Advanced Mathematical Optimization

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

INF580

About the course

- **Aims of lectures:** theory and algorithms
  won't repeat much of MAP557
- **Aims of TD: modelling abilities in practice**
  with AMPL, Python and perhaps Julia
- **Warning:**

  some disconnection between lectures and TD is normal
- **Exam:** I prefer project (max 2 people) or oral exam
  issue with timeslot: I am not free the week 190318-

`http://www.lix.polytechnique.fr/~liberti/teaching/dix/inf580-19`

Outline

Outline

What is *Mathematical Optimization*?

- Mathematics of solving optimization problems
- Formal language: Mathematical Programming (MP)
- Sentences: descriptions of optimization problems
- Interpreted by solution algorithms (“solvers”)
- As expressive as any imperative language
- Shifts focus from *algorithmics* to *modelling*

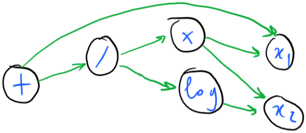
MP Formulations

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

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

- **More general than it looks:**
  - $\phi(x) = 0 \quad \Leftrightarrow \quad (\phi(x) \le 0 \wedge -\phi(x) \le 0)$
  - $L \le x \le U \quad \Leftrightarrow \quad (L - x \le 0 \wedge x - U \le 0)$
- f, g_i **represented by *expression DAGs***

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



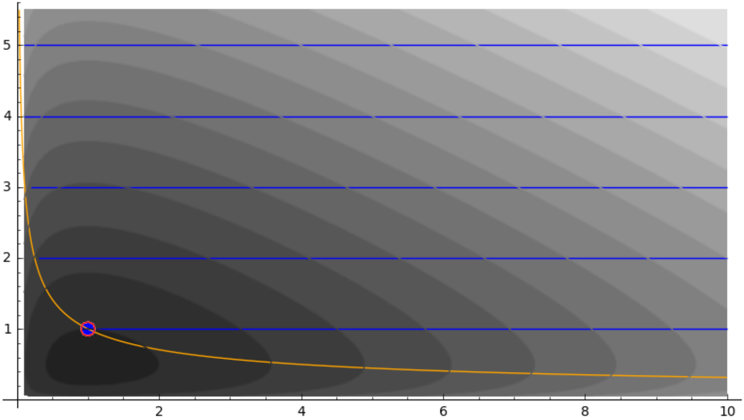
Class of all formulations P: $\mathbb{MP}$

Semantics of MP formulations

- $\llbracket P \rrbracket$ = optimum (or optima) of P
- Given $P \in \mathbb{MP}$, there are *three possibilities*:
  $\llbracket P \rrbracket$ exists, P is unbounded, P is infeasible
- P *is feasible* iff $\llbracket P \rrbracket$ exists or is unbounded
  otherwise it is infeasible
- P *has an optimum* iff $\llbracket P \rrbracket$ exists
  otherwise it is infeasible or unbounded

Example

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



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

Are feasibility and optimality really different?

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

Bisection algorithm

- $P \equiv \min\{f(x) \mid \forall i \in I \; g(x) \leq 0 \wedge x \in X\}$
- **Assume global optimum of P is between given lower/upper bounds**
- **Reformulate P to a parametrized feasibility problem** $Q(f_0) = \{x \in X \mid f(x) \leq f_0 \wedge \forall i \in I \; g(x) \leq 0\}$

Bisection algorithm

while lower and upper bounds differ by $> \epsilon$ **do**
    let f_0 be midway between bounds
    if $Q(f_0)$ is feasible **then**
        update *upper* bound to f_0
    else
        update *lower* bound to f_0
    end if
end while

Bisection algorithm for MP

initialize candidate global optimum $\hat{x}$
while lower and upper bounds differ by $> \epsilon$ **do**
  let f_0 be midway between bounds
  if $Q(f_0)$ is feasible **then**
    find a feasible point x'
    if x' improves $\hat{x}$ **then**
      update $\hat{x}$ to x'
      update *upper* bound to $f(\hat{x})$
    end if
  else
    update *lower* bound to f_0
  end if
end while

Bisection algorithm for MP (formal)

Given:

- global optimal value approximation tolerance $\epsilon > 0$
- lower bound $\underline{f}$, upper bound $\bar{f}$
- an algorithm $\mathcal{A}$ which
  finds an element in a set or certifies emptyness

Bisection algorithm for MP (formal)

let $(\hat{x}, \hat{f}) = (\mathsf{uninitialized}, \bar{f})$
while $\bar{f} - \underline{f} > \epsilon$ **do**
  let $f_0 = (\underline{f} + \bar{f})/2$
  if $Q(f_0) \neq \varnothing$ **then**
    $(x', f') = \mathcal{A}(Q)$
    if $f' < \hat{f}$ **then**
      update $(\hat{x}, \hat{f}) \leftarrow (x', f')$
      update $\bar{f} \leftarrow \hat{f}$
    end if
  else
    update $\underline{f} \leftarrow f_0$
  end if
end while

Subsection 1

MP language

Entities of a MP formulation

- Sets of indices
- Parameters
  problem input, or *instance*
- Decision variables
  will encode the solution after solver execution
- Objective function
- Constraints

Example

Linear Program (LP) in standard form

- $I = \{1, \ldots, n\}$: **row indices**
  $J = \{1, \ldots, n\}$: **col. indices**
- $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, A **an** $m \times n$ **matrix**
- $x \in \mathbb{R}^n$
- $\min_x c^\top x$
- $Ax = b \quad \wedge \quad x \geq 0$

MP language implementations

- Humans model with quantifiers ($\forall$, $\sum$,...)
  e.g. $\forall i \in I \ \sum_{j \in J} a_{ij} x_j \leq b_i$
- Solvers accept lists of explicit constraints
  e.g. $4x_1 + 1.5x_2 + x_6 \leq 2$
- **Translation from structured to flat formulation**
- MP language implementations
  AMPL, GAMS, Matlab+YALMIP,
  Python+PyOMO/cvx, Julia+JuMP, ...

AMPL

- AMPL = A Mathematical Programming Language
- Syntax similar to human notation
- Implementation sometimes somewhat buggy
- Commercial & closed-source
  - extremely rapid prototyping
  - we get free licenses for this course
  - free open-source AMPL sub-dialect in GLPK `glpsol`
- Can also use Python+PyOMO, or Julia+JuMP

Subsection 2

Solvers

Solvers

- **Solver:**
  a solution algorithm for a whole subclass of MP
- **Take formulation P as input**
- **Output** $[\![P]\!]$ *and possibly other information*
- **Trade-off between generality and efficiency**

Some subclasses of MP

(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)**

And their solvers

(i) Linear Programming (LP)
simplex algorithm, interior point method (IPM)
Implementations: CPLEX, GLPK, CLP

(ii) Mixed-Integer LP (MILP)
cutting plane alg., Branch-and-Bound (BB)
Implementations: CPLEX, GuRoBi

(iii) Nonlinear Programming (NLP)
IPM, gradient descent (cNLP), spatial BB (sBB)
Implementations: IPOPT (cNLP), Baron, Couenne

(iv) Mixed-Integer NLP (MINLP)
outer approximation (cMINLP), sBB
Implementations: Bonmin (cMINLP), Baron, Couenne

Subsection 3

MP systematics

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 $\{0, 1\}^n$)

Some more "exotic" classes:

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

Subsection 4

Some applications

Some application fields

- Production industry
  planning, scheduling, allocation, ...
- Transportation & logistics
  facility location, routing, rostering, ...
- Service industry
  pricing, strategy, product placement, ...
- Energy industry
  power flow optimization, monitoring smart grids,...
- Machine Learning & Artificial Intelligence
  clustering, approximation error minimization
- Biochemistry & medicine
  protein structure, blending, tomography, ...
- Mathematics
  Kissing number, packing of geometrical objects,...

Easy example

A bank needs to invest C gazillion dollars, and focuses on two types of investments: one, imaginatively called (a), guarantees a 15% return, while the other, riskier and called, surprise surprise, (b), is set to a 25%. At least one fourth of the budget C must be invested in (a), and the quantity invested in (b) cannot be more than double the quantity invested in (a). How do we choose how much to invest in (a) and (b) so that revenue is maximized?

Easy example

- **Parameters:**
  - budget C
  - return on investment on (a): 15%, on (b): 25%
- **Decision variables:**
  - $x_a =$ budget invested in (a)
  - $x_b =$ budget invested in (b)
- **Objective function:** $1.15\, x_a + 1.25\, x_b$
- **Constraints:**
  - $x_a + x_b = C$
  - $x_a \geq C/4$
  - $x_b \leq 2x_a$

Easy example: remarks

- *Missing trivial constraints*:
  verify that $x_a = C + 1, x_b = -1$ satisfies constraints
  forgot $x \geq 0$
- *No numbers in formulations*:
  replace numbers by parameter symbols

$$\left.\begin{array}{rcl} \max\limits_{x_a, x_b \geq 0} & c_a x_a + c_b x_b & \\ x_a + x_b & = & C \\ x_a & \geq & pC \\ d x_a - x_b & \geq & 0 \end{array}\right\}$$

- *Formulation generality*:
  extend to n investments:

$$\left.\begin{array}{rcl} \max\limits_{x \geq 0} & \sum\limits_{j \leq n} c_j x_j & \\ \sum\limits_{j \leq n} x_j & = & C \\ x_1 & \geq & pC \\ d x_1 - x_2 & \geq & 0 \end{array}\right\}$$

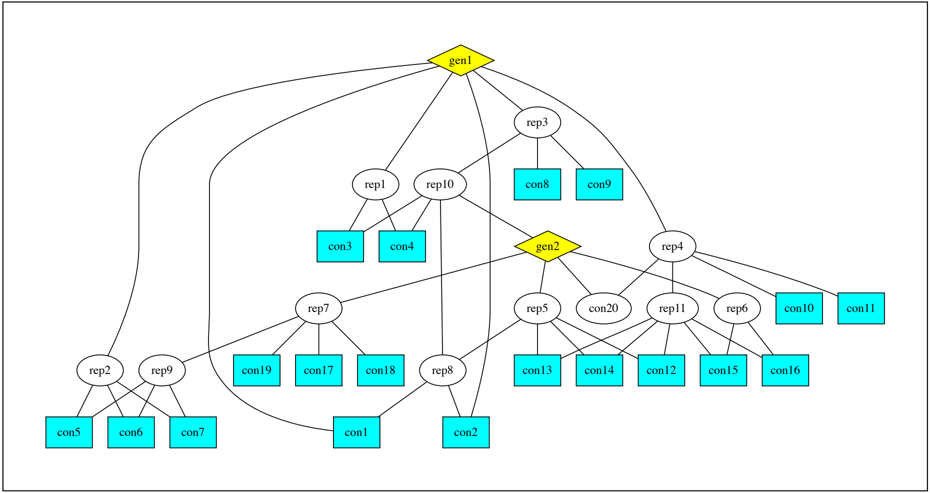
Example: monitoring an electrical grid

An electricity distribution company wants to monitor certain quantities at the lines of its grid by placing measuring devices at the buses. There are three types of buses: consumer, generator, and repeater. There are five types of devices:

- A: installed at any bus, and monitors all incident lines (cost: 0.9MEUR)
- B: installed at consumer and repeater buses, and monitors at most two incident lines (cost: 0.5MEUR)
- C: installed at generator buses only, and monitors at most one incident line (cost: 0.3MEUR)
- D: installed at repeater buses only, and monitors at most one incident line (cost: 0.2MEUR)
- E: installed at consumer buses only, and monitors at most one incident line (cost: 0.3MEUR).

Provide a least-cost installation plan for the devices at the buses, so that all lines are monitored by at least one device.

Example: the electrical grid



Example: formulation

- **Index sets:**
  - V: set of buses v
  - E: set of lines $\{u, v\}$
  - A: set of *directed* lines (u, v)
  - $\forall u \in V$ let $N_u =$ buses adjacent to u
  - D: set of device types
  - D_M: device types covering > 1 line
  - $D_1 = D \smallsetminus D_M$
- **Parameters**:
  - $\forall v \in V \quad b_v =$ bus type
  - $\forall d \in D \quad c_d =$ device cost

Example: formulation

- Decision variables
  - $\forall d \in D, v \in V \quad x_{dv} = 1$
    iff device type d installed at bus v
  - $\forall d \in D, (u,v) \in A \quad y_{duv} = 1$
    iff device type d installed at bus u measures line $\{u,v\}$
  - all variables are binary
- Objective function

$$\min_{x,y} \sum_{d \in D} c_d \sum_{v \in V} x_{dv}$$

Example: formulation

- **Constraints**
  - **device types:**

$$\begin{aligned}
\forall v \in V \quad b_v = \mathsf{gen} \quad &\rightarrow \quad x_{\mathsf{B}v} = 0 \\
\forall v \in V \quad b_v \in \{\mathsf{con}, \mathsf{rep}\} \quad &\rightarrow \quad x_{\mathsf{C}v} = 0 \\
\forall v \in V \quad b_v \in \{\mathsf{gen}, \mathsf{con}\} \quad &\rightarrow \quad x_{\mathsf{D}v} = 0 \\
\forall v \in V \quad b_v \in \{\mathsf{gen}, \mathsf{rep}\} \quad &\rightarrow \quad x_{\mathsf{E}v} = 0
\end{aligned}$$

  - **at most one device type at each bus**

$$\forall v \in V \quad \sum_{d \in D} x_{dv} \leq 1$$

Example: formulation

- **Constraints**
  - A: every line incident to installed device is monitored

$$\forall u \in V, v \in N_u \quad y_{\mathsf{A}uv} = x_{\mathsf{A}u}$$

  - B: two monitored lines incident to installed device

$$\forall u \in V \quad \sum_{v \in N_u} y_{\mathsf{B}uv} = 2x_{\mathsf{B}u}$$

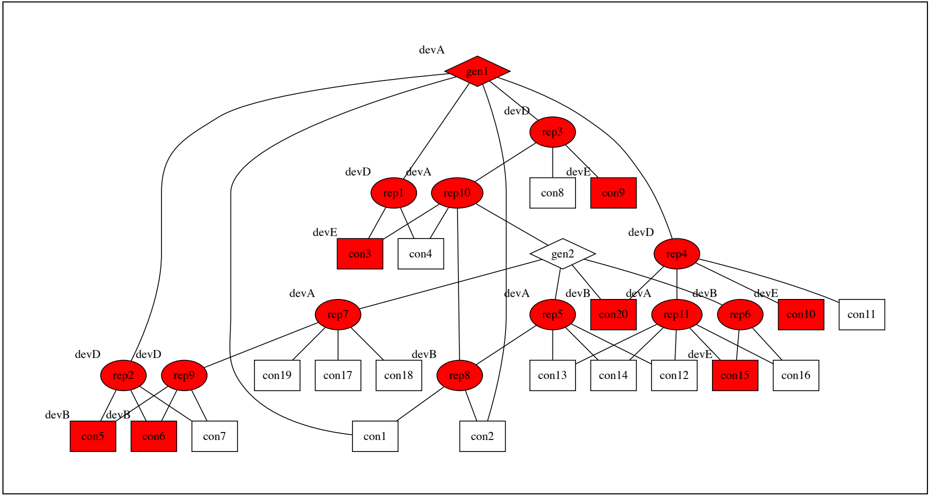
  - C,D,E: one monitored line incident to installed device

$$\forall d \in D_1, u \in V \quad \sum_{v \in N_u} y_{duv} = x_{du}$$

  - line is monitored

$$\forall \{u, v\} \in E \quad \sum_{d \in D} y_{duv} + \sum_{e \in D} y_{evu} \geq 1$$

Example: solution



all lines monitored, no redundancy, cost 9.2MEUR

Outline

Can we solve MPs?

