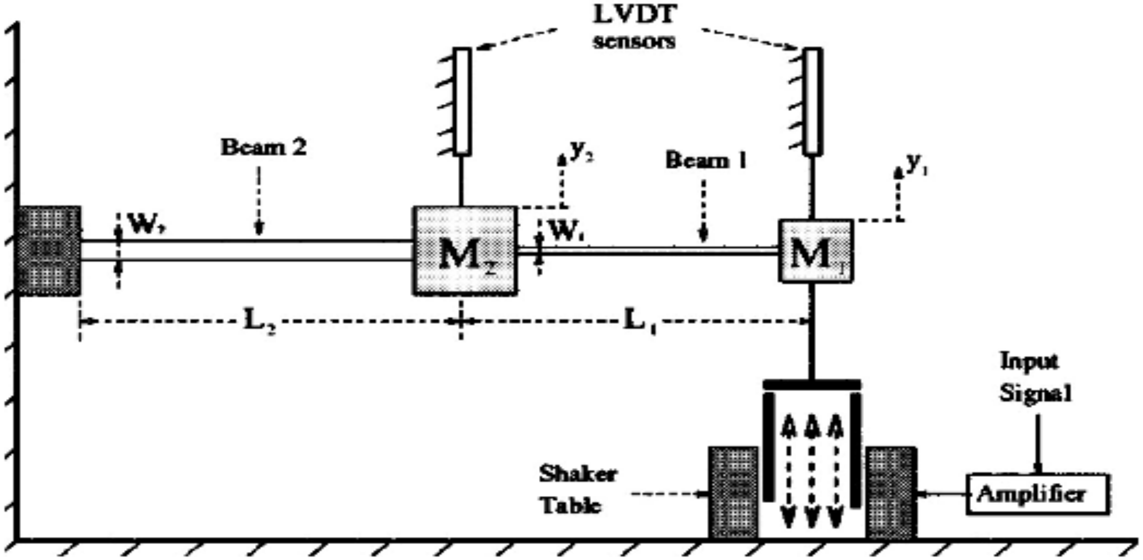
Northeastern University

The Interplay between Sparsity and Big Data in Systems Theory

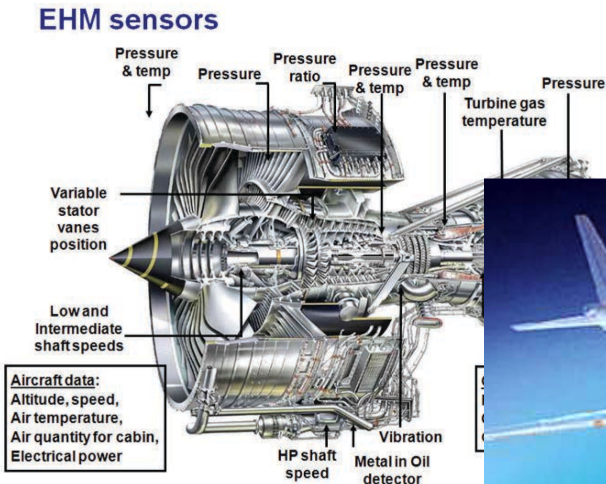
M. Sznaier
Robust Systems Lab
ECE, Northeastern University

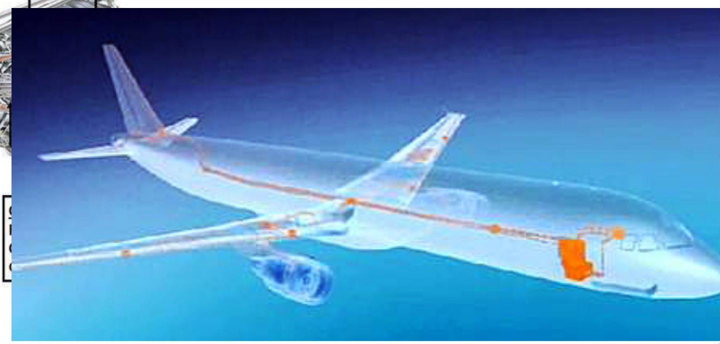
Motivation 1: SysId

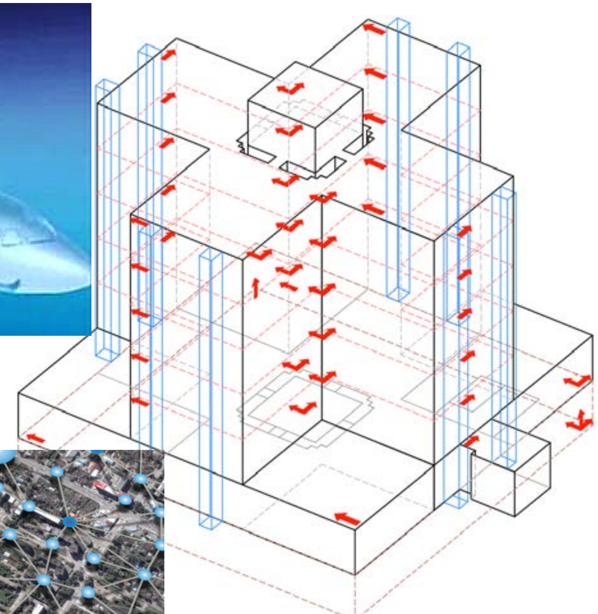


Goal: Find a low order, stable model

Motivation 2: distributed sensing & control

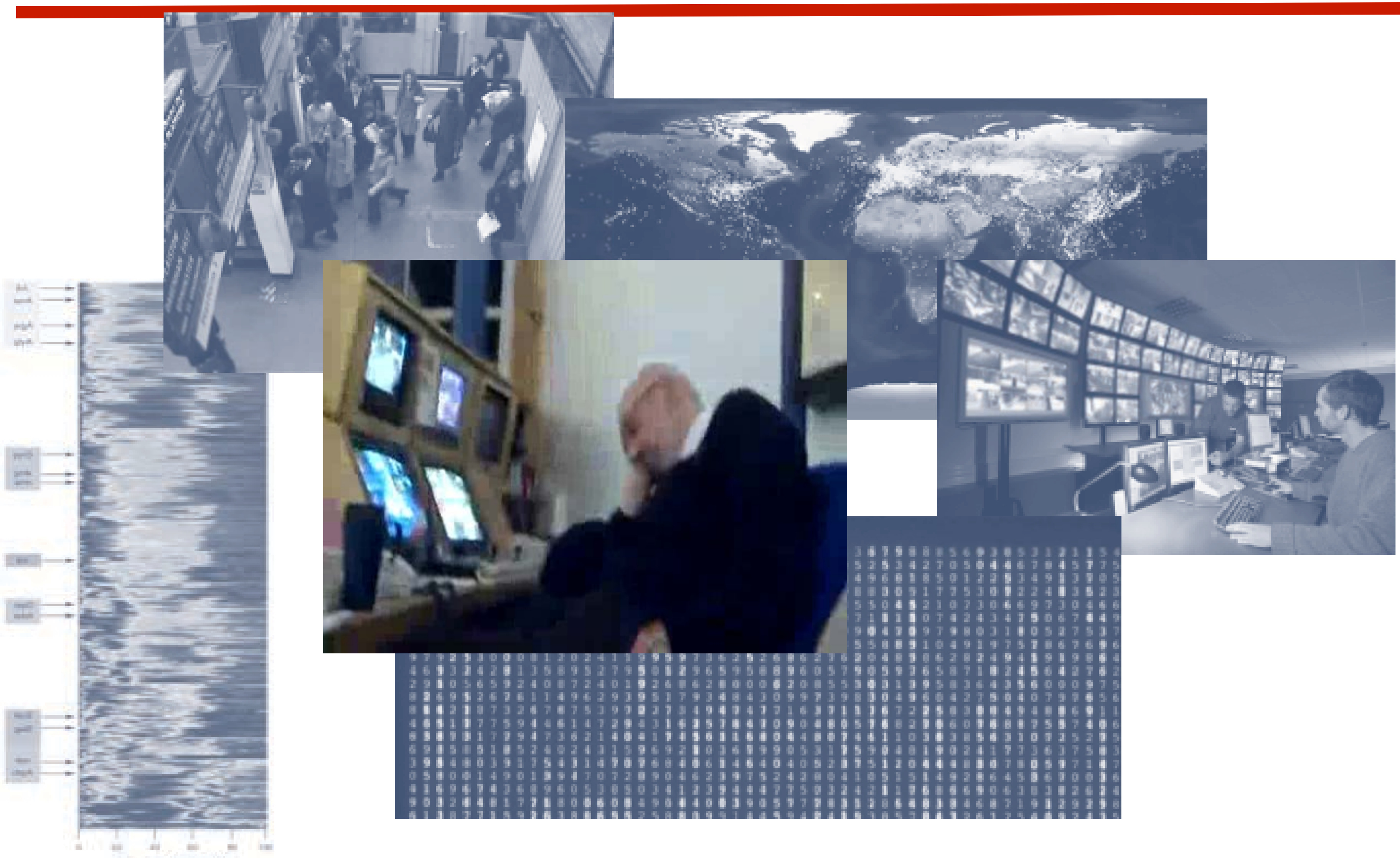






Goal: impose a sparse structure

Motivation 3: decision making



How do we make (provably) correct decisions in a "data deluged" environments? ***(a hidden hybrid SysId problem)***

Hard or Easy?

