

# Large-scale Mathematical Optimization

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

INF580



# About the course

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

*some disconnection between lectures and TD is normal*

  
*some theoretical topics do not lend themselves to implementation*
- ▶ **Lectures/TD:** (generally) on fri afternoon
- ▶ **Exam:** mini-project (individual/pairs) or oral exam

<http://www.lix.polytechnique.fr/~liberti/teaching/dix/inf580>

# Outline

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# Outline

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# 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*

# Why *Large-scale*?

- ▶ Any process can be optimized
- ▶ Social, technical and business processes are complex
- ▶ Computer power limits model precision
- ▶ Nowadays, need to solve very precise models  
⇒ *increase in model size*
- ▶ ⇒ algorithmic complexity must grow slowly with size
- ▶ Focus on LP algs and heuristics
- ▶ Investigate LP relaxations & dimensionality reduction methods

# The syllabus

- ▶ **Which optimization problems can be solved?**  
*a tour of 20th century logic*
- ▶ **Complexity of optimization problems**  
*basics of theoretical computer science*
- ▶ **Distance geometry**  
*modern large-scale optimization and data science techniques*
- ▶ **Random projections**  
*new approaches to approximately solving large-scale problems*
- ▶ **Sparsity and  $\ell_1$  minimization**  
*integrality out of continuity*
- ▶ **Further topics**  
*as time allows*

# MP Formulations

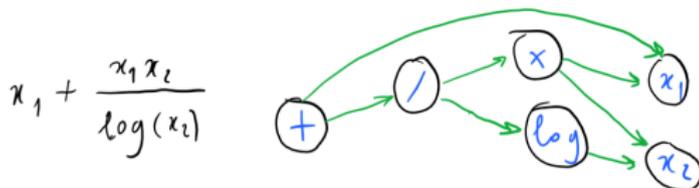
Given functions  $f, g_1, \dots, g_m : \mathbb{Q}^n \rightarrow \mathbb{Q}$  and  $Z \subseteq \{1, \dots, n\}$

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

► **More general than it looks:**

- $\phi(x) = 0 \iff (\phi(x) \leq 0 \wedge -\phi(x) \leq 0)$
- $L \leq x \leq U \iff (L - x \leq 0 \wedge x - U \leq 0)$

►  $f, g_i$  represented by *expression DAGs*



Class of all formulations  $P$ : MIP

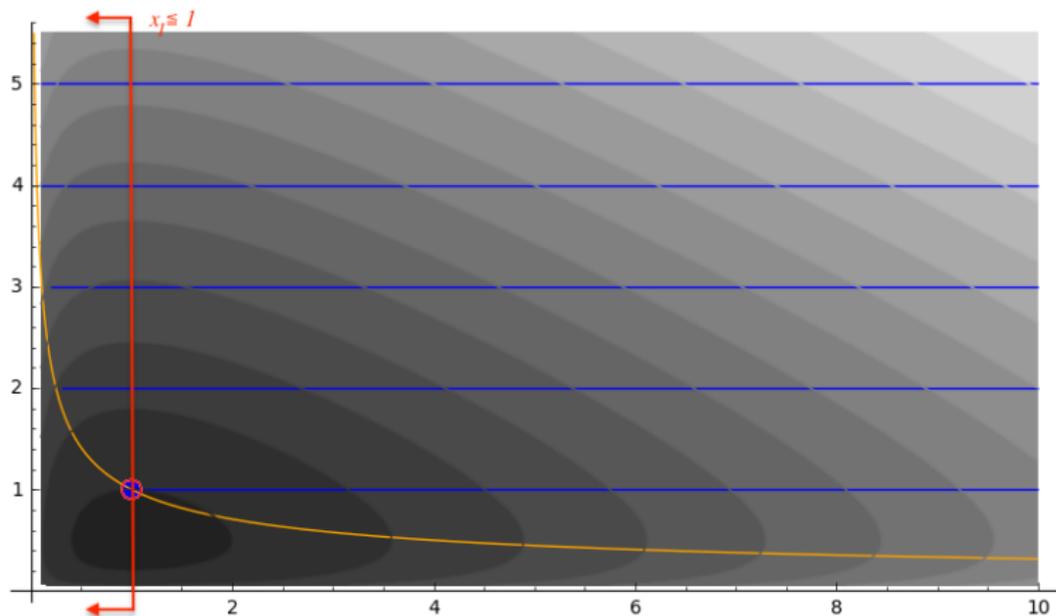
# Semantics of MP formulations

- ▶  $\llbracket P \rrbracket =$  optimum (or optima) of  $P$
- ▶ Given  $P \in \text{MP}$ , 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:

$$\left. \begin{array}{ll} \min & x_1 + 2x_2 - \log(x_1x_2) \\ & x_1x_2^2 \geq 1 \\ & x_1 \in [0, 1] \\ & x_2 \in \mathbb{N} \end{array} \right\}$$

# Example

$$P \equiv \min\{x_1 + 2x_2 - \log(x_1x_2) \mid x_1x_2^2 \geq 1 \wedge 0 \leq x_1 \leq 1 \wedge x_2 \in \mathbb{N}\}$$



$$\llbracket P \rrbracket = (\text{opt}(P), \text{val}(P))$$

$$\text{opt}(P) = (1, 1)$$

$$\text{val}(P) = 3$$

# Relationships between feasibility and optimality

- ▶ **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

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

# Bisection algorithm for optimal value

- 1: *Input:* lower & upper bound for  $f_0$
- 2: **while** lower and upper bounds differ by  $> \epsilon$  **do**
- 3:   **let**  $f_0$  **be** midway between bounds
- 4:   **if**  $Q(f_0)$  is feasible **then**
- 5:     **update upper bound to**  $f_0$
- 6:   **else**
- 7:     **update lower bound to**  $f_0$
- 8:   **end if**
- 9: **end while**

- ▶ solve an optimization problem with calls to feasibility oracle
- ▶ **need only**  $\lceil \log_2 \left( \frac{\text{UB}-\text{LB}}{\epsilon} \right) \rceil$  **calls to oracle**

# Bisection algorithm for optimum

- 1: *Input*: lower & upper bound for  $f_0$ ,  
candidate global optimum  $\hat{x}$
- 2: **while** lower and upper bounds differ by  $> \epsilon$  **do**
- 3:   **let**  $f_0$  be midway between bounds
- 4:   **if**  $Q(f_0)$  is feasible **then**
- 5:     **find** a feasible point  $x'$
- 6:     **if**  $x'$  improves  $\hat{x}$  **then**
- 7:       **update**  $\hat{x}$  to  $x'$
- 8:       **update** upper bound to  $f(\hat{x})$
- 9:     **end if**
- 10:   **else**
- 11:     **update** lower bound to  $f_0$
- 12:   **end if**
- 13: **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 emptiness up to  $\epsilon$*

# Bisection algorithm for MP (formal)

```
1: let  $(\hat{x}, \hat{f}) = (\text{uninitialized}, \bar{f})$ 
2: while  $\bar{f} - \underline{f} > \epsilon$  do
3:   let  $f_0 = (\underline{f} + \bar{f})/2$ 
4:    $(x', f') = \mathcal{A}(Q(f_0))$ 
5:   if  $(x', f') \neq (\emptyset, \emptyset)$  then
6:     if  $f' < \hat{f}$  then
7:       update  $(\hat{x}, \hat{f}) \leftarrow (x', f')$ 
8:       update  $\bar{f} \leftarrow \hat{f}$ 
9:     end if
10:  else
11:    update  $\underline{f} \leftarrow f_0$ 
12:  end if
13: 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*

- ▶  $m, n$ : number of rows and columns  
corresp. index sets  $I = \{1, \dots, m\}, J = \{1, \dots, n\}$
- ▶  $c \in \mathbb{R}^n, b \in \mathbb{R}^m, A$  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 ( $\forall, \sum, \dots$ )

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  $\llbracket P \rrbracket$  and possibly other information**
- ▶ **Trade-off between generality and efficiency**  
*fast solvers for large MP subclasses: unlikely*

# Some subclasses of MP

(i) LINEAR PROGRAMMING (LP)

$f, g_i$  linear,  $Z = \emptyset$

(ii) MIXED-INTEGER LP (MILP)

$f, g_i$  linear,  $Z \neq \emptyset$

(iii) NONLINEAR PROGRAMMING  
(NLP)

some nonlinearity in  $f, g_i, Z = \emptyset$

$f, g_i$  convex: convex NLP (cNLP)

(iv) MIXED-INTEGER NLP  
(MINLP)

some nonlinearity in  $f, g_i, Z \neq \emptyset$

$f, g_i$  convex: convex MINLP  
(cMINLP)

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

# 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)
- ▶ CPP (LP over copositive cone)
- ▶ NLP (nonlinear functions)

# Types of MP

## *Mixed-integer variables:*

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

## *Some more “exotic” classes:*

- ▶ MOP (multiple objective functions)
- ▶ BLvP (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?

## *Modelling school*

First question to ask oneself is: **What are the decision variables?**

# 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

$$\left. \begin{array}{l} \max_{x_a, x_b \geq 0} \quad c_a x_a + c_b x_b \\ \quad \quad \quad \quad \quad x_a + x_b = C \\ \quad \quad \quad \quad \quad \quad x_a \geq pC \\ \quad \quad \quad \quad \quad dx_a - x_b \geq 0 \end{array} \right\}$$

- ▶ *Formulation generality:*

extend to  $n$  investments:

$$\left. \begin{array}{l} \max_{x \geq 0} \quad \sum_{j \leq n} c_j x_j \\ \quad \quad \quad \quad \quad \sum_{j \leq n} x_j = C \\ \quad \quad \quad \quad \quad \quad x_1 \geq pC \\ \quad \quad \quad \quad \quad dx_1 - x_2 \geq 0 \end{array} \right\}$$

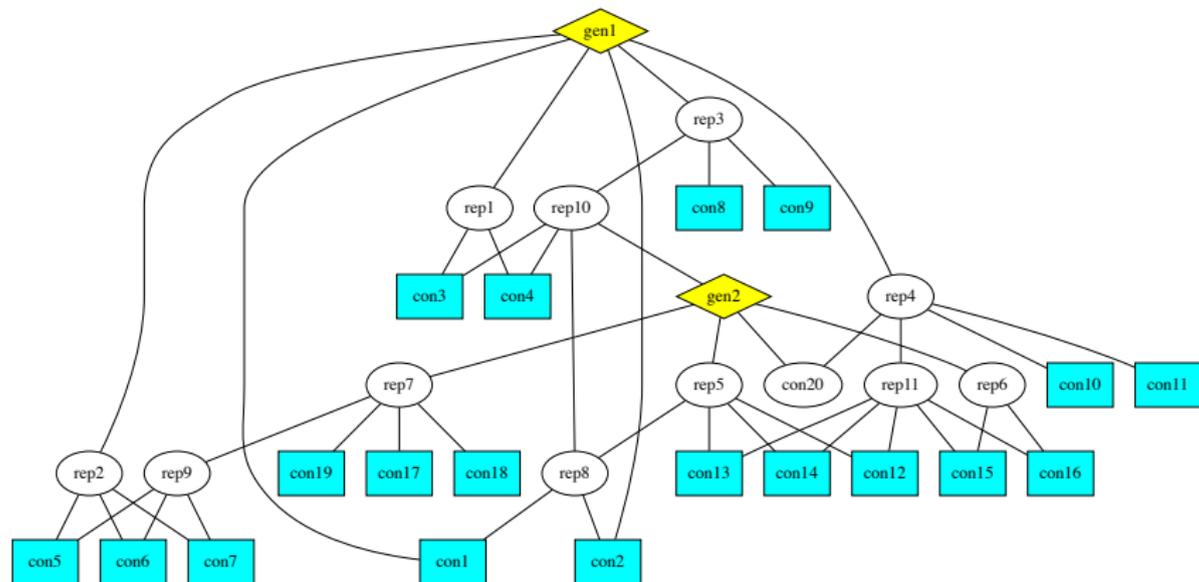
# Example: monitoring an electrical grid

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 two incident lines (cost: 0.5MEUR)
- ▶ C: installed at generator buses only, and monitors one incident line (cost: 0.3MEUR)
- ▶ D: installed at repeater buses only, and monitors one incident line (cost: 0.2MEUR)
- ▶ E: installed at consumer buses only, and monitors 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
- ▶  $D_1 = D \setminus D_M$

## ▶ Parameters:

- ▶  $\forall v \in V$   $b_v =$  bus type
- ▶  $\forall d \in D$   $c_d =$  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

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

# Example: formulation

## ► Constraints

### ► device types:

$$\forall v \in V \quad b_v = \text{gen} \quad \rightarrow \quad x_{Bv} = 0$$

$$\forall v \in V \quad b_v \in \{\text{con}, \text{rep}\} \quad \rightarrow \quad x_{Cv} = 0$$

$$\forall v \in V \quad b_v \in \{\text{gen}, \text{con}\} \quad \rightarrow \quad x_{Dv} = 0$$

$$\forall v \in V \quad b_v \in \{\text{gen}, \text{rep}\} \quad \rightarrow \quad x_{Ev} = 0$$

### ► at most one device of any 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**

$$\forall u \in V, v \in N_u \quad y_{Auv} = x_{Au}$$

- **B: two monitored lines incident to installed device**

$$\forall u \in V \quad \sum_{v \in N_u} y_{Buv} = \min(2, |N_u|)x_{Bu}$$

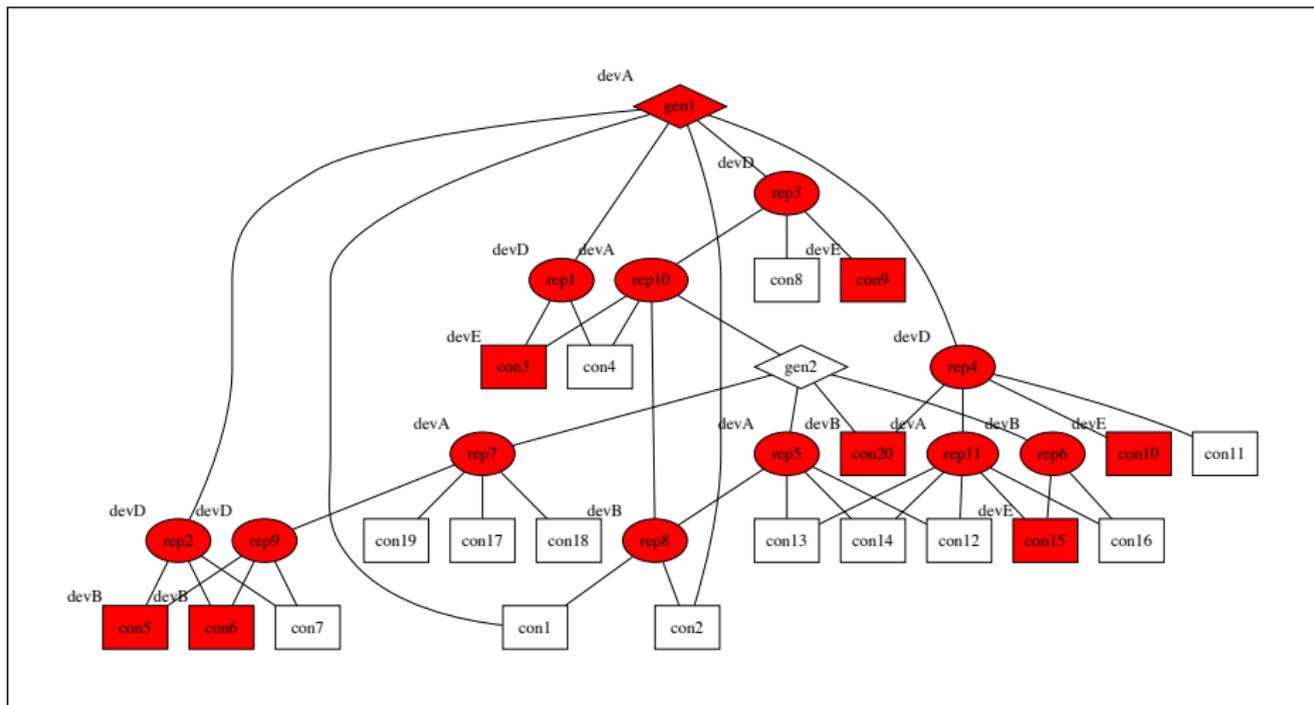
- **C,D,E: one monitored line incident to installed device**

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

- **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

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# Can we solve MPs?

- ▶ “Solve MPs”: is there an algorithm  $\mathcal{D}$  s.t.:

$$\forall P \in \text{MIP} \quad \mathcal{D}(P) = \begin{cases} \text{infeasible} & P \text{ is infeasible} \\ \text{unbounded} & P \text{ is unbounded} \\ \llbracket P \rrbracket & \text{otherwise} \end{cases}$$

- ▶ 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 *formulae* 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]

# Examples

► *Alphabets:*

$$\overline{\mathcal{B}} = \{\_, 0, 1\}, \overline{\mathcal{N}} = \overline{\mathcal{B}} \cup \{2, 3, \dots\}, \overline{\mathcal{V}} = \{x_0, x_1, x_2, \dots\}$$

$$\overline{\mathcal{E}} = \{(\_, +, -, \times, \div, \square^{\square}, \exp, \log, \sin)\} \cup \overline{\mathcal{N}} \cup \overline{\mathcal{V}}$$

$$\overline{\mathcal{PA1}} = \{\forall, \exists, \vee, \wedge, \neg, =\} \cup \overline{\mathcal{E}}, \overline{\mathcal{MP}} = \{\Sigma, \Pi, \leq, \geq\} \cup \overline{\mathcal{PA1}}$$

► *An expression grammar:*

$$\begin{aligned} \text{expr} &: \text{term} + \text{expr} \quad | \quad \text{term} - \text{expr} \quad | \quad \text{term} \\ \text{term} &: \text{factor} \times \text{term} \quad | \quad \text{factor} \div \text{term} \quad | \quad \text{factor} \\ \text{factor} &: \text{power}^{\text{power}} \quad | \quad \text{power} \\ \text{power} &: \log(\text{unr}) \quad | \quad \exp(\text{unr}) \quad | \quad \sin(\text{unr}) \quad | \quad (-\text{unr}) \quad | \quad \text{unr} \\ \text{unr} &: (\text{expr}) \quad | \quad \mathcal{N} \quad | \quad \mathcal{V} \end{aligned}$$

e.g.  $(1 - \sin(x)^2)^{(1/2)}$ :

$$\begin{aligned} \text{expr} &= \text{term} = \text{factor} = \text{power}^{\text{power}} = \text{unr}^{\text{unr}} = (\text{expr})^{(\text{expr})} = \\ &(\text{term} - \text{expr})^{(\text{term})} = (\text{factor} - \text{term})^{(\text{factor} \div \text{factor})} = \dots \end{aligned}$$

► *Axioms:* see later

► *Inference rules:*

modus ponens, symbol replacement,  $A \vdash A \wedge A$ ,  $A \vdash A \vee A$ , ...

# Example: PA1

- ▶ **Theory:** 1st order provable sentences about  $\mathbb{N}$
- ▶ **Alphabet:**  $+, \times, \wedge, \vee, \rightarrow, \forall, \exists, \neg, =, 0, S(\cdot)$  and variable names
- ▶ **Peano's Axioms:**

1.  $\forall x (0 \neq S(x))$
2.  $\forall x, y (S(x) = S(y) \rightarrow x = y)$
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  $\phi$ :**  $\forall y = (y_1, \dots, y_k) (\phi(0, y) \wedge \forall x (\phi(x, y) \rightarrow \phi(S(x), y))) \rightarrow \forall x \phi(x, y)$

- ▶ **Inference:** see

[https://en.wikipedia.org/wiki/List\\_of\\_rules\\_of\\_inference](https://en.wikipedia.org/wiki/List_of_rules_of_inference)

e.g. *modus ponens*  $(P \wedge (P \rightarrow Q)) \rightarrow Q$

- ▶ **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 of PA1 derivation

**Thm.**

$$\forall x (x \times 0 = 0)$$

**Proof**

$$x \times 0 + x \times 0 = x \times (0 + 0) = x \times 0 = x \times 0 + 0$$

$$x \times 0 = 0 \quad \text{xor} \quad x \times 0 > 0$$

$$x \times 0 > 0 \rightarrow x \times 0 + x \times 0 > x \times 0 + 0$$

$$\rightarrow x \times 0 + 0 > x \times 0 + 0 \quad \rightarrow 0 = 1$$

$$\rightarrow x \times 0 = 0 \quad \text{QED}$$

**Note that**  $A \text{ xor } B = A \vee B \wedge \neg(A \wedge B)$

# 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](https://en.wikipedia.org/wiki/List_of_rules_of_inference)

e.g. *modus ponens*  $(P \wedge (P \rightarrow Q)) \rightarrow Q$

- ▶ **Generates polynomial rings  $\mathbb{R}[X_1, \dots, X_k]$  (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:  
 $P \equiv$  provable sentences
- ▶ *Proof of  $f$* : finite sequence of sentences ending with  $f$   
*sentences*  $\in$  axioms  $\vee$  derived from predecessors by inference rules
- ▶ **PA1**: decide if sentence  $f$  about  $\mathbb{N}$  has a proof  
PA1 contains  $\exists x \in \mathbb{Z}^n \forall i p_i(x) \leq 0$  (poly  $p$ )
- ▶ **Reals**: decide if sentence  $f$  about  $\mathbb{R}$  has a proof  
Reals contains  $\exists x \in \mathbb{R}^n \forall i p_i(x) \leq 0$  (poly  $p$ )
- ▶ Formal study of MINLP/NLP feasibility

# Decidability, computability, solvability

- ▶ **Decidability:** applies to decision problems
- ▶ **Computability:** applies to function evaluation
  - ▶ Is the function mapping  $i$  to the  $i$ -th prime integer computable?
  - ▶ Is the function 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.**  $\mathcal{F}$  complete iff inconsistent

- ▶  $\phi$ : sentence “ $\phi$  not provable in  $\mathcal{F}$ ”

denoted  $\mathcal{F} \nVdash \phi$ ; it can be constructed in  $\mathcal{F}$  (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  $\neg P$ ; in particular for any  $Q$  we have  $P \vee Q$ , whence  $Q$  (since  $\neg P$  and  $P \vee Q$ ), implying  $P \wedge \neg P \rightarrow Q$

- ▶ **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  $\psi \in \text{PA1}$ ,  $\ulcorner \psi \urcorner \in \mathbb{N}$   
integer which encodes the proof  
*let me sweep the details under the carpet*
- ▶  $\ulcorner \cdot \urcorner$  is an injective map
- ▶ Inverse:  $\langle \ulcorner \phi \urcorner \rangle = \phi$   
 $\phi$  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 $\phi$

- ▶ For integers  $x, y$   $\exists g \in \mathbb{N} \langle g \rangle \equiv \text{proof}(x, y)$  :  
holds if  $\langle x \rangle$  is a proof in PA1 for the sentence  $\langle y \rangle$
- ▶ For integers  $m, n, p$   $\exists g \in \mathbb{N} \langle g \rangle \equiv \text{sost}(m, n, p) =$   
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:  $\mathbf{y} = \ulcorner 'y' \urcorner \in \mathbb{N}$ )
- ▶  $\gamma(y) \equiv \neg \exists x \in \mathbb{N} \text{proof}(x, \text{sost}(y, \mathbf{y}, y))$ :  
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  $\phi \equiv \neg \exists x \in \mathbb{N} \text{proof}(x, \text{sost}(q, \mathbf{y}, q))$

# Gödel's self-referential sentence $\phi$

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

- ▶ Let  $\psi \equiv \text{sost}(q, \mathbf{y}, q)$

*$\psi$  defined by replacing the free variable symbol 'y' in  $\langle q \rangle$  with  $q$*

- ▶  $\phi \equiv$  “there is no proof in PA1 for the sentence  $\psi$ ”

- ▶ How did we obtain  $\phi$ ?

*$\phi$  obtained by replacing the free variable  $y$  in  $\gamma(y)$  with  $q$*

i.e.  $\phi \equiv \gamma(q)$

- ▶ **Grey sentences:**  $\phi$  and  $\psi$  have almost the same meaning

- ▶ **Recall:**  $q = \ulcorner \gamma(y) \urcorner$ , i.e.  $\langle q \rangle \equiv \gamma(y)$

- ▶ **So, in fact,**  $\psi \equiv \phi$

- ▶ Hence  $\phi$  states “ $\phi$  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  $\perp$ : termination (*nontermination denoted  $\emptyset$* )
  - ▶ can model PA1
- ▶  $\exists$  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)**: given TM  $M$  & input  $x$ , is  $M(x) = \perp$ ?

*Does a given TM terminate on its input?*
- ▶ Turing's theorem: HP is undecidable

# Computable functions

- ▶ Consider TM  $T$  on input  $x$  yielding output  $y$
- ▶ **Functional view:**  $T(x) = y$
- ▶ If a TM  $T$  terminates on all input,  $T(\cdot)$  is *computable*  
*a.k.a. “total computable”*
- ▶ If a function is not computable, then it's *uncomputable*
- ▶ If  $T$  only terminates on some input,  $T(\cdot)$  is *partial computable*

# Turing's proof (formal)

- ▶ **Enumerate all TMs:**  $(M_i \mid i \in \mathbb{N})$
- ▶ **Halting function**  $\text{halt}(i, \ell) = \begin{cases} 1 & \text{if } M_i(\ell) = \perp \\ 0 & \text{if } M_i(\ell) = \emptyset \end{cases}$
- ▶ **Show  $\text{halt} \neq F$  for any total computable  $F(i, \ell)$ :**
  - ▶ **let  $G(i) = 0$  if  $F(i, i) = 0$  or undefined ( $\emptyset$ ) 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  $G(i) = 0$  (since  $G(i)$  undefined othw)
  - ▶ **consider  $\text{halt}(j, j)$ :**
    - ▶  $\text{halt}(j, j) = 1 \rightarrow M_j(j) = \perp \rightarrow G(j) = 0 \rightarrow F(j, j) = 0$
    - ▶  $\text{halt}(j, j) = 0 \rightarrow M_j(j) = \emptyset \rightarrow G(j) = \emptyset \rightarrow F(j, j) \neq 0$
  - ▶ **so  $\text{halt}(j, j) \neq F(j, j)$  for all  $j$**
- ▶ **halt is uncomputable**

# Turing and Gödel

- ▶ Consider a TM called “provable”  
with input  $\alpha \in \text{PA1}$ :