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

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

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

Subsection 1

Formal systems

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

[Smullyan, Th. of Formal Systems, 1961]

Example: PA1

- **Theory:** 1st order provable sentences about $\mathbb{N}$
- **Alphabet:** $+, \times, \wedge, \vee, \to, \forall, \exists, \neg, =, S(\cdot)$ 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* $(P \wedge (P \to Q)) \to Q$
- **Generates ring $(\mathbb{N}, +, \times)$ and arithmetical proofs**
  e.g. $\exists x \in \mathbb{N}^n\ \forall i\ (p_i(x) \leq 0)$ (polynomial MINLP feasibility)

Example: Reals

- **Theory:** **1st order provable sentences about $\mathbb{R}$**
- **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*** $(P \wedge (P \to Q)) \to Q$
- **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)**

Relevance of FSs to MP

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 $\exists x \in \mathbb{Z}^n \ \forall i \ p_i(x) \le 0$ (poly p)
- Reals: decide if sentence f about $\mathbb{R}$ has a proof
  Reals contains $\exists x \in \mathbb{R}^n \ \forall i \ p_i(x) \le 0$ (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 any $f \in \mathcal{F}$**, either** f **or** $\neg f$ **is provable**
  otherwise $\mathcal{F}$ **is** ***incomplete***
- *Decidable* **FS** $\mathcal{F}$:
  $\exists$ **algorithm** $\mathcal{D}$ **s.t.**

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

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

Example: PA1

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

Subsection 2

Gödel

Gödel's 1st incompleteness theorem

- $\mathcal{F}$: any FS extending PA1
- **Thm.** $\boxed{\mathcal{F} \text{ complete iff inconsistent}}$
- ϕ: sentence "ϕ not provable in $\mathcal{F}$"
  denoted $\mathcal{F} \nvdash \phi$; it can be constructed <u>in $\mathcal{F}$</u>; hard part of thm.
  - $\vdash$: "is provable" in PA1; $\vdash$: in meta-language
  - *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: $P \wedge \neg P$, hence P and $\neg P$, so for any Q we have $P \vee Q$,
    whence Q (since $\neg P$ and $P \vee Q$), implying $P \wedge \neg P \to Q$
- If we want PA1 to be consistent, it must be incomplete
- **Warning:** $\mathcal{F} \nvdash \phi \equiv \neg(\mathcal{F}\vdash\phi) \not\equiv \mathcal{F}\vdash\neg\phi$

Gödel's encoding

- For $\psi \in \mathsf{PA1}$, $\ulcorner \psi \urcorner \in \mathbb{N}$
  integer which encodes the proof
  let me sweep the details under the carpet
- $\ulcorner \cdot \urcorner$ is an injective map
- Inverse: $\langle \ulcorner \phi \urcorner \rangle = \phi$
  ϕ is the sentence corresponding to Gödel's number $\ulcorner \phi \urcorner$
- Encode/decode in $\mathbb{N}$ any sentence of a formal system

Gödel's self-referential sentence ϕ

- For integers x, y $\boxed{\exists g \in \mathbb{N}\ \langle g \rangle \equiv}$ $\mathsf{proof}(x, y)$:
  holds if $\langle x \rangle$ is a proof in PA1 for the sentence $\langle y \rangle$
- For integers m, n, p $\boxed{\exists g \in \mathbb{N}\ \langle g \rangle \equiv}$ $\mathsf{sost}(m, n, p) =$
  encoding in $\mathbb{N}$ of the sentence obtained by replacing in $\langle m \rangle$ the (typographical sign of the) free variable symbol $\langle n \rangle$ with the integer p
- let $\mathbf{y}$ be the encoding of the (typographical sign of the) variable symbol 'y' (remark: $\mathbf{y} = \ulcorner `y' \urcorner \in \mathbb{N}$)
- $\gamma(y) \equiv \neg\exists x \in \mathbb{N}\ \mathsf{proof}(x, \mathsf{sost}(y, \mathbf{y}, y))$:
  there is no proof in PA1 for the sentence obtained from replacing, in the sentence $\langle y \rangle$, every free variable symbol 'y' with the integer y
- let $q = \ulcorner \gamma(y) \urcorner$, replace y with q in $\gamma(y)$, get $\phi \equiv \gamma(q)$
  so $\phi \equiv \neg\exists x \in \mathbb{N}\ \mathsf{proof}(x, \mathsf{sost}(q, \mathbf{y}, q))$

Gödel's self-referential sentence ϕ

$$\boxed{\phi \equiv \neg\exists x \in \mathbb{N}\ \mathsf{proof}(x, \mathsf{sost}(q, \mathbf{y}, q))}$$

- **Let** $\psi \equiv \mathsf{sost}(q, \mathbf{y}, q)$
  ϕ **states: "there is no proof in** PA1 **for the sentence** ψ**"**
  ψ defined by replacing the free variable symbol 'y' in $\langle q \rangle$ with q
- **How did we obtain** ϕ**?**
  ϕ obtained by replacing the free variable y in $\gamma(y)$ with q,
  i.e. *$\phi \equiv \gamma(q)$*
- **Recall:** $q = \ulcorner \gamma(y) \urcorner$, **i.e.** $\langle q \rangle \equiv \gamma(y)$
- **So** $\psi \equiv \phi$
- **Hence** ϕ **states "**ϕ **is not provable in** PA1**"**

Subsection 3

Turing

Turing machines

- 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, change to state t)
  - initial tape content: input, final tape content: output
  - final state $\bot$: termination; $\varnothing$ nonterm
- $\exists$ universal TM (UTM) U s.t.
  - given the "program" of a TM T and an input x
  - U "simulates" T running on x
- $\Rightarrow$ The basis of the modern computer
- HALTING PROBLEM (HP): *does a given M terminate on input x?*

  $$\boxed{\text{Given TM } M \text{ \& input } x\text{, is } M(x) = \bot\text{?}}$$
- Turing's theorem: HP is undecidable

Turing's proof (informal)

- Suppose $\exists$ TM "halt" s.t.
  $\mathsf{halt}(T, x) = 1$ if $T(x)$ terminates, 0 othw
- Then construct function $G(x)$ as follows:
  `if` $\mathsf{halt}(G, x) = 1$ `then loop forever else stop`
- If $G(x)$ terminates then $\mathsf{halt}(G, x) = 0$, contradiction
- If $G(x)$ loops forever then $\mathsf{halt}(G, x) = 1$, contradiction
- $\Rightarrow$ TM halt cannot exist

Turing's proof (formal)

- **Enumerate all TMs:** $(M_i \mid i \in \mathbb{N})$
- **Halting function** $\mathsf{halt}(i, \ell) = \begin{cases} 1 & \text{if } M_i(\ell) = \perp \\ 0 & \text{if } M_i(\ell) = \varnothing \end{cases}$
- **Show** $\mathsf{halt} \neq F$ **for any total computable** $F(i, \ell)$**:**
  - **let** $G(i) = 0$ **if** $F(i, i) = 0$ **or undefined (**$\varnothing$**) othw**

    G is partial computable because F is computable
  - **let** M_j **be the TM computing** G

    for any i, $M_j(i) = \perp$ iff $G(i) = 0$
  - **consider** $\mathsf{halt}(j, j)$**:**
    - $\mathsf{halt}(j, j) = 1 \to M_j(j) = \perp \to G(j) = 0 \to F(j, j) = 0$
    - $\mathsf{halt}(j, j) = 0 \to M_j(j) = \varnothing \to G(j) = \varnothing \to F(j, j) \neq 0$
  - **so** $\mathsf{halt}(j, j) \neq F(j, j)$ **for all** j
- halt **is uncomputable**

Turing and Gödel

- **TM** provable **with input** $\alpha \in$ PA1**:**
  `while(1) i=0; if` $\ulcorner\alpha\urcorner$ `== i return` YES`; else i=i+1`
  provable$(\alpha) = \perp$ **iff** PA1 $\vdash \alpha$
- **termination of** provable $\Leftrightarrow$ **decidability in** PA1
- **Gödel's** ϕ **is not provable** $\Rightarrow$ PA1 **is undecidable**

PA1 **incomplete and undecidable**

Subsection 4

Tarski

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 and also decidable**
  since every complete theory is decidable (why?)

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 w/clauses $\exists x_i \; q_i(x)$ 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!

Subsection 5

Completeness and incompleteness

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 incomplete and decidable

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: $\mathsf{ACF} \cup \mathsf{C}_p \equiv [\sum_{j \leq p} 1 = 0]$ (with 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 incomplete and decidable

Complete and undecidable (impossible)

- FS $\mathcal{F}$ complete:
  $\forall\psi \in \mathcal{F}\ \exists$ proof of ψ or $\neg\psi$
- Recall proofs are finite sequences of sentences
- Algorithm $\mathcal{D}(\psi)$:
  1. iteratively generate all (countably many) proofs
     combine axioms w/inference rules and repeat
  2. for each proof, is last sentence $\equiv \psi$ or $\equiv \neg\psi$?
     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. Gödel's completeness theorem
     - A: set of sentences in $\mathcal{F}$
     - M a *model* of $\mathcal{F}$ (domain of var symbols)
     - If $\exists M$ s.t. A^M is true, then A consistent
     - If A consistent, then $\exists M$ s.t. A^M is true
- *Pay attention when reading literature/websites*

Subsection 6

MP solvability

Polynomial equations in integers

- **Consider the feasibility-only MP**

$$\min\{0 \mid \forall i \le m\ g_i(x) = 0 \wedge x \in \mathbb{Z}^n\}$$

  with $g_i(x)$ composed by arithmetical expressions $(+, -, \times, \div)$

- **Rewrite as a *Diophantine equation* (DE):**

$$\exists x \in \mathbb{Z}^n \quad \sum_{i \le m} (g_i(x))^2 = 0 \qquad (1)$$

- **Can restrict to $\mathbb{N}$ wlog, i.e. Eq. (1) $\in$ PA1**
  write $x_i = x_i^+ - x_i^-$ where $x_i^+, x_i^- \in \mathbb{N}^n$
- **Formulæ of PA1 are generally undecidable**
  but is the subclass (1) of PA1 decidable or not?

Hilbert's 10th problem

- **Hilbert:**

  Given a Diophantine equation with any number of unknowns and with rational integer coefficients: devise a process which could determine by a finite number of operations whether the equation is solvable in rational integers

- **Davis & Putnam:** *conjecture DEs are undecidable*
  - consider set $\mathbb{RE}$ of *recursively enumerable* (r.e.) sets
  - $R \subseteq \mathbb{N}$ is in $\mathbb{RE}$ if $\exists$ TM listing all and only elements in R
  - some $\mathbb{RE}$ sets are undecidable, e.g. $R = \{\ulcorner \phi \urcorner \mid \mathsf{PA1} \vdash \phi\}$

    r.e.: list all proofs; undecidable: by Gödel's thm
  - for each $R \in \mathbb{RE}$ show $\exists$ polynomial $p(r, x)$ s.t.
    $$r \in R \leftrightarrow \exists x \in \mathbb{N}^n \; p(r, x) = 0$$
  - if can prove it, $\exists$ undecidable DEs

Proof strategy

- Strategy: model recursive functions using polynomial systems
- D&P+Robinson: universal quantifiers removed, but eqn system involves exponentials
- Matiyasevich: exploits exponential growth of Pell's equation solutions to remove exponentials
- $\Rightarrow$ DPRM theorem, implying DE undecidable

Negative answer to Hilbert's 10th problem

Structure of the DPRM theorem

- **Gödel's proof of his 1st incompleteness thm.**

  r.e. sets $\equiv$ DEs with $< \infty$ $\exists$ and bounded $\forall$ quantifiers

- **Davis' normal form**

  one bounded quantifier suffices: $\exists x_0 \forall a \leq x_0 \exists x \; p(a, x) = 0$

  - (2 bnd qnt $\equiv$ 1 bnd qnt on pairs) and induction

- **Robinson's idea**

  get rid of universal quantifier by using exponent vars

  - *idea:* $[\exists x_0 \forall a \leq x_0 \exists x \; p(a, x) = 0]$ "$\rightarrow$" $\left[\exists x \prod_{a \leq x_0} p(a, x) = 0\right]$
  - precise encoding needs variables in exponents

- **Matyiasevic's contribution**

  express $c = b^a$ using polynomials

  - use Pell's equation $x^2 - dy^2 = 1$
  - solutions (x_n, y_n) satisfy $x_n + y_n\sqrt{d} = (x_1 + y_1\sqrt{d})^n$
  - x_n, y_n grow exponentially with n

MP is unsolvable

- **Consider list of all TMs** $(M_i \mid i \in \mathbb{N})$
  if $M_i(x) = \perp$ **at** t**-th execution step, write** $M_i^t(x) = \perp$
- **Yields all sets in** $\mathbb{RE} = (R_i \mid i \in \mathbb{N})$ **by dovetailing**
  at k***-th round, perform*** k***-th step of*** $M_i(1)$, $(k-1)$***-st of*** $M_i(2)$, ***…, 1-st of*** $M_i(k)$
  $\Rightarrow \forall k \in \mathbb{N}$ **and** $\ell \le k$ **if** $M_i^\ell(k-\ell+1) = \perp$
  let $R_i \leftarrow R_i \cup \{k-\ell+1\}$
  $R_i = \{k-\ell+1 \mid \exists k \in \mathbb{N}, \ell \le k \ (M_i^\ell(k-\ell+1) = \perp)\}$
- **DPRM theorem:** $\forall R \in \mathbb{RE}$, R **represented by poly eqn**
- **Let** $R_i \in \mathbb{RE}$ **s.t.** M_i **is a UTM**
  $\Rightarrow \exists$ **Universal DE (UDE), say** $U(r,x) = 0$
- $\min\{0 \mid U(r,x) = 0 \wedge (r,x) \in \mathbb{N}^{n+1}\}$**:**
  undecidable (feasibility) MP
- $\min\limits_{\substack{r\in\mathbb{N}\\ x\in\mathbb{N}^n}} (U(r,x))^2$**: unsolvable (optimization) MP**

Common misconception

"Since $\mathbb{N}$ is contained in $\mathbb{R}$, how is it possible that Reals is decidable but DE ($=$ Reals $\cap\ \mathbb{N}$) is not?"

After all, if a problem contains a hard subproblem, it's hard by inclusion, right?

- **Can you express** DE $p(x) = 0 \wedge x \in \mathbb{N}$ **in** Reals?
  - $p(x) = 0$ **belongs to both** DE **and** Reals, **OK**
  - **"$x \in \mathbb{N}$" in** Reals?

    $\Leftarrow$ **find poly** $q(x)$ **s.t.** $\exists x\, q(x) = 0$ **iff** $x \in \mathbb{N}^n$
  - $q(x) = x(x-1)\cdots(x-a)$ **only good for** $\{0, 1, \ldots, a\}$

    $q(x) = \prod_{i \in \omega} (x - i)$ **is** ∞**ly long, invalid**
  - **IMPOSSIBLE!**

    if it were possible, DE would be decidable, contradiction

MIQCP is undecidable

- **[Jeroslow 1973]: MIQCP:**

$$\left.\begin{array}{rrcl} \min & c^\top x & & \\ \forall i \le m & x^\top Q^i x + a_i{}^\top x + b_i & \ge & 0 \\ & x & \in & \mathbb{Z}^n \end{array}\right\} \qquad (\dagger)$$

is undecidable

Proof:

- **Let $U(r,x) = 0$ be an UDE**
- $P(r) \equiv \min\{u \mid (1-u)U(r,x) = 0 \wedge u \in \{0,1\} \wedge x \in \mathbb{Z}^n\}$
  $P(r)$ describes an undecidable problem
- **Linearize every product $x_i x_j$ by y_{ij} and add $y_{ij} = x_i x_j$**
  until only degree 1 and 2 left
- **Obtain MIQCP ($\dagger$)**

Some MIQCQPs are decidable

- If each Q_i is diagonal PSD, decidable [Witzgall 1963]

- If x are bounded in $[x^L, x^U] \cap \mathbb{Z}^n$, decidable

  can express $x \in \{\lceil x^L \rceil, \lceil x^L \rceil + 1, \ldots, \lfloor x^U \rfloor\}$ *by polynomial*

$$\forall i \le m \qquad \prod_{x_i^L \le i \le x_i^U} (x - i) = 0$$

  turn into poly system in $\mathbb{R}$ *(in* Reals*, decidable)*

- $\Rightarrow$ *Bounded* (vars) easier than *unbounded* (for $\mathbb{Z}$)

- [MIQP decision vers.] is decidable

$$\left.\begin{array}{rrcl} & x^\top Q x + c^\top x & \le & \gamma \\ & Ax & \ge & b \\ \forall j \in Z & x_j & \in & \mathbb{Z} \end{array}\right\} \qquad \text{(in NP [Del Pia et al. 2014])}$$

NLP is undecidable

We can't represent unbounded subsets of $\mathbb{N}$ by polynomials
But we can if we allow some transcendental functions

$$x \in \mathbb{Z} \quad \longleftrightarrow \quad \sin(\pi x) = 0$$

- **Constrained NLP is undecidable:**

$$\min\{0 \mid U(a,x) = 0 \wedge \forall j \leq n \ \sin(\pi x_j) = 0\}$$

- **Even with just one nonlinear constraint:**

$$\min\{0, \mid (U(a,x))^2 + \sum_{j \leq n} (\sin(\pi x_j))^2 = 0\}$$

- **Unconstrained NLP is undecidable:**

$$\min (U(a,x))^2 + \sum_{j \leq n} (\sin(\pi x_j))^2$$

- **Box-constrained NLP is undecidable** *(boundedness doesn't help)*:

$$\min\{(U(a, \tan x_1, \ldots, \tan x_n))^2 + \sum_{j \leq n} (\sin(\pi \tan x_j))^2 \mid -\frac{\pi}{2} \leq x \leq \frac{\pi}{2}\}$$

Some NLPs are decidable

- **All polynomial NLPs are decidable**
  by decidability of Reals

- **Quadratic Programming (QP) is decidable over $\mathbb{Q}$**

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

- *Bricks of the proof*
  - if Q is PSD, $[\![P]\!] \in \mathbb{Q}$
    1. *remove inactive constr., active are eqn, use to replace vars*
    2. *work out KKT conditions, they are linear in rational coefficients*
    3. $\Rightarrow$ *solution is rational*
  - $\exists$ polytime IPM for solving P [Renegar&Shub 1992]
  - *unbounded case treated in [Vavasis 1990]*
- $\Rightarrow$ **[QP decision version] is in NP**
  $\Rightarrow$ **QP is decidable over $\mathbb{Q}$**

Rationals

- **[Robinson 1949]:**
  RT (1st ord. theory over $\mathbb{Q}$) is undecidable
- **[Pheidas 2000]:** *existential* theory of $\mathbb{Q}$ (ERT) is open
  can we decide wether $p(x) = 0$ has solutions in $\mathbb{Q}$? Boh!
- **[Matyiasevich 1993]:**
  - equivalence between DEH and ERT
  - DEH = [DE restricted to homogeneous polynomials]
  - *but we don't know whether DEH is decidable*

Note that Diophantus solved DE in positive rationals

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

The “$O(\cdot)$” calculus

$$\forall f, g : \mathbb{N} \to \mathbb{N} \quad f <_O g \quad \leftrightarrow \quad \exists n \in \mathbb{N}\, \forall \nu > n\, (f(\nu) < g(\nu))$$

$$\forall f : \mathbb{N} \to \mathbb{N} \quad O(f) \quad = \quad \{g : \mathbb{N} \to \mathbb{N} \mid \exists C \in \mathbb{N}\, (g <_O C\, f)\}$$

$$\forall f, g : \mathbb{N} \to \mathbb{N} \quad O(f) < O(g) \quad \leftrightarrow \quad f \in O(g) \;\wedge\; g \notin O(f)$$

Polytime algorithms are "efficient"

- Why are polynomials special?
- Many different variants of Turing Machines (TM)
- Polytime is *invariant* to all definitions of TM

  e.g. TM with ∞ly many tapes: simulate with a single tape running along diagonals, similarly to dovetailing
- 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
- Remarks
  - There are problems which no algorithm can solve
  - A problem can be solved by different algorithms
- Given prob. P find complexity of *best alg.* solving P

$$\min_{<_O}\{\tau(\mathcal{A}) \mid \mathcal{A} \text{ solves } P\}$$

- We (generally) don't know how to search over all algs for P
  when we do, we find <u>*lower bounds*</u> *for complexity (usually hard)*

Complexity classes: P, NP

- 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 algorithm exists

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

Equivalent definition of NP

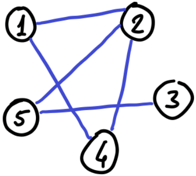
- $\boxed{\textbf{NP}\text{: problems solved by } \textit{nondeterministic} \text{ polytime TM}}$
- ($\Rightarrow$) Assume $\exists$ polysized certificate for every YES instance. Nondeterministic polytime algorithm: concurrently explore all possible polysized certificates, call verification oracle for each, determine YES/NO.
- ($\Leftarrow$) Run nondeterministic polytime algorithm: trace will look like a tree (branchings at tests, loops unrolled) with polytime depth. If YES there will be a terminating polysized sequence of steps from start to termination, serving as a polysized certificate

Subsection 1

Some combinatorial problems in NP

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 $O(k^2) < O(n^2)$ (k fixed)
- 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?

