_h^K(t) + c_0 & \leq & 0 & \text{(ii)} \\
& \sum\limits_{h \in H} c_h G_h^K(1) + c_0 & \leq & 1 & \text{(iii)} \\
\forall\, h \in H & c_h & \geq & 0 & \text{(conic comb.)}
\end{array}\right\}$$

- *Discretize $[-1, z]$ by finite T, solve LP, check validity (again)*

Delsarte's and Pfender's theorem compared

- Delsarte & Pfender's theorem look similar:

Delsarte	**Pfender**
(i) $F(t)$ **G. poly comb**	**(i)** $f(t)$ **G. poly comb**
(ii) $\forall t \in [-1, z]\ F(t) \leq 0$	**(ii)** $\forall t \in [-1, z]\ f(t) + c_0 \leq 0$
	(iii) $f(1) + c_0 \leq 1$
$\Rightarrow \mathsf{kn}(K) \leq \frac{F(1)}{d_0}$	$\Rightarrow \mathsf{kn}(K) \leq \frac{1}{c_0}$

- **Try setting** $F(t) = f(t) + c_0$: **condition (ii) is the same**
- By condition (iii) in Pfender's theorem

$$\mathsf{kn}(K) \leq \frac{F(1)}{d_0} = \frac{f(1) + c_0}{c_0} \leq \frac{1}{c_0}$$

$\Rightarrow$ *Delsarte bound at least as tight as Pfender's*

- **Delsarte (i)** $\Rightarrow \int_{[-1,1]} F(t)dt \geq 0 \Rightarrow \int_{[-1,1]} (f(t) + c_0)dt \geq 0$
  Pfender (i) $\Rightarrow \int_{[-1,1]} f(t)dt \geq 0$ **more stringent**
- *Delsarte requires weaker condition and yields tighter bound*

Conditioned on $F(t) = f(t) + c_0$, not a proof! Verify computationally

The final, easy improvement

- However you compute your upper bound B:
- The number of surrounding balls is *integer*
- If $\mathsf{kn}(K) \leq B$, then in fact $\mathsf{kn}(K) \leq \lfloor B \rfloor$

THE END