- **Claim 1: These problems are (NP!) hard**

Hard or Easy?

- **Claim 1: These problems are (NP!) hard**
- **Claim 2: These problems can be solved in polynomial time**

Hard or Easy?

- **Claim 1: These problems are (NP!) hard**
- **Claim 2: These problems can be solved in polynomial time**

Both can't be right, can they?

Hard or Easy?

- **Claim 1: These problems are generically NP-hard**
- **Claim 2: Many of these problems can be solved in polynomial time**

Hard or Easy?

- **Q: What makes a problem easy?**
- **A: Convexity?**

Hard or Easy?

- Q: What makes a problem easy?
- A: Convexity? Not Necessarily!

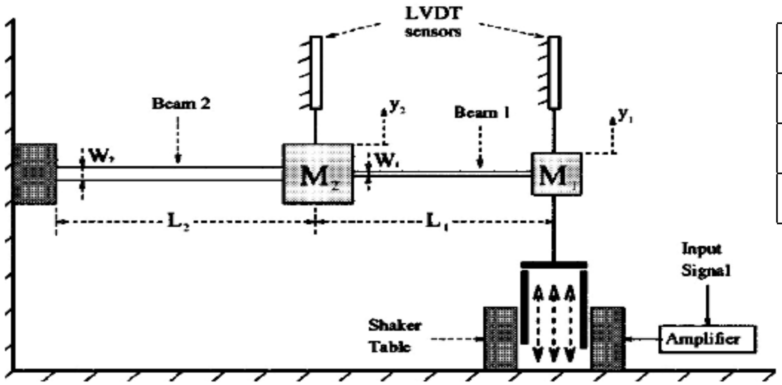
Optimization over co-Positive matrices is NP-hard

Hard or Easy?

- **Q: What makes a problem easy?**
- **A: Convexity + Self-Concordance?**

Hard or Easy?

- **Q: What makes a problem easy?**
- **A: Convexity + Self-Concordance? Not Necessarily!**



Horizon	ADMM (secs)	SDP solver(secs)
280	1071.8	4177.0
350	1828.0	12686.9
420	2657.7	out of memory

In (convex) SysId Big Data may be as low as 10^2

Hard or Easy?

- **Q: What makes a problem hard?**
- **A: Lack of Convexity?**

Hard or Easy?

- **Q: What makes a problem hard?**
- **A: Lack of Convexity? Not Necessarily!**

$$\min \sum \mathrm{c_i x_i x_{i+1}} \ \text{subject to} \ \mathrm{x_i} = \pm 1$$

Non-convex but solving for 100000 variables takes 50 secs on a Mac

Hard or Easy?

- Q: What makes a problem hard/easy?
- A: Structure
  - Self Similarity
  - Sparsity
- Both observed in many practical problems
  - Often they induces “good” convexity
  - Exploited in Machine Learning for “static” problems

Hard or Easy?

- **Challenge**
  - Separate easy/hard problems
  - Understand where does the complexity come from
  - Use this understanding to design "easy" problems

Main point of this talk: These issues are related to the sparsity structure of the problem

Intuition: look at QCQP

$$p^* = \min_x \mathbf{x}'\mathbf{Q_o}\mathbf{x} \; s.t. \; \mathbf{x}'\mathbf{Q_i}\mathbf{x} \leq 0 \; i = 1, ..n$$

$$\Downarrow$$

$$p^* = \min_x \mathbf{Trace}(\mathbf{Q_o}\mathbf{x}\mathbf{x}') \; s.t. \mathbf{Trace}(\mathbf{Q_i}\mathbf{x}\mathbf{x}') \leq 0 \; i = 1, ..n$$

$$\Downarrow$$

$$p_{SDP} = \min_x \mathbf{Trace}(\mathbf{Q_o}\mathbf{X}) \; s.t. \mathbf{Trace}(\mathbf{Q_i}\mathbf{X}) \leq 0, \; \mathbf{X} \succeq 0$$

Clearly $\mathrm{p_{SDP}} \leq \mathrm{p}^*$ **and** $\mathrm{p_{SDP}} = \mathrm{p}^*$ **if rank(X)=1**

Intuition: look at QCQP

$$p^* = \min_x \mathbf{x}'\mathbf{Q_o}\mathbf{x} \; s.t. \; \mathbf{x}'\mathbf{Q_i}\mathbf{x} \leq 0 \; i = 1, ..n$$

$$\Downarrow$$

$$p^* = \min_x \mathbf{Trace}(\mathbf{Q_o}\mathbf{x}\mathbf{x}') \; s.t. \mathbf{Trace}(\mathbf{Q_i}\mathbf{x}\mathbf{x}') \leq 0 \; i = 1, ..n$$

$$\Downarrow$$

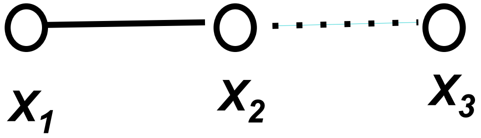
$$p_{SDP} = \min_x \mathbf{Trace}(\mathbf{Q_o}\mathbf{X}) \; s.t. \mathbf{Trace}(\mathbf{Q_i}\mathbf{X}) \leq 0, \; \mathbf{X} \succeq 0$$

Clearly $\mathrm{p_{SDP}} \leq \mathrm{p}^*$ **and** $\mathrm{p_{SDP}} = \mathrm{p}^*$ **if rank(X)=1**

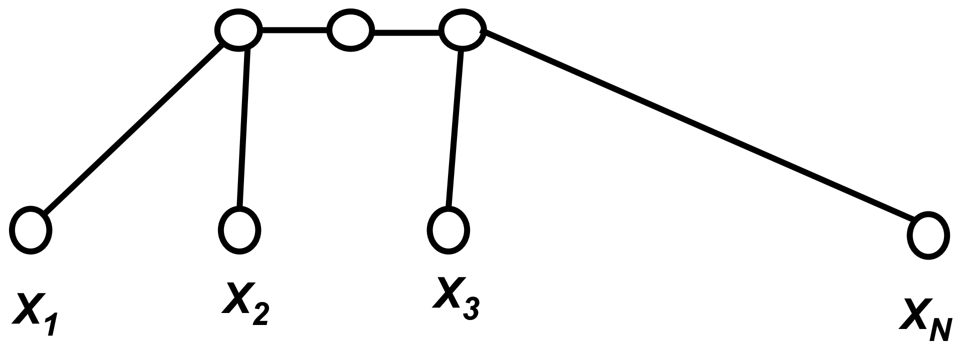
Q: Can we get this for (almost) free?

Exploiting sparsity in QCQP

- **Complexity related to the topology of a graph:**
  - **Each vertex corresponds to a variable**
  - **There is an edge (i,j) if there are terms involving $x_i x_j$**



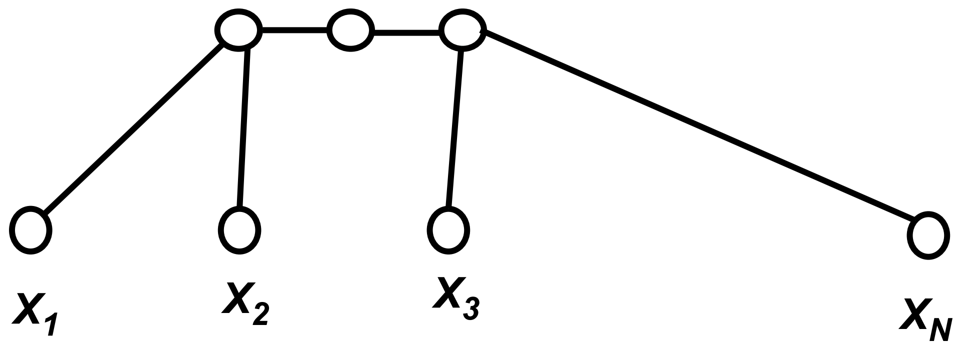
Exploiting sparsity in QCQP



- **If the graph is a tree, then the SDP relaxation is exact**

J. Lavaei, 2014

Exploiting sparsity in QCQP



- **If the graph is a tree, then the SOCP relaxation is exact**
- **Example:** $\min \sum \mathbf{c_i x_i x_{i+1}}$ **subject to** $\mathbf{x_i = \pm 1}$

Solving for 100,000 variables takes 50 secs on a Mac
Structure and Sparsity Matter

Sparse polynomial optimization

- **Many problems have a sparse structure (running intersection)**

$$\min_x p_1(x) + p_2(x) + \dots p_m(x) \;\; s.t.$$

$$f_1(x^\alpha) \leq 0$$

$$f_2(x^\alpha) \leq 0$$

$$\vdots$$

$$f_m(x^\alpha) \leq 0$$

where each $p_i(.)$, $f_i(.)$ depends only on a subset of variables such that

P_1

f_1

$x_1, x_2, .. x_k .. x_d, x_{d+1}, \dots x_{d+k}, \dots\dots\dots\dots x_{n-d+1}, x_n$