- **Decision variables**: $\forall j \in V \quad x_j = \begin{cases} 1 & j \in k\text{-}\mathbf{clique} \\ 0 & \mathbf{otherwise} \end{cases}$

MP formulations for CLIQUE

Variables? Objective? Constraints?

- **Decision variables:** $\forall j \in V \quad x_j = \begin{cases} 1 & j \in k\text{-}\mathbf{clique} \\ 0 & \mathbf{otherwise} \end{cases}$
- no objective (pure feasibility MP)

MP formulations for CLIQUE

Variables? Objective? Constraints?

- **Decision variables**: $\forall j \in V \quad x_j = \begin{cases} 1 & j \in k\text{-}\mathbf{clique} \\ 0 & \mathbf{otherwise} \end{cases}$
- **no objective (pure feasibility MP)**
- **Constraints**: **"if** $x_i = x_j = 1$, **then** $\{i, j\} \in E$**"**

MP formulations for CLIQUE

Variables? Objective? Constraints?

- **Decision variables**: $\forall j \in V \quad x_j = \begin{cases} 1 & j \in k\text{-}\textbf{clique} \\ 0 & \textbf{otherwise} \end{cases}$
- **no objective (pure feasibility MP)**
- **Constraints**: **"if** $x_i = x_j = 1$, **then** $\{i, j\} \in E$**"**

$$\forall i \neq j \in V \quad x_i x_j = \begin{cases} 1 & \{i, j\} \in E \\ 0 & \textbf{otherwise} \end{cases}$$

- **Issue:** ***nonlinear term in equality constr*** $\Rightarrow$ ***nonconvex***

MP formulations for CLIQUE

Variables? Objective? Constraints?

- **Decision variables:** $\forall j \in V \quad x_j = \begin{cases} 1 & j \in k\text{-}\mathbf{clique} \\ 0 & \mathbf{otherwise} \end{cases}$
- **no objective (pure feasibility MP)**
- **Constraints: "if $x_i = x_j = 1$, then $\{i,j\} \in E$"**

$$\forall i \neq j \in V \quad x_i x_j = \begin{cases} 1 & \{i,j\} \in E \\ 0 & \mathbf{otherwise} \end{cases}$$

- **Issue:** ***nonlinear term in equality constr*** $\Rightarrow$ ***nonconvex***
- **Prop.:** C **clique in** $G \Leftrightarrow C$ **stable in** $\bar{G}$
- **Use constraints for** k**-stable in** $\bar{G}$ **instead:**
  "if $\{i,j\} \in E(\bar{G})$, **then** $x_i = 1$ **or** $x_j = 1$ **or neither but not both"**

MP formulations for CLIQUE

Variables? Objective? Constraints?

- **Decision variables:** $\forall j \in V \quad x_j = \begin{cases} 1 & j \in k\text{-clique} \\ 0 & \text{otherwise} \end{cases}$
- no objective (pure feasibility MP)
- **Constraints:** "if $x_i = x_j = 1$, then $\{i,j\} \in E$"

$$\forall i \neq j \in V \quad x_i x_j = \begin{cases} 1 & \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

- **Issue:** *nonlinear term in equality constr $\Rightarrow$ nonconvex*
- **Prop.:** C clique in $G \Leftrightarrow C$ stable in $\bar{G}$
- **Use constraints for k-stable in $\bar{G}$ instead:**
  "if $\{i,j\} \in E(\bar{G})$, then $x_i = 1$ or $x_j = 1$ or neither but not both"

$$\forall i \neq j \in V \text{ with } \{i,j\} \notin E \quad x_i + x_j \leq 1$$

- Any other constraint?

MP formulations for CLIQUE

- *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

- *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:

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

AMPL code for Max Clique

File `clique.mod`

```
# clique.mod
param n integer, > 0;
set V := 1..n;
set E within {V,V};
var x{V} binary;
maximize clique_card: sum{j in V} x[j];
subject to notstable{i in V, j in V : i<j and (i,j) not in E}:
  x[i] + x[j] <= 1;
```

File `clique.dat`

```
# clique.dat
param n := 5;
set E := (1,2) (1,4) (2,4) (2,5) (3,5);
```

AMPL code for Max Clique

File `clique.run`:

```
# clique.run
model clique.mod;
data clique.dat;
option solver cplex;
solve;
printf "C =";
for {j in V : x[j] > 0} {
  printf " %d", j;
}
printf "\n";
```

Run with "`ampl clique.run`" on command line

```
CPLEX 12.8.0.0: optimal integer solution; objective 3
0 MIP simplex iterations
0 branch-and-bound nodes
C = 1 2 4
```

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 **with** $J = \{1, 2, 3\}$

  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*:

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

AMPL code for SUBSET-SUM

File subsetsum.mod

```
# subsetsum.mod
param n integer, > 0;
set N := 1..n;
param a{N} integer, >= 0;
param b integer, >= 0;
var x{N} binary;
subject to subsetsum: sum{j in N} a[j]*x[j] = b;
```

File subsetsum.dat

```
# subsetsum.dat
param n := 5;
param a :=
1  1
2  1
3  1
4  4
5  5
;
param b := 3;
```

Code your own subsetsum.run!

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

---

- $c = (5, 6, 7), w = (3, 4, 5), K = 11$
  - $c(J) \leq 11$ feasible for J in $\varnothing, \{j\}, \{1, 2\}$
  - $w(\varnothing) = 0, w(\{1, 2\}) = 3 + 4 = 7, w(\{j\}) \leq 5$ for $j \leq n$

    $\Rightarrow J_{\mathsf{max}} = \{1, 2\}$
- $K = 4$: 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?

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

AMPL code for KNAPSACK

File knapsack.mod

```
# knapsack.mod
param n integer, > 0;
set N := 1..n;
param c{N} integer;
param w{N} integer;
param K integer, >= 0;
var x{N} binary;
maximize value: sum{j in N} w[j]*x[j];
subject to knapsack: sum{j in N} c[j]*x[j] <= K;
```

File knapsack.dat

```
# knapsack.dat
param n := 3;
param : c w :=
1  5 3
2  6 4
3  7 5 ;
param K := 11;
```

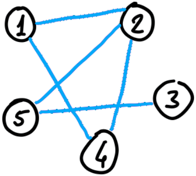
Code your own knapsack.run!

Hamiltonian Cycle

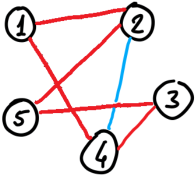
- **Instance:** $G = (V, E)$
- **Problem: does G have a *Hamiltonian cycle*?**

  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?

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

$$\forall j \in V \quad \sum_{\substack{i \in V \\ \{i,j\} \in E}} x_{ij} = 1 \qquad (3)$$

$$\forall \varnothing \subsetneq S \subsetneq V \quad \sum_{\substack{i \in S, j \notin S \\ \{i,j\} \in E}} x_{ij} \geq 1 \qquad (4)$$

***WARNING*: Eq. (4) is a second order statement!**

quantified over sets

yields exponentially large set of constraints

AMPL code for Hamiltonian Cycle

File hamiltonian.mod

```
# hamiltonian.mod
param n integer, > 0;
set V default 1..n, ordered;
set E within {V,V};
set A := E union {i in V, j in V : (j,i) in E};
# index set for nontrivial subsets of V
set PV := 1..2**n-2;
# nontrivial subsets of V
set S{k in PV} := {i in V: (k div 2**(ord(i)-1)) mod 2 = 1};

var x{A} binary;
subject to successor{i in V} :
  sum{j in V : (i,j) in A} x[i,j] = 1;
subject to predecessor{j in V} :
  sum{i in V : (i,j) in A} x[i,j] = 1;

# breaking non-hamiltonian cycles
subject to breakcycles{k in PV}:
  sum{i in S[k], j in V diff S[k]: (i,j) in A} x[i,j] >= 1;
```

Code your own .dat and .run files!

Satisfiability (sat)

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

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

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

- **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

Variables? Objective? Constraints?

MP formulation for SAT

Variables? Objective? Constraints?

Algorithm $\hat{\rho}$ to generate MP from $\bigwedge\limits_{i \le m} \bigvee\limits_{j \in C_i} \ell_j$:

MP formulation for SAT

Variables? Objective? Constraints?

Algorithm $\hat\rho$ to generate MP from $\bigwedge\limits_{i\le m}\bigvee\limits_{j\in C_i}\ell_j$:

- **Literals $\ell_j \in \{x_j, \bar x_j\}$:** *decision variables in* $\{0,1\}$

$$\hat\rho(\ell_j) \quad \longmapsto \quad \begin{cases} x_j & \textbf{if } \ell_j \equiv x_j \\ 1 - x_j & \textbf{if } \ell_j \equiv \bar x_j \end{cases}$$

- **Clauses $\Gamma_i \equiv \bigvee_{j\in C_i}\ell_j$:** *constraints*

$$\hat\rho(\Gamma_i) \quad \longmapsto \quad \sum_{j\in C_i}\hat\rho(\ell_j) \ge 1$$

- **Conjunction:** *feasibility-only ILP*

$$\hat\rho(\bigwedge_i \Gamma_i) \quad \longmapsto \quad \forall i \le m \quad \hat\rho(\Gamma_i)$$

MP formulation for SAT

- **Prop.:** SAT instance q is YES iff ILP instance $\hat{\rho}(q)$ is YES
- ***Proof:*** Let $L = (\ell_1, \ldots, \ell_n)$ be a solution of SAT. Then $x^* = (x_1^*, \ldots, x_n^*)$ where $x_j^* = 1$ iff $\ell_j = \mathsf{true}$ and $x_j^* = 0$ iff $\ell_j = \mathsf{false}$ is a feasible solution of ILP (satisfies each clause constraint by definition of $\hat{\rho}$).

  Conversely: if x solves ILP, then form solution L of SAT by mapping $x_j^* = 1$ to true and $x_j^* = 0$ to false, result follows again by defn of $\hat{\rho}$.

AMPL code for SAT?

Without a numeric encoding of SAT instances, we can only write AMPL code for single instances (i.e. "we are $\hat{\rho}$")

Example: file `sat.run` (flat formulation) for instance
$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2)$

```
# sat.run
var x{1..3} binary;
subject to con1: x[1] + (1-x[2]) + x[3] >= 1;
subject to con2: (1-x[1])+ x[2] >= 1;
option solver cplex;
solve;
display x, solve_result;
```

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
- Prob. P is $\boxed{\textbf{NP-}\textit{hard}}$ if $\forall Q \in \mathbf{NP}\ \exists$ polytime alg. ρ_Q:
  1. $q \in Q \mapsto \rho_Q(q) \in P$ with q YES iff $\rho_Q(q)$ YES
  2. run best alg. for P on $\rho_Q(q)$, get answer $\alpha \in \{\mathsf{YES}, \mathsf{NO}\}$
  3. return α as answer for q

  ρ_Q is called a *polynomial reduction* from Q to P

  P hardest since othw, using ρ_Q, Q would be "easier than itself"!
- If P is in NP and is NP-hard, it is called NP-*complete*
- *Reduction idea: "model" Q using "language" of P*
- 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:

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

$G^t_{i,j}$ 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!

The MP version of Cook's theorem

Thm.

Any problem in **NP** can be polynomially reduced to a MILP

Proof

(*Sketch*) Model the dynamics of a nondeterministic polytime TM using binary variables and constraints involving sums and products; and then linearize the products of binary variables by means of Fortet's inequalities

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: $(Q \smallsetminus F \times \Sigma) \times (Q \times \Sigma \times \{-1, 1\})$
  δ: state ℓ, symbol $j \mapsto$ state ℓ', symbol j', direction d
- 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* <u>tape</u> cell** i **contains symbol** j **at step** k
- $\forall i \in I, k \in K$
  $h_{ik} = 1$ ***iff* <u>head</u> is at tape cell** i **at step** k
- $\forall \ell \in Q, k \in K$
  $q_{\ell k} = 1$ ***iff*** M **is in <u>state</u>** 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

1. *Initialization:*

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

2. *Execution:*

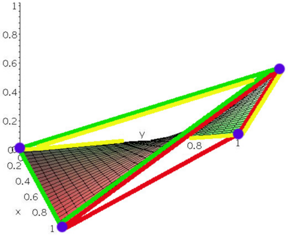
   2.1 $\forall i, k \qquad \sum_j t_{ijk} = 1$
   2.2 $\forall i, j \neq j', k < p(n) \qquad t_{ijk}\, t_{i,j',k+1} = h_{ik}$
   2.3 $\forall k \qquad \sum_i h_{ik} = 1$
   2.4 $\forall i, \ell, j, k$
   $$|\delta(\ell, j)|\, h_{ik}\, q_{\ell k}\, t_{ijk} = \sum_{((\ell,j),(\ell',j',d))\in\delta} h_{i+d,k+1}\,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

3. *Termination:*

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

Cook's theorem: conclusion

- **MP in previous slide MINLP not MILP**
- **Fortet's inequalities for products of binary vars:**
  For $x, y \in \{0,1\}$ **and** $z \in [0,1]$
  $z = xy \Leftrightarrow z \leq x \wedge z \leq y \wedge z \geq x + y - 1$



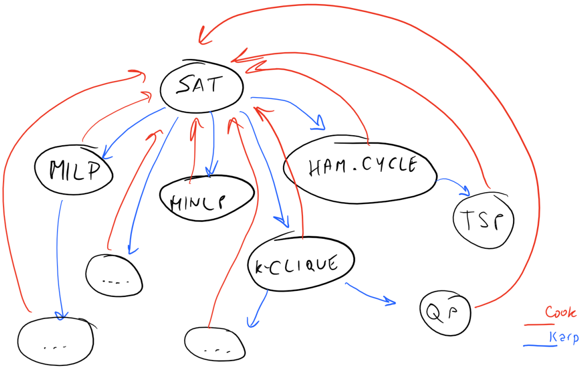
- **MILP is feasibility only**
- **MILP has polynomial size**
- $\Rightarrow$ *MILP is* **NP**-*hard*

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_Q from Q to P [Karp 1972]

Why it works: suppose P easier than Q, solve Q by calling $\text{Alg}_P \circ \rho_Q$, 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)

$$\bar{G} = (V, \bar{E} = \{\{i, j\} \mid i, j \in V \wedge \{i, j\} \not\in E\})$$

  - **Prop.**: 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

Subsection 3

Complexity of solving MP formulations

LP is in P

- Khachian's algorithm (Ellipsoid method)
- Karmarkar's algorithm
- IPM with crossover
  IPM: penalize $x \geq 0$ by $-\beta \log(x)$, polysized sequence of subproblems
  crossover: polytime number of simplex pivots get to opt
- No known pivot rule makes simplex alg. polytime!
  greedy pivot has exponential complexity on Klee-Minty cube

(Recall) MILP is NP-hard

- SAT NP-hard by Cook's theorem, reduce from SAT

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

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

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

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

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

- SAT is YES iff MILP is feasible

Complexity of Quadratic Programming (QP)

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

- **Quadratic obj, linear consts, continuous vars**
- **Many applications (e.g. portfolio selection)**
- **If Q has at least one negative eigenvalue, NP-hard**
- **Decision problem: "is the min. obj. fun. value ≤ 0?"**
- **If Q PSD then objective is convex, problem is in P**
  KKT conditions become linear system, data in $\mathbb{Q} \Rightarrow$ soln in $\mathbb{Q}$