```
while(1) {i=0; if proof(i,  $\ulcorner \alpha \urcorner$ ) return YES; else i=i+1}
```

- ▶  $\text{provable}(\alpha) = \text{YES}$  iff  $\text{PA1} \vdash \alpha$
- ▶ termination of provable  $\Rightarrow$  decidability in PA1
- ▶ Gödel's  $\phi$  is not provable  $\Rightarrow$  PA1 is undecidable

**PA1 incomplete and undecidable**

## Subsection 4

Tarski

# Example: Reals

- ▶ **Tarski's theorem:** Reals is complete
- ▶ **Algorithm:**  
*constructs solution sets (YES) or derives contradictions(NO)*  
⇒ provides proofs or contradictions for all sentences
- ▶ ⇒ Reals is complete **and also decidable**  
*since every complete theory is decidable (why?)*

# Completeness $\Rightarrow$ decidability

- ▶ Given  $\phi \in \mathcal{F}$

$i = 0$

**while 1 do**

**if** proof( $i, \lceil \phi \rceil$ ) **then**

    return YES

**else if** proof( $i, \lceil \neg \phi \rceil$ ) **then**

    return NO

**end if**

$i = i + 1$

**end while**

- ▶ Since  $\mathcal{F}$  complete, alg. terminates on all  $\phi$

# 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  $\phi$  in DLO (*similar to Reals in one dimension*)

- ▶ Reduce to DNF w/clauses  $\exists x_i q_i(x)$  s.t.  $q_i = \bigwedge q_{ij}$
- ▶ 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:  
 $q_i$  conjunction 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?

*case already discussed, answer is NO*

# 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*
- ▶ **Theorem:** ACF is incomplete
  - ▶  $\text{ACF}_p$ :  $\text{ACF} \wedge C_p \equiv [\sum_{j \leq p} 1 = 0]$  (with  $p$  prime)
  - ▶ **Claim:**  $\forall p$  (prime)  $C_p$  independent of ACF
    - ▶ suppose proof of  $C_p$  or  $\neg C_p$  possible for  $p$
    - ▶ then either  $\text{ACF} \wedge C_p$  or  $\text{ACF} \wedge \neg C_p$  inconsistent
    - ▶ but  $\exists$  field of characteristic  $p$
    - ▶  $\text{ACF} \wedge C_p$  and  $\text{ACF} \wedge \neg C_p$  consistent
- ▶ **Theorem:** ACF is decidable  
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  $\psi$   
*but replace  $\sum_{j \leq p} 1$  by 0 whenever possible*
- ▶  $\Rightarrow$  ACF is incomplete and 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  $\phi$  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
- ▶ **Complete** FS:  $\forall M (A^M) \Rightarrow \mathcal{F} \vdash A$
- ▶ *Gödel's completeness theorem*:  
1<sup>st</sup> order logic is **complete**

▶ *Pay attention when reading literature/websites*

## Subsection 6

### MP solvability

# Polynomial equations in integers

- ▶ Consider the feasibility-only MP

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

with  $g_i(x)$  composed by arithmetical expressions (+, -, ×, ÷)

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

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

- ▶ 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 integer coefficients: devise a process which could determine by a finite number of operations whether the equation is solvable in 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  $R$

let  $\mathbb{TM} = \{T : \mathbb{N} \rightarrow \mathbb{N} \mid T \text{ is a TM}\}$ ; then  $\forall R \in \mathbb{RE} \exists T \in \mathbb{TM} (\text{ran } T = R)$

- ▶ some  $\mathbb{RE}$  sets are undecidable, e.g.  $R = \{\ulcorner \phi \urcorner \mid \text{PA1} \vdash \phi\}$

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

*othw for any  $r \in \mathbb{N}$  can decide if  $r \in R$  by finding  $x \in \mathbb{N}^n$  s.t.  $p(r, x) = 0$*

# Proof strategy

- ▶ **Strategy:** “model” r.e. sets with polynomial equations in integers
- ▶ **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

- ▶ **Matyiasievic'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

- ▶ **Claim:**  $\forall T \in \text{TM} \text{ ran } T \in \mathbb{RE}$
- ▶ **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  $M_i(k)$   
 $\Rightarrow \forall k \in \mathbb{N}, \ell \leq k$  if  $M_i^\ell(k-\ell+1) = \perp$  then let  $R_i \leftarrow R_i \cup \{k-\ell+1\}$   
 $\Rightarrow R_i = \{k-\ell+1 \mid \exists k \in \mathbb{N}, \ell \leq k (M_i^\ell(k-\ell+1) = \perp)\}$
- ▶ **DPRM theorem:**  $\forall R \in \mathbb{RE}$ ,  $R$  represented by poly eqn
- ▶ **By claim, can choose UTM  $M_i$  with  $\text{ran } M_i = R_i \in \mathbb{RE}$**   
 $\Rightarrow \exists$  Universal DE (UDE), say  $U(r, x) = 0$
- ▶  $\min\{0 \mid U(r, x) = 0 \wedge x \in \mathbb{N}^n\}$ :  
**undecidable (feasibility) MP**
- ▶  $\min_{x \in \mathbb{N}^n} (U(r, x))^2$ : **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  $\{0, 1, \dots, a\}$   
 $q(x) = \prod_{i \in \omega} (x-i)$  is  $\infty$  long, invalid
  - ▶ **IMPOSSIBLE!**
    - if it were possible, DE would be decidable, contradiction*
  - ▶  $\Rightarrow$  Reals  $\not\subseteq$  DE

# MIQCP is undecidable

- ▶ [Jeromlow 1973]: MIQCP:

$$\left. \begin{array}{l} \min \\ \forall i \leq m \quad x^\top Q^i x + a_i^\top x + b_i \geq 0 \\ x \in \mathbb{Z}^n \end{array} \right\} \quad (\dagger)$$

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 unsolvable 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  $x \in \{\lceil x^L \rceil, \lceil x^L \rceil + 1, \dots, \lfloor x^U \rfloor\}$  by polynomial*

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

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

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

- ▶ [MIQP decision vers.] is **decidable**

$$\left. \begin{array}{rcl} x^\top Qx + c^\top x & \leq & \gamma \\ Ax & \geq & b \\ \forall j \in Z & x_j & \in \mathbb{Z} \end{array} \right\} \quad (\text{in NP [Del Pia et al. 2014]})$$

# NLP is undecidable

We can't represent unbounded subsets of  $\mathbb{N}$  by polynomials

But we can if we allow some transcendental functions

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

- ▶ **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*):**

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

# Some NLPs are decidable

- ▶ All polynomial NLPs are **decidable**

*by decidability of Reals*

---

- ▶ QUADRATIC PROGRAMMING (QP) is **decidable over  $\mathbb{Q}$**

$$\min \left. \begin{array}{l} x^\top Qx + c^\top x \\ Ax \geq b \end{array} \right\} \quad (P)$$

- ▶ *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 whether  $p(x) = 0$  has solutions in  $\mathbb{Q}$ ? Boh!*
- ▶ [Matyiasевич 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

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# Worst-case algorithmic complexity

- ▶ Computational complexity theory:  
*worst-case time/space taken by an algorithm to complete*
- ▶ Given an 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  $\text{cpu}(\mathcal{A})$  vary with  $\nu$ ?
  - ▶  $\text{cpu}(\mathcal{A}) = O(\log \nu)$ : logarithmic (sublinear)
  - ▶  $\text{cpu}(\mathcal{A}) = O(\log^k \nu)$  for fixed  $k$ : polylogarithmic
  - ▶  $\text{cpu}(\mathcal{A}) = O(\nu)$ : linear
  - ▶  $\text{cpu}(\mathcal{A}) = O(\nu^2)$ : quadratic
  - ▶  $\text{cpu}(\mathcal{A}) = O(\nu^k)$  for fixed  $k$ : polytime
  - ▶  $\text{cpu}(\mathcal{A}) = O(2^\nu)$ : exponential
- ▶ polytime  $\leftrightarrow$  efficient
- ▶ exponential  $\leftrightarrow$  inefficient

# The “ $O(\cdot)$ ” calculus

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

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

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

# Are polytime algorithms “efficient”?

- ▶ **Why are polynomials special?**
- ▶ **Many different variants of Turing Machines (TM)**  
*more tapes, more heads, ...*
- ▶ **Polytime is *invariant* to all definitions of TM**  
*e.g. TM with  $\infty$  many tapes: simulate with a single tape running along diagonals, similarly to dovetailing*
- ▶ **In practice,  $O(n)$ - $O(n^3)$  is an acceptable range covering most practically useful efficient algorithms**
- ▶ **Many exponential algorithms are also usable in practice for limited sizes**
- ▶ **Sublinear algorithms aren't allowed to read their whole input!**

# Instances and problems

- ▶ An input to an algorithm  $\mathcal{A}$ : *instance*
- ▶ Collection of all inputs for  $\mathcal{A}$ : *problem*  
*in general, a problem  $P$  is an infinite set of instances*
- ▶  $\mathcal{A}$  solves  $P$  if  $\mathcal{A}$  solves **every instance** of  $P$ 
  - ▶ There are problems which no algorithm can solve
  - ▶ A problem can be solved by different algorithms
- ▶ Given  $P$  find complexity of *best alg.*  $\mathcal{A}$  solving  $P$

$$\min_{<0} \{ \text{cpu}(\mathcal{A}) \mid \mathcal{A} \text{ solves } P \}$$

- ▶ We (generally) don't know how to search over all algs for  $P$   
*sometimes we can find complexity bounds*

# Complexity classes: P, NP

- ▶ Focus on *decision problems*
- ▶ If  $\exists$  polytime algorithm for  $P$ , then  $P \in P$
- ▶ If there is a polytime checkable *certificate* for all YES instances of  $P$ , then  $P \in NP$

*e.g. problem finding a path with fewer than  $K$  edges in a graph: path itself is a certificate: it can be checked whether it has fewer than  $K$  edges in time proportional to  $K$ , which is obviously smaller than the size of the graph*

- ▶ No-one knows whether  $P = NP$  (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, ...*

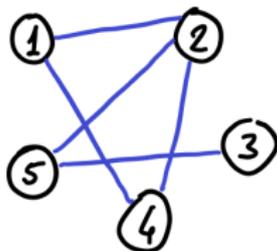


## Subsection 1

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 with  $\geq 1$  vertices 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  $P = NP$*

# MP formulations for CLIQUE

Variables? Objective? Constraints?

# MP formulations for CLIQUE

Variables? Objective? Constraints?

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

# MP formulations for CLIQUE

Variables? Objective? Constraints?

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

# MP formulations for CLIQUE

Variables? Objective? Constraints?

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

# MP formulations for CLIQUE

Variables? Objective? Constraints?

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

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

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

# 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:** “ $x_i = x_j = 1$  iff  $\{i, j\} \in E$ ”

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

- ▶ **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”

# 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:** “ $x_i = x_j = 1$  iff  $\{i, j\} \in E$ ”

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

- ▶ **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:*

$$\left. \begin{array}{l} \sum_{i \in V} x_i = k \\ \forall \{i, j\} \notin E \quad x_i + x_j \leq 1 \\ x \in \{0, 1\}^n \end{array} \right\}$$

# MP formulations for CLIQUE

- ▶ *Pure feasibility problem:*

$$\left. \begin{array}{l} \sum_{i \in V} x_i = k \\ \forall \{i, j\} \notin E \quad x_i + x_j \leq 1 \\ x \in \{0, 1\}^n \end{array} \right\}$$

- ▶ **MAX CLIQUE:**

$$\left. \begin{array}{l} \max \sum_{i \in V} x_i \\ \forall \{i, j\} \notin E \quad x_i + x_j \leq 1 \\ x \in \{0, 1\}^n \end{array} \right\}$$

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

- ▶  $a = (1, 1, 1, 4, 5), b = 3$ : **YES** with  $J = \{1, 2, 3\}$   
*all  $b \in \{0, \dots, 12\}$  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:*

$$\left. \begin{array}{l} \sum_{j \leq n} a_j x_j = b \\ x \in \{0, 1\}^n \end{array} \right\}$$

# 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  $J \subseteq \{1, \dots, n\}$  s.t.  $c(J) \leq K$  and  $w(J)$  is maximum**
  - ▶ *notation:*  $c(J) = \sum_{j \in J} c_j$  (similarly for  $w(J)$ )
  - ▶ natively expressed as an optimization problem
- 

- ▶  $n = 3, c = (5, 6, 7), w = (3, 4, 5), K = 11$ 
  - ▶  $c(J) \leq 11$  **feasible for  $J$  in  $\emptyset, \{j\}, \{1, 2\}$**
  - ▶  $w(\emptyset) = 0, w(\{1, 2\}) = 3 + 4 = 7, w(\{j\}) \leq 5$  **for  $j \leq 3$**   
 $\Rightarrow J_{\max} = \{1, 2\}$
- ▶  **$K = 4$ : trivial solution ( $J = \emptyset$ )**

# MP formulation for KNAPSACK

**Variables? Objective? Constraints?**

# MP formulation for KNAPSACK

Variables? Objective? Constraints?

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

# 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!**

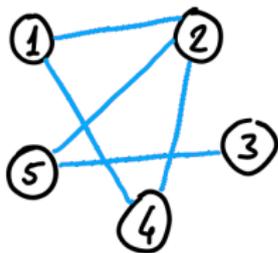
# HAMILTONIAN CYCLE

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

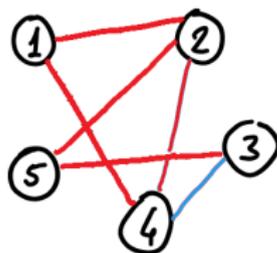
*cycle covering every  $v \in V$  exactly once*

---

NO



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



# MP formulation for HAMILTONIAN CYCLE

**Variables? Objective? Constraints?**

# MP formulation for HAMILTONIAN CYCLE

Variables? Objective? Constraints?

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

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

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

**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

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

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

- ▶ Problem: is there  $\phi : x \rightarrow \{0, 1\}^n$  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

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

$\phi$	$x = (1, 1)$	$x = (0, 0)$	$x = (1, 0)$	$x = (0, 1)$
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  $\bigwedge_{i \leq m} \bigvee_{j \in C_i} \ell_j$ :

# MP formulation for SAT

Variables? Objective? Constraints?

Algorithm  $\hat{\rho}$  to generate MP from  $\bigwedge_{i \leq m} \bigvee_{j \in C_i} \ell_j$ :

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

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

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

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

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

$$\hat{\rho}\left(\bigwedge_i \Gamma_i\right) \longmapsto \forall i \leq m \quad \hat{\rho}(\Gamma_i)$$

# MP formulation for SAT

- ▶ **Prop.:** SAT instance  $q$  is YES iff ILP instance  $\hat{\rho}(q)$  is YES

- ▶ **Proof:** Let  $L = (\ell'_1, \dots, \ell'_n)$  be a solution of SAT. Then  $x^* = (x_1^*, \dots, x_n^*)$  where  $x_j^* = 1$  iff  $\ell'_j = \text{true}$  and  $x_j^* = 0$  iff  $\ell'_j = \text{false}$  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 the machines executing  $\hat{\rho}$ ”*