Sparse polynomial optimization

- **Many problems have a sparse structure (running intersection)**

$$\min_x p_1(x) + p_2(x) + \dots p_m(x) \;\; s.t.$$
$$f_1(x^\alpha) \leq 0$$
$$f_2(x^\alpha) \leq 0$$
$$\vdots$$
$$f_m(x^\alpha) \leq 0$$

where each $p_i(.)$, $f_i(.)$ depends only on a subset of variables such that

P_1

f_1

$x_1, x_2, .. x_k .. x_d, x_{d+1}, ... x_{d+k}, ... x_{n-d+1}, x_n$

Sparse polynomial optimization

- **Many problems have a sparse structure (running intersection)**

$$\min_x p_1(x) + p_2(x) + \ldots p_m(x) \;\; s.t.$$

$$f_1(x^\alpha) \leq 0$$

$$f_2(x^\alpha) \leq 0$$

$$\vdots$$

$$f_m(x^\alpha) \leq 0$$

where each $p_i(.)$, $f_i(.)$ depends only on a subset of variables such that

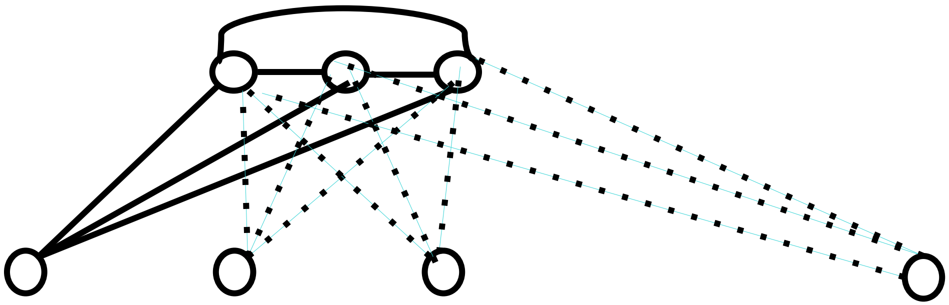
P_1 $\qquad$ P_2

f_1 $\qquad$ f_2

$x_1, x_2, .. x_k .. x_d, x_{d+1}, \ldots x_{d+k} \ldots\ldots\ldots\ldots x_{n-d+1}, x_n$

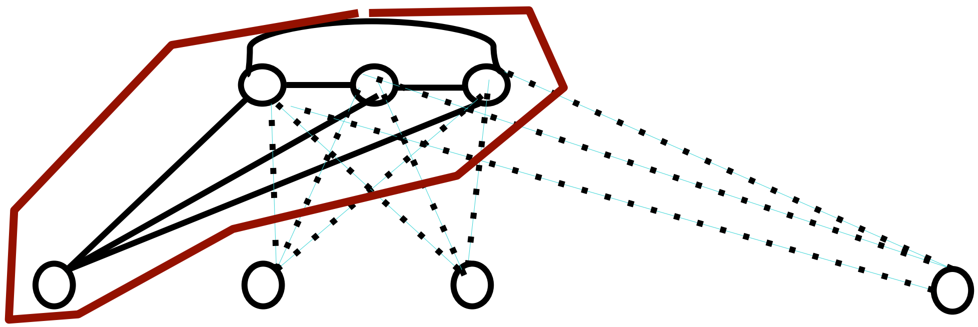
Sparse polynomial optimization

- **Running intersection is related to cliques in the (chordal completion of the) csp graph**



Sparse polynomial optimization

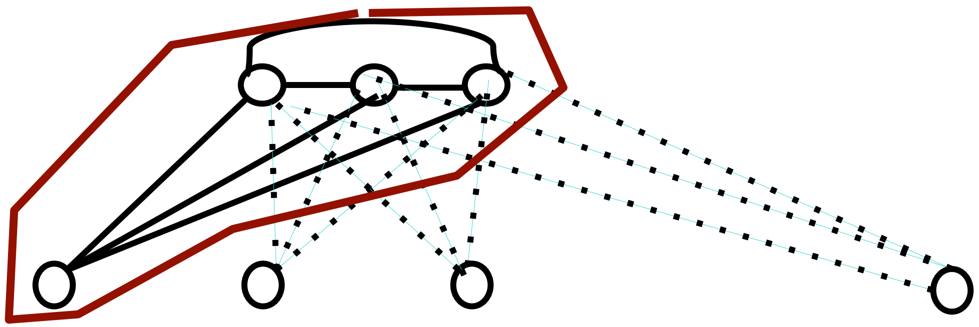
- **Running intersection is related to cliques in the (chordal completion of the) csp graph**



Size of the running intersection is given by the tree width

Sparse polynomial optimization

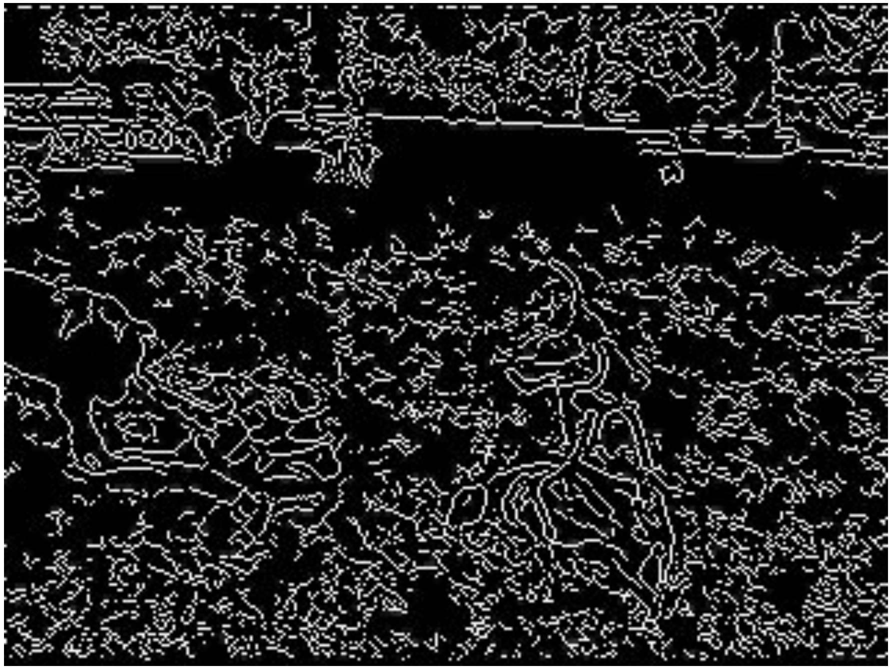
- **Running intersection is related to cliques in the (chordal completion of the) csp graph**



Complexity dominated by the size of the clique, not the size of the problem

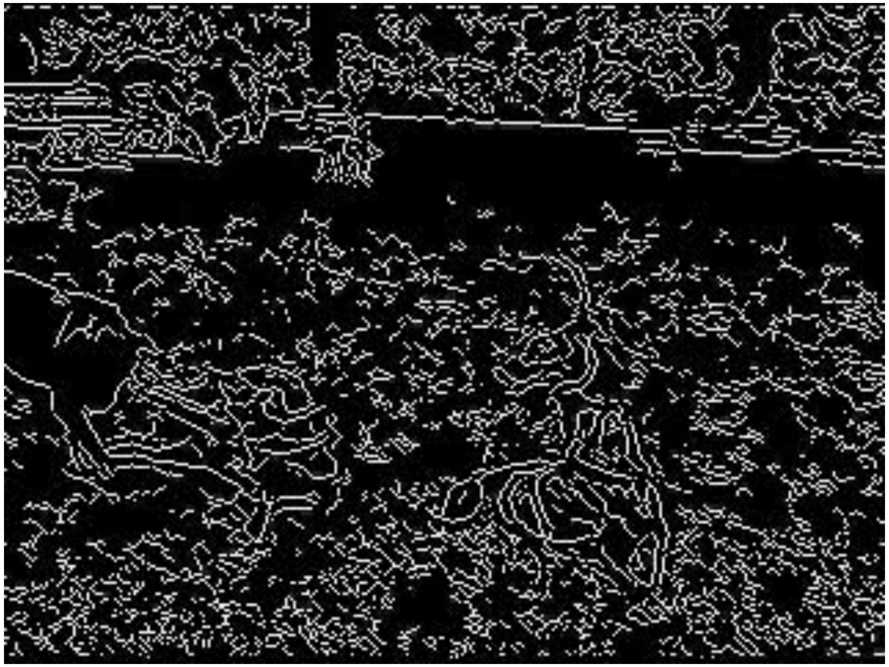
Connecting Information, Sparsity & Dynamics

Where should we pay attention?:



Features (edges, regions, etc.) are important.