QP is NP-hard

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

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

- **Claim:** σ is YES iff val($\dagger$)$\equiv$ opt. obj. fun. val. of ($\dagger$) $= 0$
- *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, val($\dagger$) $= 0$
  - assume σ NO, suppose 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

$$\left.\min_{x\in[x^L,x^U]} \quad x^\top Q x \quad + \quad c^\top x \right\}$$

- **Add surplus vars v to SAT→MILP constraints:**
  $\hat{\rho}(\sigma, x) - 1 - v = 0$
  (denote by $\forall i \leq m\ (a_i^\top x - b_i - v_i = 0)$)
- **Consider special QP subclass**

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

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

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 special QP subclass with $f(x) = \sum_j x_j(1 - x_j)$

- 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 $\mathsf{opt}(\text{cQKP}) = x^*$ with $f(x^*) = 0$
  - then $x^* \in \{0,1\}^n$ because $f(x^*) = 0$
  - feasibility of $x^* \to J = \mathsf{supp}(x^*)$ solves σ, contradiction $\Rightarrow f(x^*) > 0$

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

  Motzkin-Straus formulation (MSF):

$$\max\{\sum_{\{i,j\} \in E} x_i x_j \mid \sum_{j \in V} x_j = 1 \wedge x \ge 0\}$$

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

Let C **be the maximum clique of the instance** $G = (V, E)$ **of MAX CLIQUE**
$\exists x^* \in \mathsf{opt}\,(\mathbf{MSF})$ **with** $f^* = f(x^*) = \frac{1}{2} - \frac{1}{2\omega(G)}$
$\forall j \in V \qquad x^*_j = \begin{cases} \frac{1}{\omega(G)} & \textbf{if } j \in C \\ 0 & \textbf{otherwise} \end{cases}$

- $\omega(G)$**: size of max clique in** G

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 $\{1,2\} \not\in E$, then $x^*_1, x^*_2 > 0$, can perturb by $\epsilon \in [-x^*_1, x^*_2]$, get $x^\epsilon = (x^*_1 + \epsilon, x^*_2 - \epsilon, \ldots)$, feasible w.r.t. simplex and bounds
   - $\{1,2\} \not\in E \Rightarrow x_1 x_2$ 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
   - setting $\epsilon = -x^*_1$ 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

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

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 $j \notin C \Rightarrow$

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

- $\psi(x) = 1$ 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

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

so $f(x^*)$ attains max $\frac{1}{2} - \frac{1}{2\omega(G)}$ when $|C| = \omega(G) \Rightarrow \forall j \in C \ x_j = \frac{1}{\omega(G)}$

Copositive programming

- StQP: $\min x^\top Q x : \sum_j x_j = 1 \wedge x \geq 0$
  NP-hard by Motzkin-Straus
- **Linearize:** $X = xx^\top$
  replace $x_i x_j$ by X_{ij} and add constraints $X_{ij} = x_i x_j$
- **Define** $A \bullet B = \mathsf{tr}(A^\top B) = \sum_{i,j} A_{ij} B_{ij}$
  write StQP (linearized) objective as $\min Q \bullet X$
- **Let** $C = \{X \mid X = xx^\top \wedge x \geq 0\}$, $\bar{C} = \mathsf{conv}(C)$
- $\sum_j x_j = 1 \Leftrightarrow (\sum_j x_j)^2 = 1^2 \Leftrightarrow \mathbf{1} \bullet X = 1$
- StQP $\equiv \min Q \bullet X : \mathbf{1} \bullet X = 1 \wedge X \in \bar{C}$
  linear obj. $\Rightarrow$ *optima attained at extrema of feas. set*
  $\Rightarrow$ *can replace* C *by its convex hull* $\bar{C}$

  $\bar{C}$ is a completely positive cone
- **Dual** $\equiv \max y : Q - y\mathbf{1} \in \bar{C}^* = \{A \mid \forall x \geq 0 \; (x^\top A x \geq 0)\}$

  $\bar{C}^*$ is a copositive cone
- $\Rightarrow$ **Pair of NP-hard cNLPs!**

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

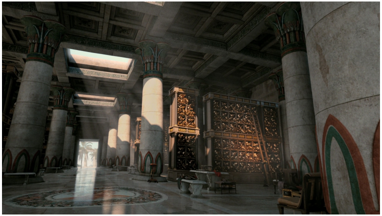
> *You, a private investment banker, are seeing a customer. She tells you "I have 3,450,000$ I don't need in the next three years. Invest them in low-risk assets so I get at least 2.5% return per year."*
>
> *Model the problem of determining the required portfolio. Missing data are part of the fun (and of real life).*

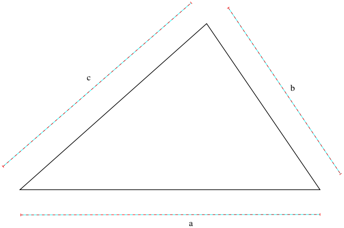
[Hint: what are the decision variables, objective, constraints? What data are missing?]

Outline

A gem in Distance Geometry

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



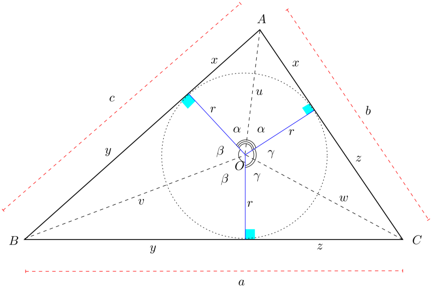


$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

- $A =$ 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:*

$$\begin{aligned} & d_{ik} \leq d_{ij} + d_{jk} \quad \wedge \quad d_{jk} \leq d_{ij} + d_{ik} \\ \Rightarrow \quad & d_{ik} - d_{jk} \leq d_{ij} \quad \wedge \quad d_{jk} - d_{ik} \leq d_{ij} \\ \Rightarrow \quad & |d_{ik} - d_{jk}| \leq d_{ij} \end{aligned}$$

If valid $\forall i, j$ **then valid for** $\max$

- $\max |d_{ik} - d_{jk}|$ **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: $d_{ij}^2 = \|x_i - x_j\|_2^2 = (x_i - x_j)(x_i - x_j) = x_i x_i + x_j x_j - 2x_i x_j$ $\quad (*)$
- Aim at "inverting" $(*)$ to express $x_i x_j$ in function of d_{ij}^2
- Sum $(*)$ over i: $\sum_i d_{ij}^2 = \sum_i x_i x_i + n x_j x_j - 2x_j \cancel{\sum_i x_i}$ 0 by zero centroid
- Similarly for j and divide by n, get:

$$\frac{1}{n}\sum_{i\leq n} d_{ij}^2 = \frac{1}{n}\sum_{i\leq n} x_i x_i + x_j x_j \quad (\dagger)$$

$$\frac{1}{n}\sum_{j\leq n} d_{ij}^2 = x_i x_i + \frac{1}{n}\sum_{j\leq n} x_j x_j \quad (\ddagger)$$

- 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 $(*)$, $(\dagger)$, $(\ddagger)$ as follows:

$$
\begin{aligned}
2x_i x_j &= x_i x_i + x_j x_j - d_{ij}^2 && (5)\\
x_i x_i &= \frac{1}{n}\sum_j d_{ij}^2 - \frac{1}{n}\sum_j x_j x_j && (6)\\
x_j x_j &= \frac{1}{n}\sum_i d_{ij}^2 - \frac{1}{n}\sum_i x_i x_i && (7)
\end{aligned}
$$

- Replace LHS of Eq. (6)-(7) in RHS of Eq. (5), 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

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

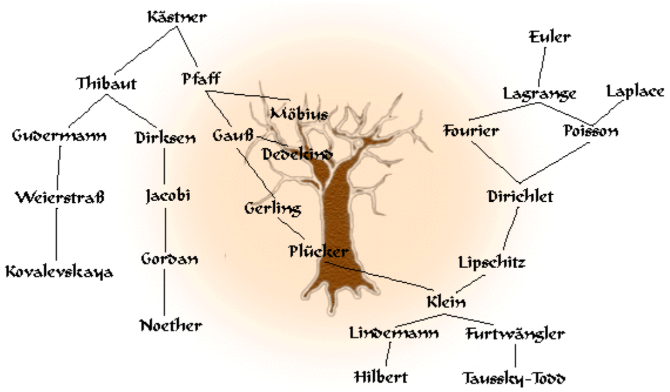
which expresses $x_i x_j$ in function of D

Principal Component Analysis (PCA)

- **Given an approximate distance matrix** D
- **find** $x = \textbf{MDS}(D)$
- **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

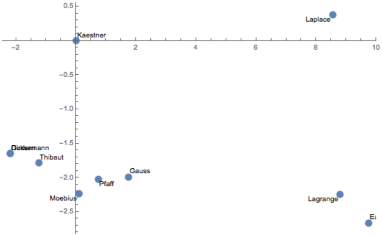
A partial view

	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

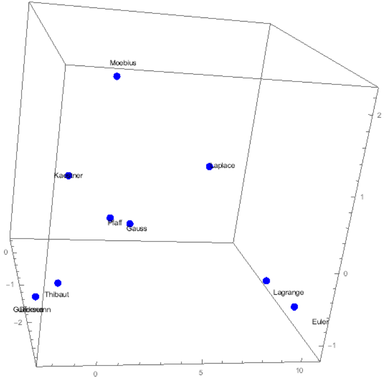
$$D = \begin{pmatrix} 0 & 10 & 1 & 1 & 9 & 8 & 2 & 2 & 2 & 2 \\ 10 & 0 & 11 & 9 & 1 & 3 & 10 & 12 & 12 & 8 \\ 1 & 11 & 0 & 2 & 10 & 10 & 3 & 1 & 1 & 3 \\ 1 & 9 & 2 & 0 & 8 & 8 & 1 & 3 & 3 & 1 \\ 9 & 1 & 10 & 8 & 0 & 2 & 9 & 11 & 11 & 7 \\ 8 & 3 & 10 & 8 & 2 & 0 & 9 & 11 & 11 & 7 \\ 2 & 10 & 3 & 1 & 9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

Example 3/3

In 2D



In 3D



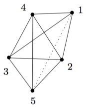
Subsection 3

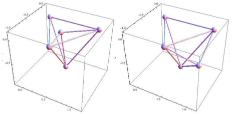
Distance geometry problem

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)
- *finding graph embeddings*

Clock synchronization

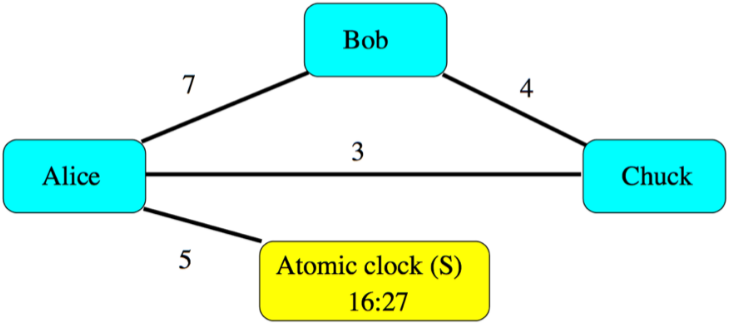
From [Singer, *Appl. Comput. Harmon. Anal.* 2011]

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

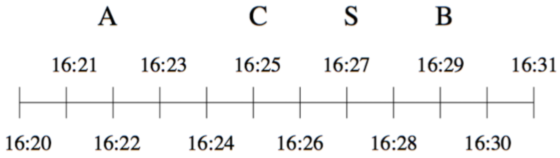
- $K = 1$
- V: timestamps
- $\{u, v\} \in E$ if known time difference between u, v
- d: values of the time differences

Used in time synchronization of distributed networks

Clock synchronization



↓



Sensor network localization

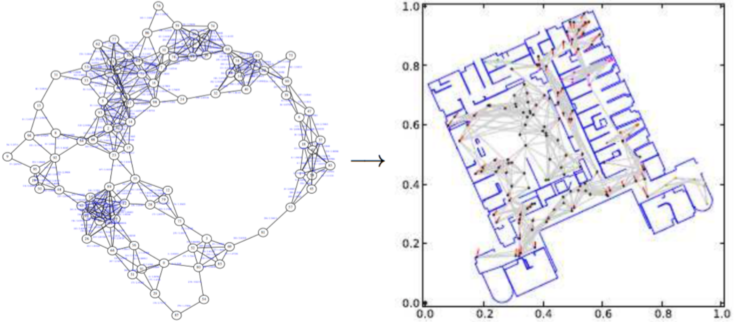
From [Yemini, *Proc. CDSN*, 1978]

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

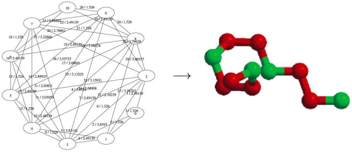
Used whenever GPS not viable (e.g. underwater)
$d_{uv} \propto$ battery consumption in P2P communication betw. u, v

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

Graph embeddings

- Relational knowledge best represented by graphs
- We have fast algorithms for clustering vectors
- **Task:** represent a graph in $\mathbb{R}^n$
- "Graph embeddings" and "distance geometry": almost synonyms
- Used in Natural Language Processing (NLP)
  obtain "word vectors" & "concept vectors"
- **Project:** create a graph-of-words from a sentence, enrich it with semantic distances, then use MP formulations for DG to embed the graph in a low-dimensional space

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

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

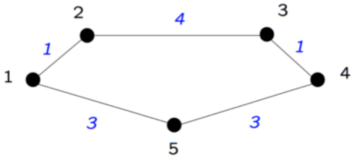
    in polytime
- DGP_K may not be in NP for $K > 1$
  don't know how to verify $\|x_i - x_j\|_2 = d_{ij}$ for $x \notin \mathbb{Q}^{nK}$

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*** $x_{n+1} = x_1$

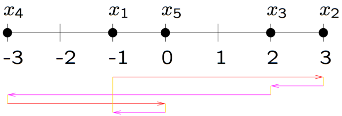
PARTITION is YES $\Rightarrow$ DGP$_1$ is YES

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

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

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 $E(C) \smallsetminus F$ ($\leftarrow$)**



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

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

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

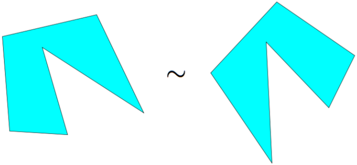
- **So J solves Partition instance, contradiction**
- **$\Rightarrow$ DGP is NP-hard, DGP$_1$ is NP-complete**

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 $|X| \neq \aleph_0$

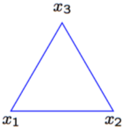
- System S of polynomial equations of degree 2

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

- 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

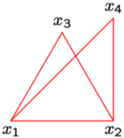
Examples

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



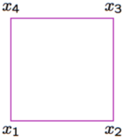
ρ congruence in $\mathbb{R}^2$
$\Rightarrow \rho x$ valid realization
$|X| = 1$

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



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

$$\forall \{u,v\} \in E \quad \|x_u - x_v\|^2 = d_{uv}^2 \qquad \textbf{(8)}$$

Computationally: useless
reformulate using slacks:

$$\min_{x,s} \Big\{ \sum_{\{u,v\}\in E} s_{uv}^2 \;\Big|\; \forall \{u,v\} \in E \quad \|x_u - x_v\|^2 = d_{uv}^2 + s_{uv} \Big\} \qquad \textbf{(9)}$$

Unconstrained Global Optimization

$$\min_x \sum_{\{u,v\}\in E} (\|x_u - x_v\|^2 - d_{uv}^2)^2 \qquad \textbf{(10)}$$

Globally optimal obj. fun. value of (10) is 0 iff x solves (8)

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

- $\min_x \sum_{\{u,v\}\in E} |\|x_u - x_v\|^2 - d_{uv}^2|$ **exactly reformulates (8)**
- **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*

$$\left.\begin{array}{rl} \max_x & \sum_{\{u,v\}\in E} \|x_u - x_v\|^2 \\ \forall \{u,v\}\in E & \|x_u - x_v\|^2 \le d_{uv}^2 \end{array}\right\} \qquad (11)$$

Theorem (Activity)

At a glob. opt. x^* of a YES *instance, all constraints of* (11) *are active*

Linearization

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

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

Relaxation

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

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

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

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:
  - $\mathbf{LDE}(x) = \max_{\{i,j\}\in E} |\, \|x_i - x_j\| - d_{ij}\,|$
  - $\mathbf{MDE}(x) = \frac{1}{|E|} \sum_{\{i,j\}\in E} |\, \|x_i - x_j\| - d_{ij}\,|$

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

Efficiency vs. mathematical rigor

- Today we wish to solve problems with very large sizes
- We need methods that work computationally
- But we'd also like methods that are mathematically sound
  exactness, guaranteed approximation ratios, etc
- Unfortunately, there is no correlation beteween the efficiency of a methodology and the ease of proving approximation guarantees
- **In industry:** we care FIRST about the empirical efficiency, and NEXT about the proofs
- **In academia:** often the opposite, but mostly both
- In practice, we use inductive/abductive inference in order to guide us in looking for an efficient algorithm
  sometimes these inferences can lead to approximation proofs in probability

Retrieving realizations in $\mathbb{R}^K$

- SDP relaxation yields $n \times n$ PSD matrix X^*
- We need $n \times K$ realization matrix x^*
- Recall *PSD* $\Leftrightarrow$ *Gram*
- Apply PCA to X^*, keep K largest comps, get x'
- This yields solutions with errors
- Use x' as starting pt for local NLP solver

When SDP solvers hit their size limit

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

Diagonally dominant matrices

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

$$\forall i \leq n \quad X_{ii} \geq \sum_{j \neq i} |X_{ij}|.$$

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

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

- **linearize $|\cdot|$ by additional matrix var T**
  $\Rightarrow$ ***write $|X|$ as T***
- **$\Rightarrow (*)$ becomes**

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

- **add "sandwich" constraints $-T \le X \le T$**
- **Can easily prove $(*)$ by sandwich constraints:**

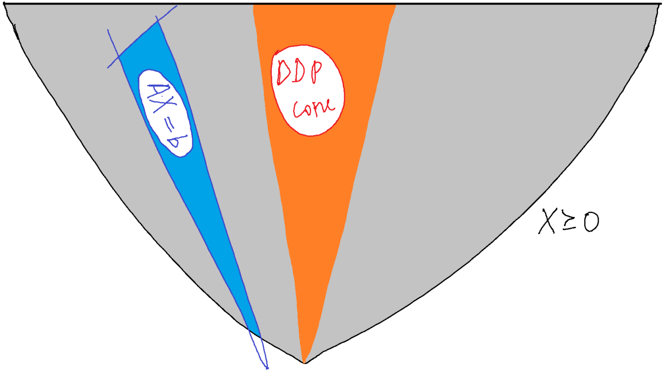
$$\begin{aligned} X_{ii} &\ge \sum_{j \ne i} T_{ij} \ge \sum_{j \ne i} X_{ij} \\ X_{ii} &\ge \sum_{j \ne i} T_{ij} \ge \sum_{j \ne i} -X_{ij} \end{aligned}$$

DD Programming (DDP)

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

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

The issue with inner approximations



DDP could be infeasible!