**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  $P \neq NP$
- ▶ Next best thing: define *hardest problem in NP*
- ▶ **Prob.  $P$  is NP-hard if  $\forall Q \in NP \exists$  polytime alg.  $\rho_Q$ :**  

$q \in Q \mapsto \rho_Q(q) \in P$  with  $q$  YES iff  $\rho_Q(q)$  YES

$\rho_Q$  is called a *polynomial reduction* from  $Q$  to  $P$
- ▶ Why is such a  $P$  hardest for NP?
  1. run best alg. for  $P$  on  $\rho_Q(q)$ , get answer  $\alpha \in \{\text{YES}, \text{NO}\}$
  2. return  $\alpha$  as answer for  $q$
  3. so  $Q$  is no harder than  $P$
  4. hence no problem in NP is harder than  $P$
- ▶ If  $P$  is in NP and is NP-hard, it is called *NP-complete*
- ▶ *Reduction*: “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:

$P_{s,t}^i$  for  $1 \leq i \leq \ell, 1 \leq s, t \leq T$ .

$P_{s,t}^i$  is true iff tape square number  $s$  at step  $t$  contains the symbol  $\sigma_i$ .

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

$S_{s,t}$  for  $1 \leq s, t \leq T$  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:

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

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

$$G_{i,j}^t = \bigwedge_{s=1}^T (\neg Q_t^i \vee \neg S_{s,t} \vee \neg P_{s,t}^j \vee Q_{t+1}^k)$$

**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  $P$  in NP can be polynomially reduced to a MILP

**Proof**

Since  $P \in \text{NP}$ , every YES instance  $\pi \in P$  must have a polynomial-time (say  $p^P(|\pi|)$ ) verifiable certificate  $c_\pi$  (wlog assume it is a  $\{0, 1\}$  string), with length bounded by a polynomial (say  $q^P(|\pi|)$ ). This means that a deterministic TM  $M_P$  verifying  $c_\pi$  will reach termination in polytime  $p^P(\pi)$ . Let  $\kappa$  be such that  $p^P, q^P \in O(n^\kappa)$ . We define a MILP on binary variables holding the content of the tape of  $M_P$  as it changes according to the transition function of  $M_P$ , such that the tape contains: (i) “NO” in the first cell, and  $\pi$  in the subsequent  $|\pi|$  cells, at the initial step  $k = 0$ ; (ii) “YES” in the first cell at the final step  $k = n^\kappa$ . Then the MILP is feasible iff  $\pi$  is a YES instance of  $P$ . This provides a polytime reduction from  $P$  to MILP.

# Cook's theorem: sets and params

- ▶ **Model a deterministic TM dynamics using MILP**
- ▶  $M_P$  is a 5-tuple  $(Q, \Sigma, s, F, \delta)$ :  
*states, alphabet, initial, final, transition*
- ▶ **Transition function**  $\delta : Q \setminus F \times \Sigma \rightarrow Q \times \Sigma \times \{-1, 1\}$   
 *$\delta$ : state  $\ell$ , symbol  $j \mapsto$  state  $\ell'$ , symbol  $j'$ , direction  $d$*
- ▶  $M_P$  **polytime**: terminates in  $n^\kappa$ , where  $n = |\pi|$
- ▶ **Index sets:**  
states  $Q$ , characters  $\Sigma$ , tape cells  $I$ , steps  $K$
- ▶ **Parameters:**  
initial tape string  $(\text{NO}, \pi)$   
YES written in cell 1 when  $M_P$  in final state

# Cook's theorem: decision vars

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

# Cook's theorem: constraints (informal)

## 1. *Initialization:*

- 1.1 initial string  $(\mathbf{NO}, \pi)$  on tape at step  $k = 0$
- 1.2  $M_P$  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  $\forall k$ :  $M_P$  is in exactly one state  $\ell$
- 2.3  $\forall k$ : tape head  $M_P$  is at exactly one cell  $i$
- 2.4  $\forall i, k$ : if cell  $i$  changes symbol between steps  $k$  and  $k + 1$ , head must be on cell  $i$  at step  $k$
- 2.5  $\forall k, i, j$ : cell  $i$  and symbol  $j$  in state  $k$  lead to cells, symbol and states given by transition function  $\delta$

## 3. *Termination:*

- 3.1  $M_P$  terminates at step  $k \leq n^k$  w/YES written in cell 1

# Cook's theorem: constraints

## 1. Initialization:

$$1.1 \quad (t_{1,NO,0} = 1) \wedge (\forall i > 1 \quad t_{i,\pi_i,0} = 1)$$

$$1.2 \quad q_{s,0} = 1$$

$$1.3 \quad h_{0,0} = 1$$

## 2. Execution:

$$2.1 \quad \forall i, k \quad \sum_j t_{ijk} = 1$$

$$2.2 \quad \forall k \quad \sum_\ell q_{\ell k} = 1$$

$$2.3 \quad \forall k \quad \sum_i h_{ik} = 1$$

$$2.4 \quad \forall i, j \neq j', k < n^\kappa \quad t_{ijk} t_{i,j',k+1} \leq h_{ik}$$

$$2.5 \quad \forall i, \ell, \ell', j, j', k, d \text{ s.t. } (\ell', j', d) = \delta(\ell, j)$$

$$h_{ik} q_{\ell k} t_{ijk} = h_{i+d,k+1} q_{\ell',k+1} t_{i,j',k+1}$$

## 3. Termination:

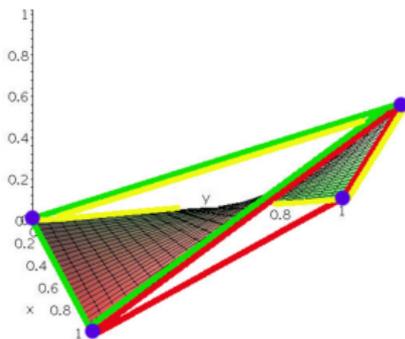
$$3.1 \quad (t_{1,YES,n^\kappa} = 1) \wedge \left( \sum_{f \in F, k} q_{fk} = 1 \right)$$

# 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  $z \in [0, 1]$

$$z = xy \Leftrightarrow z \leq x \wedge z \leq y \wedge z \geq x + y - 1$$



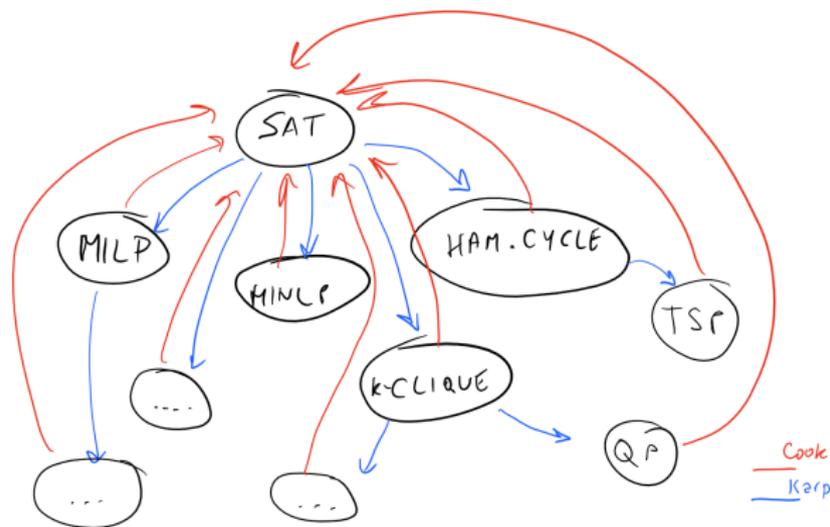
- ▶ MILP is feasibility only
- ▶ MILP has polynomial size
- ▶  $\Rightarrow$  *MILP is NP-hard*

# 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  $\rho_Q$  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 since  $Q$  hardest in NP



# Example of polynomial reduction

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

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

- ▶ **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 NP-hard
- ▶ STABLE is also in NP
  - $U \subseteq V$  is a stable set iff  $E(G[U]) = \emptyset$  (polytime verification)
- ▶  $\Rightarrow$  STABLE is 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**

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

where  $\ell_j$  is either  $x_j$  or  $\bar{x}_j \equiv \neg x_j$

- ▶ **Polynomial reduction  $\hat{\rho}$**

$$\begin{array}{c|cccc} \text{SAT} & x_j & \bar{x}_j & \vee & \wedge \\ \hline \text{MILP} & x_j & 1 - x_j & + & \geq 1 \end{array}$$

- ▶ **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)

$$\min \left. \begin{array}{l} x^\top Qx + c^\top x \\ Ax \geq b \end{array} \right\}$$

- ▶ Quadratic obj, linear constrs, 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  $\leq 0$ ?”**
- ▶ **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  $\sigma$  be an instance of SAT
- ▶  $\hat{\rho}(\sigma, x) \geq 1$ : linear constraints of (SAT  $\rightarrow$  MILP) reduction
- ▶ Consider QP subclass

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

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

# Box-constrained QP is NP-hard

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

- ▶ Add surplus vars  $v$  to SAT  $\rightarrow$  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

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

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

# cQKP is NP-hard

- ▶ CONTINUOUS QUADRATIC KNAPSACK PROBLEM (cQKP)

$$\left. \begin{aligned} \min \quad & f(x) = x^\top Qx + c^\top x \\ & \sum_{j \leq n} a_j x_j = \gamma \\ & x \in [0, 1]^n, \end{aligned} \right\}$$

- ▶ **Reduction from SUBSET-SUM**

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

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

- ▶  $\sigma$  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

- ▶  $\sigma$  is a NO instance of SUBSET-SUM

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

# QP on a simplex is NP-hard

$$\left. \begin{array}{l} \min f(x) = x^\top Qx + c^\top x \\ \sum_{j \leq n} x_j = 1 \\ \forall j \leq n \quad x_j \geq 0 \end{array} \right\}$$

- ▶ Reduce MAX CLIQUE to QP subclass  $f(x) = - \sum_{\{i,j\} \in E} x_i x_j$

*Motzkin-Straus formulation (MSF):*

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

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

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

$\exists x^* \in \text{opt (MSF)}$  with  $f^* = f(x^*) = \frac{1}{2} - \frac{1}{2\omega(G)}$

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

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

# Proof of the Motzkin-Straus theorem

$$x^* = \text{opt} \left( \max_{\substack{\sum_j x_j = 1 \\ x \geq 0}} \sum_{ij \in E} x_i x_j \right) \text{ s.t. } |C = \{j \in V \mid x_j^* > 0\}| \text{ smallest } (\ddagger)$$

## 1. *C is a clique*

- ▶ Suppose  $1, 2 \in C$  but  $\{1, 2\} \notin E$ , then  $x_1^*, x_2^* > 0$ , can perturb by  $\epsilon \in [-x_1^*, x_2^*]$ , get  $x^\epsilon = (x_1^* + \epsilon, x_2^* - \epsilon, \dots)$ , feasible w.r.t. simplex and bound constraints
- ▶  $\{1, 2\} \notin E \Rightarrow x_1 x_2$  does not appear in  $f(x) \Rightarrow f(x^\epsilon)$  depends at worst linearly on  $\epsilon$ ; by optimality of  $x^*$ ,  $f$  achieves max for  $\epsilon = 0$ , in interior of its range  $\Rightarrow f(\epsilon)$  constant
- ▶ setting  $\epsilon = -x_1^*$  or  $\epsilon = x_2^*$  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

$$x^* = \text{opt} \left( \max_{\substack{\sum_j x_j = 1 \\ x \geq 0}} \sum_{ij \in E} x_i x_j \right) \text{ s.t. } |C| = |\{j \in V \mid x_j^* > 0\}| \text{ smallest } (\ddagger)$$

2.  $|C| = \omega(G)$

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

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

- ▶ by construction  $x_j^* = 0$  for  $j \notin C \Rightarrow$

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

- ▶  $\psi(x) = 1$  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**:  $\min x^\top Qx : \sum_j x_j = 1 \wedge x \geq 0$

*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**  $A \bullet B = \text{tr}(A^\top B) = \sum_{i,j} A_{ij} B_{ij}$

write StQP (linearized) objective as  $\min Q \bullet X$

- ▶ **Let**  $C = \{X \mid X = xx^\top \wedge x \geq 0\}$ ,  $\bar{C} = \text{conv}(C)$

- ▶  $\sum_j x_j = 1 \Leftrightarrow (\sum_j x_j)^2 = 1^2 \Leftrightarrow \mathbf{1} \bullet X = 1$

- ▶ **StQP**  $\equiv \min Q \bullet X : \mathbf{1} \bullet X = 1 \wedge X \in C$

*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**  $\equiv \max y : Q - y\mathbf{1} \in \bar{C}^* = \{A \mid \forall x \geq 0 (x^\top Ax \geq 0)\}$

$\bar{C}^*$  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

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

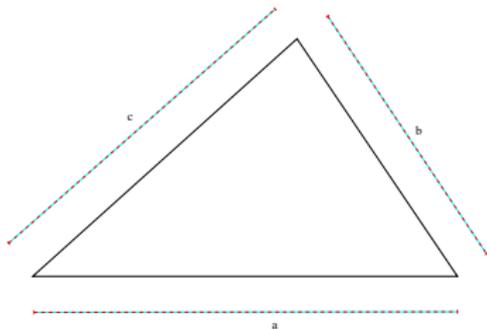
### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# 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.**  $2\alpha + 2\beta + 2\gamma = 2\pi \Rightarrow \alpha + \beta + \gamma = \pi$

$$r + ix = ue^{i\alpha}$$

$$r + iy = ve^{i\beta}$$

$$r + iz = we^{i\gamma}$$

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

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

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

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

$$s - a = x + y + z - y - z = x$$

$$s - b = x + y + z - x - z = y$$

$$s - c = x + y + z - x - y = z$$

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

## 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  $x_i = (d_{i1}, \dots, d_{in})$  of  $D$
- ▶ Embed  $i \in X$  by vector  $U^D(i) = x_i \in \mathbb{R}^n$   
define  $U^D : \{1, \dots, n\} \rightarrow \mathbb{R}^n$  s.t.  $U^D(i) = x_i$
- ▶ **Thm.:**  $(U^D, \ell_\infty)$  is a metric space with dist. matrix  $D$   
*i.e.  $\forall i, j \leq n \ \|x_i - x_j\|_\infty = d_{ij}$*
- ▶  $U^D$  is called **Universal Isometric Embedding (UIE)**
- ▶ **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

$$\|x_i - x_j\|_\infty = \max_{k \leq n} |d_{ik} - d_{jk}| \leq \max_{k \leq n} |d_{ij}| = d_{ij}$$

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

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

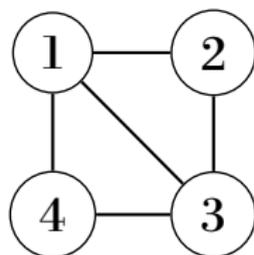
If valid  $\forall i, j$  then valid for max

- ▶  $\max |d_{ik} - d_{jk}|$  over  $k \leq n$  achieved when  $k \in \{i, j\}$

$$\Rightarrow \|x_i - x_j\|_\infty = d_{ij}$$

# UIE from incomplete metrics

- ▶ If your metric space is missing some distances
- ▶ **Get incomplete distance matrix  $D$**
- ▶ Cannot define vectors  $U^D(i)$  in UIE
- ▶ Note:  $D$  defines a graph

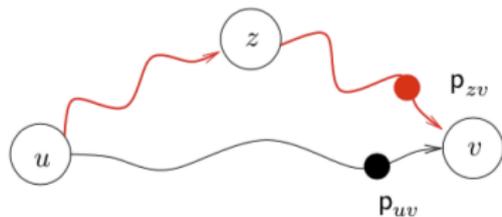


$$D = \begin{pmatrix} 0 & 1 & \sqrt{2} & 1 \\ 1 & 0 & 1 & ? \\ \sqrt{2} & 1 & 0 & 1 \\ 1 & ? & 1 & 0 \end{pmatrix}$$

- ▶ **Complete this graph with shortest paths:  $d_{24} = 2$**

# Floyd-Warshall algorithm 1/2

- ▶ Given  $n \times n$  partial matrix  $D$  computes *all shortest path lengths*
- ▶ For each triplet  $u, v, z$  of vertices in the graph, test: *when going  $u \rightarrow v$ , is it convenient to pass through  $z$ ?*



- ▶ If so, then change the path length

# Floyd-Warshall algorithm 2/2

*# initialization*

**for**  $u \leq n, v \leq n$  **do**

**if**  $d_{uv} = ?$  **then**

$d_{uv} \leftarrow \infty$

**end if**

**end for**

*# main loop*

**for**  $z \leq n$  **do**

**for**  $u \leq n$  **do**

**for**  $v \leq n$  **do**

**if**  $d_{uv} > d_{uz} + d_{zv}$  **then**

$d_{uv} \leftarrow d_{uz} + d_{zv}$

**end if**

**end for**

**end for**

**end for**

## 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<sup>1</sup> corresponding to  $n$  points  $x_1, \dots, x_n \in \mathbb{R}^K$  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**

---

<sup>1</sup>Euclidean Distance Matrix

# Gram in function of EDM

- ▶  $x = (x_1, \dots, x_n) \subseteq \mathbb{R}^K$ , 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:*

$$G = -\frac{1}{2}JD^2J \quad (\S)$$

- ▶ where  $D^2 = (d_{ij}^2)$  and

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

# Multidimensional scaling (MDS)

- ▶ Often get approximate EDMs  $\tilde{D}$  from raw data (*dissimilarities, discrepancies, differences*)
- ▶  $\tilde{G} = -\frac{1}{2}J\tilde{D}^2J$  is an approximate Gram matrix
- ▶ Approximate Gram  $\Rightarrow$  spectral decomposition  $P\tilde{\Lambda}P^\top$  has  $\tilde{\Lambda} \not\geq 0$
- ▶ Let  $\Lambda$  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:  $G = -\frac{1}{2}JD^2J$

# Proof of lemma

## ▶ $\text{Gram} \subseteq \text{PSD}$

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

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

- ▶  $\Rightarrow G \succeq 0$

## ▶ $\text{PSD} \subseteq \text{Gram}$

- ▶ 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:  $d_{ij}^2 = \|x_i - x_j\|_2^2 = (x_i - x_j)(x_i - x_j) = x_i x_i + x_j x_j - 2x_i x_j$  (\*)
- ▶ Aim at “inverting” (\*) to express  $x_i x_j$  in function of  $d_{ij}^2$