Where should we pay attention?:



***Dynamics* are important too!.**

Sparse signal recovery:

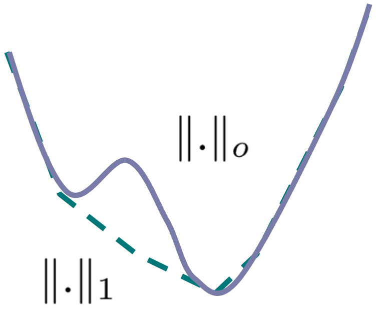
- **Strong prior:**
  - Signal has a sparse representation

$$f = \sum c_i \psi_i$$

only a few $c_i \neq 0$

- **Signal Recovery:**
  - "sparsify" the coefficients

$$\begin{aligned} &min\|[c_1, \ldots, c_n]\|_o \\ &\text{subject to} : f(x_i) = y_i \end{aligned}$$



Sparse signal recovery:

- **Strong prior:**
  - Signal has a sparse representation

$$f = \sum c_i \psi_i$$

only a few $c_i \neq 0$

- **Signal Recovery:**
  - "sparsify" the coefficients

$$min\|[c_1, \ldots, c_n]\|_o$$
$$\text{subject to}: f(x_i) = y_i$$

:

Sparse information extraction

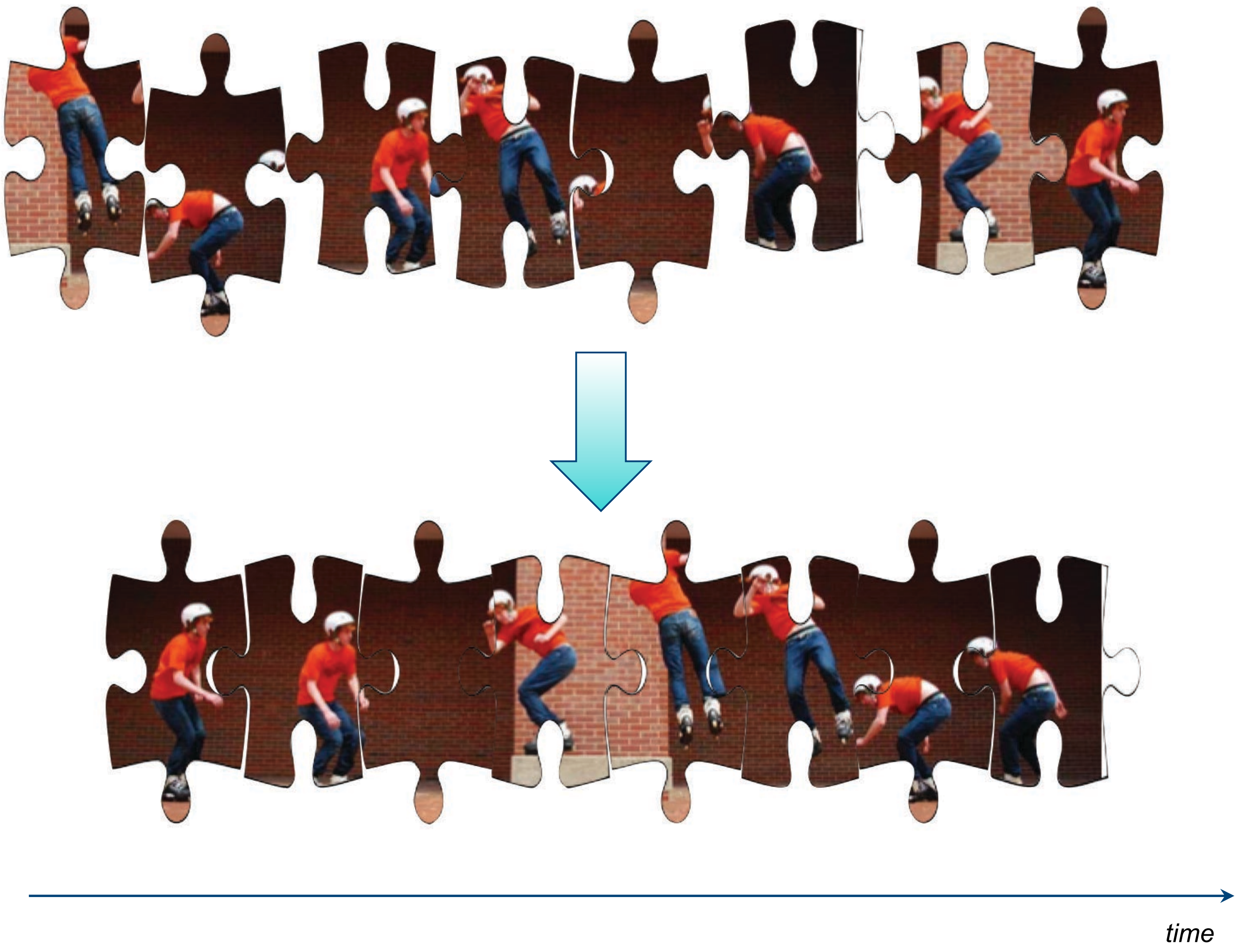
- **Strong prior:**
  - Actionable information is generated by low complexity dynamical systems.

- **Information extraction:**
  - "sparsify" the dynamics

$$\min_{\mathbf{y}} \{\mathbf{rank}[\mathbf{M}(\mathbf{y})] + \lambda \|\mathbf{E}(\mathbf{y})\|_o\}$$

  - Where M(.), E(.) are affine in y

Example: Solving "Temporal Puzzles"



Example: Solving "Temporal Puzzles"

y

$$\min_{\mathbf{c}} \|\mathbf{c}\|_1 \text{ subject to:}$$

$$\mathbf{v} = \mathbf{D}\mathbf{c}$$

$$\|\mathbf{P}\mathbf{y} - \mathbf{v}\| \leq \epsilon$$

$$\mathbf{P} \in \mathcal{P}$$

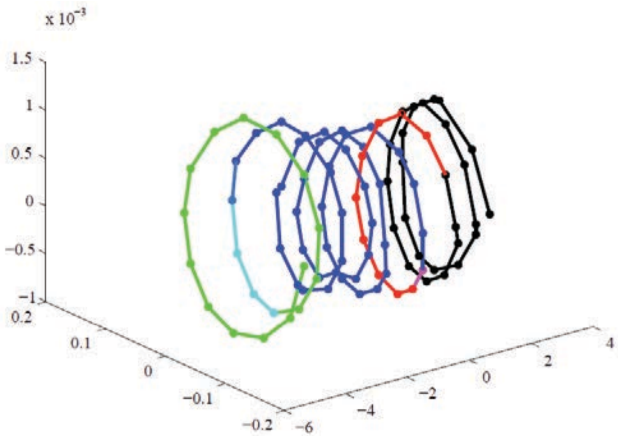
v

D *is a suitably chosen dynamic dictionary*

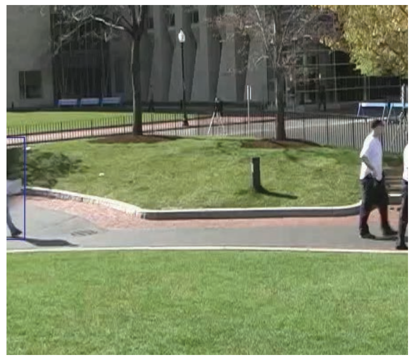
Example: Solving "Temporal Puzzles"

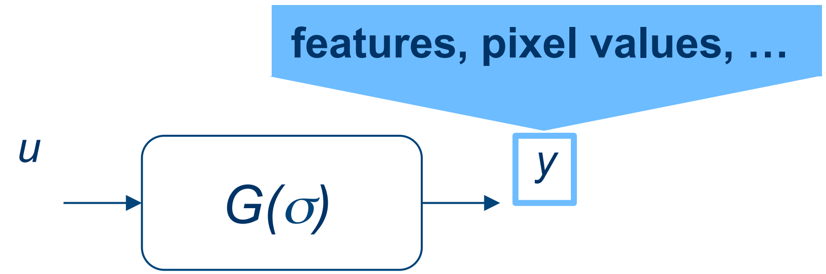
Dynamic sparsification

Information Extraction as an ID problem

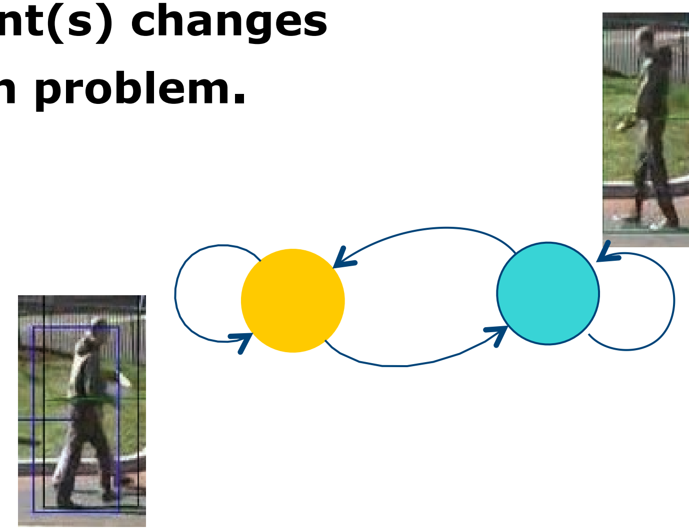


Information extraction as an Id problem:



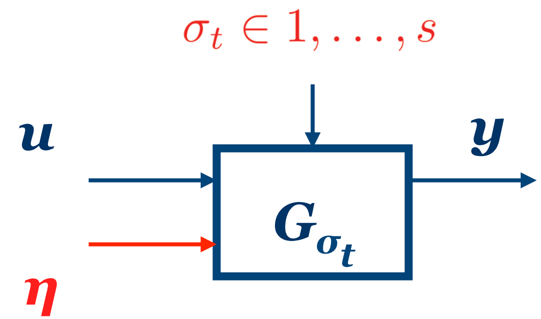


- Model data streams as outputs of switched systems
- "Interesting" events ⇔ Model invariant(s) changes
- An identification/model (in)validation problem.