Exploit push-and-pull

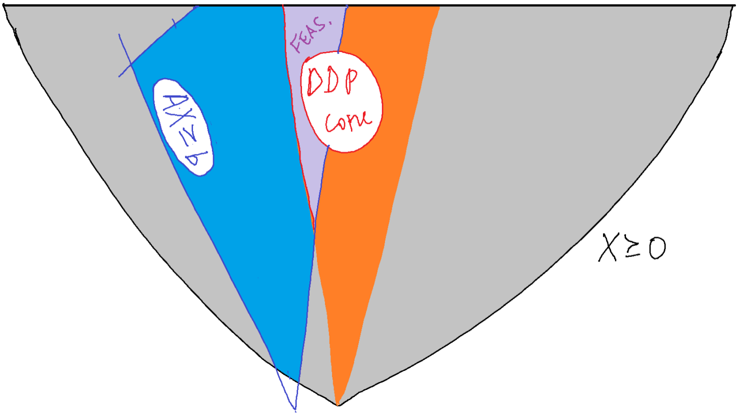
- Enlarge the feasible region
- From

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

- 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

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

Solve, then retrieve solution in $\mathbb{R}^K$ with PCA

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

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

all vectors making acute angles with elements of C

- 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

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

- *$\mathbb{PSD}$ (resp. $\mathbb{DD}$) are cones of PSD (resp. DD) matrices: prove it*

Representations of $\mathbb{DD}$

- **Consider $E_{ii}, E_{ij}^{+}, E_{ij}^{-}$ in $\mathbb{S}_n$**
  Define $\mathcal{E}_0 = \{E_{ii} \mid i \leq n\}, \mathcal{E}_1 = \{E_{ij}^{\pm} \mid i < j\}, \mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_1$
- $E_{ii} = \mathsf{diag}(0, \ldots, 0, 1_i, 0, \ldots, 0)$
- 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**
  $\mathcal{X}_{\mathbb{DD}} = \{e_i \mid i \leq n\} \cup \{(e_i \pm e_j) \mid j < \ell \leq n\}$
  Pf. *Verify* $E_{ii} = e_i e_i^{\top}, E_{ij}^{\pm} = (e_i \pm e_j)(e_i \pm e_j)^{\top}$, *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

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

- ($\supseteq$) 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 defn. of Y)
  - whence $Y \in C^*$ (by defn. of C^*)
- ($\subseteq$) 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. $\forall x \in \mathcal{X}'\ (Z \bullet xx^\top < 0)$
  - 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:**

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

- **Cor.** $\mathbb{DD}^* \supset \mathbb{PSD}$

  Pf. $X \in \mathbb{PSD}$ iff $\forall v \in \mathbb{R}^n \; vXv \geq 0$, so certainly valid $\forall v \in \mathcal{X}$
- 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

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

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

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

Note we went back to equality "pull" constraints (why?)

Quantifier $\forall \{i,j\} \notin E$ *should be* $\forall i < j$ *but we already have those constraints* $\forall \{i,j\} \in E$

Properties

- SDP relaxation of original problem
- DualDDP relaxation of SDP
  hence also of original problem
- Yields extremely tight obj fun bounds w.r.t. SDP
- Solutions may have large negative rank
  in some applications, retrieving feasible solutions may be 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.

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

where $\mathsf{E}(\cdot)$ is w.r.t. given Borel measure μ over X

$\equiv$ *“discrepancy from mean is unlikely”*

Barvinok's theorem

Consider:

- **for each** $k \leq m$**, manifolds** $\mathcal{X}_k = \{x \in \mathbb{R}^n \mid x^\top Q^k x = a_k\}$
  where $m \leq \mathsf{poly}(n)$
- **feasibility problem** $F \equiv \left[\bigcap_{k \leq m} \mathcal{X}_k \stackrel{?}{\neq} \varnothing\right]$
- **SDP relaxation** $\forall k \leq m \ (Q^k \bullet X = a_k) \wedge X \succeq 0$ **with soln.** $\bar{X}$
- **Algorithm:** $T \leftarrow \mathsf{factor}(\bar{X})$; $\quad y \sim \mathcal{N}^n(0,1)$; $\quad x' \leftarrow Ty$

Then:

- $\exists c > 0, n_0 \in \mathbb{N}$ **such that** $\forall n \geq n_0$

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

Algorithmic application

- x' **is "close" to each** $\mathcal{X}_k$**:** ***try local descent from*** x'
- $\Rightarrow$ **Feasible QP solution from an SDP relaxation**

Elements of Barvinok's formula

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

- $\sqrt{\|\bar{X}\|_2}$ **arises from** T **(a factor of** $\bar{X}$**)**
- $\sqrt{\ln n}$ **ensures concentration of measure**
- 0.9 **follows by adjusting parameter values in "union bound"**

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:** $x \in \mathbb{R}^{Kn}$, **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

Proof structure

- Show that, on average, $\forall k \le m \; (Ty)^\top Q^k (Ty) = Q^K \bullet \bar{X} = a_k$
  - compute multivariate integrals
  - bilinear terms disappear because y normally distributed
  - decompose multivariate int. to a sum of univariate int.
- Exploit concentration of measure to show errors happen rarely
  - a couple of technical lemmata yielding bounds
  - $\Rightarrow$ bound Gaussian measure μ of ε-neighbourhoods of

$$
\begin{aligned}
A_i^- &= \{y \in \mathbb{R}^{n\times K} \mid \mathcal{Q}^i(Ty) \le Q^i \bullet \bar{X}\} \\
A_i^+ &= \{y \in \mathbb{R}^{n\times K} \mid \mathcal{Q}^i(Ty) \ge Q^i \bullet \bar{X}\} \\
A_i &= \{y \in \mathbb{R}^{n\times K} \mid \mathcal{Q}^i(Ty) = Q^i \bullet \bar{X}\}.
\end{aligned}
$$

  - use "union bound" for measure of $A_i^-(\varepsilon) \cap A_i^+(\varepsilon)$
  - show $A_i^-(\varepsilon) \cap A_i^+(\varepsilon) = A_i(\varepsilon)$
  - use "union bound" to measure intersections of $A_i(\varepsilon)$
  - appropriate values for some parameters $\Rightarrow$ result

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**

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

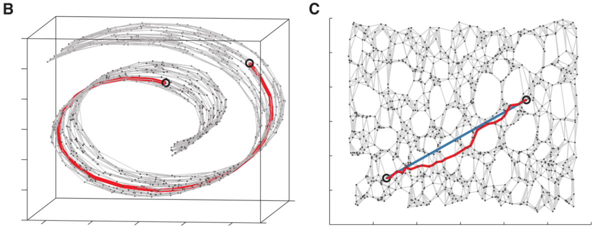
and return improved solution x

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* EDM $\tilde{D}$
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

Why it works

- G represented by weighted partial adj. matrix D'
- don't know full EDM, approximate to $\tilde{D}$
- $\Rightarrow$ get $\tilde{B}$, not generally Gram
- $\leq K$ largest nonnegative eigenvalues
  $\Rightarrow$ "closest PSD matrix" B' to $\tilde{B}$ having rank $\leq K$
- Factor it to get $\tilde{x} \in \mathbb{R}^{Kn}$

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/DDP/DualDDP to find a realization $\bar{x} \in \mathbb{R}^n$, use its sqEDM**

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

Subsection 8

Summary

Matrix reformulations

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

Solution rank reduction methods

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

All provide good starting points for local NLP descent

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

Outline

The gist of random projections

- **Let A be a $m \times n$ data matrix (columns in $\mathbb{R}^m$, $m \gg 1$)**
- **T short & fat, normally sampled componentwise**

$$\underbrace{\begin{pmatrix} \vdots\,\vdots\,\vdots & \vdots\,\vdots\,\vdots \end{pmatrix}}_{T}\underbrace{\begin{pmatrix}\vdots&\vdots&\vdots\\ \vdots&\vdots&\vdots\end{pmatrix}}_{A} = \underbrace{\begin{pmatrix}\vdots&\vdots&\vdots\end{pmatrix}}_{TA}$$

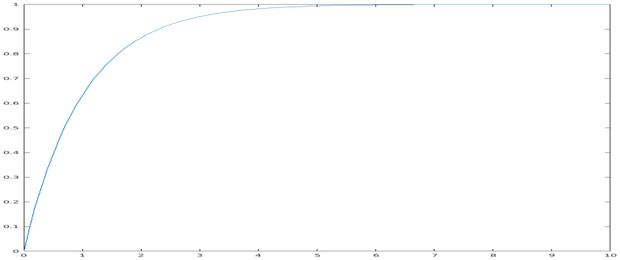
- **Then $\forall i < j \ \|A_i - A_j\|_2 \approx \|TA_i - TA_j\|_2$ "wahp"**

wahp

"wahp" = "with arbitrarily high probability"

the probability of E_k (depending on some parameter k) approaches 1 *"exponentially fast"* as k increases

$$\mathbf{P}(E_k) \approx 1 - O(e^{-k})$$



Johnson-Lindenstrauss Lemma (JLL)

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.**

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

If $k \times m$ **matrix** T **is sampled componentwise from** $N(0, \frac{1}{\sqrt{k}})$**, then**

$$\mathbf{P}(A \text{ and } TA \text{ approximately congruent}) \ge \frac{1}{n}$$

(nontrivial) — result follows by probabilistic method

Note that $1/\sqrt{k}$ *is the standard deviation, not the variance*

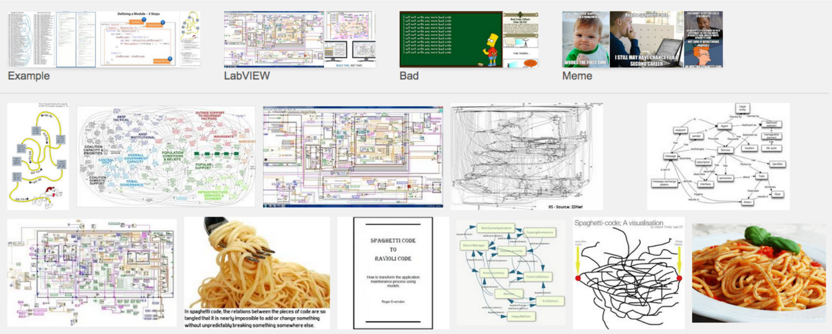
In practice

- $\mathbf{P}(A$ **and** TA **approximately congruent**$) \geq \frac{1}{n}$
- **re-sampling sufficiently many times gives wahp**
- **Empirically, sample** T **few times (once will do)**
  $\mathbb{E}_T(\|Tx - Ty\|) = \|x - y\|$
  probability of error decreases wahp

Surprising fact:

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

Clustering Google images



[L. & Lavor, 2017]

Clustering without random projections

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


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

Too slow!

Clustering 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

Projecting formulations

- **Given:**
  - $F(p,x)$: **MP formulation with params p & vars x**
  - $\mathsf{sol}(F)$: **solution of F**
  - $\mathscr{C}$: **formulation class (e.g. LP, NLP, MILP, MINLP)**
  - R **rnd proj operator if R, F commute:**
$$R\,F(p,x) \triangleq F\,R(p,x)$$
- **"Thm.":** $\forall F \in \mathscr{C}\ \mathsf{sol}(F) \approx \mathsf{sol}(RF)$ **wahp**
- *Low distortion result holds for a formulation*

- **Today we see this for $\mathscr{C} =$ LP**
- **I'm working on QP**
  we have theoretical results (no tests) for SDP/SOCP

Subsection 1

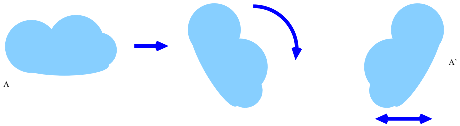
Random projection theory

The shape of a set of points

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

  A and A' "have almost the same shape"

- **What is the shape of a set of points?**



congruence $\Leftrightarrow$ *same shape:* $\|A_i - A_j\| = \|A'_i - A'_j\|$

- *Approximate congruence* $\equiv$ *small distortion:*
  A, A' have *almost the same shape* if

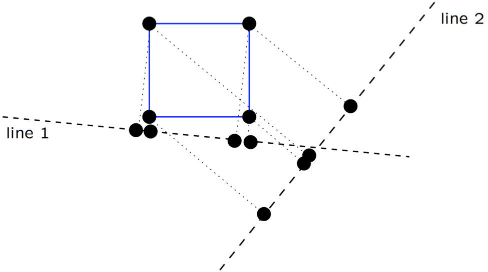
  $$\forall i < j \leq n \quad (1-\varepsilon)\|A_i - A_j\| \;\leq\; \|A'_i - A'_j\| \;\leq\; (1+\varepsilon)\|A_i - A_j\|$$

  for some small $\varepsilon > 0$

Assume norms are all Euclidean

Losing dimensions = “projection”

In the plane, hopeless



In 3D: no better

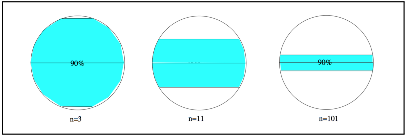
Recall the JLL

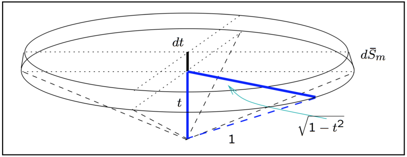
Thm.

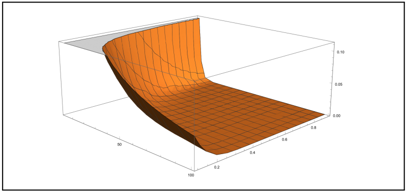
Given $A \subseteq \mathbb{R}^m$ **with** $|A| = n$ **and** $\varepsilon > 0$ **there is** $k \sim O(\frac{1}{\varepsilon^2} \ln n)$ **and a** $k \times m$ **matrix** T **s.t.**

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

Sketch of a JLL proof by pictures

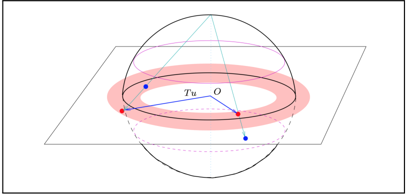






Thm.