- ▶ Sum (\*) over  $i$ :  $\sum_i d_{ij}^2 = \sum_i x_i x_i + n x_j x_j - 2x_j \sum_i x_i$   $\xrightarrow{0 \text{ by zero centroid}}$
- ▶ 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 ( $\ddagger$ ) over  $j$ , get:

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

- ▶ Divide by  $n$ , get:

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

# Schoenberg's theorem proof (2/2)

- ▶ Rearrange (\*), (†), (‡) as follows:

$$2x_i x_j = x_i x_i + x_j x_j - d_{ij}^2 \quad (5)$$

$$x_i x_i = \frac{1}{n} \sum_j d_{ij}^2 - \frac{1}{n} \sum_j x_j x_j \quad (6)$$

$$x_j x_j = \frac{1}{n} \sum_i d_{ij}^2 - \frac{1}{n} \sum_i x_i x_i \quad (7)$$

- ▶ 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

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

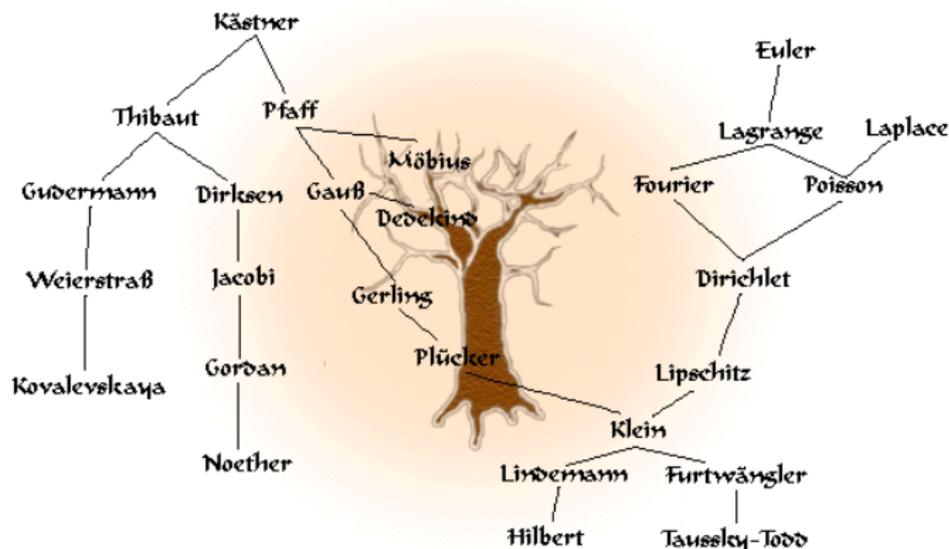
which expresses  $x_i x_j$  in function of  $D$

# Principal Component Analysis (PCA)

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

# Example 1/3

## Mathematical genealogy skeleton



# Example 2/3

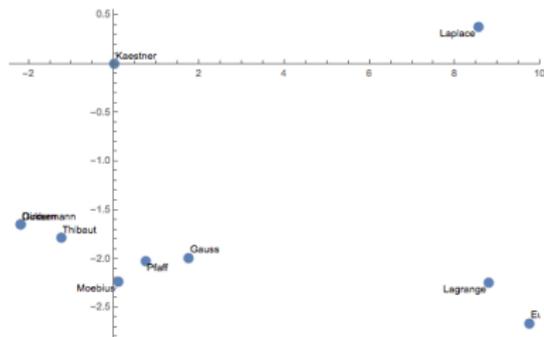
*A partial view*

	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

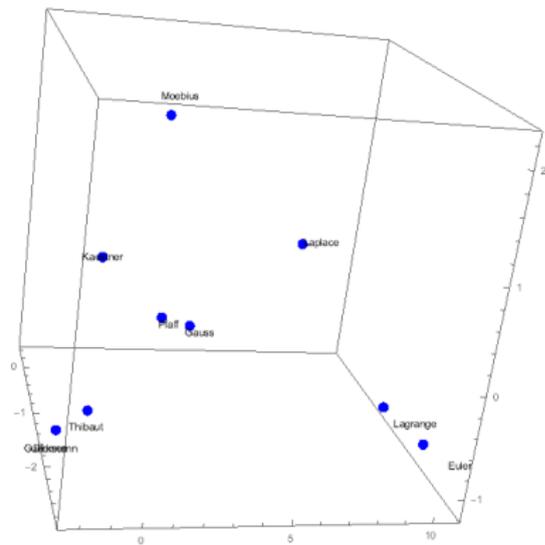
$$D = \begin{pmatrix} 0 & 10 & 1 & 1 & 9 & 8 & 2 & 2 & 2 & 2 \\ 10 & 0 & 11 & 9 & 1 & 3 & 10 & 12 & 12 & 8 \\ 1 & 11 & 0 & 2 & 10 & 10 & 3 & 1 & 1 & 3 \\ 1 & 9 & 2 & 0 & 8 & 8 & 1 & 3 & 3 & 1 \\ 9 & 1 & 10 & 8 & 0 & 2 & 9 & 11 & 11 & 7 \\ 8 & 3 & 10 & 8 & 2 & 0 & 9 & 11 & 11 & 7 \\ 2 & 10 & 3 & 1 & 9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

# Example 3/3

In 2D



In 3D



## Subsection 3

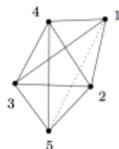
### Distance geometry problem

# The Distance Geometry Problem (DGP)

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

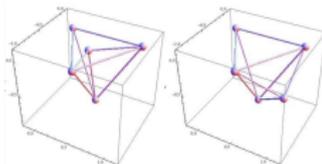
$$\forall \{i, j\} \in E \quad \|x_i - x_j\|_2^2 = d_{ij}^2$$

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

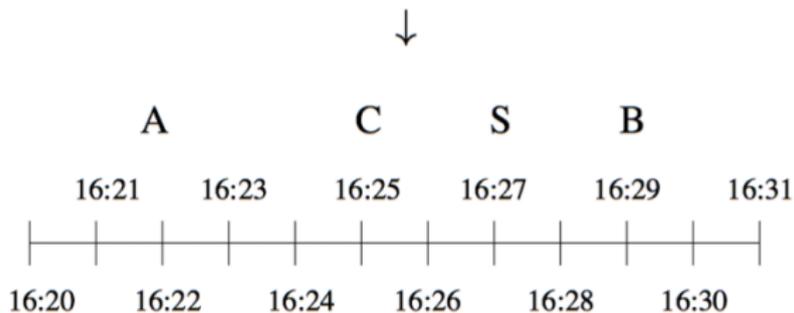
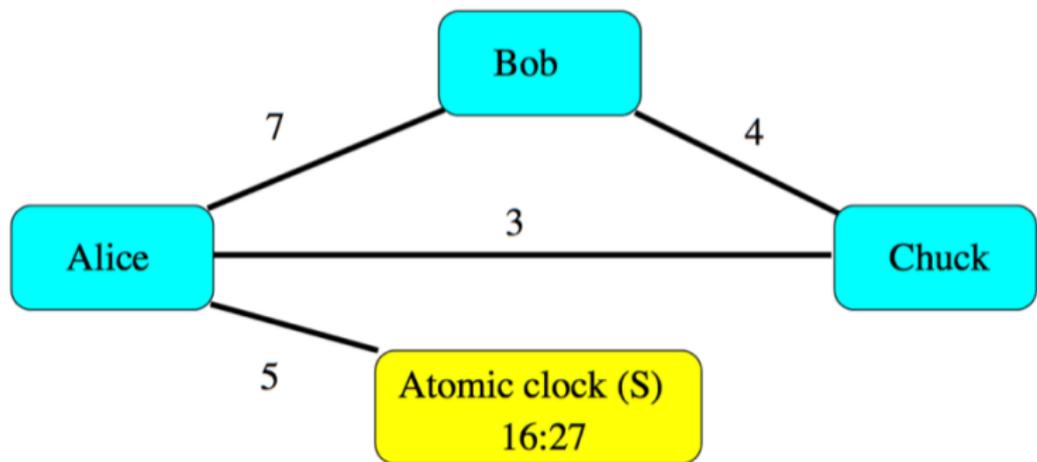
From [Singer, *Appl. Comput. Harmon. Anal.* 2011]

*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]

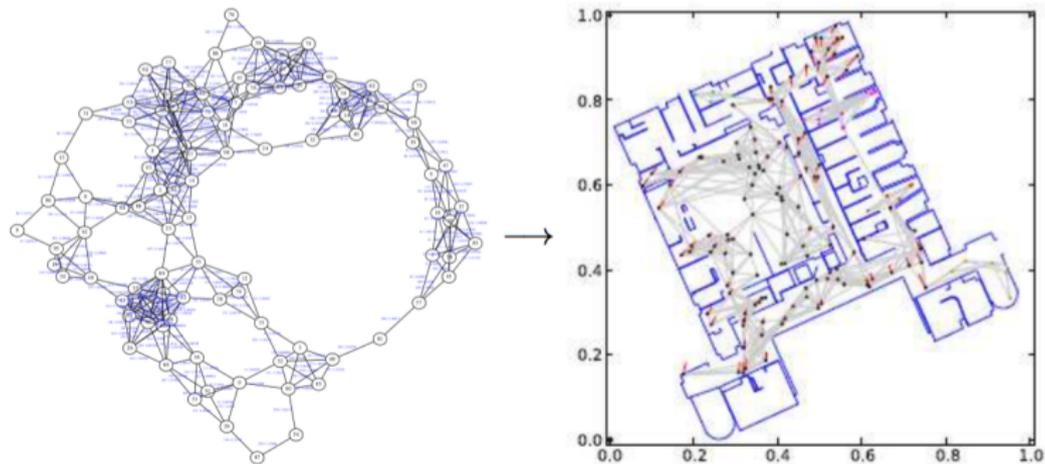
*The positioning problem arises when it is necessary to locate a set of geographically distributed objects using measurements of the distances between some object pairs*

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

Used whenever GPS not viable (e.g. underwater)

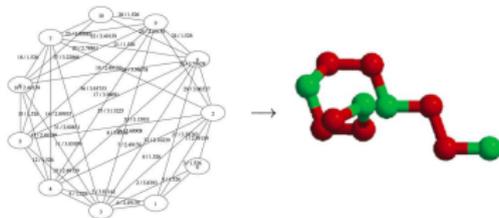
$d_{uv} \propto$  battery consumption in P2P communication betw.  $u, v$

# 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  $\approx 5.5$  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<sub>1</sub> with  $d : E \rightarrow \mathbb{Q}_+$  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  $\ell = |\text{sh. path } p : i \rightarrow j| = \sum_{\{u,v\} \in p} (-1)^{s_{uv}} d_{uv} \in \mathbb{Q}$   
for some  $s_{uv} \in \{0, 1\}$
- ▶ then  $x_j = x_i \pm \ell \rightarrow x_j \in \mathbb{Q}$
- ▶  $\Rightarrow$  **verification of**

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

**in polytime**

▶ **DGP<sub>K</sub> may not be in NP for  $K > 1$**

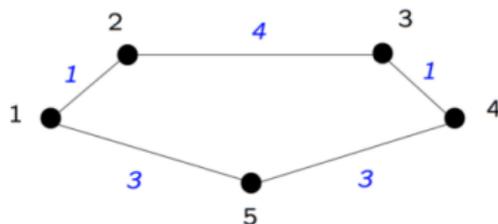
don't know how to check  $\|x_i - x_j\|_2 = d_{ij}$  in polytime for  
 $x \notin \mathbb{Q}^{nK}$

# Hardness

## PARTITION is NP-hard

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

- ▶ Reduce PARTITION to  $DGP_1$
- ▶  $a \rightarrow$  cycle  $C$   
 $V(C) = \{1, \dots, n\}$ ,  $E(C) = \{\{1, 2\}, \dots, \{n, 1\}\}$
- ▶ 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:*



# PARTITION is YES $\Rightarrow$ DGP<sub>1</sub> is YES

- ▶ **Given:**  $I \subset \{1, \dots, n\}$  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  $x_{n+1} = x_1$*

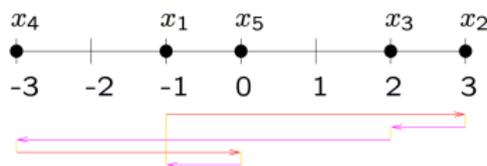
# PARTITION is YES $\Rightarrow$ DGP<sub>1</sub> is YES

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

$$\begin{aligned}(1) = (2) &\Rightarrow \sum_{i \in I} (x_{i+1} - x_i) = \sum_{i \notin I} (x_i - x_{i+1}) \Rightarrow \sum_{i \leq n} (x_{i+1} - x_i) = 0 \\ &\Rightarrow (x_{n+1} - x_n) + (x_n - x_{n-1}) + \cdots + (x_3 - x_2) + (x_2 - x_1) = 0 \\ &\hspace{20em} \Rightarrow x_{n+1} = x_1\end{aligned}$$

# PARTITION is NO $\Rightarrow$ DGP<sub>1</sub> is NO

- ▶ By contradiction: suppose DGP<sub>1</sub> is YES,  $x$  realization of  $C$
- ▶  $F = \{\{u, v\} \in E(C) \mid x_u \leq x_v\}$ ,  
 $E(C) \setminus F = \{\{u, v\} \in E(C) \mid x_u > x_v\}$
- ▶ Trace  $x_1, \dots, x_n$ : follow edges in  $F$  ( $\rightarrow$ ) and in  $E(C) \setminus F$  ( $\leftarrow$ )



DGP<sub>1</sub> instance is YES  $\Rightarrow$

$$\sum_{\{u,v\} \in F} (x_v - x_u) = \sum_{\{u,v\} \notin F} (x_u - x_v)$$

$$\sum_{\{u,v\} \in F} |x_u - x_v| = \sum_{\{u,v\} \notin F} |x_u - x_v|$$

$$\sum_{\{u,v\} \in F} d_{uv} = \sum_{\{u,v\} \notin F} d_{uv}$$

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

$$\Rightarrow \sum_{i \in J} a_i = \sum_{i \notin J} a_i$$

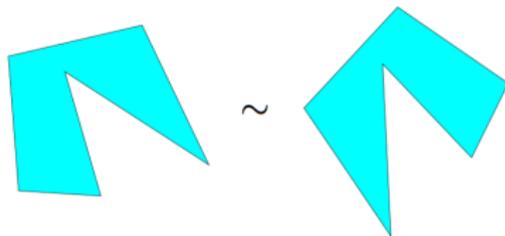
- ▶ So  $J$  solves Partition instance, contradiction
- ▶  $\Rightarrow$  DGP is NP-hard, DGP<sub>1</sub> is NP-complete

# 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  $\rho$
- ▶  $\Rightarrow$  if  $|\tilde{X}| > 0$ ,  $|\tilde{X}| = 2^{N_0}$

# 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

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

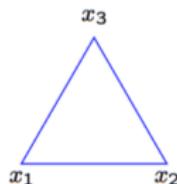
- ▶ Let  $X$  be the set of  $x \in \mathbb{R}^{nK}$  satisfying  $S$
- ▶ **Number of connected components of  $X$  is  $O(3^{nK})$**   
[Milnor 1964]
- ▶ Assume  $|X|$  is countable; then  $G$  cannot be flexible  
⇒ *each incongruent rltz is in a separate component*  
⇒ by Milnor's theorem, there's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

# Examples

$$V^1 = \{1, 2, 3\}$$

$$E^1 = \{\{u, v\} \mid u < v\}$$

$$d^1 = 1$$



$\rho$  congruence in  $\mathbb{R}^2$

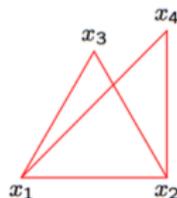
$\Rightarrow \rho x$  valid realization

$$|X| = 1$$

$$V^2 = V^1 \cup \{4\}$$

$$E^2 = E^1 \cup \{\{1, 4\}, \{2, 4\}\}$$

$$d^2 = 1 \wedge d_{14} = \sqrt{2}$$



$\rho$  reflects  $x_4$  wrt  $\overline{x_1x_2}$

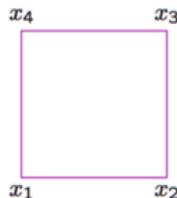
$\Rightarrow \rho x$  valid realization

$$|X| = 2 (\triangle, \diamond)$$

$$V^3 = V^2$$

$$E^3 = \{\{u, u+1\} \mid u \leq 3\} \cup \{1, 4\}$$

$$d^1 = 1$$



$\rho$  rotates  $\overline{x_2x_3}$ ,  $\overline{x_1x_4}$  by  $\theta$

$\Rightarrow \rho x$  valid realization

$|X|$  is uncountable

$$(\square, \diamond, \text{parallelogram}, \text{trapezoid}, \dots)$$

## 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

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

*Computationally: useless*  
**reformulate using slacks:**

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

# Unconstrained Global Optimization

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

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

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

- ▶ GO solvers from  $\approx 15$  years ago
- ▶ randomly generated protein data:  $\leq 50$  atoms
- ▶ cubic crystallographic grids:  $\leq 64$  atoms

# Constrained global optimization

- ▶  $\min_x \sum_{\{u,v\} \in E} \left| \|x_u - x_v\|^2 - d_{uv}^2 \right|$  **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*

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

## Theorem (Activity)

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

# Linearization

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

$$\Rightarrow \begin{cases} \forall \{i, j\} \in E & X_{ii} + X_{jj} - 2X_{ij} = d_{ij}^2 \\ & X = x x^\top \end{cases}$$

# Relaxation

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

$$\text{(relax)} \quad \Rightarrow \quad X - x x^\top \preceq 0$$

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

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

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

# SDP relaxation

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

How do we choose  $F$ ?

$$F \bullet X = \text{tr}(F^\top X)$$

# 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

$$\min \text{tr}(X)$$

$$\text{tr}(X) = \text{tr}(P^{-1}\Lambda P) = \text{tr}(P^{-1}P\Lambda) = \text{tr}(\Lambda) = \sum_i \lambda_i$$

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

# 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  $+\eta \text{tr}(X)$  to objective, with  $\eta \ll 1$   
*might help minimize solution rank*
- ▶  $\min \sum_{\{i,j\} \in E} (X_{ii} + X_{jj} - 2X_{ij}) + \eta \text{tr}(X)$

# 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 between 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

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

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

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



# Gershgorin's circle theorem

- ▶ Let  $A$  be symmetric  $n \times n$
- ▶  $\forall i \leq n$  let  $R_i = \sum_{j \neq i} |A_{ij}|$  and  $I_i = [A_{ii} - R_i, A_{ii} + R_i]$
- ▶ Then  $\forall \lambda$  eigenvalue of  $A$   $\exists i \leq n$  s.t.  $\lambda \in I_i$

## Proof

- ▶ Let  $\lambda$  be an eigenvalue of  $A$  with eigenvector  $x$
- ▶ Normalize  $x$  s.t.  $\exists i \leq n$   $x_i = 1$  and  $\forall j \neq i$   $|x_j| \leq 1$   
*let  $i = \operatorname{argmax}_j |x_j|$ , divide  $x$  by  $\operatorname{sgn}(x_i)|x_i|$   
(associate this  $i$  to  $\lambda$  in the rest of the proof)*
- ▶  $Ax = \lambda x \Rightarrow \sum_{j \neq i} A_{ij}x_j + A_{ii}x_i = \sum_{j \neq i} A_{ij}x_j + A_{ii} = \lambda x_i = \lambda$
- ▶ Hence  $\sum_{j \neq i} A_{ij}x_j = \lambda - A_{ii}$
- ▶ Triangle inequality and  $|x_j| \leq 1$  for all  $j \neq i \Rightarrow$   
 $|\lambda - A_{ii}| = \left| \sum_{j \neq i} A_{ij}x_j \right| \leq \sum_{j \neq i} |A_{ij}| |x_j| \leq \sum_{j \neq i} |A_{ij}| = R_i$   
hence  $\lambda \in I_i$

# DD $\Rightarrow$ PSD

- ▶ Assume  $A$  is DD,  $\lambda$  an eigenvalue of  $A$
- ▶  $\Rightarrow \forall i \leq n \quad A_{ii} \geq \sum_{j \neq i} |A_{ij}| = R_i$
- ▶  $\Rightarrow \forall i \leq n \quad A_{ii} - R_i \geq 0$
- ▶ By Gershgorin's circle theorem  $\lambda \geq 0$
- ▶  $\Rightarrow A$  is PSD

# DD Linearization

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

- ▶ linearize  $|\cdot|$  by additional matrix var  $T$   
 $\Rightarrow$  write  $|X|$  as  $T$

- ▶  $\Rightarrow (*)$  becomes

$$X_{ii} \geq \sum_{j \neq i} T_{ij}$$

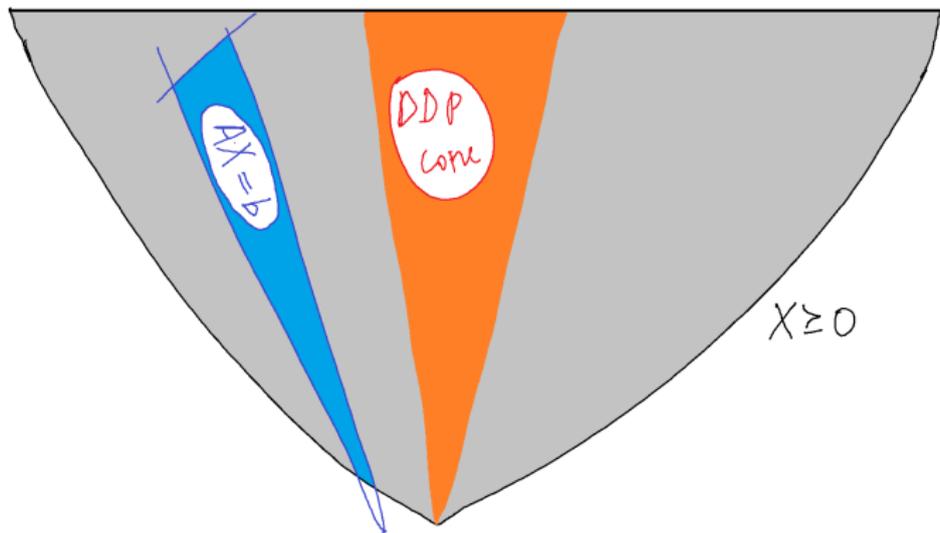
- ▶ add “sandwich” constraints  $-T \leq X \leq T$
- ▶ Can easily prove  $(*)$  in case  $X \geq 0$  or  $X \leq 0$ :

$$X_{ii} \geq \sum_{j \neq i} T_{ij} \geq \sum_{j \neq i} X_{ij}$$

$$X_{ii} \geq \sum_{j \neq i} T_{ij} \geq \sum_{j \neq i} -X_{ij}$$



# The issue with inner approximations



*DDP could be infeasible!*

# Exploit push-and-pull

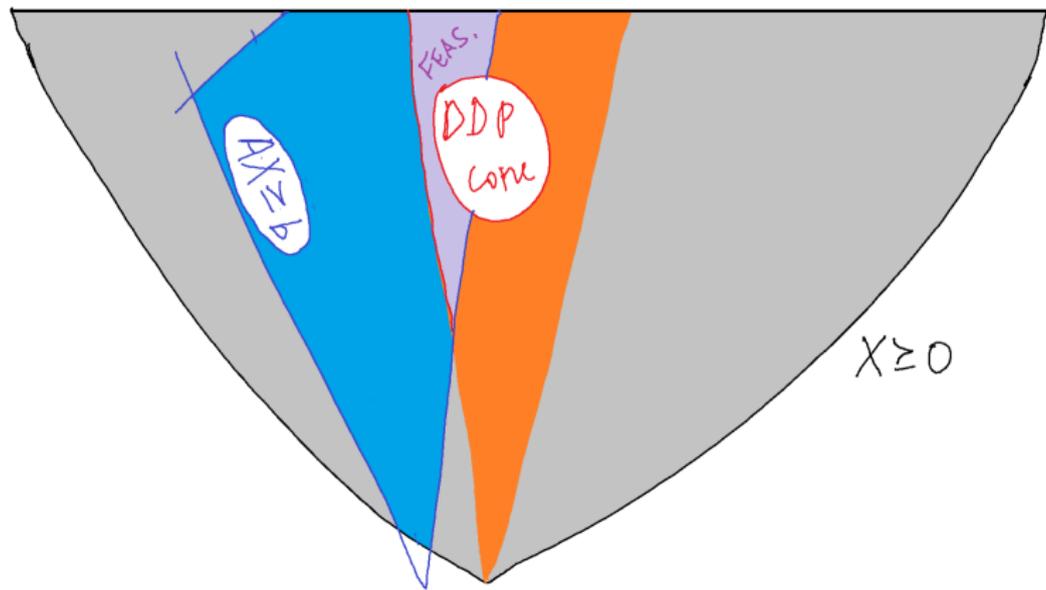
- ▶ **Enlarge the feasible region**
- ▶ **From**

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

- ▶ 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

$$\left. \begin{array}{l} \min \sum_{\{i,j\} \in E} (X_{ii} + X_{jj} - 2X_{ij}) \\ \forall \{i, j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} \geq d_{ij}^2 \\ \forall i \leq n \quad \sum_{\substack{j \leq n \\ j \neq i}} T_{ij} \leq X_{ii} \\ -T \leq X \leq T \\ T \geq 0 \end{array} \right\}$$

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

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

*vectors making acute angles with all 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

$$\mathcal{X} = \{x_1, \dots, x_p\} \wedge \quad \forall X \in C \exists \delta \in \mathbb{R}_+^p \quad X = \sum_{\ell \leq p} \delta_\ell x_\ell x_\ell^\top$$

# Representations of $\mathbb{DD}$

- ▶ Consider  $E_{ii}, E_{ij}^+, E_{ij}^-$  in  $\mathbb{S}_n$

Define  $\mathcal{E}_0 = \{E_{ii} \mid i \leq n\}$ ,  $\mathcal{E}_1 = \{E_{ij}^\pm \mid i < j\}$ ,  $\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_1$

- ▶  $E_{ii} = \text{diag}(0, \dots, 0, 1_i, 0, \dots, 0)$
- ▶  $E_{ij}^+$  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.**  $\mathbb{DD} = \text{cone generated by } \mathcal{E}$  [Barker & Carlson 1975]

*Pf. Rays in  $\mathcal{E}$  are extreme, all DD matrices generated by  $\mathcal{E}$*

- ▶ **Cor.**  $\mathbb{DD}$  finitely gen. by

$$\mathcal{X}_{\mathbb{DD}} = \{e_i \mid i \leq n\} \cup \{(e_i \pm e_j) \mid i < j \leq n\}$$

*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 representation

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

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

recall  $C^* \triangleq \{Y \in \mathbb{S}^n \mid \forall X \in C \ Y \bullet X \geq 0\}$

- ▶ ( $\supseteq$ ) 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  $Z \in C^* \setminus \{Y \mid \forall x \in \mathcal{X} (Y \bullet xx^\top \geq 0)\}$ 
  - ▶ 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:**  $X \bullet vv^\top = v^\top Xv$
- ▶ Use finitely generated dual cone theorem
- ▶ Decision variable matrix  $X$
- ▶ Constraints:

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

- ▶ **Cor.**  $\text{DD}^* \supset \text{PSD}$   
Pf.  $X \in \text{PSD}$  iff  $\forall v \in \mathbb{R}^n \quad v^\top Xv \geq 0$ , so certainly valid  $\forall v \in \mathcal{X}$
- ▶ If  $|\mathcal{X}|$  polysized, get compact formulation  
*otherwise use column generation*
- ▶  $|\mathcal{X}_{\text{DD}}| = |\mathcal{E}| = O(n^2)$

# Dual cone DDP formulation for DGP

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

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

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

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 relaxes original problem
- ▶ DualDDP relaxes SDP  
*hence also relaxes 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  $f : X \rightarrow \mathbb{R}$  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.

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

where  $\mathbb{E}(\cdot)$  is w.r.t. given Borel measure  $\mu$  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  $m \leq \text{poly}(n)$
- ▶ feasibility problem  $F \equiv [\bigcap_{k \leq m} \mathcal{X}_k \stackrel{?}{\neq} \emptyset]$