SARX Id problem:

- **Given:**
  - **Bounds on noise ($||\eta||_\infty \leq \varepsilon$), sub-system order ($n_o$)**
  - **Input/output data (u,y)**
  - **Number of sub-models**

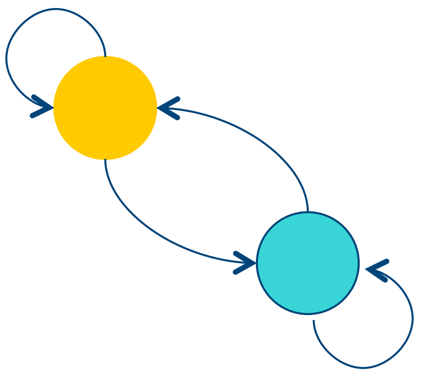


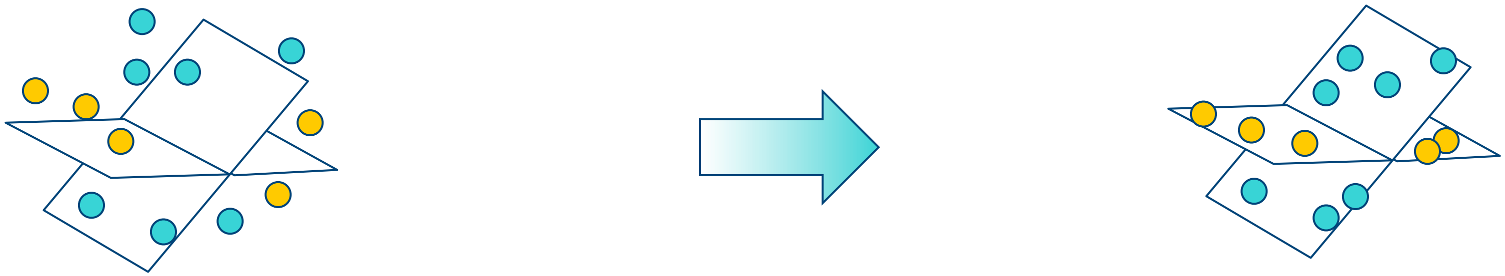
- **Find:**
  - **A piecewise affine model such that:**

$$y_t = \sum_{i=1}^{n_a} a_i(\sigma_t) y_{t-i} + \sum_{i=1}^{n_c} c_i(\sigma_t) u_{t-i} + f(\sigma_t) + \eta_t$$

$$\Updownarrow$$

$$0 = \mathrm{b}(\sigma_t)^T \mathbf{r}_t + \eta_t$$

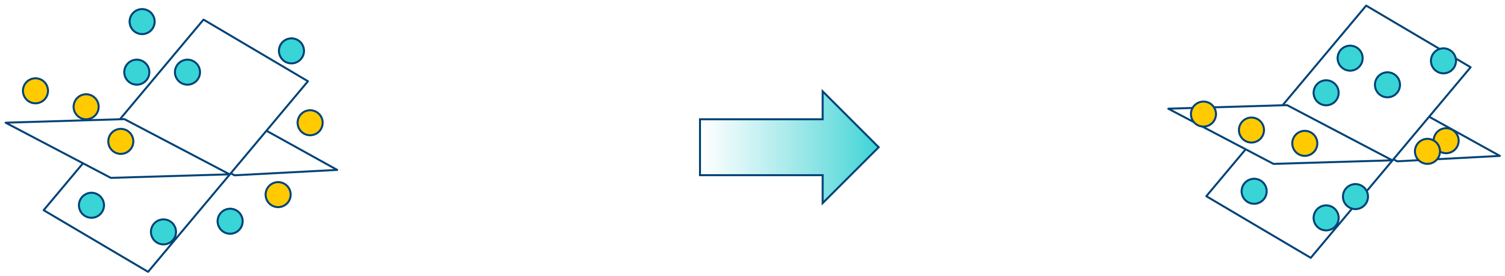




- **Given N points in R^n, fit them to hyperplanes**
- **"Chicken and egg" problem**
  - **Do not known the point "labels"**
  - **Do not know the hyperplanes.**

NP Hard !

Reformulation:



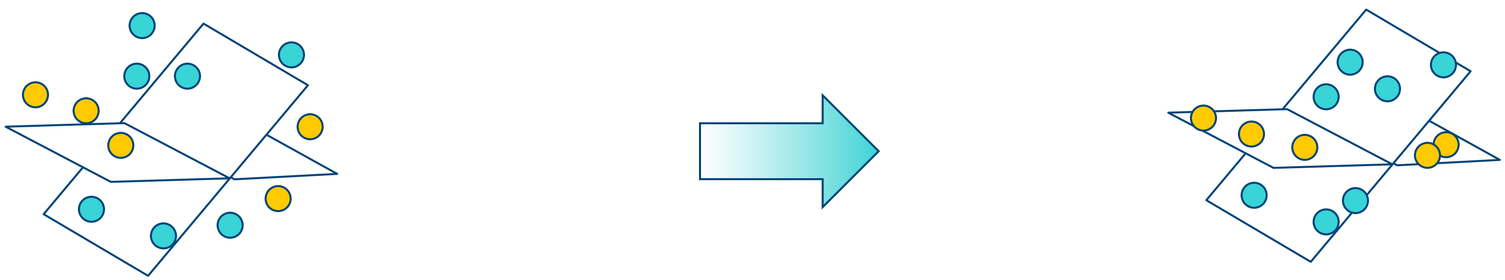
$$\mathbf{y}_t + \boldsymbol{\eta}_t - \sum_{i=1}^{n_a} \mathbf{A}_i(\sigma_1)\mathbf{y}_{t-i} - \sum_{i=1}^{n_c} \mathbf{C}_i(\sigma_1)\mathbf{u}_{t-i} = 0$$

or

$$\mathbf{y}_t + \boldsymbol{\eta}_t - \sum_{i=1}^{n_a} \mathbf{A}_i(\sigma_2)\mathbf{y}_{t-i} - \sum_{i=1}^{n_c} \mathbf{C}_i(\sigma_2)\mathbf{u}_{t-i} = 0$$

A hidden QCQP problem

QCQP reformulation:



$$\mathbf{s_{1,t}}(\mathbf{y}_t + \boldsymbol{\eta}_t - \sum_{i=1}^{n_a} \mathbf{A}_i(\sigma_1)\mathbf{y}_{t-i} - \sum_{i=1}^{n_c} \mathbf{C}_i(\sigma_1)\mathbf{u}_{t-i}) = 0$$

and

$$\mathbf{s_{2,t}}(\mathbf{y}_t + \boldsymbol{\eta}_t - \sum_{i=1}^{n_a} \mathbf{A}_i(\sigma_2)\mathbf{y}_{t-i} - \sum_{i=1}^{n_c} \mathbf{C}_i(\sigma_2)\mathbf{u}_{t-i}) = 0$$

Subject to: $s_{i,t} = s_{i,t}^2, \text{ and } \sum_i s_{i,t} = 1$

$s \in \{0, 1\}$

QCQP reformulation:

$$\begin{cases} \left|s_{i,j}\mathbf{r}_i^T\mathbf{x}_j\right| \leq \epsilon s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\ s_{i,j}^2 = s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\ \Sigma_{i=1}^{N_s} s_{i,j} = 1, \forall_{j=1}^{N_p} \\ \mathbf{r}_i^T\mathbf{r}_i = 1, \forall_{i=1}^{N_s} \\ \mathbf{r}_1(1) \geq \mathbf{r}_2(1) \geq \cdots \geq \mathbf{r}_{N_s}(1) \geq 0 \end{cases}$$

QCQP reformulation:

$$
\begin{cases}
|s_{i,j}\mathbf{r}_i^T\mathbf{x}_j| \leq \epsilon s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\
s_{i,j}^2 = s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\
\Sigma_{i=1}^{N_s} s_{i,j} = 1, \forall_{j=1}^{N_p} \\
\mathbf{r}_i^T\mathbf{r}_i = 1, \forall_{i=1}^{N_s} \\
\mathbf{r}_1(1) \geq \mathbf{r}_2(1) \geq \cdots \geq \mathbf{r}_{N_s}(1) \geq 0
\end{cases}
$$

$\mathbf{x}_j$ is an inlier in $\mathcal{S}_i$ if $s_{ij} = 1$

QCQP reformulation:

$$
\begin{cases}
|s_{i,j}\mathbf{r}_i^T\mathbf{x}_j| \le \epsilon s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\
s_{i,j}^2 = s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\
\Sigma_{i=1}^{N_s} s_{i,j} = 1, \forall_{j=1}^{N_p} \\
\mathbf{r}_i^T\mathbf{r}_i = 1, \forall_{i=1}^{N_s} \\
\mathbf{r}_1(1) \ge \mathbf{r}_2(1) \ge \cdots \ge \mathbf{r}_{N_s}(1) \ge 0
\end{cases}
$$