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

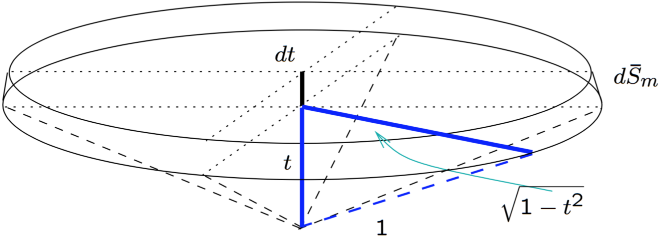


Surface area of a slice of hypersphere

$$\bar{S}_m(r) = \frac{2\pi^{m/2} r^{m-1}}{\Gamma(m/2)} \qquad S_m \triangleq \bar{S}_m(1)$$

Lateral surface of infinitesimally high hypercylinder

$$d\bar{S}_m(t) = S_{m-1}(1-t^2)^{\frac{m-2}{2}} dt$$

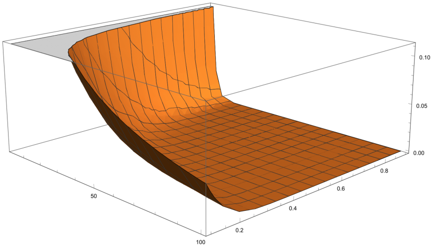


Area of polar caps

$$\mathcal{A}^{\mathbf{pc}} = \int_t^1 d\bar{S}_m(s) = 2S_{m-1}\int_t^1 (1-s^2)^{\frac{m-2}{2}} ds$$

$$1 + x \le e^x \textbf{ for all } x \qquad \textbf{and} \qquad \int\limits_t^1 f(s)ds \le \int\limits_t^\infty f(s)ds \textbf{ for } f \ge 0$$

$$\Rightarrow \mathcal{A}^{\mathbf{pc}} \le 2S_{m-1}\int_t^\infty e^{-\frac{m-2}{2}s^2} ds = \frac{2S_{m-1}}{\sqrt{m-2}}\sqrt{\frac{\pi}{2}}\,\mathsf{erfc}\left(\frac{\sqrt{m-2}t}{\sqrt{2}}\right) = O(e^{-t^2})$$



- **Polar caps area decreases wahp as** $m \to \infty$
- **Concentration of measure**

Rnd proj preserve norms on avg

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

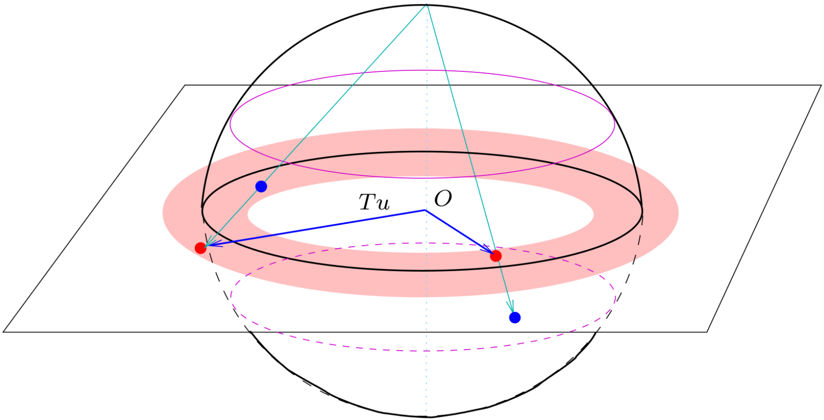
Proof

- $\forall i \leq k$ **let** $v_i = \sum_{j \leq n} T_{ij} u_j$
- $\mathbb{E}(v_i) = \mathbb{E}\left(\sum_{j \leq m} T_{ij} u_j\right) = \sum_{j \leq m} \mathbb{E}(T_{ij}) u_j = 0$
- $\mathsf{Var}(v_i) = \sum_{j \leq m} \mathsf{Var}(T_{ij} u_j) = \sum_{j \leq m} \mathsf{Var}(T_{ij}) u_j^2 = \sum_{j \leq m} \frac{u_j^2}{k} = \frac{1}{k}\|u\|^2 = \frac{1}{k}$
- $\frac{1}{k} = \mathsf{Var}(v_i) = \mathbb{E}(v_i^2 - (\mathbb{E}(v_i))^2) = \mathbb{E}(v_i^2 - 0) = \mathbb{E}(v_i^2)$
- $\mathbb{E}(\|Tu\|^2) = \mathbb{E}(\|v\|^2) = \mathbb{E}\left(\sum_{i \leq k} v_i^2\right) = \sum_{i \leq k} \mathbb{E}(v_i^2) = \sum_{i \leq k} \frac{1}{k} = 1$

Can we argue that the variance decreases wahp?

An intuitive explanation

- **Polar caps area** $\mu(\mathcal{A}_t^m) = \mu(\{u \in \mathbb{S}^{m-1} \mid |u_m| \geq t\})$ **decreases wahp**
- **Can we infer the same for** $\mu(\mathcal{B}_t^m) = \mu(\{u \in \mathbb{S}^{m-1} \mid \left| \|Tu\|^2 - 1 \right| \geq t\})$**?**



Intermezzo: The union bound

- Events $E_1, \ldots, E_k$ such that $\mathbf{P}(E_i) \geq 1 - t$ for each $i \leq k$

- What is $\mathbf{P}(\text{all } E_i)$?

- $\mathbf{P}(\text{all } E_i) = 1 - \mathbf{P}(\text{at least one } \neg E_i) \Rightarrow$

$$\begin{aligned} \mathbf{P}\Big(\bigcap_{i \leq k} E_i\Big) &= 1 - \mathbf{P}\Big(\bigcup_{i \leq k} (\neg E_i)\Big) \geq \\ \geq 1 - \sum_{i=1}^{k} \mathbf{P}(\neg E_i) &= 1 - \sum_{i=1}^{k} (1 - (1 - t)) = 1 - kt \end{aligned}$$

- So $\mathbf{P}(\text{all } E_i) \geq 1 - kt$

Where the $\varepsilon^{-2} \ln n$ comes from

- B = set of unit vectors; by "intuitive explanation"
  $\Rightarrow \forall u \in B\ \exists \mu, \nu' > 0$ s.t. $\mathbf{P}(1 - t \leq \|Tu\| \leq 1 + t) \geq 1 - \mu e^{-\nu' m t^2}$
- Fix[1] m and union bound:
  $\Rightarrow\ \exists \nu$ s.t. $\mathbf{P}(\forall u \in B\ 1 - t \leq \|Tu\| \leq 1 + t) \geq 1 - |B| \mu e^{-\nu t^2}$
- Prob. $\in [0, 1] \Rightarrow$ require $1 - |B| \mu e^{-\nu t^2} > 0$:
  $\Rightarrow |B| \mu e^{-\nu t^2} < 1$
- Let $t = \varepsilon \sqrt{k}$:
  $\Rightarrow |B| \mu e^{-\nu \varepsilon^2 k} < 1$
- $\Rightarrow k > \nu \varepsilon^{-2} \ln(|B|) + \chi$, where $\chi = \frac{\ln \mu}{\nu \varepsilon^2}$ is a fixed constant
- $\Rightarrow \exists$ constant C s.t. $\boxed{k > C \varepsilon^{-2} \ln(|B|)}$

[1] In this explanation, $C = C(m)$; but C can be made independent of m

Apply to vector differences

- **Let** $A \subset \mathbb{R}^m$**,** $|A| = n$
- $\forall x, y \in A$ **we have**

$$\|Tx - Ty\|^2 = \|T(x-y)\|^2 = \|x-y\|^2 \Big\|T\frac{x-y}{\|x-y\|}\Big\|^2 = \|x-y\|^2\|Tu\|^2$$

- $\mathbb{E}(\|Tu\|^2) = \|u\| = 1 \Rightarrow \boxed{\mathbb{E}(\|Tx-Ty\|^2) = \|x-y\|^2}$
- **Let** $B = \{\frac{x-y}{\|x-y\|} \mid x, y \in A\}$
  $|B| = O(n^2) \Rightarrow k = C\varepsilon^{-2}\ln(n)$ **for some constant** C
- **By concentration of measure on** $\mathcal{B}^m$**,** $\exists \varepsilon \in (0,1)$ **s.t.**

$$(1-\varepsilon)\|x-y\|^2 \le \|Tx-Ty\|^2 \le (1+\varepsilon)\|x-y\|^2 \quad (*)$$

  holds with positive probability
- **Probabilistic method:** $\exists T$ **such that** $(*)$ **holds**
  This is the statement of the Johnson-Lindenstrauss Lemma

Randomized algorithm

- Distortion has low probability [Gupta 02]:

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

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

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

JLL follows by probabilistic method

Subsection 2

Projecting feasibility

Easy cases

Thm.

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

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

by linearity of T

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

by JLL applied to $\|b - \sum_i x_i A_i\|$

(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**)**

by union bound

[Vu et al., Discr. Appl. Math. 2019]

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 $A \subseteq \mathbb{R}^m$
- **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
- Can also project point-to-cone distances

Projection of separating hyperplanes

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:

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

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 ("squared variant") 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 $\langle Tc, TA_i\rangle \geq 0$

[Vu et al., Math. OR, 2018]

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 3

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** 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.)

more precisely, $(*)$ **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. (12) equivalent to P for $\delta = 0$

$$\left.\begin{array}{rcl} cx & \leq & v(P) - \delta \\ Ax & = & b \\ x & \geq & 0 \end{array}\right\} \qquad (12)$$

  Note: for $\delta > 0$, Eq. (12) 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 $\delta > 0$

- Restate: $cx < v(P) - \delta \wedge TAx = Tb \wedge x \geq 0$ 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 4

Solution retrieval

Projected solutions are infeasible in P

- $Ax = b \Rightarrow TAx = Tb$ by linearity
- However,
  Thm.
  $\boxed{\text{For } x \geq 0 \text{ s.t. } TAx = Tb,\ Ax = b \text{ with probability zero}}$

  if not, an x belonging to $(n-k)$-dim. subspace would belong to an $(n-m)$-dim. subspace (with $k \ll m$) with positive probability
- Can't get solution for original LFP using projected LFP!

Solution retrieval by duality

- Primal $\min\{c^\top x \mid Ax = b \wedge x \geq 0\} \Rightarrow$ dual $\max\{b^\top y \mid A^\top y \leq c\}$
- Let $x' = \mathsf{sol}(TP)$ and $y' = \mathsf{sol}(\mathsf{dual}(TP))$
- $\Rightarrow (TA)^\top y' = (A^\top T^\top)y' = A^\top(T^\top y') \leq c$
- $\Rightarrow T^\top y'$ is a solution of $\mathsf{dual}(P)$
- $\Rightarrow$ **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 $A_H^\top A_H x_H = A_H^\top b$
  $\Rightarrow x_H = (A_H^\top A_H)^{-1} A_H^\top b$
- let $x = (x_H, 0)$
- *Can prove small feasibility error wahp*
- **ISSUE**: may be slightly infeasible
  empirically: $x \not\geq 0$ *but* $x^- = \min(0, x) \to 0$ *as* $k \to \infty$

Subsection 5

Application to quantile regression

Conditional random variables

- **random variable B conditional on $A_1, \ldots, A_p$**
- **assume B depends linearly on $\{A_j \mid j \leq p\}$**
- **want to find $x_1, \ldots, x_n \in \mathbb{R}$ s.t.**

$$B = \sum_{j \leq p} x_j A_j$$

- **use samples $b, a_1, \ldots, a_p \in \mathbb{R}^m$ to find estimates**
- **$a^i =$ row, $a_j =$ column**

Sample statistics

- expectation:

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

- **conditional expectation (*linear regression*):**

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

- sample median:

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

- **conditional sample median: *similarly***

Quantile regression

- sample τ-quantile:

$$\hat{\xi} = \mathop{\mathbf{arg\,min}}_{\xi \in \mathbb{R}} \sum_{i \le m} \left(\tau \,\mathbf{max}(b_i - \xi, 0) - (1-\tau)\,\mathbf{min}(b_i - \xi, 0)\right)$$

- **conditional sample τ-quantile (*quantile regression*):**

$$\hat{\beta} = \mathop{\mathbf{arg\,min}}_{\beta \in \mathbb{R}^p} \sum_{i \le m} \left(\tau \,\mathbf{max}(b_i - \beta a^i, 0) - (1-\tau)\,\mathbf{min}(b_i - \beta a^i, 0)\right)$$

Linear Programming formulation

Let $A = (a_j \mid j \leq n)$**; then**

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

- **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 `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 pixels
  - $m = 14596, p = 24300$
  - 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
  - could not load dataset in R
- Results ⇒ LP too large, projected LP can be solved

Electricity prices

- Every hour over 365 days in 2015 (8760 rows)
- From 22 countries (columns) from the European zone

	orig	proj
1	5.82e-01	5.69e-01
2	9.46e-02	0
3	0	0
4	1.06-01	1.18e-01
5	2.73e-04	6.92e-05
6	-4.81e-06	-2.07e-05
7	1.32e-01	1.36e-01
8	0	0
9	0	0
10	0	0
11	-3.46e-08	-2.45e-05
12	0	0
13	5.66e-02	5.49e-02
14	-2.50e-04	2.91e-03
15	2.86e-02	2.81e-02
16	0	0
17	0	0
18	0	9.35e-02
19	0	0
20	2.23e-09	0
21	0	-7.99e-06

- Permutation (18,2) (21,20) applied to proj gives same nonzero pattern and reduces ℓ_2 error from 0.13 to 0.01
- For every proj solution I found I could always find a permutation with this property!!
- …On closer inspection, many columns reported equal data
- *Small numerical error*
- *Approximate solutions respect Nonzero pattern*
- LP too small for approximation to have an impact on CPU time

Outline

Subsection 1

Motivation

Coding problem for costly channels

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

  1. **Sample full rank $m \times n$ encoding matrix A with $m \leq n$**
     both parties know A

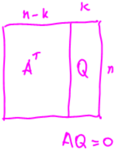
  2. **Encode $b = Ay \in \mathbb{R}^m$**

  3. **Send b**

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

Coding problem for noisy channels

- **Need to send a "word" $w \in \mathbb{R}^d$ on a noisy channel**
- **Encoding $n \times d$ matrix Q, with $n > d$, 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 $m \times n$ matrix A s.t. $m = n - d$ and $AQ = 0$**
  - **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 $\|w - w'\|$?

Subsection 2

Basis pursuit

Sparsest solution of a linear system

- **Problem $P^0(A,b) \equiv \min\{\|x\|_0 \mid Ax = b\}$ is NP-hard**

  Reduction from Exact Cover by 3-Sets [Garey&Johnson 1979, A6[MP5]]
- **Relax to $P^1(A,b) \equiv \min\{\|x\|_1 \mid Ax = b\}$**
- **Reformulate to LP:**

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

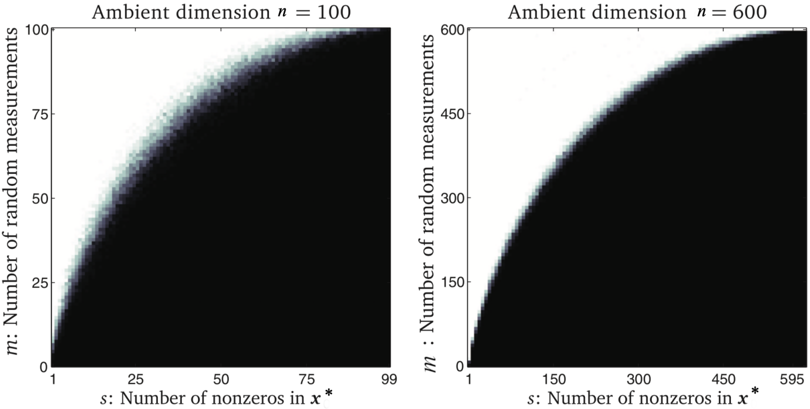
- **Empirical observation: can often find optimum**

  Too often for this to be a coincidence
- **Theoretical justification by Candès, Tao, Donoho**

  "Mathematics of sparsity", "Compressed sensing", "Compressive sampling"

Phase transition in sparse recovery

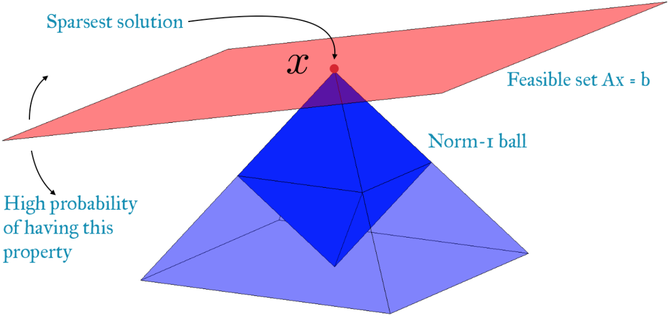
Consider $P^1(A, b)$ where A is $m \times n$



Probability that solution x^* of randomly generated P has sparsity s

[Tropp et al., Information and Inference, 2014]

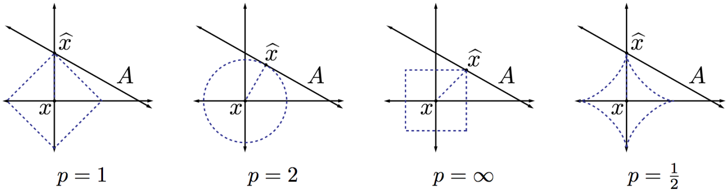
Graphical intuition 1



- **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!