- ▶ SDP relaxation  $\forall k \leq m (Q^k \bullet X = a_k) \wedge X \succeq 0$  with soln.  $\bar{X}$
- ▶ **Algorithm:**  $T \leftarrow \text{factor}(\bar{X}); \quad y \sim \mathcal{N}^n(0, 1); \quad x' \leftarrow Ty$

## Then:

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

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

# 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

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

- ▶  $\sqrt{\|\bar{X}\|_2}$  arises from  $T$  (a factor of  $\bar{X}$ )
- ▶  $\sqrt{\ln n}$  arises from 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  $\bigcap_{\{i,j\} \in E} \mathcal{X}_{ij} \stackrel{?}{\neq} \emptyset$
- ▶ SDP relaxation  $X_{ii} + X_{jj} - 2X_{ij} = d_{ij}^2 \wedge X \succeq 0$  with soln.  $\bar{X}$

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

# Proof structure

- ▶ **Show that, on average,  $\forall k \leq m \operatorname{tr}((Ty)^\top Q^k(Ty)) = Q^K \bullet \bar{X} = a_k$** 
  - ▶ 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  $\mu$  of  $\varepsilon$ -neighbourhoods of

$$A_i^- = \{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) \leq Q^i \bullet \bar{X}\}$$

$$A_i^+ = \{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) \geq Q^i \bullet \bar{X}\}$$

$$A_i = \{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) = Q^i \bullet \bar{X}\}.$$

- ▶ 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.**  $T = \text{factor}(\bar{X})$   
**b.**  $y \sim \mathbf{N}^{nK}(0, \frac{1}{\sqrt{K}})$   
**c.**  $x' = Ty$
3. Use  $x'$  as starting point for a local NLP solver on formulation

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

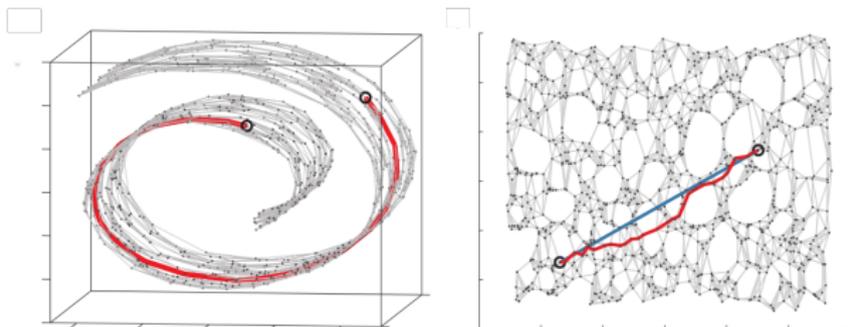
and return improved solution  $x$

## 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  $\Lambda, P$  so  $\tilde{B} = P^\top \Lambda P$
  - (c) Keep  $\leq K$  largest nonneg. eigenv. of  $\Lambda$  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  $\bar{x}$  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

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# The gist of random projections

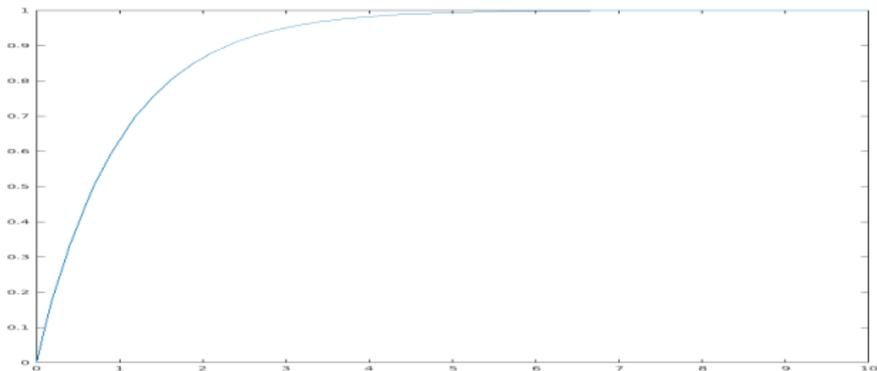
- ▶ Let  $A$  be a  $m \times n$  data matrix (columns in  $\mathbb{R}^m$ ,  $m \gg 1$ )
- ▶  $T$  short & fat, normally sampled componentwise

$$\underbrace{\begin{pmatrix} \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \end{pmatrix}}_T \underbrace{\begin{pmatrix} \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{pmatrix}}_A = \underbrace{\begin{pmatrix} \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{pmatrix}}_{TA}$$

- ▶ Then  $\forall i < j \|A_i - A_j\|_2 \approx \|TA_i - TA_j\|_2$  “wahp”

“wahp” = “with arbitrarily high probability”  
the probability of  $E_k$  (depending on some parameter  $k$ )  
approaches 1 “*exponentially fast*” as  $k$  increases

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



# Johnson-Lindenstrauss Lemma (JLL)

**Thm.**

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

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

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

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

**(nontrivial)** — result follows by probabilistic method

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

# In practice

- ▶  $\mathbb{P}(A \text{ and } TA \text{ approximately congruent}) \geq \frac{1}{n}$
- ▶ 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"];  
VKing = JohnsonLindenstrauss[VHimg, 0.1];  
VKcl = Timing[ClusteringComponents[VKing, 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  $x$
- ▶  $\text{sol}(F)$ : solution of  $F$
- ▶  $\mathcal{C}$ : 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.”**:  $\forall F \in \mathcal{C} \text{ sol}(F) \approx \text{sol}(RF)$  wahp
  - ▶ *Low distortion result holds for a formulation*
- 

- ▶ Today we see this for  $\mathcal{C} = \text{LP}$
- ▶ Can also be applied to QP, SDP

## Subsection 1

# Random projection theory

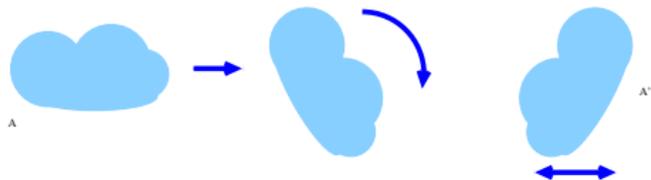
# The shape of a set of points

- ▶ **Lose dimensions but not too much accuracy**

Given  $A_1, \dots, A_n \in \mathbb{R}^m$  find  $k \ll m$  and  $A'_1, \dots, A'_n \in \mathbb{R}^k$  s.t.

$A$  and  $A'$  “have almost the same shape”

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



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

- ▶ *Approximate congruence*  $\equiv$  *small distortion*:

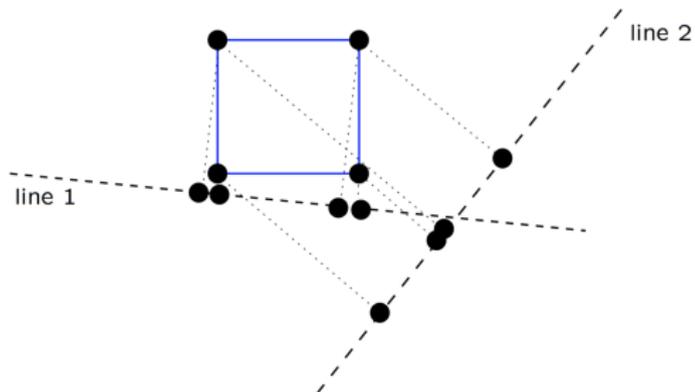
$A, A'$  have almost the same shape if

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

for some small  $\varepsilon > 0$

# Losing dimensions = “projection”

In the plane, hopeless



In 3D: no better

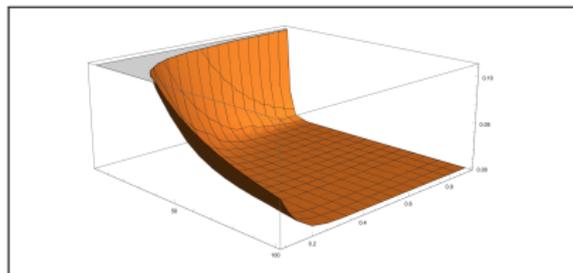
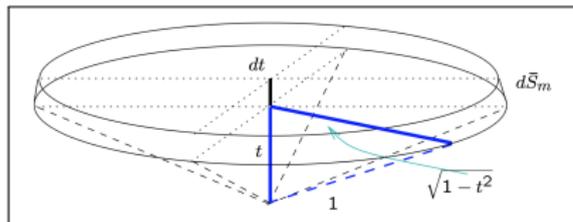
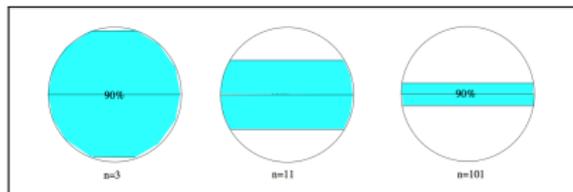
# Recall the JLL

**Thm.**

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

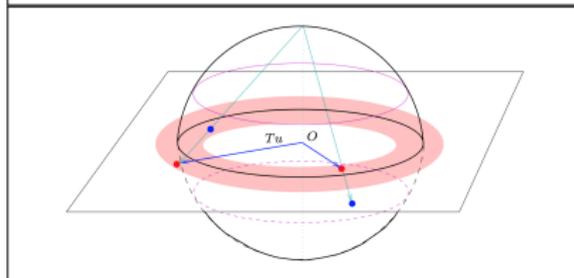
$$\forall x, y \in A \quad (1 - \varepsilon)\|x - y\| \leq \|Tx - Ty\| \leq (1 + \varepsilon)\|x - y\|$$

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



# Rnd proj preserve norms on avg

**Thm.**

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

**Proof**

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

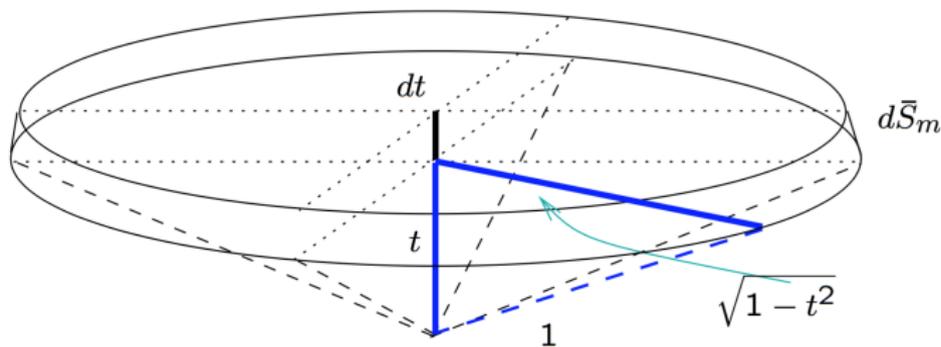
*Can we argue that the variance decreases wahp?*

# Surface area of a slice of hypersphere

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

## Lateral surface of infinitesimally high hypercylinder

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

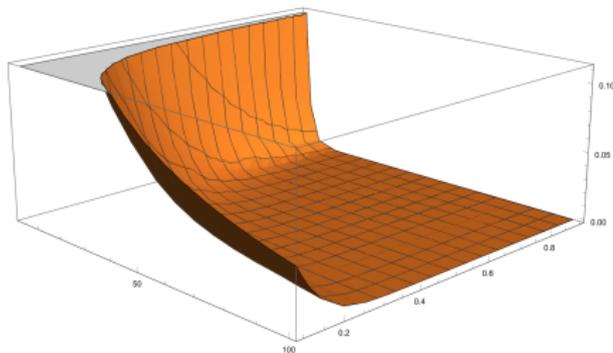


# Area of polar caps

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

$$1+x \leq e^x \text{ for all } x \quad \text{and} \quad \int_t^1 f(s)ds \leq \int_t^\infty f(s)ds \text{ for } f \geq 0$$

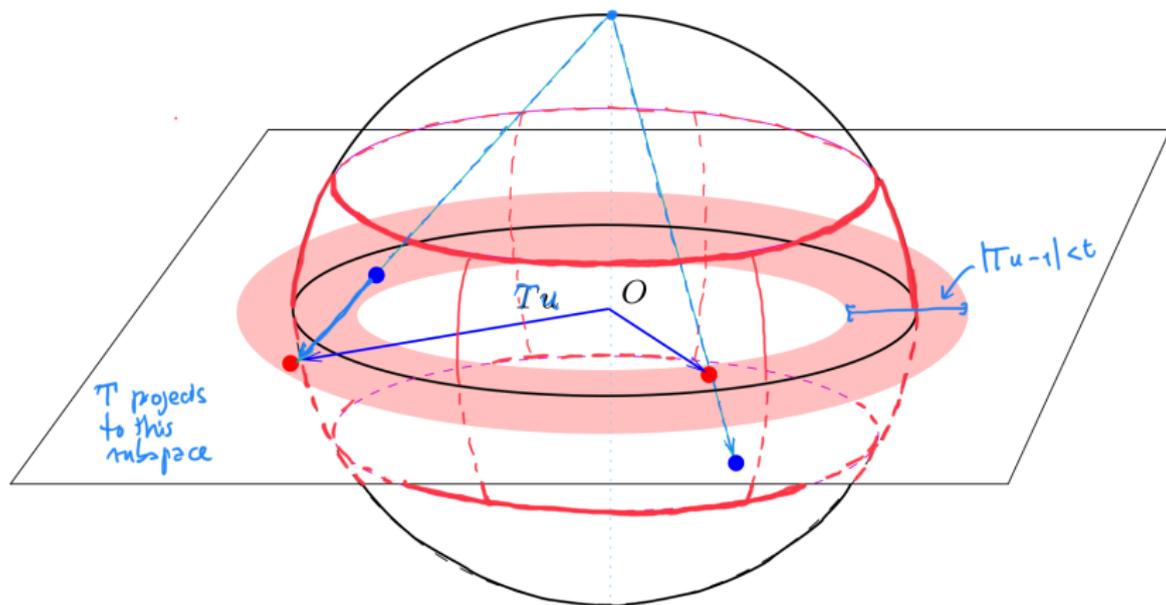
$$\Rightarrow \mathcal{A}^{\text{pc}} \leq 2S_{m-1} \int_t^\infty e^{-\frac{m-2}{2}s^2} ds = \frac{2S_{m-1}}{\sqrt{m-2}} \sqrt{\frac{\pi}{2}} \operatorname{erfc}\left(\frac{\sqrt{m-2}t}{\sqrt{2}}\right) \text{ “}\approx\text{” } O(e^{-t^2})$$



- ▶ **Polar caps area decreases as  $m \rightarrow \infty$  with  $t$  fixed**
- ▶ **Concentration of measure of the equatorial band**

# An intuitive explanation

- ▶ Polar caps area  $\mu(\mathcal{A}_t^m) = \mu(\{u \in \mathbb{S}^{m-1} \mid |u_m| > t\})$  decreases with  $t$  fixed as  $m \rightarrow \infty \Rightarrow$  **area of equatorial band  $\bar{\mathcal{A}}_t^m$  increases with same conditions**
- ▶  $T\bar{\mathcal{A}}_t^m = \{u \in \mathbb{S}^{m-1} \mid ||Tu\|^2 - 1\} \leq t\}$  has concentration of measure (if  $T$  is stereographic)



## Intermezzo: The union bound

- ▶ Events  $E_1, \dots, E_k$  such that  $\mathbf{P}(E_i) \geq 1 - t$  for each  $i \leq k$
- ▶ What is  $\mathbf{P}(\text{all } E_i)$ ?
- ▶  $\mathbf{P}(\text{all } E_i) = 1 - \mathbf{P}(\text{at least one } \neg E_i) \Rightarrow$

$$\begin{aligned} \mathbf{P}\left(\bigwedge_{i \leq k} E_i\right) &= 1 - \mathbf{P}\left(\bigvee_{i \leq k} (\neg E_i)\right) \geq \\ &\geq 1 - \sum_{i=1}^k \mathbf{P}(\neg E_i) \geq 1 - \sum_{i=1}^k (1 - (1 - t)) = 1 - kt \end{aligned}$$

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

# A syntactical explanation for $k = O(\varepsilon^{-2} \ln n)$

- ▶  $B =$  set of unit vectors; by “intuitive explanation”

$$\Rightarrow \forall u \in B \exists C > 0 \text{ s.t. } \mathbf{P}(1 - t \leq \|Tu\| \leq 1 + t) \geq 1 - e^{-Ct^2}$$

- ▶ By union bound:

$$\Rightarrow \mathbf{P}(\forall u \in B \ 1 - t \leq \|Tu\| \leq 1 + t) \geq 1 - |B|e^{-Ct^2}$$

- ▶  $\text{Prob.} \in [0, 1] \Rightarrow$  require  $1 - |B|e^{-Ct^2} > 0$ :

$$\Rightarrow |B|e^{-Ct^2} < 1$$

- ▶ Arbitrarily let  $t = \varepsilon\sqrt{k}$ :

$$\Rightarrow |B|e^{-C\varepsilon^2 k} < 1$$

- ▶  $\Rightarrow$   $k > C\varepsilon^{-2} \ln(|B|)$

# Apply to vector differences

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

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

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

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

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]:

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

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

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

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

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

- ▶ **Algorithm:**

- ▶  $\mathbf{P}(\exists x, y \in A \text{ with large JLL discrepancy}) \leq 1 - 1/n$
- ▶ Consider  $t > 1$  samplings of random projector (RP)  $T$
- ▶ Probability that all have large discrepancy:  $\leq (1 - 1/n)^t$
- ▶ Choose  $t$  so at least one  $T$  will be good with prob.  $\geq 0.99$ :  
 $(1 - 1/n)^t \leq 1 - 0.99 = 0.01$  yields  $t \geq \ln(0.01)/\ln(1 - 1/n)$

## Subsection 2

### Projecting feasibility

# Pure feasibility LP: easy cases

**Thm.**

$T : \mathbb{R}^m \rightarrow \mathbb{R}^k$  a JLL random projection, and  $b, A_1, \dots, A_n \in \mathbb{R}^m$ . 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  $\mathbf{P}\left(Tb \neq \sum_{i=1}^n x_i T A_i\right) \geq 1 - 2e^{-Ck}$

*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

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

for some constant  $C > 0$  (independent of  $n, k$ )

*by union bound*

# Separating hyperplanes

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

- ▶ **Convex  $C \subseteq \mathbb{R}^m, x \notin C$ : then  $\exists$  hyperplane  $c$  separating  $x, C$**
- ▶ In particular, true if  $C = \text{cone}(A_1, \dots, A_n)$  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  $c, b, A_1, \dots, A_n \in \mathbb{R}^m$  of unit norm s.t.  $b \notin \text{cone}\{A_1, \dots, A_n\}$  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:

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

for some constant  $C$ .

Proof

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

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

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

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

# The optimality projection theorem

- ▶ Assume  $\text{feas}(P)$  is bounded
- ▶ Assume all optima of  $P$  satisfy  $\sum_j x_j \leq \theta$  for some given  $\theta > 0$   
(prevents unboundedness)

**Thm.**

**Given  $\gamma > 0$ ,**

$$v(P) - \gamma \leq v(TP) \leq v(P) \quad (*)$$

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

more precisely, (\*) holds with prob.  $1 - 4ne^{-C(\varepsilon^2 - \varepsilon^3)k}$  where  $\varepsilon = \gamma / (2(\theta + 1)\eta)$  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  $\gamma = 0$

$$\left. \begin{array}{l} cx \leq v(P) - \gamma \\ Ax = b \\ x \geq 0 \end{array} \right\} \quad (12)$$

**Note:** for  $\gamma > 0$ , Eq. (12) is infeasible

- ▶ By feasibility projection theorem,

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

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

- ▶ **Restate:**  $cx < v(P) - \gamma \wedge TAx = Tb \wedge x \geq 0$  infeasible w.a.h.p.
- ▶  $\Rightarrow cx \geq v(P) - \gamma$  holds w.a.h.p. for  $x \in \text{feas}(TP)$
- ▶  $\Rightarrow v(P) - \gamma \leq v(TP)$

## Subsection 4

### Solution retrieval

# Projected solutions are infeasible in $P$

▶  $Ax = b \Rightarrow TAx = Tb$  by linearity

▶ However,  
**Thm.**

**For  $x \geq 0$  s.t.  $TAx = Tb, Ax = b$  with probability 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  $x' = \text{sol}(TP)$  and  $y' = \text{sol}(\text{dual}(TP))$
- ▶  $\Rightarrow (TA)^\top y' = (A^\top T^\top)y' = A^\top (T^\top y') \leq c$
- ▶  $\Rightarrow T^\top y'$  is a solution of  $\text{dual}(P)$
- ▶  $\Rightarrow$  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  $x^- = \min(0, x) \rightarrow 0$  as  $k \rightarrow \infty$*

## Subsection 5

### Application to quantile regression

# Conditional random variables

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

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

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

# Sample statistics

- ▶ expectation:

$$\hat{\mu} = \arg \min_{\mu \in \mathbb{R}} \sum_{i \leq m} (b_i - \mu)^2$$

- ▶ conditional expectation (*linear regression*):

$$\hat{\nu} = \arg \min_{\nu \in \mathbb{R}^p} \sum_{i \leq m} (b_i - \nu a^i)^2$$

- ▶ sample median:

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

- ▶ conditional sample median: *similarly*

# Quantile regression

- ▶ sample  $\tau$ -quantile:

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

- ▶ conditional sample  $\tau$ -quantile (*quantile regression*):

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

# Linear Programming formulation

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

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

- ▶ **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**
  - ▶  $m = 3783, p = 855$
  - ▶  $A = \text{hf1995f}, b = \log \text{avg}(A)$
  - ▶ 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  $90 \times 90$  pixels**
  - ▶  $m = 14596, p = 24300$
  - ▶ each row of  $A$  is an image vector,  $b = \sum A$
  - ▶ 62.4% dense
  - ▶ CPLEX killed by OS after  $\approx 30$ min (presumably for lack of RAM) on 16GB
  - ▶ could not load dataset in R
- ▶ **Results  $\Rightarrow$  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  $\ell_2$  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

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# Subsection 1

## Motivation

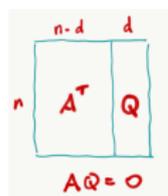
# Coding problem for costly channels

- ▶ **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  $z = Qw \in \mathbb{R}^n$   
*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  $w' = (Q^T Q)^{-1} Q^T z'$  (pseudoinverse)  
*What is the 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

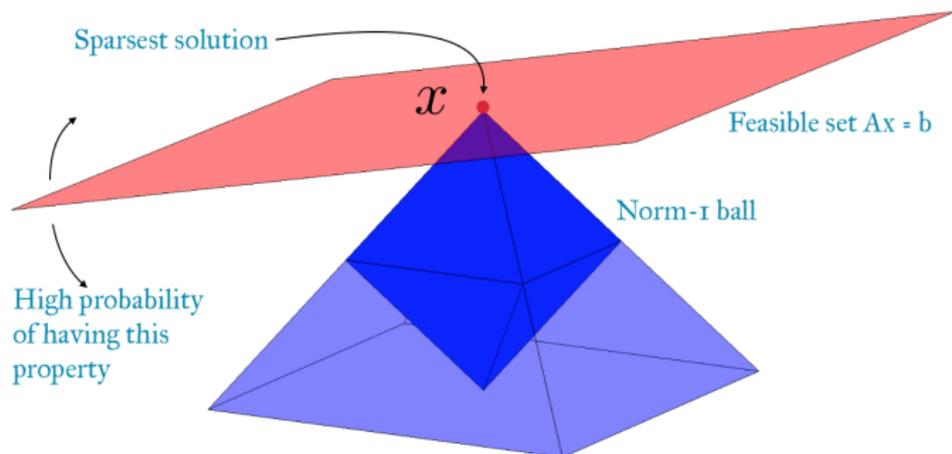
*Reduction from EXACT COVER BY 3-SETS [Garey&Johnson 1979, A6[MP5]]*

- ▶ **Relax to**  $P^1(A, b) \equiv \min\{\|x\|_1 \mid Ax = b\}$
- ▶ **Reformulate to LP:**

$$\left. \begin{array}{l} \min \quad \sum_{j \leq n} s_j \\ \forall j \leq n \quad -s_j \leq x_j \leq s_j \\ Ax = b \end{array} \right\} \quad (\dagger)$$