$\mathbf{x}_j$ is an inlier in $\mathcal{S}_i$ if $s_{ij} = 1$

$s_{ij} \in \{0, 1\}$

QCQP reformulation:

$$\begin{cases} |s_{i,j}\mathbf{r}_i^T\mathbf{x}_j| \leq \epsilon s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\ s_{i,j}^2 = s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\ \Sigma_{i=1}^{N_s} s_{i,j} = 1, \forall_{j=1}^{N_p} \\ \mathbf{r}_i^T\mathbf{r}_i = 1, \forall_{i=1}^{N_s} \\ \mathbf{r}_1(1) \geq \mathbf{r}_2(1) \geq \cdots \geq \mathbf{r}_{N_s}(1) \geq 0 \end{cases}$$

$\mathbf{x}_j$ is an inlier in $\mathcal{S}_i$ if $s_{ij} = 1$

$s_{ij} \in \{0, 1\}$

each sample is assigned to one subspace

QCQP reformulation:

$$\begin{cases} |s_{i,j}\mathbf{r}_i^T\mathbf{x}_j| \le \epsilon s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\ s_{i,j}^2 = s_{i,j}, \forall_{i=1}^{N_s}\forall_{j=1}^{N_p} \\ \Sigma_{i=1}^{N_s} s_{i,j} = 1, \forall_{j=1}^{N_p} \\ \mathbf{r}_i^T\mathbf{r}_i = 1, \forall_{i=1}^{N_s} \\ \mathbf{r}_1(1) \ge \mathbf{r}_2(1) \ge \cdots \ge \mathbf{r}_{N_s}(1) \ge 0 \end{cases}$$

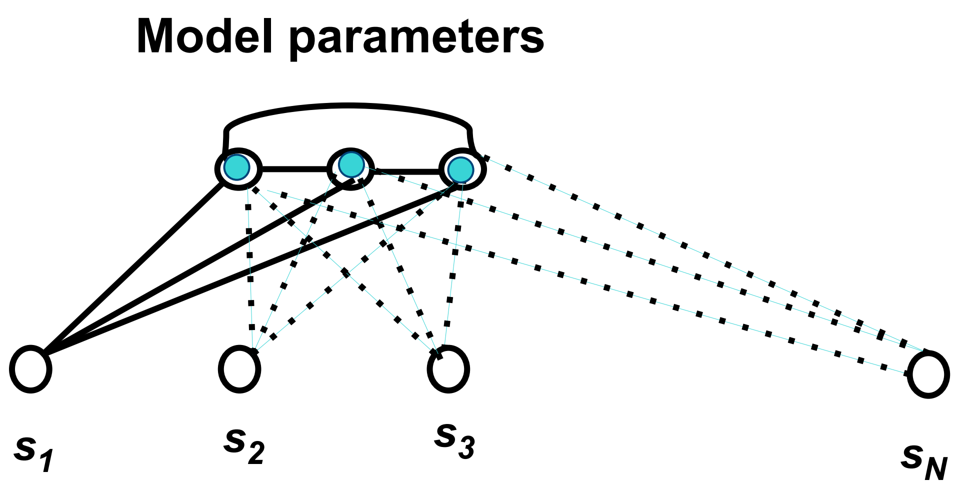
$\mathbf{x}_j$ is an inlier in $\mathcal{S}_i$ if $s_{ij} = 1$

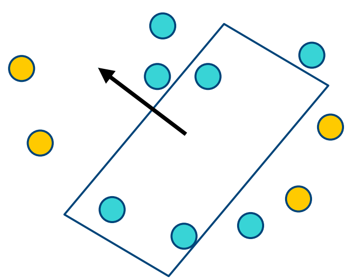
$s_{ij} \in \{0, 1\}$

each sample is assigned to one subspace

Solvable using SoS / Moments techniques

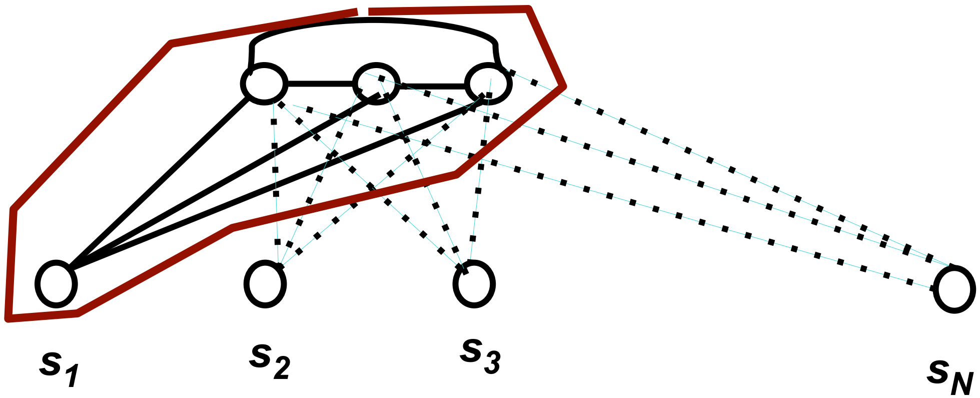
Hidden Sparse Structure:

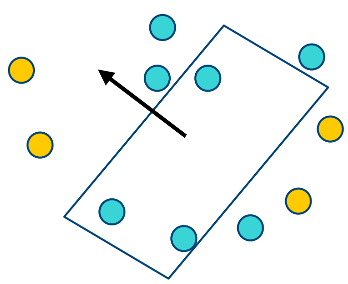




$$\mathbf{s_1}_{,\mathbf{t}}\left(\mathbf{y}_t + \boldsymbol{\eta}_t - \sum_{i=1}^{n_a} \mathbf{A}_i(\sigma_1)\mathbf{y}_{t-i} - \sum_{i=1}^{n_c} \mathbf{C}_i(\sigma_1)\mathbf{u}_{t-i}\right) = 0$$

Hidden Sparse Structure:





Complexity determined by the order of the model.

Linear in the number of data points

Exploiting the Sparse Structure:

Original problem:
Scales as $O((N_pN_s)^6)$

$$\begin{cases} P_0 : \begin{cases} \mathbf{r}_i^T\mathbf{r}_i = 1, \forall_{i=1}^{N_s} \\ \mathbf{r}_1(1) \geq \mathbf{r}_2(1) \geq \cdots \geq \mathbf{r}_{N_s}(1) \geq 0 \end{cases} \\ \forall_{j=1}^{N_p} : P_j : \begin{cases} \left|s_{i,j}\mathbf{r}_i^T\mathbf{x}_j\right| \leq \epsilon s_{i,j}, \forall_{i=1}^{N_s} \\ s_{i,j}^2 = s_{i,j}, \forall_{i=1}^{N_s} \\ \Sigma_{i=1}^{N_s} s_{i,j} = 1 \end{cases} \end{cases}$$

Reduced problem:
Scales as $O(N_p(N_s)^6)$

$$\begin{cases} \mathrm{Tr}(\bar{\mathbf{Q}}_{k,0}\mathbf{M}_0) \leq 0, \forall_{k=1}^{K_0} \\ \mathbf{M}_0 \succeq \mathbf{0}, \mathbf{M}_0(1,1) = 1 \\ \mathrm{rank}(\mathbf{M}_0) = 1 \\ \forall_{j=1}^{N_p} : \begin{cases} \mathrm{Tr}(\bar{\mathbf{Q}}_{k,j}\mathbf{M}_j) \leq 0, \forall_{k=1}^{K_j} \\ \mathbf{M}_j \succeq \mathbf{0}, \mathbf{M}_j(1,1) = 1 \\ \mathbf{M}_j(1:nN_s+1, 1:nN_s+1) = \mathbf{M}_0 \end{cases} \end{cases}$$

Linear in the number of data points

Exploiting the Sparse Structure:

Original problem:
Scales as $O((N_pN_s)^6)$

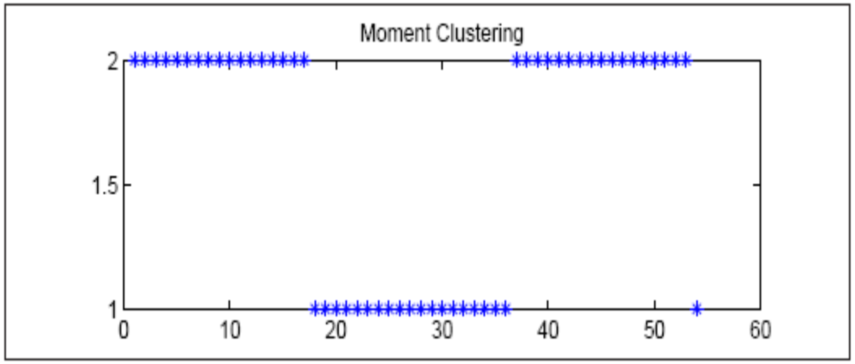
$$\begin{cases} P_0 : \begin{cases} \mathbf{r}_i^T\mathbf{r}_i = 1, \forall_{i=1}^{N_s} \\ \mathbf{r}_1(1) \geq \mathbf{r}_2(1) \geq \cdots \geq \mathbf{r}_{N_s}(1) \geq 0 \end{cases} \\ \forall_{j=1}^{N_p} : P_j : \begin{cases} \left|s_{i,j}\mathbf{r}_i^T\mathbf{x}_j\right| \leq \epsilon s_{i,j}, \forall_{i=1}^{N_s} \\ s_{i,j}^2 = s_{i,j}, \forall_{i=1}^{N_s} \\ \Sigma_{i=1}^{N_s} s_{i,j} = 1 \end{cases} \end{cases}$$