Subsection 3

Theoretical results

Main theorem and proof structure

- **Thm. If:**
  - $\hat{x} \in \mathbb{R}^n$ has t nonzeros and $n-t$ near-zeros or zeros
  - $\bar{x}$ closest approx of $\hat{x}$ with exactly t nonzeros
  - $A \sim \mathsf{N}(0,1)^{mn}$ with $m < n$ but not too small
  - $b = A\hat{x}$ and x^* is the unique min of $P^1(A,b)$

  then x^* is a good approximation of $\bar{x}$
- **Prop.** If A has the null space property (NSP), result follows
- **Prop.** If A has restricted isometry property (RIP), NSP follows
- **Prop.** If $A \sim \mathsf{N}(0,1)^{mn}$, then A has RIP

Some notation

- **Consider** $Ax = b$ **where** A **is** $m \times n$ **with** $m < n$
  $\Rightarrow$ *if feasible it has uncountably many solutions*
- **Let** $x \in \mathbb{R}^n$ **s.t.** $Ax = b$, $N_A = \mathsf{null}(A)$, $N_A^0 = N_A \smallsetminus \{0\}$
  $\Rightarrow \forall y \in N_A$ **we have** $A(x+y) = Ax + Ay = Ax + 0 = b$
- **For** $z \in \mathbb{R}^n$ **and** $S \subseteq N = \{1, \ldots, n\}$ **let** $S' = N \smallsetminus S$
  define $z[S] = ((z_j \text{ iff } j \in S) \text{ xor } 0 \mid j \leq n)$
  restriction of z *to* S
- **Note that** $\forall z \in \mathbb{R}^n$ **we have** $z = z[S] + z[S']$

Null space property

- **Defn.** $\mathsf{NSP}_s(A) \equiv$
  $\forall S \subseteq \{1, \ldots, n\} \big(|S| = s \;\rightarrow\; \forall y \in N_A^0 \;\; \|y[S]\|_1 < \|y[S']\|_1\big)$
  A has the *null space property of order* s

- **Choose solution x^* of $Ax = b$ with min ℓ_1 norm**
  Let $S = \mathsf{supp}(x^*)$ and suppose $|S| = t$

- **Prop.** $\forall x^* \in \mathbb{R}^n$ **with** $|\mathsf{supp}(x^*)| = t$ **and** $b = Ax^*$
  x^* **unique min of** $P^1(A, b)$ **iff** $\mathsf{NSP}_t(A)$

Proof of the proposition ($\Rightarrow$)

$[\forall x^*$ **(**x^* **uniq min of** $P^1(A, Ax^*)$**)]** $\Rightarrow \mathsf{NSP}_t(A)$

- Suppose x^* unique soln of $P^1(A, b)$ with $b = Ax^*$
- Let $y \in N_A^0$ and $S \subseteq \{1, \ldots, n\}$ with $|S| = t$
- Since $|\mathsf{supp}(y[S])| = t$
  $y[S]$ unique min of $P^1(A, A\, y[S])$ by hypothesis
- $y = y[S] + y[S'] \in N_A \Rightarrow 0 = Ay = A(y[S]) + A(y[S'])$
  $\Rightarrow A(-y[S']) = Ay[S] \neq 0$
- By uniqueness, $\|A(-y[S'])\|_1 > \|Ay[S]\|_1$ as claimed

Proof of the proposition ($\Leftarrow$)

$\mathsf{NSP}_t(A) \Rightarrow \forall x^*$, **unique min** $P^1(A, Ax^*)$ **is** x^*

- **Let** $S = \mathsf{supp}(x^*)$ **and** $|S| = t$
- **Let** $\bar{x}$ **soln. of** $Ax = b$**, then** $\bar{x} = x^* - y$ **with** $y \in N_A$

$$
\begin{aligned}
\|x^*\|_1 = \|(x^* - \bar{x}[S]) + \bar{x}[S]\|_1 \quad &\leq \quad \text{[by triangle inequality]} \\
\leq \|x^* - \bar{x}[S]\|_1 + \|\bar{x}[S]\|_1 \quad &= \quad \text{[since } S = \mathsf{supp}(x^*)\text{]} \\
= \|x^*[S] - \bar{x}[S]\|_1 + \|\bar{x}[S]\|_1 \quad &= \quad \text{[since } x^* - \bar{x} = y\text{]} \\
= \|y[S]\|_1 + \|\bar{x}[S]\|_1 \quad &< \quad \text{[by } \mathsf{NSP}_t(A)\text{]} \\
< \|y[S']\|_1 + \|\bar{x}[S]\|_1 \quad &= \quad \text{[since } x^*[S'] = 0 \wedge y = x^* - \bar{x}\text{]} \\
= \|-\bar{x}[S']\|_1 + \|\bar{x}[S]\|_1 \quad &= \quad \text{[since } \|-z\|_1 = \|z\|_1 \wedge z[S] + z[S'] = z\text{]} \\
&= \quad \|\bar{x}\|_1
\end{aligned}
$$

A variant of the null space property

- **Motivation: "almost sparse solutions"**
  given $\hat{x}$ **with** $\mathsf{supp}(\hat{x}) \geq t$ **and** $b = A\hat{x}$**, assume**
  $\exists \epsilon > 0$ **s.t.** $\bar{x} = \max(0, x - \mathbf{1}\epsilon)$ **has** $\mathsf{supp}(\bar{x}) = t$
  i.e. $\hat{x}$ *"almost" has support size* t

- **Find closest approx** x^* **of** $\hat{x}$ **with** $\mathsf{supp}(x^*) = t$

- **Adapt null space property:** $\mathsf{NSP}_t^\rho(A) \Leftrightarrow$
  $\exists \rho \in (0,1)\ \forall S \subseteq N\ (|S| = t \rightarrow \|y[S]\|_1 \leq \rho \|y[S']\|_1)$

- **Prop.** $\mathsf{NSP}_t^\rho(A) \Rightarrow$ **if** x^* **min of** $P^1(A, A\hat{x})$ **then**

$$\|x^* - \hat{x}\|_1 \leq 2\tfrac{1+\rho}{1-\rho}\|\bar{x} - \hat{x}\|_1$$

  Moreover, if $\mathsf{supp}(\hat{x}) = t$ **then** $x^* = \hat{x} = \bar{x}$

$$\boxed{\textit{i.e. } x^* \textit{ is a good approximation of } \bar{x}}$$

Pf. see Thm. 5.8 in [Damelin & Miller 2012]

Restricted isometry property

- $\mathsf{RIP}_t^\delta(A) \quad \Leftrightarrow \quad \forall x \in \mathbb{R}^n$ **s.t.** $\mathsf{supp}(x) = t$ **we have**

$$(1-\delta)\,\|x\|_2^2 \quad \leq \quad \|Ax\|_2^2 \quad \leq \quad (1+\delta)\,\|x\|_2^2$$

- **Prop.** $\mathsf{RIP}_{2t}^\delta(A) \ \wedge \ \rho = \frac{\sqrt{2}\delta}{1-\delta} < 1 \quad \Rightarrow \quad \mathsf{NSP}_t^\rho(A)$

  Pf. see Thm. 5.12 in [Damelin & Miller 2012]

- **It suffices that** $\delta < \frac{1}{1+\sqrt{2}} \approx 0.4142$

RIP and eigenvalues

- **Recall** $\mathsf{RIP}_t^\delta(A)$**:** $\forall x$ **with** $S = \mathsf{supp}(x)$ **and** $|S| = t$
$$(1-\delta)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\delta)\|x\|_2^2$$
- **Let** $A_T = (A_i \mid i \in T)$**, where** A_i **is the** i**-th col. of** A
- $\|Ax\|_2^2 = \langle Ax, Ax\rangle = \langle A_S x[S], A_S x[S]\rangle = \langle A_S^\top A_S x[S], x[S]\rangle$
- **Since** A_S **is** $m \times t$, $B^S = A_S^\top A_S$ **is** $t \times t$ **and PSD**
- **Moreover,** $\lambda_{\mathsf{min}}(B^S)\|x\|_2^2 \le \langle B^S x, x\rangle \le \lambda_{\mathsf{max}}(B^S)\|x\|_2^2$
- **Let** $\lambda^L = \min\limits_{|S|=t} \lambda_{\mathsf{min}}(B^S)$, $\lambda^U = \max\limits_{|S|=t} \lambda_{\mathsf{max}}(B^S)$
- $\Rightarrow 1-\delta \le \lambda^L \le \lambda^U \le 1+\delta$

RIP and $P^0(A, b)$

- **Recall $P^0(A, b) \equiv \min\{\|x\|_0 \mid Ax = b\}$ is NP-hard**
  find solution to $Ax = b$ with smallest support size

- **Thm. Let $\hat{x} \in \mathbb{R}^n$ with $|\mathsf{supp}(x)| = t$, $\delta < 1$, A s.t. $\mathsf{RIP}^{\delta}_{2t}(A)$, $x^* = \mathsf{arg}P^0(A, A\hat{x})$; then $x^* = \hat{x}$**

  Pf. Suppose false, let $y = x^* - \hat{x} \neq 0$; by defn of x^* we have $\|x^*\|_0 \leq \|\hat{x}\|_0 \leq t$, hence $\|y\|_0 \leq 2t$, so since A has RIP get $\|Ay\|_2^2 \in (1 \pm \delta)\|y\|_2^2$, but $Ay = Ax^* - A\hat{x} = 0$ while $y \neq 0$, and $\delta \in (0, 1) \to 1 \pm \delta > 0$, hence $0 \in (\alpha, \beta)$ where $\alpha, \beta > 0$, contradiction

  Thm. 23.6 [Shwartz & Ben-David, 2014]

Construction of A s.t. $\mathsf{RIP}_t^\delta(A)$

- **Need $\lambda \approx 1$ for each eigenvalue λ of B^S**
- $\Rightarrow$ **Need** $\quad \forall S \subseteq N \quad |S| = t \;\rightarrow\; A_S^\top A_S \approx I_t$
- $\Rightarrow$ **Need**

$$\begin{array}{rcl} \forall i < j \leq n \quad A_i^\top A_j & \approx & 0 \\ \forall i \leq n \quad A_i^\top A_i = \|A_i\|_2^2 & \approx & 1 \end{array}$$

- **Sufficient condition:** A **sampled from** $\mathsf{N}\big(0, \frac{1}{\sqrt{m}}\big)^{mn}$
- *Difference with JLL*

  RIP holds for uncountably many vectors x with $|\mathsf{supp}(x)| = t$
  JLL holds for given sets of finitely many vectors with any support

Random matrices with orthogonal columns

1. **Defn.** Rnd vect $A_i \in \mathbb{R}^m$ is *isotropic* iff $\mathsf{cov}(A_i) = I_m$
   remark: (a) $\mathsf{cov}(X) = \mathsf{E}(XX^\top)$; (b) if $A_i \sim \mathsf{N}(0,1)^m$ then A_i isotropic
2. An isotropic rnd vect A_i is s.t. $\forall x \in \mathbb{R}^m\ \mathsf{E}(\langle A_i, x\rangle^2) = \|x\|_2^2$
   For two sq. symm. matrices B, C we have $B = C$ iff $\forall x\ (x^\top Bx = x^\top Cx)$; hence $x^\top \mathsf{E}(A_i A_i^\top)x = x^\top I_m x$; LHS is $\mathsf{E}(\langle A_i, x\rangle^2)$, RHS is $\|x\|_2^2$
3. An isotropic rnd vect A_i in $\mathbb{R}^m$ is s.t. $\mathsf{E}(\|X\|_2^2) = m$
   $\mathsf{E}(\|X\|_2^2) = \mathsf{E}(X^\top X) = \mathsf{E}(\mathsf{tr}(X^\top X)) = \mathsf{E}(\mathsf{tr}(XX^\top)) = \mathsf{tr}(\mathsf{E}(XX^\top)) = \mathsf{tr}(I_m) = m$
4. Indep isotr rnd vect A_i, A_j in $\mathbb{R}^m$ have $\mathsf{E}(\langle A_i, A_j\rangle^2) = m$
   By conditional expectation $\mathsf{E}(\langle A_i, A_j\rangle^2) = \mathsf{E}_{A_j}(\mathsf{E}_{A_i}(\langle A_i, A_j\rangle^2 \mid A_j))$; by Item 2 inner expectation is $\|A_j\|_2^2$, by Item 3 outer is m
5. If $A_i \sim \mathsf{N}(0,1)^m$, $\|A_i\|_2 \sim \sqrt{m}$ wahp
   by Thm. 3.1.1 in [Vershynin, 2018]
6. Independent rnd vectors are almost orthogonal
   Results above $\Rightarrow \|A_i\|_2, \|A_j\|_2, \langle A_i, A_j\rangle \sim \sqrt{m}$, normalize A_i, A_j to $\bar{A}_i, \bar{A}_j$ to get $\langle \bar{A}_i, \bar{A}_j\rangle \sim 1/\sqrt{m} \Rightarrow$ for m large $\langle \bar{A}_i, \bar{A}_j\rangle \approx 0$

Construction of A s.t. $\mathsf{RIP}_t^\delta(A)$

- **Thm. For $A \sim \mathsf{N}(0,1)^{m\times n}$ and $\delta \in (0,1)$ $\exists c_1, c_2 > 0$ depending on δ s.t.**

$$\forall t < m \left(\frac{t}{c_1} \ln\left(\frac{n}{t}\right) \le m \to \mathsf{Prob}(\mathsf{RIP}_t^\delta(A)) \ge 1 - e^{c_2 m} \right)$$

Pf. see Thm. 5.17 in [Damelin & Miller, 2012]

Remark: extra $\sqrt{m}$ factor in A comes from $\|\cdot\|_2 \le \|\cdot\|_1 \le \sqrt{m}\|\cdot\|_2$

- **In practice:**
  - $\mathsf{Prob}(\mathsf{RIP}_t^\delta(A)) = 0$ for m too small w.r.t. t fixed
  - as m increases $\mathsf{Prob}(\mathsf{RIP}_t^\delta(A)) > 0$
  - as m increases even more $\mathsf{Prob}(\mathsf{RIP}_t^\delta(A)) \to 1$ wahp
  - achieve logarithmic compression for large n and fixed t
  - $A \sim \mathsf{N}(0,1)^{mn} \wedge m \ge 10t \ln \frac{n}{t} \Rightarrow \mathsf{RIP}_t^{\frac{1}{3}}(A)$ wahp, Lem. 5.5.2 [Moitra 2018]
  - works better than worst case bounds ensured by theory

Some literature

1. Damelin & Miller, The mathematics of signal processing, CUP, 2012
2. Vershynin, High-dimensional probability, CUP, 2018
3. Moitra, Algorithmic aspects of machine learning, CUP, 2018
4. Shwartz & Ben-David, Understanding machine learning, CUP, 2014
5. Hand & Voroninski, `arxiv.org/pdf/1611.03935v1.pdf`
6. Candès & Tao
   `statweb.stanford.edu/~candes/papers/DecodingLP.pdf`
7. Candès
   `statweb.stanford.edu/~candes/papers/ICM2014.pdf`
8. Davenport *et al.*, `statweb.stanford.edu/~markad/publications/ddek-chapter1-2011.pdf`
9. Lustig *et al.*, `people.eecs.berkeley.edu/~mlustig/CS/CSMRI.pdf`

and many more (look for "compressed sensing")

Subsection 4

Application to noisy channel encoding

Noisy channel encoding procedure

Algorithm:

1. **message:** ***character string*** s
2. $w = \mathsf{string2bitlist}(s) \in \{0, 1\}^d$
3. **send** $z = Qw$**, receive** $\bar{z} = z + \hat{x}$**, let** $b = A\bar{z}$
   $\Delta =$ *density of* $\hat{x}$*,* $\quad Q$ *is* $n \times d$ *full rank with* $n > d$
4. $x^* = \arg P^1(A, b)$
5. $z^* = \bar{z} - x^*$
6. $w^* = \mathsf{cap}(\mathsf{round}(\,(Q^\top Q)^{-1} Q^\top z^*), [0, 1])$
7. $s^* = \mathsf{bitlist2string}(w^*)$
8. **evaluate** $s_{\mathsf{err}} = \|s - s^*\|$

Parameter choice [Matousek]:

- $\Delta = 0.08$
- $n = 4d$

Finding orthogonal A, Q

- [Matousek, Gärtner 2007]:
  - sample A componentwise from $N(0,1)$
  - then "find Q s.t. $QA = 0$"
  - *Gaussian elim. on underdet. system* $AQ = 0$
- Faster:
  - sample $n \times n$ matrix M from uniform distr
    full rank with probability 1
  - find eigenvector matrix of $M^\top M$ (orthonormal basis)
    random rotation of standard basis (used in original JLL proof)
  - Concatenate d eigenvectors to make Q, and $m = n - d$ to make A
    $AQ = 0$ *by construction!*

Subsection 5

Improvements

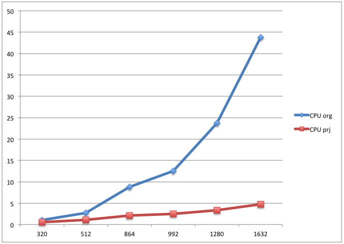
LP size reduction

- **Motivation**
  - Reduce CPU time spent on LP
  - $n = 4d$ redundancy for $\Delta = 0.08$ error seems excessive
- **Size of basis pursuit LP**
  - $Ax = b$ is an $m \times n$ system where $m = n - d$
  - If $n \gg d$, m "relatively close" to n
  - *Recall random projections for LP: use them!*

Computational results

d	n	Δ	ϵ	α	$s_{\text{err}}^{\text{org}}$	$s_{\text{err}}^{\text{prj}}$	CPU$^{\text{org}}$	CPU$^{\text{prj}}$
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

- $d = |s|, n = 4d, \Delta = 0.08, \epsilon = 0.2$
- $\alpha =$ Achlioptas density
  $\mathsf{P}(T_{ij} = -1) = \mathsf{P}(T_{ij} = 1) = \frac{\alpha}{2}$
  $\mathsf{P}(T_{ij} = 0) = 1 - \alpha$
- $s_{\text{err}} =$ 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 $n = (1+\Delta)d$?*
- $m = n - d \approx \Delta d$ **is very small**
- **Makes $Ax = b$ very short and fat**
- **Prevents compressed sensing from working correctly**
  not enough constraints
- **Would need both m and d to be $\approx n$ and $AQ = 0$: impossible**
  $\mathbb{R}^n$ *too small to host* $m + d \approx 2n$ *orthogonal vectors*
- **Relax to $AQ \approx 0$?**