- ▶ **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*”
- ▶ *We always assume  $b \neq 0$*

# Graphical intuition 1

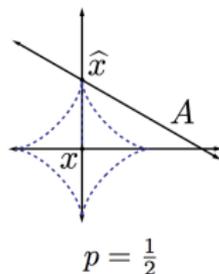
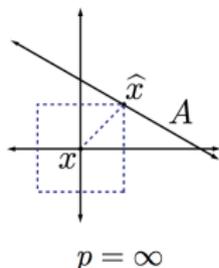
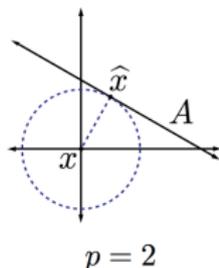
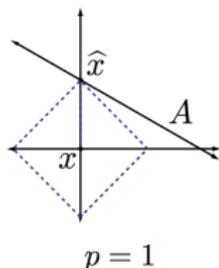


- ▶ Wouldn't work with  $\ell_2, \ell_\infty$  norms

*$Ax = b$  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

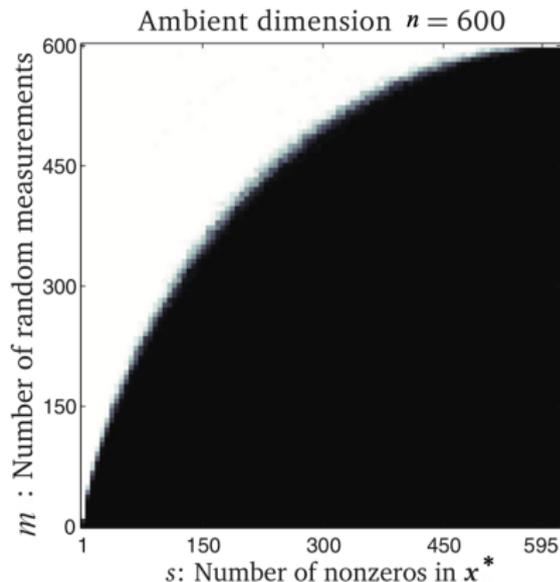
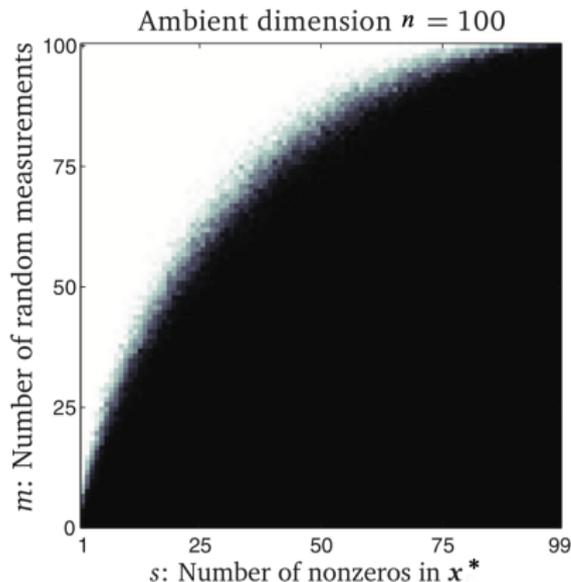


- ▶  $\hat{x}$  such that  $A\hat{x} = b$  approximates  $x$  in  $\ell_p$  norms
- ▶  $p = 1$  only convex case zeroing some components

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

# 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  $s$  undergoes a phase transition

## Subsection 3

### Theoretical results

# Main theorem and proof structure

▶ **Thm. If:**

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

then  $x^*$  is a “good approximation” of  $\bar{x}$     (★)

- ▶ **Prop.** If  $A$  has the null space property (NSP), (★) follows
- ▶ **Prop.** If  $A$  has restricted isometry property (RIP), NSP follows
- ▶ **Prop.** If  $A \sim N(0, 1)^{mn}$ , 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.  $Ax = b$ ,  $N_A = \text{null}(A)$ ,  $N_A^0 = N_A \setminus \{0\}$   
 $\Rightarrow \forall y \in N_A$  we have  $A(x + y) = Ax + Ay = Ax + 0 = b$
- ▶ For  $z \in \mathbb{R}^n$  and  $S \subseteq [n] = \{1, \dots, n\}$  let  $\bar{S} = [n] \setminus S$   
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  $z = z[S] + z[\bar{S}]$

# Null space property

- ▶ **Defn.**  $\text{NSP}_s(A) \equiv$

$$\forall S \subseteq [n] \ (|S| = s \rightarrow \forall y \in N_A^0 \ \|y[S]\|_1 < \|y[\bar{S}]\|_1)$$

*A has the null space property of order s*

- ▶ **Choose solution  $x^*$  of  $Ax = b$  with min  $\ell_1$  norm**

Let  $S = \text{supp}(x^*)$  and suppose  $|S| \leq s$

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

# Strength of $\text{NSP}_t$ as $t$ grows

*NSP Prop. states  $|\text{supp}(x^*)| \leq s$  but  $\text{NSP}_s(A)$  assumes  $|S| = s$ : why?*

## Lemma

$$\forall A \in \mathbb{R}^{m \times n}, t < s \leq n \quad \text{NSP}_s(A) \Rightarrow \text{NSP}_t(A)$$

## Proof

$\text{NSP}_s(A) \equiv \forall S \subseteq [n] (|S| = s \rightarrow \forall y \in N_A^0 \|y[S]\|_1 < \|y[\bar{S}]\|_1)$ , hence:  
given  $T, U \subseteq [n]$  with  $T, U$  nontrivial disjoint,  $|T| = t$  and  $|T \cup U| = s$ ,  
–  $\forall y \in N_A^0 \|y[T \cup U]\|_1 < \|y[\overline{T \cup U}]\|_1 = \|y[[n] \setminus (T \cup U)]\|_1 \Rightarrow$   
–  $\forall y \in N_A^0 \|y[T]\|_1 + \|y[U]\|_1 < \|y\|_1 - \|y[T]\|_1 - \|y[U]\|_1 \Rightarrow$   
–  $\forall y \in N_A^0 \|y[T]\|_1 < \|y[\bar{T}]\|_1 - 2\|y[U]\|_1$   
– **whence**  $\forall T \subseteq [n] (|T| = t \rightarrow \forall y \in N_A^0 \|y[T]\|_1 < \|y[\bar{T}]\|_1)$   
    **since**  $\|y[U]\|_1 > 0$ , and so  $\text{NSP}_t(A)$

# Proof of the NSP proposition ( $\Rightarrow$ )

$\forall x$  ( $x$  **uniq min** of  $P^1(A, Ax)$  and  $|\text{supp}(x)| = s$ )  $\Rightarrow$   $\text{NSP}_s(A)$

▶ **Let**  $y \in N_A^0$  **and**  $S \subseteq [n]$  **with**  $|S| = s$   
*assume wlog*  $Ay[S] \neq 0$

▶ **Note**  $|\text{supp}(y[S])| = s$  **since**  $|S| = s$   
*hence*  $y[S]$  **unique min** of  $P^1(A, Ay[S])$  **by hypothesis**

▶  $y = y[S] + y[\bar{S}] \in N_A^0 \Rightarrow 0 = Ay = Ay[S] + Ay[\bar{S}]$   
 $\Rightarrow A(-y[\bar{S}]) = Ay[S] \neq 0$

▶  $y[S] \neq -y[\bar{S}]$  **othw** by  $y = y[S] + y[\bar{S}]$  both would be scalings  
of  $y$  and hence both in  $N_A^0$ , which cannot happen as  $Ay[S] \neq 0$

▶  $\|y[S]\|_1$  **uniq min** and  $-y[\bar{S}]$  **feas** in  $P^1(A, Ay[S]) \Rightarrow$   
 $\| -y[\bar{S}]\|_1 = \|y[\bar{S}]\|_1 > \|y[S]\|_1$

# Proof of the NSP proposition ( $\Leftrightarrow$ )

$\text{NSP}_s(A) \Rightarrow \forall x^* (x^* \text{ uniq min } P^1(A, Ax^*) \wedge |\text{supp}(x^*)| = s)$

► Let  $x^* \in \mathbb{R}^n, b = Ax^*, S = \text{supp}(x^*)$  and  $|S| = s$

► Let  $\bar{x}$  soln. of  $Ax = b$ , then  $\bar{x} = x^* - y$  with  $y \in N_A$

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

$\ell_1$  norm strictly convex  $\Rightarrow x^* = \bar{x} \Rightarrow x^*$  uniq min

# A variant of the null space property

- ▶ **Motivation: “almost sparse solutions”**

given  $\hat{x}$  with  $|\text{supp}(\hat{x})| \geq s$  and  $b = A\hat{x}$ , assume  
 $\exists \epsilon > 0$  s.t.  $\bar{x} = \max(0, |\hat{x}| - \mathbf{1}\epsilon)$  has  $|\text{supp}(\bar{x})| = s$   
i.e.  $\hat{x}$  “almost” has support size  $s$

- ▶ Find closest approx  $x^*$  of  $\hat{x}$  with  $|\text{supp}(x^*)| = s$

- ▶ Adapt null space property:  $\text{NSP}_s^\rho(A) \Leftrightarrow$

$$\boxed{\exists \rho \in (0, 1) \forall S \subseteq [n] (|S| = s \rightarrow \forall y \in N_A^0 \|y[S]\|_1 \leq \rho \|y[\bar{S}]\|_1)}$$

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

$$\|x^* - \hat{x}\|_1 \leq 2 \frac{1+\rho}{1-\rho} \|\bar{x} - \hat{x}\|_1 \leq (n-s)\epsilon$$

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

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

- ▶ Moreover, if  $|\text{supp}(\hat{x})| = s$  then  $x^* = \hat{x} = \bar{x}$

# Proof of the $\text{NSP}_s^\rho$ proposition

- ▶  $x^*$  feasible in  $Ax = A\hat{x}$  so  $\exists! y \in N_A$  ( $x^* = \hat{x} + y$ )
- ▶  $\Rightarrow \|x^*\|_1 = \|\hat{x} + y\|_1 \leq \|\hat{x}\|_1$  since  $x^*$  min of  $P^1(A, A\hat{x})$
- ▶  $\|\hat{x} + y\|_1 = \sum_{j \in S} |\hat{x}_j + y_j| + \sum_{j \in \bar{S}} |\hat{x}_j + y_j|$   
 $\geq \sum_{j \in S} (|\hat{x}_j| - |y_j|) + \sum_{j \in \bar{S}} (|y_j| - |\hat{x}_j|)$  by triangle ineq
- ▶  $= \|\hat{x}[S]\|_1 - \|y[S]\|_1 + \|y[\bar{S}]\|_1 - \|\hat{x}[\bar{S}]\|_1$   
 $= \|\hat{x}\|_1 + \|y[\bar{S}]\|_1 - 2\|\hat{x}[\bar{S}]\|_1 - \|y[S]\|_1$   
 $= \|x\|_1 - 2\|\hat{x} - \bar{x}\|_1 + \|y[\bar{S}]\|_1 - \|y[S]\|_1 \quad (*)$
- ▶ Hence  $(*) \leq \|\hat{x} + y\|_1 \leq \|\hat{x}\|_1$ , whence  
 $\|\hat{x}\|_1 \geq \|\hat{x}\|_1 - 2\|\hat{x} - \bar{x}\|_1 + \|y[\bar{S}]\|_1 - \|y[S]\|_1$   
 $\Rightarrow 2\|\hat{x} - \bar{x}\|_1 \geq \|y[\bar{S}]\|_1 - \|y[S]\|_1$
- ▶ By  $\text{NSP}_s^\rho$ ,  $-\|y[S]\|_1 \geq \rho\|y[\bar{S}]\|_1$ , hence  
 $2\|\hat{x} - \bar{x}\|_1 \geq (1 - \rho)\|y[\bar{S}]\|_1$  whence  $\|y[\bar{S}]\|_1 \leq \frac{2}{1-\rho}\|\hat{x} - \bar{x}\|_1 \quad (\dagger)$
- ▶  $x^* = \hat{x} + y \Rightarrow \|x^* - \hat{x}\|_1 = \|y\|_1 = \|y[S]\|_1 + \|y[\bar{S}]\|_1$   
 by  $\text{NSP}_s^\rho$   $\|y[S]\|_1 \leq \rho\|y[\bar{S}]\|_1$  hence  $\|x^* - \hat{x}\|_1 \leq (1 + \rho)\|y[\bar{S}]\|_1$   
 by  $(\dagger)$   $\|x^* - \hat{x}\|_1 \leq 2\frac{1+\rho}{1-\rho}\|\hat{x} - \bar{x}\|_1$
- ▶ Further,  $\|\hat{x} - \bar{x}\|_1 = \|\hat{x} - \max(0, |\hat{x}| - \mathbf{1}\epsilon)\|_1 = \|\hat{x} - \hat{x}[S]\|_1 = \|\hat{x}[\bar{S}]\|_1 \leq |\bar{S}|\epsilon = (n - s)\epsilon$

# Restricted isometry property

- ▶  $\text{RIP}_s^\delta(A) \Leftrightarrow \forall x \in \mathbb{R}^n$  s.t.  $|\text{supp}(x)| = s$  we have

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

- ▶ **Prop.**  $\text{RIP}_{2s}^\delta(A) \wedge \rho = \frac{\sqrt{2}\delta}{1-\delta} < 1 \Rightarrow \text{NSP}_s^\rho(A)$

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

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

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

- ▶ **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  $|\text{supp}(\hat{x})| = s$ ,  $\delta < 1$ ,  $A$  s.t.  $\text{RIP}_{2s}^\delta(A)$ ,  $x^* = \arg P^0(A, A\hat{x})$ ; then  $x^* = \hat{x}$

**Pf.** Suppose false, let  $y = x^* - \hat{x} \neq 0$ ; by defn of  $x^*$  we have  $\|x^*\|_0 \leq \|\hat{x}\|_0 \leq s$ , hence  $\|y\|_0 \leq 2s$ ; 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  $\delta \in (0, 1) \rightarrow 1 \pm \delta > 0$ , hence  $0 \in (\alpha, \beta)$  where  $\alpha, \beta > 0$ , contradiction

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

- ▶ **Result of limited scope**, since we don't know if  $P^0(A, b)$  can be solved efficiently if  $A$  has the RIP

# Necessary eigenvalue conditions for RIP

- ▶ Recall  $\text{RIP}_s^\delta(A)$ :  $\forall x$  with  $S = \text{supp}(x)$  and  $|S| = s$

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

- ▶ Let  $A_I = (A_i \mid i \in I)$ , 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  $m \times s$ ,  $B(S) = A_S^\top A_S$  is  $s \times s$  and PSD

- ▶  $\Rightarrow 0 \leq \lambda_{\min}(B(S))\|x\|_2^2 \leq \langle B(S)x, x \rangle \leq \lambda_{\max}(B(S))\|x\|_2^2$   
*easy to see with  $B(S)$  replaced by diagonal  $PB(S)P^\top$*

- ▶ Let  $\lambda^L = \min_{|S|=s} \lambda_{\min}(B(S))$ ,  $\lambda^U = \max_{|S|=s} \lambda_{\max}(B(S))$

- ▶  $\Leftarrow \exists \delta > 0$  s.t.  $1 - \delta \leq \lambda^L \leq \lambda^U \leq 1 + \delta$   
*i.e. all eigenvalues of  $B(S)$  close to 1 for all  $S \subset [n]$  with  $|S| = s$*

# Construction of $A$ s.t. $\text{RIP}_s^\delta(A)$

- ▶ Need  $\lambda \approx 1$  for each eigenvalue  $\lambda$  of  $B(S)$
- ▶  $\Rightarrow$  Need  $\forall S \subseteq N \quad |S| = s \rightarrow A_S^\top A_S \approx I_s$
- ▶  $\Rightarrow$  Need

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

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

RIP holds for uncountably many vectors  $x$  with  $|\text{supp}(x)| = s$

JLL holds for given sets of finitely many vectors with any support

# Isotropic vectors

1. **Defn. Rnd vect**  $A_i \in \mathbb{R}^m$  **is isotropic iff**  $\text{cov}(A_i) = I_m$

remark: (a)  $\text{cov}(X) = E(XX^\top)$ ; (b) if  $A_i \sim N(0, 1)^m$  then  $A_i$  isotropic

2. **An isotropic rnd vect**  $A_i$  **is s.t.**  $\forall x \in \mathbb{R}^m \ E(\langle A_i, x \rangle^2) = \|x\|_2^2$

For two sq. symm. matrices  $B, C$  we have  $B = C$  iff  $\forall x (x^\top Bx = x^\top Cx)$ ; hence  $x^\top E(A_i A_i^\top) x = x^\top I_m x$ ; LHS is  $E(\langle A_i, x \rangle^2)$ , RHS is  $\|x\|_2^2$

3. **An isotropic rnd vect**  $x$  **in**  $\mathbb{R}^m$  **is s.t.**  $E(\|x\|_2^2) = m$

$E(\|x\|_2^2) = E(x^\top x) = E(\text{tr}(x^\top x)) = E(\text{tr}(xx^\top)) = \text{tr}(E(xx^\top)) = \text{tr}(I_m) = m$

4. **Indep isotr rnd vect**  $A_i, A_j$  **in**  $\mathbb{R}^m$  **have**  $E(\langle A_i, A_j \rangle^2) = m$

By conditional expectation  $E(\langle A_i, A_j \rangle^2) = E_{A_j}(E_{A_i}(\langle A_i, A_j \rangle^2 | A_j))$ ; by Item 2 inner expectation is  $\|A_j\|_2^2$ , by Item 3 outer is  $m$

5. **If**  $A_i \sim N(0, 1)^m$ ,  $\|A_i\|_2 = O(\sqrt{m})$  **wahp**

by Thm. 3.1.1 in [Vershynin, 2018]

6. **Independent rnd vectors are almost orthogonal**

Results above  $\Rightarrow \|A_i\|_2, \|A_j\|_2, \langle A_i, A_j \rangle = O(\sqrt{m})$ , normalize  $A_i, A_j$  to  $\bar{A}_i, \bar{A}_j$  to get  $\langle \bar{A}_i, \bar{A}_j \rangle = O(1/\sqrt{m}) \Rightarrow$  for  $m$  large  $\langle \bar{A}_i, \bar{A}_j \rangle \approx 0$

# Construction of $A$ s.t. $\text{RIP}_s^\delta(A)$

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

$$\forall s < m \left( \frac{s \ln(n/s)}{c_1} \leq m \rightarrow \text{Prob}(\text{RIP}_s^\delta(A)) \geq 1 - e^{-c_2 m} \right)$$

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

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

## ▶ In practice:

- ▶  $\text{Prob}(\text{RIP}_s^\delta(A)) = 0$  for  $m$  too small w.r.t.  $s$  fixed
- ▶ as  $m$  increases  $\text{Prob}(\text{RIP}_s^\delta(A)) > 0$
- ▶ as  $m$  increases even more  $\text{Prob}(\text{RIP}_s^\delta(A)) \rightarrow 1$  wahp
- ▶ achieve logarithmic compression for large  $n$  and fixed  $s$
- ▶  $A \sim \mathcal{N}(0, 1)^{m \times n} \wedge m \geq 10s \ln \frac{n}{s} \Rightarrow \text{RIP}_s^{1/3}(A)$  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](https://arxiv.org/pdf/1611.03935v1.pdf)
6. Candès & Tao  
[statweb.stanford.edu/~candes/papers/DecodingLP.pdf](https://statweb.stanford.edu/~candes/papers/DecodingLP.pdf)
7. Candès  
[statweb.stanford.edu/~candes/papers/ICM2014.pdf](https://statweb.stanford.edu/~candes/papers/ICM2014.pdf)
8. Davenport *et al.*, [statweb.stanford.edu/~markad/publications/ddek-chapter1-2011.pdf](https://statweb.stanford.edu/~markad/publications/ddek-chapter1-2011.pdf)
9. Lustig *et al.*, [people.eecs.berkeley.edu/~mlustig/CS/CSMRI.pdf](https://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*  $\mu$
2.  $w = \text{string2bitlist}(\mu) \in \{0, 1\}^d$
3. **send**  $z = Qw$ , **receive**  $\bar{z} = z + \hat{x}$ , **let**  $b = A\bar{z}$   
 $\Delta = s/n = \text{density of } \hat{x}$ ,  $Q$  is  $n \times d$  full rank with  $n > d$
4.  $x^* = \arg P^1(A, b)$
5.  $z^* = \bar{z} - x^*$
6.  $w^* = \text{cap}(\text{round}((Q^\top Q)^{-1}Q^\top z^*), [0, 1])$
7.  $\mu^* = \text{bitlist2string}(w^*)$
8. **evaluate**  $\mu_{\text{err}} = \|\mu - \mu^*\|$

## Parameter choice [Matousek]:

- ▶ **noise**  $\Delta = 0.08$
- ▶ **redundancy**  $n = R d$ , where  $R = 4$

# 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  $M^T M$  (orthonormal basis)  
*random rotation of standard basis* (used in original JLL proof)
  - ▶ Concatenate  $d$  eigenvectors to make  $Q$   
Concatenate  $m = n - d$  eigenvectors to make  $A$   
 *$AQ = 0$  by construction!*

# Subsection 5

## Improvements

# LP size reduction

- ▶ **Motivation**

- ▶ Reduce CPU time spent on LP
- ▶  $R = 4$  redundancy for  $\Delta = 0.08$  noise seems excessive

- ▶ **Size of basis pursuit LP**

- ▶  $Ax = b$  is an  $m \times n$  system where  $m = n - d$
- ▶ If  $n = Rd \gg d$ ,  $m$  “relatively close” to  $n$
- ▶ *Recall random projections for LP: use them!*

# Computational results

$d$	$n$	$\Delta$	$\epsilon$	$\alpha$	$\mu_{\text{err}}^{\text{org}}$	$\mu_{\text{err}}^{\text{prj}}$	CPU <sup>org</sup>	CPU <sup>prj</sup>
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

▶  $d = \lfloor \mu \rfloor, n = 4d, \Delta = 0.08, \epsilon = 0.2$

▶  $\alpha =$  **Achlioptas density**

$$P(T_{ij} = -1) = P(T_{ij} = 1) = \frac{\alpha}{2}$$

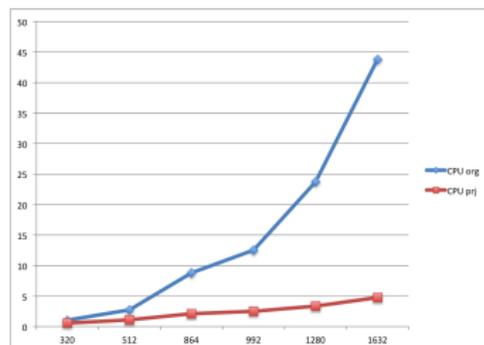