Reduced problem:
Scales as $O(N_p(N_s)^6)$

$$\begin{cases} \mathrm{Tr}(\bar{\mathbf{Q}}_{k,0}\mathbf{M}_0) \leq 0, \forall_{k=1}^{K_0} \\ \mathbf{M}_0 \succeq \mathbf{0}, \mathbf{M}_0(1,1) = 1 \\ \boxed{\mathrm{rank}(\mathbf{M}_0) = 1} \\ \forall_{j=1}^{N_p} : \begin{cases} \mathrm{Tr}(\bar{\mathbf{Q}}_{k,j}\mathbf{M}_j) \leq 0, \forall_{k=1}^{K_j} \\ \mathbf{M}_j \succeq \mathbf{0}, \mathbf{M}_j(1,1) = 1 \\ \mathbf{M}_j(1:nN_s+1, 1:nN_s+1) = \mathbf{M}_0 \end{cases} \end{cases}$$

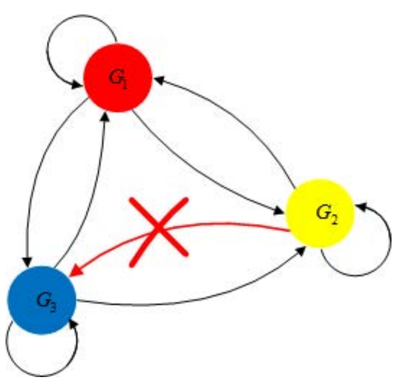
Caveat: still need to deal with a rank constraint

Example: Human Activity Analysis

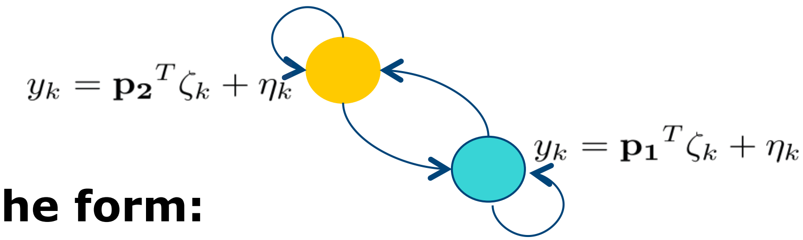


WALK ***BEND*** ***WALK***

(In)Validating SARX Models



Model (In)validation of SARX Systems



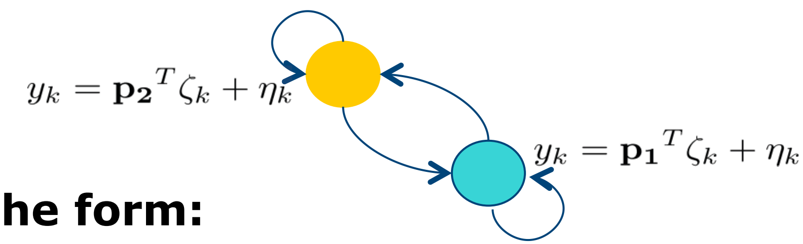
- **Given:**
  - **A nominal switched model of the form:**

$$
\begin{aligned}
\mathbf{y}_t &= \textstyle\sum_{k=1}^{n_a} \mathbf{A}_k(\sigma_t)\mathbf{y}_{t-k} + \sum_{k=1}^{n_c} \mathbf{C}_k(\sigma_t)\mathbf{u}_{t-k} + \mathbf{f}(\sigma_t) \\
\tilde{\mathbf{y}}_t &= \mathbf{y}_t + \boldsymbol{\eta}_t
\end{aligned}
$$

  - **A bound on the noise ($\|\eta\|_\infty \leq \varepsilon$)**
  - **Experimental Input/Output Data** $\{\mathbf{u}_t, \tilde{\mathbf{y}}_t\}_{t=t_0}^{T}$

- **Determine:**
  - **whether there exist noise and switching sequences consistent with a priori information and experimental data**

Model (In)validation of SARX Systems



- **Given:**
  - **A nominal switched model of the form:**

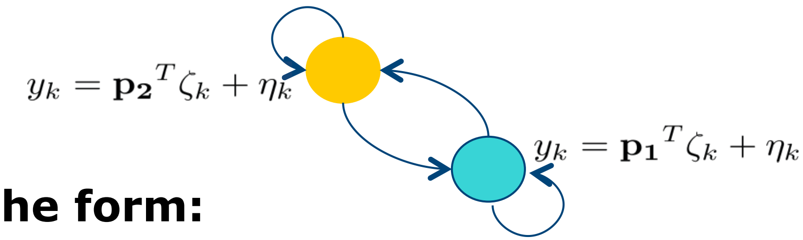
$$
\begin{aligned}
\mathbf{y}_t &= \textstyle\sum_{k=1}^{n_a} \mathbf{A}_k(\sigma_t)\mathbf{y}_{t-k} + \sum_{k=1}^{n_c} \mathbf{C}_k(\sigma_t)\mathbf{u}_{t-k} + \mathbf{f}(\sigma_t) \\
\tilde{\mathbf{y}}_t &= \mathbf{y}_t + \boldsymbol{\eta}_t
\end{aligned}
$$

  - **A bound on the noise ($\|\eta\|_\infty \leq \varepsilon$)**
  - **Experimental Input/Output Data** $\{\mathbf{u}_t, \tilde{\mathbf{y}}_t\}_{t=t_0}^T$

- **Determine:**
  - **whether there exist noise and switching sequences consistent with a priori information and experimenta data**

Reduces to SDP via Putinar's Positivstellensatz

Model (In)validation of SARX Systems



- **Given:**
  - **A nominal switched model of the form:**

$$\begin{aligned} \mathbf{y}_t &= \textstyle\sum_{k=1}^{n_a} \mathbf{A}_k(\sigma_t)\mathbf{y}_{t-k} + \sum_{k=1}^{n_c} \mathbf{C}_k(\sigma_t)\mathbf{u}_{t-k} + \mathbf{f}(\sigma_t) \\ \tilde{\mathbf{y}}_t &= \mathbf{y}_t + \boldsymbol{\eta}_t \end{aligned}$$

  - **A bound on the noise ($\|\eta\|_\infty \leq \varepsilon$)**
  - **Experimental Input/Output Data** $\{\mathbf{u}_t, \tilde{\mathbf{y}}_t\}_{t=t_0}^{T}$

- **Determine:**
  - **whether there exist noise and switching sequences consistent with a priori information and experimenta data**

Reduces to SDP via Putinar's Positivstellensatz

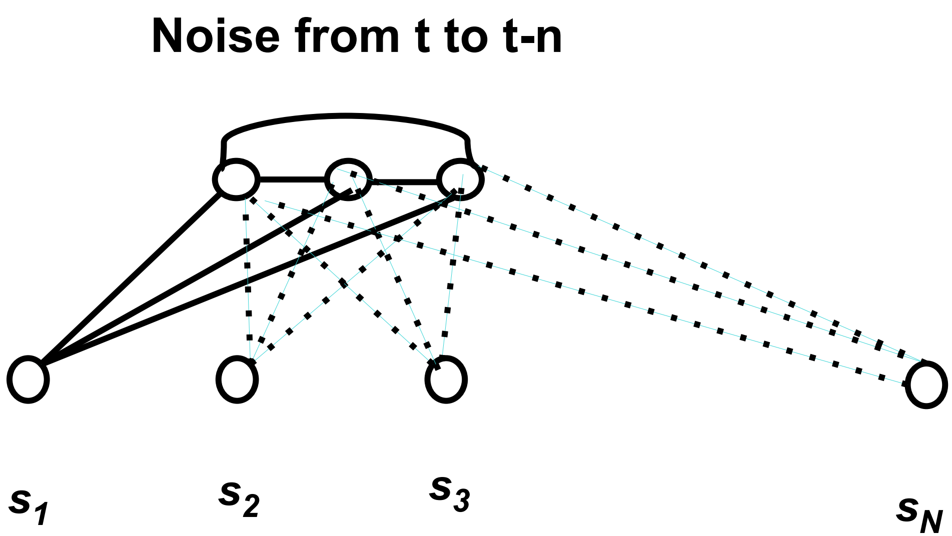
Guaranteed convergence for the n=T relaxation

(In)validation Certificates:

- **The model is invalid if and only if**

$$d^* \doteq \left\{ \begin{array}{l} \min_{\mathbf{s},\boldsymbol{\eta}} \sum_{t=1}^{T} \sum_{i=1}^{n_s} e_{i,t}^2 \\ \text{subject to:} \\ \mathbf{s_{i,t}}(\mathbf{g}_{i,t} + \mathbf{h}_{i,t}\boldsymbol{\eta}_{t-n_a:t}) = \mathbf{e}_{i,t} \\ \sum_i s_{i,t} = 1 \\ s_{i,t}^2 = 1 \\ \|\boldsymbol{\eta}\|_\infty \leq \epsilon \end{array} \right\} > 0$$