Almost orthogonality by the JLL

Aim at $A^\top, Q$ with $m + d \approx 2n$ and $AQ \approx 0$

- **JLL Corollary:** $\exists O(e^k)$ *approx orthog vectors in* $\mathbb{R}^k$

  Pf. Let T be a $k \times p$ random projector (RP), use conc. meas. on $\|z\|_2^2$

  $$\mathsf{Prob}(\,(1-\varepsilon)\|z\|_2^2 \le \|Tz\|_2^2 \le (1+\varepsilon)\|z\|_2^2\,) \ge 1 - 2e^{-\mathcal{C}(\varepsilon^2-\varepsilon^3)k}$$

  given $x, y \in \mathbb{R}^n$ apply to $x+y, x-y$ and union bound:

  $$\begin{array}{rcl} |\langle Tx, Ty\rangle - \langle x, y\rangle| & = & \frac{1}{4}\left|\|T(x+y)\|^2 - \|T(x-y)\|^2 - \|x+y\|^2 + \|x-y\|^2\right| \\ & \le & \frac{1}{4}\left|\|T(x+y)\|^2 - \|x+y\|^2\right| + \frac{1}{4}\left|\|T(x-y)\|^2 - \|x-y\|^2\right| \\ & \le & \frac{\varepsilon}{4}(\|x+y\|^2 + \|x-y\|^2) = \frac{\varepsilon}{2}(\|x\|^2 + \|y\|^2) \end{array}$$

  with prob $\ge 1 - 4e^{-\mathcal{C}\varepsilon^2 k}$; apply to std basis mtx I_p, get

  $$-\varepsilon \le \langle T\mathbf{e}_i, T\mathbf{e}_j\rangle - \langle \mathbf{e}_i, \mathbf{e}_j\rangle \le \varepsilon$$

  $\Rightarrow \exists p$ almost orthogonal vectors in $\mathbb{R}^k$, and $k = O(\frac{1}{\varepsilon^2}\ln p) \Rightarrow p = O(e^k)$

- **Algorithm:** $k = n, p = \lceil e^n \rceil$, get $2k$ columns from $T\,I_p$

Also see [`https://terrytao.wordpress.com/2013/07/18/a-cheap-version-of-the-kabatjanskii-levenstein-bound-for-almost-orthogonal-vectors/`]

Almost orthogonality by the JLL

- **Aim at $m \times n$ A and $n \times m$ Q s.t. $AQ \approx 0$**
  with $n = (1 + \Delta')m$ and Δ' "small" (say $\Delta' < 1$)
- **Need $2m$ approx orthog vectors in $\mathbb{R}^n$ with $n < 2m$**
  JLL errors too large for such "small" sizes

- **Note we only need $AQ = 0$:**
  can accept non-orthogonality in rows of A & cols of Q

Almost orthogonality by LP

- **Sample Q and compute A using an LP**
  WLOG: we could sample A and compute Q
- $\max \sum_{\substack{i \le m \\ j \le n}} \mathsf{Uniform}(-1,1) A_{ij}$
- **subject to** $AQ = 0$ **and** $A \in [-1, 1]$
- **for** $m = 328$ **and** $n = 590$ **(i.e.** $\Delta' = 0.8$**):**
  - **error:** $\sum A_i Q^j = O(10^{-10})$
  - **rank: full** (*not really, but* $|A| = O(\epsilon)$)
  - **CPU: 688s (*meh*)**
- **for** $m = 328$ **and** $n = 492$ **(i.e.** $\Delta' = 0.5$**):** ***the same***
- **for** $m = 328$ **and** $n = 426$ **(i.e.** $\Delta' = 0.3$**):** ***CPU 470s***
- **Reduce CPU time by solving m LPs deciding A_i (for $i \le m$)**

Computational results

m	n	Δ'	$s_{\text{err}}^{\text{org}}$	$s_{\text{err}}^{\text{prj}}$	CPU$^{\text{org}}$	CPU$^{\text{prj}}$
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!

In summary

- 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]

$m = 1896, n = 2465$

$\Delta' = 0.3$: 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

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!
Research Scientist - AWS New Artificial Intelligence Team! ...views - 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

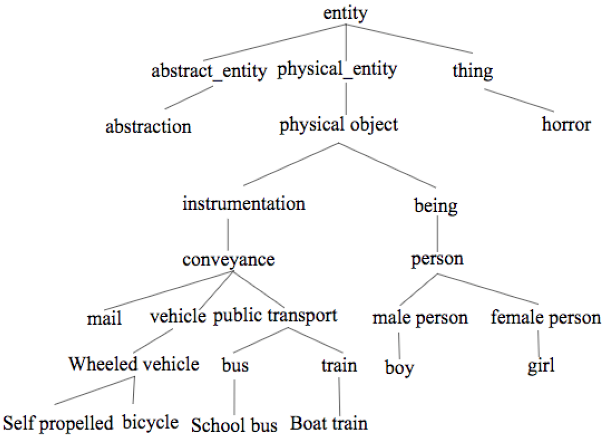
Exploit NER to infer relations

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

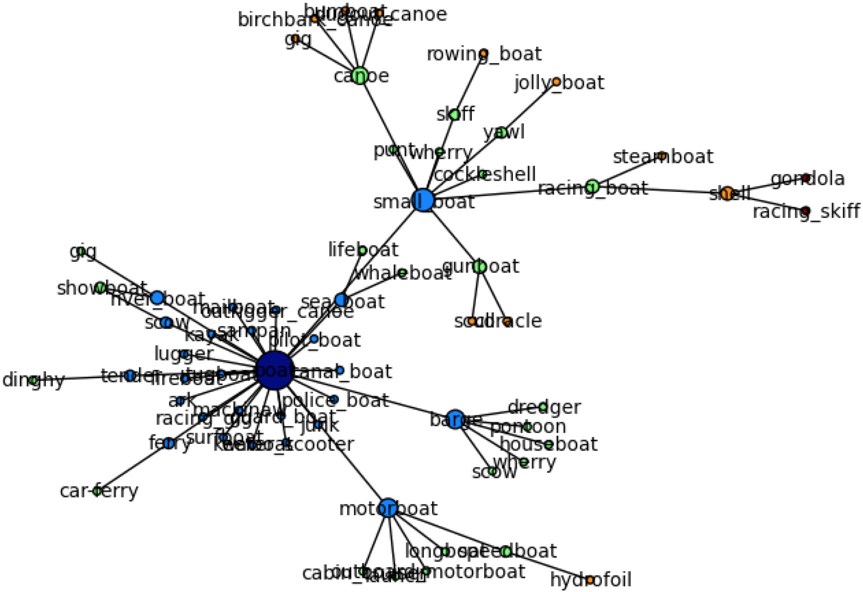
The named entities

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

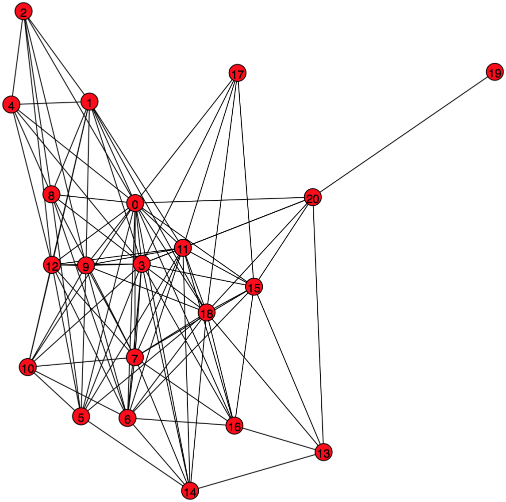
$$
\begin{pmatrix}
1.00 & 0.63 & 0.51 & 0.51 & 0.66 & 0.45 & 0.46 & 0.47 & 0.72 & 0.58 & 0.54 & 0.50 & 0.72 & 0.38 & 0.49 & 0.47 & 0.47 & 0.44 & 0.54 & 0.31 & 0.44 \\
0.63 & 1.00 & 0.45 & 0.45 & 0.54 & 0.40 & 0.42 & 0.42 & 0.57 & 0.49 & 0.46 & 0.45 & 0.59 & 0.35 & 0.43 & 0.42 & 0.42 & 0.41 & 0.47 & 0.32 & 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



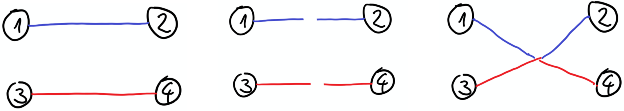
$G = (V, E)$, **weighted adjacency matrix** A

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"

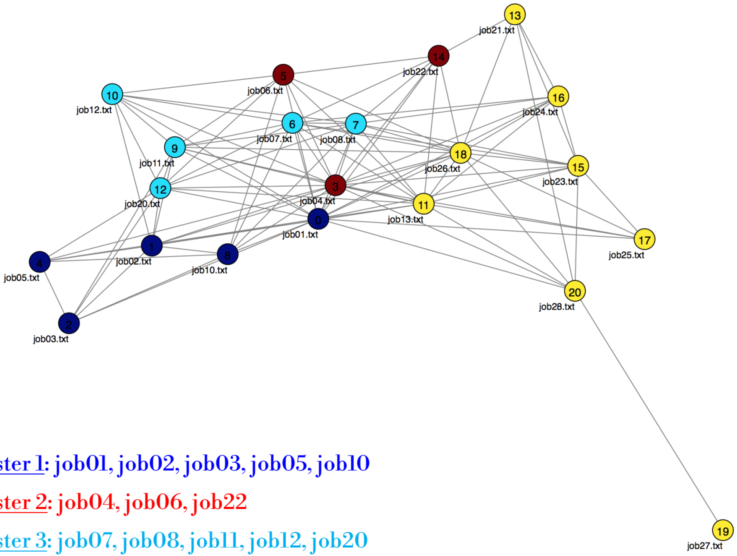
$$\left.\begin{array}{rl} \max & \displaystyle\sum_{\{u,v\}\in E} \mathsf{mod}(u,v)x_{uv} \\ \forall u \in V, v \in V & x_{uv} \in \{0,1\} \end{array}\right\}$$

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

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

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

The resulting clustering



<u>cluster 1</u>: job01, job02, job03, job05, job10

<u>cluster 2</u>: job04, job06, job22

<u>cluster 3</u>: job07, job08, job11, job12, job20

<u>cluster 4</u>: 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
```

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

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 $C_1, \ldots, C_k$

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

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

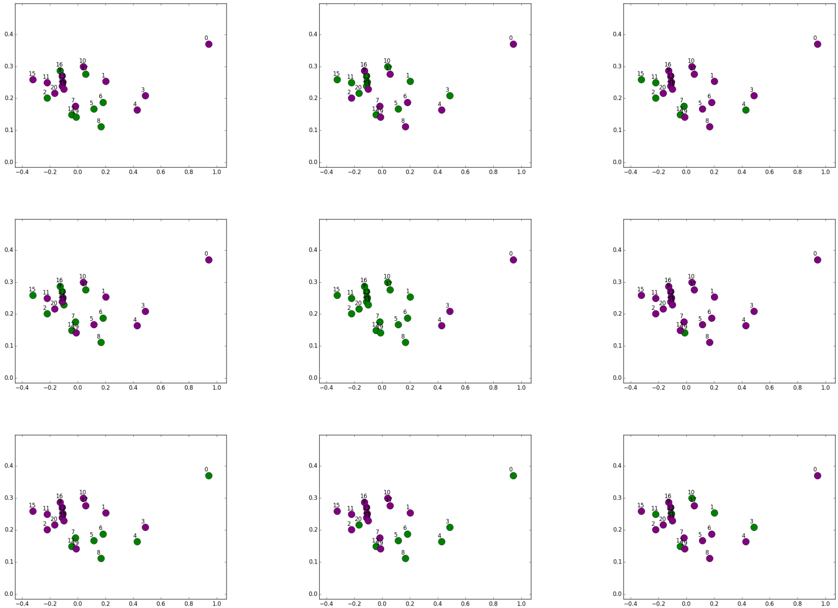
Deezer	Vinci
Elektrobit	AXA
Google	Accenture
Aledade	Alstom
	Expedia
	Ford
	Marriott
	Llamasoft
	Amazon 1-3
	CSX
	WestRock
	MITRE
	Clarity
	fragments 1-2

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

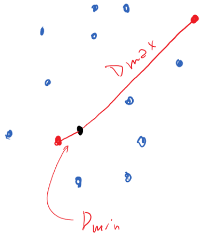
(b) $\forall y \in \mathcal{Y}, x \in \mathcal{X} \quad (d(z,y) \leq d(z,x))$

- **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**
  - $D^m_{\mathbf{min}} = \min_{i \leq \ell} d^m(Z^m, X_i^m)$
  - $D^m_{\mathbf{max}} = \max_{i \leq \ell} d^m(Z^m, X_i^m)$

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**

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

  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. $\mathbf{D}^m = ((d^m(Z^m, X_i^m))^p \mid i \le \ell) \to_{\mathbb{P}} \mathbf{1}$ (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 $D^m_{\mathbf{max}} \ge D^m_{\mathbf{min}}$)

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 <u>bounded</u> 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:**

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

- **$\Rightarrow$ 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:

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

- $\Rightarrow$ Everything's linear now!

[L. et al. 2009]

MSSC is a convex MINLP

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

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

Random projections again

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?

Compare them with clustering measures
e.g. "adjusted mutual information score"

Outline

Definition

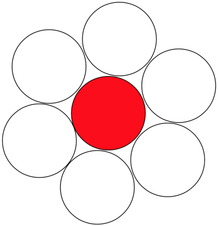
- *Optimization version.* **Given $K \in \mathbb{N}$, determine the maximum number $\mathsf{kn}(K)$ of unit spheres that can be placed adjacent to a central unit sphere so their interiors do not overlap**
- *Decision version.* **Given $n, K \in \mathbb{N}$, is $\mathsf{kn}(K) \leq n$?**

  in other words, determine whether n unit spheres can be placed adjacent to a central unit sphere so that their interiors do not overlap

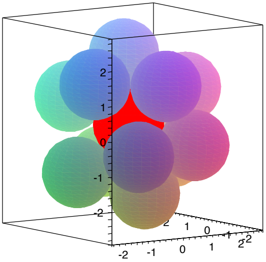
Funny story: *Newton and Gregory went down the pub...*

Some examples

$n = 6, K = 2$



$n = 12, K = 3$



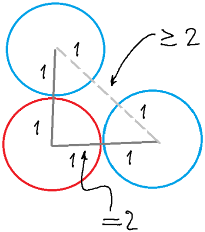
more dimensions

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:**

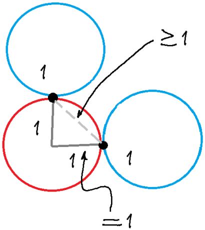
$$\begin{array}{rrcl} \forall i \le n & \|x_i\|_2^2 & = & 4 \\ \forall i < j \le n & \|x_i - x_j\|_2^2 & \ge & 4 \end{array}$$



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:**

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

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:**

$$\begin{array}{lrcl}
\forall i < j \le n & \beta_{ij} & \le & \alpha_i \\
\forall i < j \le n & \beta_{ij} & \le & \alpha_j \\
\forall i < j \le n & \beta_{ij} & \ge & \alpha_i + \alpha_j - 1
\end{array}$$

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

- $\forall i \leq n \qquad \alpha_i \in \{0,1\} \Leftrightarrow \alpha_i^2 = \alpha_i$
- **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*

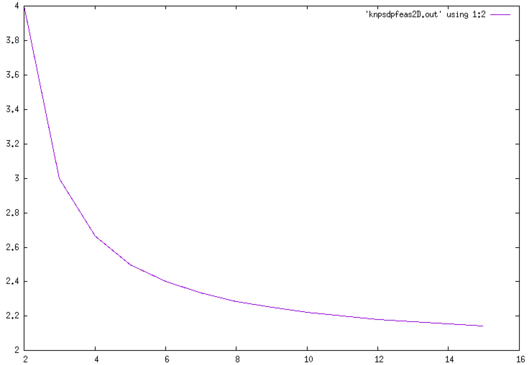
SDP relaxation of [KBLM07]

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

Computational experiments

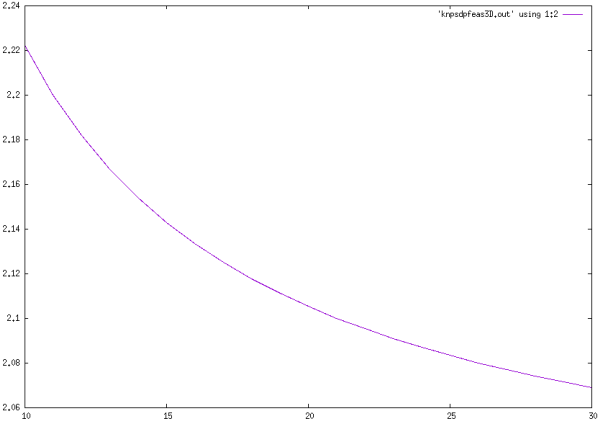
With $K = 2$

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

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

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

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

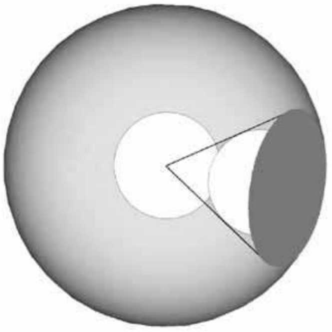
Subsection 3

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

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

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

$$\int_{[-1,1]} \sigma_t dt = \sum_{t \in [-1,1]} \sigma_t = \frac{1}{n}|\textbf{all pairs}| = \frac{n^2}{n} \quad = \quad n$$

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

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

  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:**

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

- **Dual:**

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

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

Focus on normalized [DelD]

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

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

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>* **[DelD]**$_T$*:*

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

- **Risk:** $\mathsf{val}(\textbf{[DelD]}_T) < \min F(1)/d_0$
  not a valid bound to $\mathsf{kn}(K)$
- **Happens if soln. of [DelD]$_T$ infeasible in [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** $\epsilon = \max\{a_i^\top x^* - b_i \mid i \in \bar{I}\}$
  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**

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

   get solution d^*

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

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

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

Subsection 5

Pfender's upper bound

Pfender's upper bound theorem

Thm.

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

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

([Pfender 2006])

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

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

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