$$P(T_{ij} = 0) = 1 - \alpha$$

▶  $\mu_{\text{err}}$  = **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  $k \times p$  RP, use conc. meas. on  $\|z\|_2^2$

$\text{Prob}((1 - \varepsilon)\|z\|_2^2 \leq \|Tz\|_2^2 \leq (1 + \varepsilon)\|z\|_2^2) \geq 1 - 2e^{-C(\varepsilon^2 - \varepsilon^3)k}$   
given  $x, y \in \mathbb{R}^n$  apply to  $x + y, x - y$  and union bound:

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

with prob  $\geq 1 - 4e^{-C(\varepsilon^2 - \varepsilon^3)k}$ ; apply to std basis mtx  $I_p$ , get

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

$\Rightarrow \exists p$  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  $\Delta'$  “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  $Q$*
- ▶  $\max \sum_{\substack{i \leq m \\ j \leq n}} \text{Uniform}(-1, 1) A_{ij}$
- ▶ **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  $i \leq m$ )**

# Computational results

$m$	$n$	$\Delta'$	$\mu_{\text{err}}^{\text{org}}$	$\mu_{\text{err}}^{\text{prj}}$	CPU <sup>org</sup>	CPU <sup>prj</sup>
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  $\mu$  is short, set  $\Delta' = \Delta$  and use compressed sensing (CS)
- ▶ If  $\mu$  is longer, try increasing  $\Delta'$  and use CS
- ▶ If  $\mu$  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

<i>method</i>	<i>error</i>	<i>CPU</i>
CS	0	29.67s
JLL-CS	2	17.13s

*These results are consistent over 3 samplings*

*Technique applies to all sparse retrieval problems*

# Outline

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

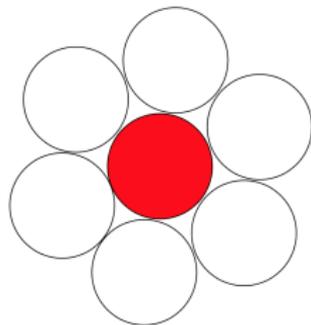
# Definition

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

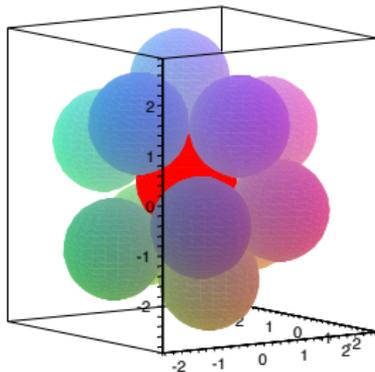
Funny story: *Newton and Gregory went down the pub...*

# Some examples

$$n = 6, K = 2$$



$$n = 12, K = 3$$



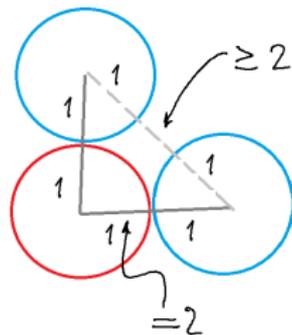
*more dimensions*

$n$	$\tau$ (lattice)	$\tau$ (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  $x_1, \dots, x_n \in \mathbb{R}^K$  such that:

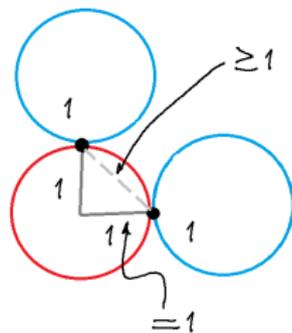
$$\begin{aligned} \forall i \leq n \quad \|x_i\|_2^2 &= 4 \\ \forall i < j \leq n \quad \|x_i - x_j\|_2^2 &\geq 4 \end{aligned}$$



# Contact point formulation

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

$$\begin{aligned} \forall i \leq n \quad \|x_i\|_2^2 &= 1 \\ \forall i < j \leq n \quad \|x_i - x_j\|_2^2 &\geq 1 \end{aligned}$$



# Spherical codes

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

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

- ▶ we aim at maximizing  $\text{kn}_z(K) \triangleq |\mathcal{C}|$

let  $\text{kn}(K) = \text{kn}_{\frac{1}{2}}(K)$

# 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  $\text{kn}(K)$

## Variables:

- ▶  $x_i \in \mathbb{R}^K$ : contact pt. of  $i$ -th surrounding sphere
- ▶  $\alpha_i = 1$  iff sphere  $i$  in configuration

$$\left. \begin{array}{ll} \max & \sum_{i=1}^n \alpha_i \\ \forall i \leq n & \|x_i\|^2 = \alpha_i \\ \forall i < j \leq n & \|x_i - x_j\|^2 \geq \alpha_i \alpha_j \\ \forall i \leq n & x_i \in [-1, 1]^K \\ \forall i \leq n & \alpha_i \in \{0, 1\} \end{array} \right\}$$

# Reformulating the binary products

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

$$\forall i < j \leq n \quad \beta_{ij} \leq \alpha_i$$

$$\forall i < j \leq n \quad \beta_{ij} \leq \alpha_j$$

$$\forall i < j \leq n \quad \beta_{ij} \geq \alpha_i + \alpha_j - 1$$

# 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

$$\left. \begin{array}{ll} \max_{x, \alpha} & \alpha \\ \forall i \leq n & \|x_i\|^2 = 1 \\ \forall i < j \leq n & \|x_i - x_j\|^2 \geq \alpha \\ \forall i \leq n & x_i \in [-1, 1]^K \\ & \alpha \geq 0 \end{array} \right\}$$

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

[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** ( $\arccos \alpha = \text{min. angular sep}$ )

**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?

- ▶  $\forall i \leq n \quad \mathbf{x}_i = (x_{i1}, \dots, x_{iK}) \mapsto (\vartheta_{i1}, \dots, \vartheta_{i,K-1})$
- ▶ **Formulation**

$$(\dagger) \quad \forall k \leq K \quad \rho \sin \vartheta_{i,k-1} \prod_{h=k}^{K-1} \cos \vartheta_{ih} = x_{ik}$$

$$(\ddagger) \quad \forall i < j \leq n \quad \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \geq \rho^2$$

$$\forall i \leq n, k \leq K \quad (\sin(\vartheta_{ik}))^2 + (\cos(\vartheta_{ik}))^2 = 1$$

$$\text{(optional)} \quad \rho = 1$$

- ▶ **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

$$\forall i, j \leq n \quad x_i \cdot x_j \longrightarrow X_{ij}$$

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:**

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

# SDP relaxation of binary constraints

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

# SDP relaxation of [MMS95]

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

# 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*

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

*(then eliminate all constraints in  $A$ )*

- ▶ *integrality of  $\alpha$  completely lost*

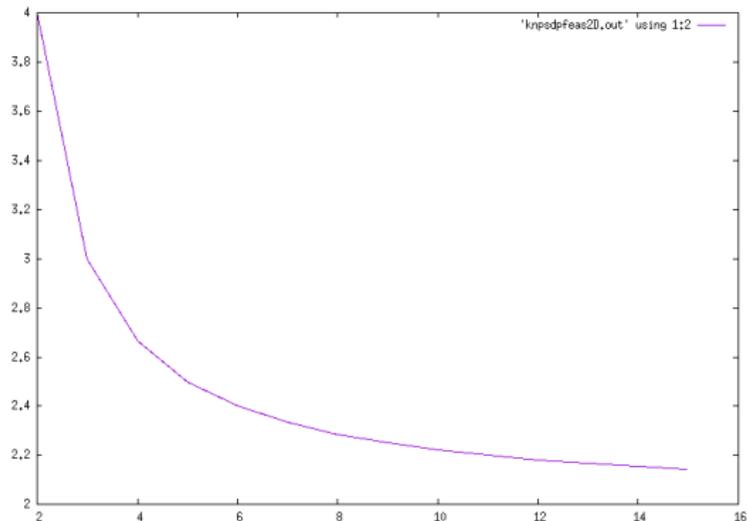
# SDP relaxation of [KBLM07]

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

# Computational experiments

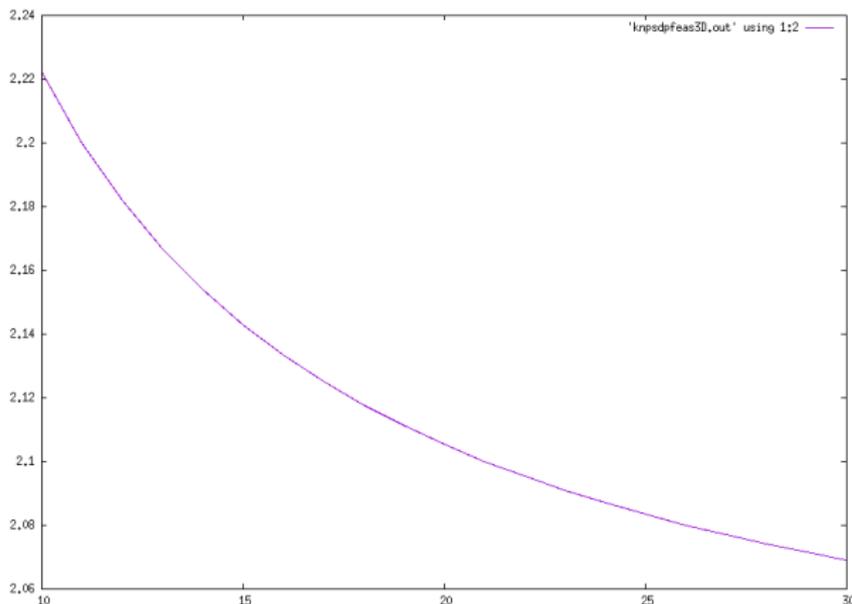
With  $K = 2$

$n$	$\alpha^*$
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*  $\rightarrow 2$ ?

# 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  $0, \frac{n}{n-1}, \dots, \frac{n}{n-1}$

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

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

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

# Dual SDP

$$\begin{aligned} \min \quad & \sum_i u_i + 2 \sum_{i < j} (y_{ij} + z_{ij}) \\ \sum_i u_i E_{ii} + \sum_{i < j} ((y_{ij} - z_{ij})(E_{ij} - E_{ji}) + w_{ij} A_{ij}) \quad & \succeq \quad 0 \\ \sum_{i < j} w_{ij} \quad & \geq \quad 1 \\ w, y, z \geq 0 \end{aligned}$$

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

$$\begin{aligned} \min \quad & \sum_i u_i + 2 \sum_{i < j} |v_{ij}| \\ \sum_i u_i E_{ii} + \sum_{i < j} (v_{ij}(E_{ij} - E_{ji}) + w_{ij} A_{ij}) \quad & \succeq \quad 0 \\ \sum_{i < j} w_{ij} \quad & \geq \quad 1 \\ w, v \geq 0 \end{aligned}$$

# Pulling a solution out of a hat

$$\alpha^* = \frac{2n}{n-1}$$

$$X^* = \frac{n}{n-1}I_n - \frac{1}{n-1}\mathbf{1}_n$$

$$u^* = \frac{2}{n-1}$$

$$w^* = \frac{1}{n(n-1)}$$

$$v^* = 0$$

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

# Solution verification

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

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

# Corollary

$$\lim_{n \rightarrow \infty} v(n, [\mathbf{KBLM07}]) = \lim_{n \rightarrow \infty} \frac{2n}{n-1} = 2$$

*as observed in computational experiments*

## Subsection 3

### Gregory's upper bound

# Surface upper bound

*Gregory 1694, Szpiro 2003*

Consider a  $kn(3)$  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  $kn(3)$ .



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

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

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

$$\int_{[-1,1]} \sigma_t dt = \sum_{\substack{t \in [-1,1] \\ \sigma_t \neq 0}} \sigma_t = \frac{1}{n} |\text{all pairs}| = \frac{n^2}{n} = n$$

$$\sigma_1 = \frac{1}{n} n = 1$$

$$\forall t \in (z, 1) \quad \sigma_t = 0$$

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

$$|\{\sigma_t > 0 \mid t \in [-1, 1]\}| < \infty$$

# Growing Delsarte's LP

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

$$\begin{aligned}\max |\mathcal{C}| = \max n &= \max_{\sigma} \sum_{\substack{t \in [-1, 1] \\ \sigma_t \neq 0}} \sigma_t \\ &= \sigma_1 + \max_{\sigma} \sum_{\substack{t \in [-1, z] \\ \sigma_t \neq 0}} \sigma_t = 1 + \max_{\sigma} \sum_{\substack{t \in [-1, z] \\ \sigma_t \neq 0}} \sigma_t\end{aligned}$$

*Note  $n$  not a parameter in this formulation*

- ▶ **Constraints:**

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

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

# The general approach

- ▶ We need  $\sigma$  to encode the fact that

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

- ▶ We use the algebraic theory in [Delsarte et al., “Spherical codes and designs”, *Geometriae Dedicata* 6:363-388, 1977]
- ▶ It involves the expression of a non-negative polynomial by means of a linear combination of Gegenbauer polynomials weighted by the  $\sigma_t$   
I will skip over the details
- ▶ You can also see the proof in [Odlyzko, Sloane, “New bounds on the number of unit spheres that can touch a unit sphere in  $n$  dimensions”, *J. of Comb. Theory A*, 26:210-214, 1979]

# Gegenbauer cuts

- ▶ Look for function family  $\mathcal{F} : [-1, 1] \rightarrow \mathbb{R}$  s.t.

$$\forall \phi \in \mathcal{F} \quad \sum_{\substack{t \in [-1, z] \\ \sigma_t \neq 0}} \phi(t) \sigma_t \geq 0$$

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

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

- ▶ Matlab knows them:  $G_h^K(t) = \text{gegenbauerC}(h, (K - 2)/2, t)$
- ▶ Octave knows them:  $G_h^K(t) = \text{gsl\_sf\_gegenpoly\_n}(h, \frac{K-2}{2}, t)$   
*need command `pkg load gsl` before function call*
- ▶ **They encode dependence on  $K$**

# Delsarte's LP

- **Primal:** (given some Gegenbauer polynomial index set  $H$ )

$$\left. \begin{array}{l} 1 + \max \sum_{t \in [-1, \frac{1}{2}]} \sigma_t \\ \forall h \in H \sum_{\substack{t \in [-1, \frac{1}{2}] \\ \sigma_t \neq 0}} G_h^K(t) \sigma_t \geq -G_h^K(1) \\ \forall t \in [-1, z] \sigma_t \geq 0. \end{array} \right\} \text{[DeIP]}$$

*MP syntax error: decision variables  $\sigma$  in sum quantifier!*

- **Dual:**

$$\left. \begin{array}{l} 1 + \min \sum_{h \in H} (-G_h^K(1)) d_h \\ \forall t \in [-1, z] \sum_{h \in H} G_h^K(t) d_h \geq 1 \\ \forall h \in H d_h \leq 0. \end{array} \right\} \text{[DeID]}$$

# Delsarte's theorem

- ▶ [Delsarte *et al.*, 1977; Odlyzko & Sloane, 1979]

## Theorem

Let  $d_0 > 0$  and  $F : [-1, 1] \rightarrow \mathbb{R}$  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  $\text{kn}_z(K) \leq \frac{F(1)}{d_0}$

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

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

## ► Primal:

$$\left. \begin{array}{l} 1 + \max \quad \sum_{\substack{t \in [-1, \frac{1}{2}] \\ \sigma_t \neq 0}} \sigma_t \\ \forall h \in H \quad \sum_{\substack{t \in [-1, \frac{1}{2}] \\ \sigma_t \neq 0}} G_h^K(t) \sigma_t \geq -1 \\ \forall t \in [-1, \frac{1}{2}] \quad \sigma_t \geq 0 \end{array} \right\} \text{[DelP]}$$

## ► Dual:

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

►  $d_0 = 1 \Rightarrow$  remove 0 from  $H$

# Focus on normalized [DeID]

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

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

**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 relaxation  $[\text{DeID}]_T$ :

$$\text{val}([\text{DeID}]_T) \leq \text{val}([\text{DeID}])$$

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

# SILP feasibility

- ▶ **Given SILP**  $\bar{S} \equiv \min\{c^\top x \mid \forall t \in \bar{T} \ a^\top(t)x \leq b(t)\}$
- ▶ **Relax to LP**  $S \equiv \min\{c^\top x \mid \forall t \in T \ a^\top(t)x \leq b(t)\}$   
with  $T \subsetneq \bar{T}$  and  $|T| < \infty$
- ▶ **Solve  $S$ , get solution  $x^*$**
- ▶ **Let  $\epsilon = \max_t\{a^\top(t)x^* - b(t) \mid t \in \bar{T}\}$**   

<i>continuous optimization w.r.t. single var. <math>t</math></i>
--
- ▶ **If  $\epsilon \leq 0$  then  $x^*$  feasible in  $\bar{S}$**   
 $\Rightarrow \text{val}(\bar{S}) \leq c^\top x^*$
- ▶ **If  $\epsilon > 0$  refine  $S$  and repeat**
- ▶ **Apply to  $[\text{DelD}]_T$ , get solution  $d^*$  feasible in  $[\text{DelD}]$**

# [DeID] feasibility

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

$$\left. \begin{array}{l} 1 + \min \sum_{h \in H} d_h \\ \forall t \in T \quad \sum_{h \in H} G_h^K(t) d_h \leq -1 \\ \forall h \in H \quad d_h \geq 0 \end{array} \right\} \text{[DeID]}_T$$

get solution  $d^*$

3. **Solve PP**  $\epsilon = \max_t \{1 + \sum_{h \in H} G_h^K(t) d_h^* \mid t \in [-1, z]\}$
4. If  $\epsilon \leq 0$  then  $d^*$  feasible in [DeID]  
 $\Rightarrow \text{kn}_z(K) \leq 1 + \sum_{h \in H} d_h^*$
5. Else refine  $T$  and repeat from Step 2

## Subsection 5

### Pfender's upper bound

# Pfender's upper bound theorem

Thm.

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

(i)  $\sum_{i,j \leq n} f(x_i \cdot x_j) \geq 0$       (ii)  $f(t) + c_0 \leq 0$  for  $t \in [-1, z]$       (iii)  $f(1) + c_0 \leq 1$

Then  $\text{kn}_z(K) = n \leq \frac{1}{c_0}$

([Pfender 2006])

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

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

# Pfender's LP

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

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

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

- ▶ Get SILP

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

- ▶ *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) $\forall t \in [-1, z] f(t) + c_0 \leq 0$
$\Rightarrow \text{kn}_z(K) \leq \frac{F(1)}{d_0}$	(iii) $f(1) + c_0 \leq 1$ $\Rightarrow \text{kn}_z(K) \leq \frac{1}{c_0}$

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

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

$\Rightarrow$  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

*If  $\mathcal{F}$  are Gegenbauer polynomials, Delsarte requires weaker condition and yields tighter bound; but Pfender allows for more general  $\mathcal{F}$ , can get improved results  
see [Pfender, "Improved Delsarte bounds for spherical codes in small dimensions",  
*J. Comb. Theory A*, 114:1133-1147, 2007]*

# The final, easy improvement

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

# Outline

## Introduction

- MP language
- Solvers
- MP systematics
- Some applications

## Decidability

- Formal systems
- Gödel
- Turing
- Tarski
- Completeness and incompleteness
- MP solvability

## Efficiency and Hardness

- Some combinatorial problems in NP
- NP-hardness
- Complexity of solving MP formulations

## Distance Geometry

- The universal isometric embedding
- Dimension reduction
- Distance geometry problem
- Distance geometry in MP
- DGP cones
- Barvinok's Naive Algorithm
- Isomap for the DGP

## Summary

### Random projections in LP

- Random projection theory
- Projecting feasibility
- Projecting optimality
- Solution retrieval
- Application to quantile regression

### Sparsity and $\ell_1$ minimization

- Motivation
- Basis pursuit
- Theoretical results
- Application to noisy channel encoding
- Improvements

### Kissing Number Problem

- Lower bounds
- Upper bounds from SDP?
- Gregory's upper bound
- Delsarte's upper bound
- Pfender's upper bound

### Clustering in Natural Language

- Clustering on graphs
- Clustering in Euclidean spaces
- Distance instability
- MP formulations
- Random projections again

# Job offers



## Optimisation / Operations Senior Manager

VINCI Airports

Rueil-Malmaison, Ile-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](http://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](http://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! ★★★★★ reviews - 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  $\text{NER}(D)$  = named entity words

## Subsection 1

# Clustering on graphs

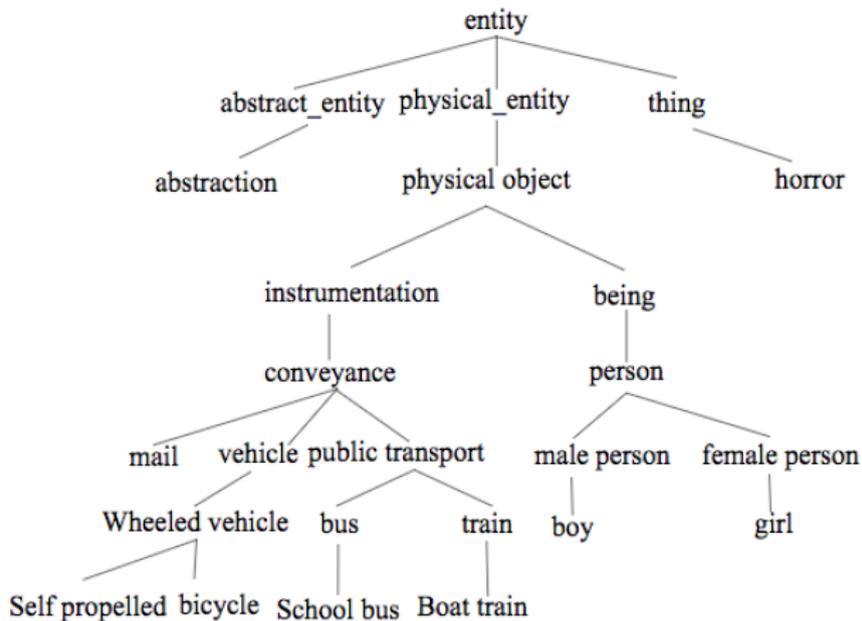
# Exploit NER to infer relations

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 Miers 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 Java 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





# Wu-Palmer word similarity

*Semantic WordNet distance between words  $w_1, w_2$*

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

- ▶ **lcs:** *lowest common subsumer*

earliest common word in paths from both words to WordNet root

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

**Example:**  $\text{wup}(\text{dog}, \text{boat})?$

`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

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

# Extensions of Wu-Palmer similarity

- ▶ to lists of words  $H, L$ :

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

- ▶ to pairs of documents  $D_1, D_2$ :

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

- ▶  $\text{wup}$  and its extensions are always in  $[0, 1]$

# The similarity matrix

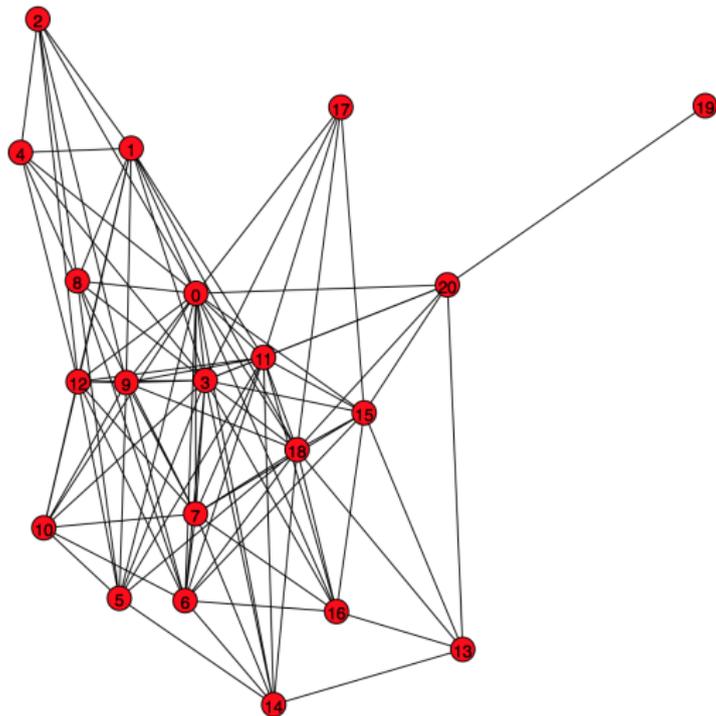
1.00	0.63	0.51	0.51	0.66	0.45	0.46	0.47	0.72	0.58	0.54	0.50	0.72	0.38	0.49	0.47	0.47	0.44	0.54	0.31	0.44
0.63	1.00	0.45	0.45	0.54	0.40	0.42	0.42	0.57	0.49	0.46	0.45	0.59	0.35	0.43	0.42	0.42	0.41	0.47	0.32	0.40
0.51	0.45	1.00	0.40	0.53	0.35	0.37	0.37	0.58	0.47	0.43	0.40	0.59	0.28	0.39	0.37	0.38	0.35	0.43	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

# The similarity matrix

*Too uniform! Try zeroing values below median*

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

# The graph



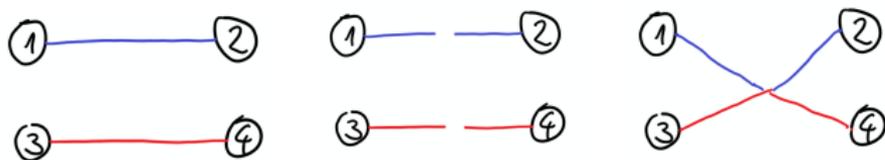
$G = (V, E)$ , weighted adjacency matrix  $A$

*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* =  $(k_1, \dots, k_n)$  where  $k_i = |N(i)|$
- ▶ *“expected”* = over all possible “half-edge” recombinations



- ▶ expected edges between  $u, v$ :  $k_u k_v / (2m)$  where  $m = |E|$
- ▶  $\text{mod}(u, v) = \frac{1}{2m} (A_{uv} - k_u k_v / (2m))$
- ▶  $\text{mod}(G) = \sum_{\{u,v\} \in E} \text{mod}(u, v) x_{uv}$   
 $x_{uv} = 1$  if  $u, v$  in the same cluster and 0 otherwise
- ▶ *“Natural extension”* to weighted graphs:  $k_u = \sum_v A_{uv}$ ,  $m = \sum_{u,v} A_{uv}$

# Use modularity to define clustering

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

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

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

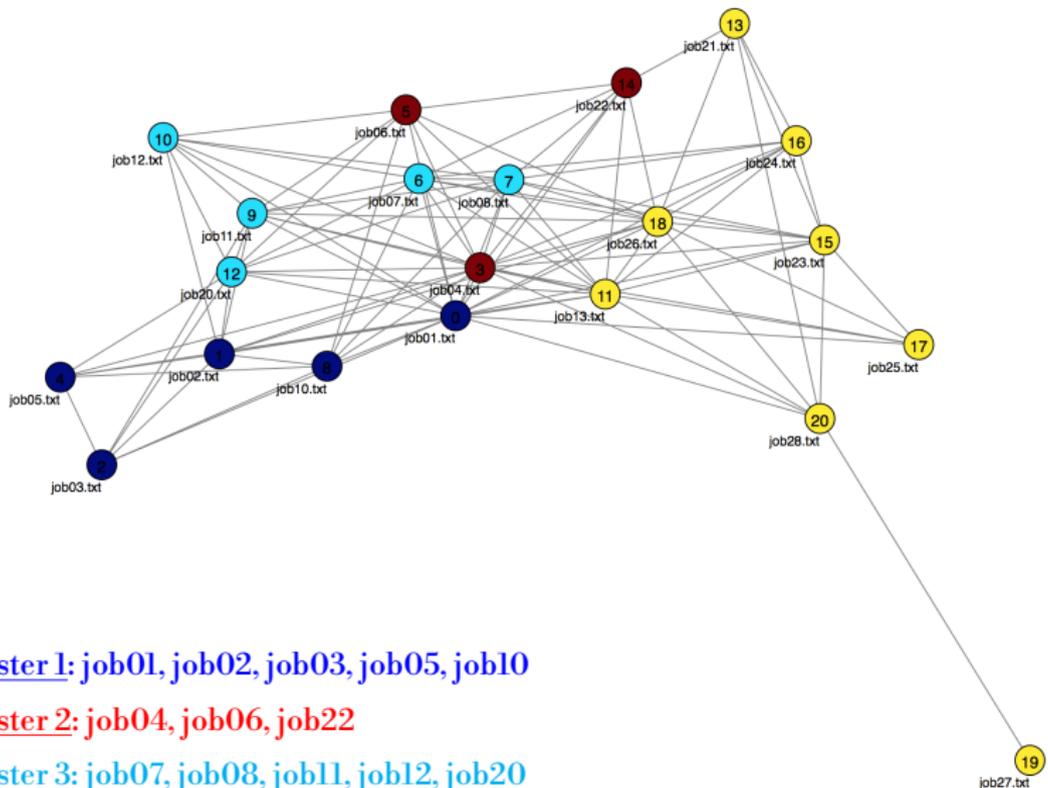
$$\forall i < j < k \quad x_{ij} + x_{jk} - x_{ik} \leq 1$$

$$\forall i < j < k \quad x_{ij} - x_{jk} + x_{ik} \leq 1$$

$$\forall i < j < k \quad -x_{ij} + x_{jk} + x_{ik} \leq 1$$

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

# The resulting clustering



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

cluster 2: job04, job06, job22

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

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

# Is it good?

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

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

## Subsection 2

# Clustering in Euclidean spaces

# Clustering vectors

Most frequent words  $w$  over collection  $C$  of documents  $d$

`./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

$$\text{tfidf}_C(w, d) = \frac{|(t \in d \mid t = w)| |C|}{|\{h \in C \mid w \in h\}|}$$

$\text{keyword}_C(i, d) =$  *word  $w$  having  $i^{\text{th}}$  best  $\text{tfidf}_C(w, d)$  value*

$$\text{vec}_C^m(d) = (\text{tfidf}_C(\text{keyword}_C(i, d), d) \mid i \leq m)$$

*Transforms documents to vectors*

# Minimum sum-of-squares clustering

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

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

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

- ▶  **$k$ -means alg.:** given initial clustering  $C_1, \dots, C_k$ 
  - 1:  $\forall j \leq k$  compute  $y_j = \text{centroid}(C_j)$
  - 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  
Deezer  
Accenture  
Expedia  
Google  
Aledade  
Llamasoft

AXA  
Alstom  
Elektrobit  
Ford  
Marriott  
Amazon 1-3  
CSX  
WestRock  
MITRE  
Clarity  
fragments 1-2

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

Deezer  
Elektrobit  
Google  
Aledade

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

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

AXA

Deezer

Expedia

Ford

Marriott

Llamasoft

Amazon 1-3

CSX

WestRock

MITRE

Clarity

fragments 1-2

Vinci

Accenture

Alstom

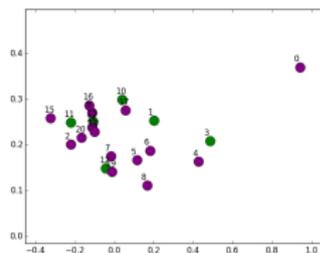
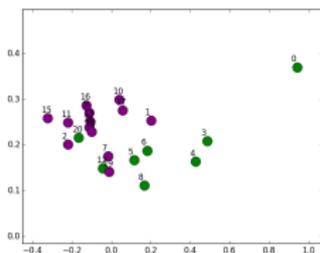
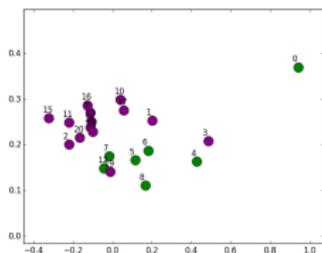
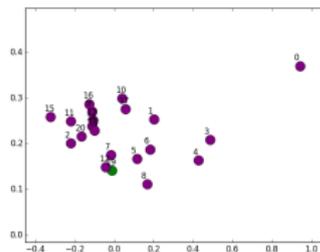
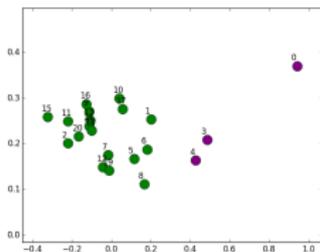
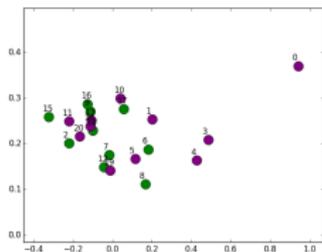
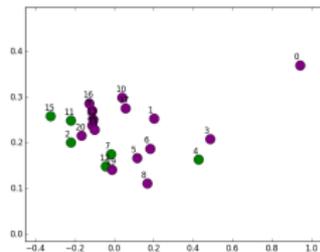
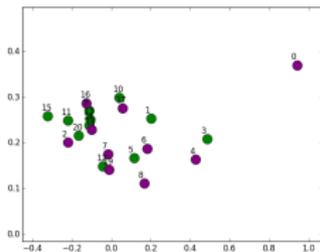
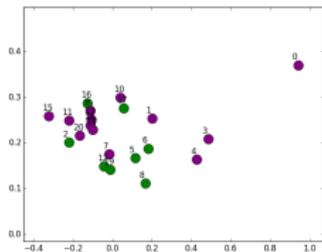
Elektrobit

Google

Aledade

*A fickle algorithm*

# We can't trust $k$ -means: why?



## Subsection 3

### Distance instability

# Nearest Neighbours

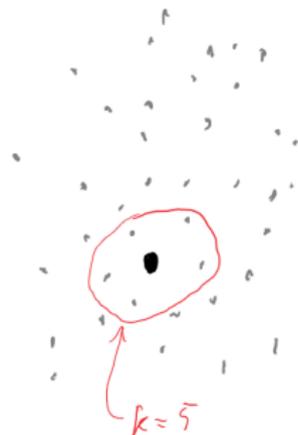
## $k$ -NEAREST NEIGHBOURS ( $k$ -NN).

Given:

- ▶  $k \in \mathbb{N}$
- ▶ a distance function  $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+$
- ▶ 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:

- $|\mathcal{Y}| = k$
- $\forall y \in \mathcal{Y}, x \in \mathcal{X} \quad (d(z, y) \leq d(z, x))$



- ▶ **basic problem in data science**

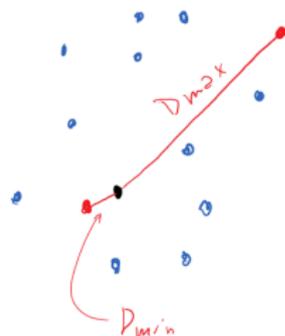
- ▶ pattern recognition, computational geometry, machine learning, data compression, robotics, recommender systems, information retrieval, natural language processing and more

- ▶ **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  $\{d^m : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+\}_m$
- ▶ 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  $X_i^m$
  - ▶  $D_{\min}^m = \min_{i \leq \ell} d^m(Z^m, X_i^m)$
  - ▶  $D_{\max}^m = \max_{i \leq \ell} d^m(Z^m, X_i^m)$



# Distance Instability Theorem

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

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

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

$$\forall i \leq \ell \lim_{m \rightarrow \infty} \text{Var} \left( \frac{(d^m(Z^m, X_i^m))^p}{\mathbb{E}((d^m(Z^m, X_i^m))^p)} \right) = 0$$

then, for any  $\varepsilon > 0$ ,

$$\lim_{m \rightarrow \infty} \mathbb{P}(D_{\max}^m \leq (1 + \varepsilon) D_{\min}^m) = 1$$

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  $\lim_{m \rightarrow \infty} \mathbb{E}(B^m) = b \wedge \lim_{m \rightarrow \infty} \text{Var}(B^m) = 0$ ; **then**

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

denoted  $B^m \rightarrow_{\mathbb{P}} b$

- ▶ **Slutsky's theorem.**  $\{B^m\}_m$  seq. of rnd. vars and  $g$  a continuous function; if  $B^m \rightarrow_{\mathbb{P}} b$  and  $g(b)$  exists, **then**  $g(B^m) \rightarrow_{\mathbb{P}} g(b)$
- ▶ **Corollary.** If  $\{A^m\}_m, \{B^m\}_m$  seq. of rnd. vars. s.t.  $A^m \rightarrow_{\mathbb{P}} a$  and  $B^m \rightarrow_{\mathbb{P}} b \neq 0$  **then**  
 $\frac{A^m}{B^m} \rightarrow_{\mathbb{P}} \frac{a}{b}$

# Proof

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

# When it applies

- ▶ iid random variables from any distribution
- ▶ Particular forms of correlation  
e.g.  $U_i \sim \text{Uniform}(0, \sqrt{i})$ ,  $X_1 = U_1$ ,  $X_i = U_i + (X_{i-1}/2)$  for  $i > 1$
- ▶ Variance tending to zero  
e.g.  $X_i \sim N(0, 1/i)$
- ▶ 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  $\approx$ ) 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

# Why?

- ▶ With k-means being so fast, why bother with MP?
- ▶ **Principle:**  
*changing an MP is easier than changing an algorithm*
- ▶ **Side constraints**  
e.g. clusters are spheres, or other shapes
- ▶ **Clustering subproblems**  
e.g. assign resources subject to optimal clustering
- ▶ **MP delivers a bound**  
“can’t do better than bound” guarantee

# MP formulation

$$\left. \begin{array}{ll} \min_{x,y,s} & \sum_{i \leq n} \sum_{j \leq k} \|p_i - y_j\|_2^2 x_{ij} \\ \forall j \leq k & \frac{1}{s_j} \sum_{i \leq n} p_i x_{ij} = y_j \\ \forall i \leq n & \sum_{j \leq k} x_{ij} = 1 \\ \forall j \leq k & \sum_{i \leq n} x_{ij} = s_j \\ \forall j \leq k & y_j \in \mathbb{R}^m \\ & x \in \{0, 1\}^{nk} \\ & s \in \mathbb{N}^k \end{array} \right\} \text{(MSSC)}$$

*MINLP: nonconvex terms; continuous, binary and integer variables*

# Reformulation

The (MSSC) formulation has the same optima as:

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

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

# Products of binary and continuous vars.

- ▶ Suppose term  $xy$  appears in a formulation
- ▶ Assume  $x \in \{0, 1\}$  and  $y \in [0, 1]$  is bounded
- ▶ means “either  $z = 0$  or  $z = y$ ”
- ▶ Replace  $xy$  by a new variable  $z$
- ▶ Adjoin the following constraints:

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

- ▶  $\Rightarrow$  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  $y \in [y^L, y^U]$  is bounded
- ▶ means “either  $z = 0$  or  $z = y$ ”
- ▶ Replace  $xy$  by a new variable  $z$
- ▶ Adjoin the following constraints:

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

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

[L. et al. 2009]

# MSSC is a convex MINLP

$$\begin{aligned}
 & \min_{x,y,P,\chi,\xi} \sum_{i \leq n} \sum_{j \leq k} \chi_{ij} \\
 & \forall i \leq n, j \leq k \quad 0 \leq \chi_{ij} \leq P_{ij} \\
 & \forall i \leq n, j \leq k \quad P_{ij} - (1 - x_{ij})P^U \leq \chi_{ij} \leq x_{ij}P^U \\
 & \forall i \leq n, j \leq k \quad \|p_i - y_j\|_2^2 \leq P_{ij} \quad \leftarrow \text{convex} \\
 & \forall j \leq k \quad \sum_{i \leq n} p_i x_{ij} = \sum_{i \leq n} \xi_{ij} \\
 & \forall i \leq n, j \leq k \quad y_j - (1 - x_{ij}) \max(|y^L|, |y^U|) \leq \xi_{ij} \leq y_j + (1 - x_{ij}) \max(|y^L|, |y^U|) \\
 & \forall i \leq n, j \leq k \quad -x_{ij} \max(|y^L|, |y^U|) \leq \xi_{ij} \leq x_{ij} \max(|y^L|, |y^U|) \\
 & \forall i \leq n \quad \sum_{j \leq k} x_{ij} = 1 \\
 & \forall j \leq k \quad y_j \in [y^L, y^U] \\
 & \quad \quad \quad x \in \{0, 1\}^{nk} \\
 & \quad \quad \quad P \in [0, P^U]^{nk} \\
 & \quad \quad \quad \chi \in [0, P^U]^{nk} \\
 & \forall i \leq n, j \leq k \quad \xi_{ij} \in [\min(y^L, 0), \max(y^U, 0)]
 \end{aligned}$$

$y_j, \xi_{ij}, y^L, y^U$  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 do better than *bound*  
all BB algorithms provide it

# BONMIN's first solution

Alstom  
Elektrobit  
Ford  
Llamasoft  
Amazon 2  
CSX  
MITRE  
Clarity  
fragment 2

Vinci  
AXA  
Deezer  
Accenture  
Expedia  
Google  
Aledade  
Marriott  
Amazon 1 & 3  
WestRock  
fragment 1

# Couple of things left to try

- ▶ **Approximate  $\ell_2$  by  $\ell_1$  norm**  
 *$\ell_1$  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*

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

- ▶ Replace each  $|\cdot|$  term by new vars.  $Q_{ija} \in [0, P^U]$

*Adjust  $P^U$  in terms of  $\|\cdot\|_1$*

- ▶ Adjoin constraints

$$\forall i \leq n, j \leq k \quad \sum_{a \leq d} Q_{ija} \leq P_{ij}$$

$$\forall i \leq n, j \leq k, a \leq d \quad -Q_{ija} \leq p_{ia} - y_{ja} \leq Q_{ija}$$

- ▶ Obtain a MILP

*Most advanced MILP solver: CPLEX*

# CPLEX's first solution

*objective 112.24, bound 39.92, in 44.74s*

AXA

Deezer

Ford

Marriott

Amazon 1-3

Llamasoft

CSX

WestRok

MITRE

Clarity

fragments 1-2

Vinci

Accenture

Alstom

Expedia

Elektrobit

Google

Aledade

*Interrupted after 28ls with bound 59.68*

## Subsection 5

### Random projections again

# Works on the MSSC MP formulation too!

$$\left. \begin{array}{l} \min_{x,y,s} \sum_{i \leq n} \sum_{j \leq d} \|T p_i - T y_j\|_2^2 x_{ij} \\ \forall j \leq d \quad \frac{1}{s_j} \sum_{i \leq n} T p_i x_{ij} = T y_j \\ \forall i \leq n \quad \sum_{j \leq d} x_{ij} = 1 \\ \forall j \leq d \quad \sum_{i \leq n} x_{ij} = s_j \\ \forall j \leq d \quad y_j \in \mathbb{R}^m \\ \quad \quad \quad x \in \{0, 1\}^{nd} \\ \quad \quad \quad s \in \mathbb{N}^d \end{array} \right\}$$

where  $T$  is a  $k \times m$  random projector  
replace  $Ty$  by  $y'$

# Works on the MSSC MP formulation too!

$$\left. \begin{array}{l} \min_{x, y', s} \sum_{i \leq n} \sum_{j \leq d} \|T p_i - y'_j\|_2^2 x_{ij} \\ \forall j \leq d \quad \frac{1}{s_j} \sum_{i \leq n} T p_i x_{ij} = y'_j \\ \forall i \leq n \quad \sum_{j \leq d} x_{ij} = 1 \\ \forall j \leq d \quad \sum_{i \leq n} x_{ij} = s_j \\ \forall j \leq d \quad y'_j \in \mathbb{R}^k \\ \quad \quad \quad x \in \{0, 1\}^{nd} \\ \quad \quad \quad s \in \mathbb{N}^d \end{array} \right\} \text{(MSSC')}$$

- ▶ where  $k = O(\frac{1}{\epsilon^2} \ln n)$
- ▶ **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

Ford

Amazon 1-3

CSX

MITRE

fragment 1

Vinci

AXA

Accenture

Alstom

Expedia

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  
Deezer  
Expedia  
Google  
Aledade  
Ford  
Amazon 1-3  
CSX  
Clarity

AXA  
Accenture  
Alstom  
Elektrobit  
Marriott  
Llamasoft  
WestRock  
MITRE  
fragment 1-2

# Many clusterings?

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

**THE END**