Model (In)validation of SARX Systems



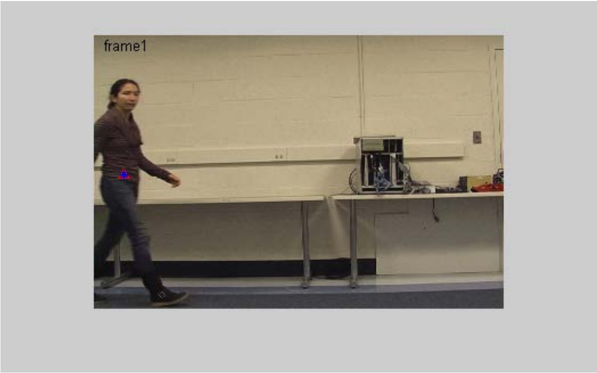
Hidden sparse structure similar to the Id case

Complexity dominated by the order of the model

Example: Activity Monitoring

- **A priori switched model: walking and waiting, 4% noise**
- **Test sequences of hybrid behavior:**

WALK, WAIT

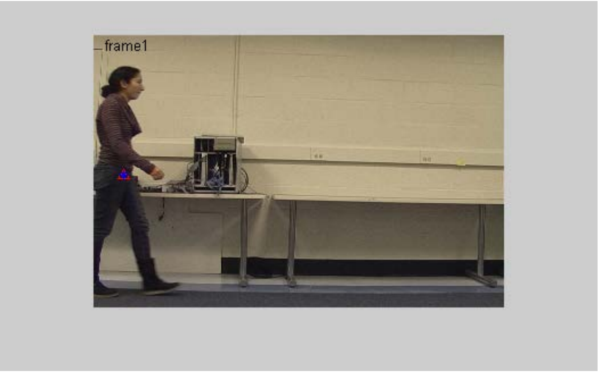


Not Invalidated

RUN

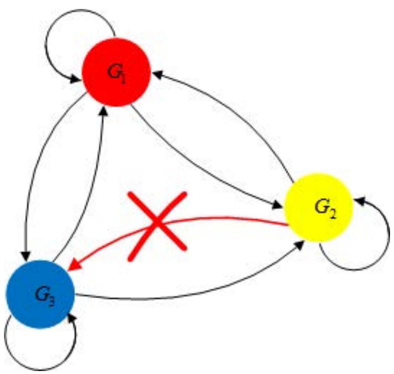
Invalidated

WALK, JUMP



Invalidated

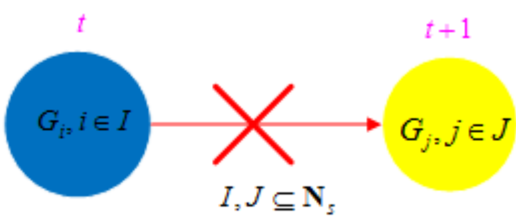
Adding topological constraints:



- **The model is invalid if and only if**

$$d^* \doteq \left\{ \begin{array}{l} \min_{\mathbf{s},\boldsymbol{\eta}} \sum_{t=1}^{T} \sum_{i=1}^{n_s} e_{i,t}^2 \\ \text{subject to:} \\ \mathbf{s_{i,t}}(\mathbf{g}_{i,t} + \mathbf{h}_{i,t}\boldsymbol{\eta}_{t-n_a:t}) = \mathbf{e}_{i,t} \\ \sum_i s_{i,t} = 1 \\ s_{i,t}^2 = 1 \\ \|\boldsymbol{\eta}\|_\infty \leq \epsilon \end{array} \right\} > 0$$

plus additional linear constraints:

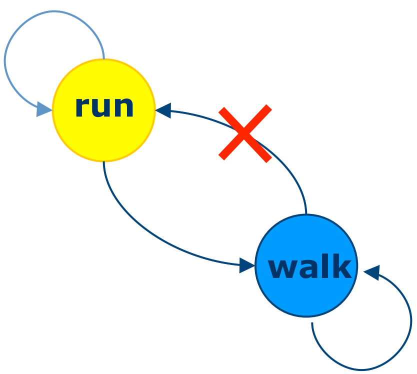


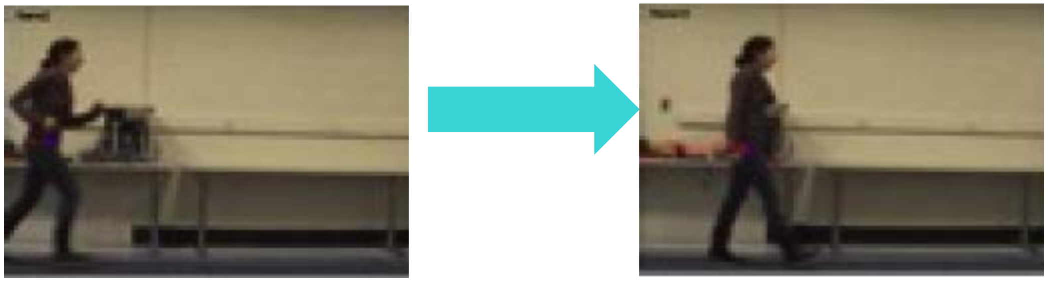
$$s_{i,t} + s_{j,t+1} \leq 1, \forall i \in I, \forall j \in J$$

These destroy sparsity patterns!

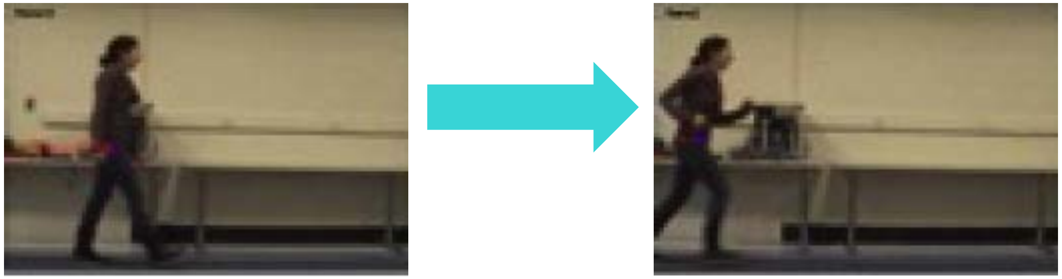
Example: Activity Monitoring

A Priori information





Not Invalidated (d=-3e-8)



Invalidated (d=0.175)

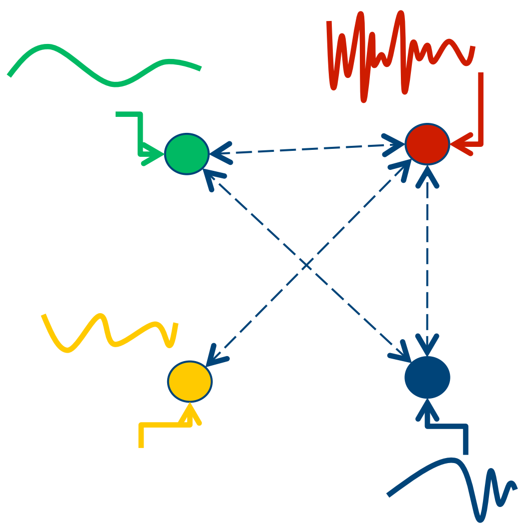
Identifying Sparse Dynamical Networks

Who is in the same team?
Who reacts to whom?

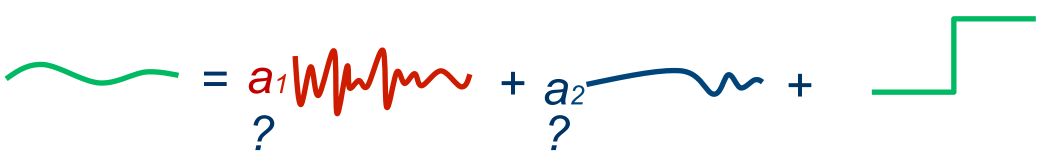
Formalization as a graph id problem:

Each time series becomes a **node** in a graph

Each edge is **a dynamical system**



$$x_i(t) = \sum_{j=1, j\neq i}^{P} \sum_{n=k}^{N} (a_{ji}(n) x_j(t-n)) + u_i(t) + \eta_i(t)$$



A Sparsification Problem:

- **Find block sparse solutions to:**

$$\mathbf{x} = [\mathbf{X}, \mathbf{I}]\,[\mathbf{a}^t \mathbf{u}^t]^t + \eta$$

- **Efficient solutions using atomic norm minimization**
- **Atoms are the time series at other nodes**
- **Projection free Frank-Wolfe algorithm**

$$\mathbf{x} = [\mathbf{X}, \mathbf{I}]\,[\mathbf{a}^t \mathbf{u}^t]^t + \eta \quad \Rightarrow \quad \begin{array}{ll} \min\limits_{\mathbf{z}} & \|\mathbf{z} - \mathbf{x}_j\|_2 \\ \text{s.t.} & \|\mathbf{z}\|_{s\mathcal{A}} \le \tau \end{array}$$

Algorithm

$$\min_{\mathbf{z}} \quad f(\mathbf{z})$$
$$\text{s.t.} \quad \|\mathbf{z}\|_{\mathcal{A}} \leq \tau$$

$$\min_{\mathbf{z}} \quad \|\mathbf{z} - \mathbf{x}_j\|_2$$
$$\text{s.t.} \quad \|\mathbf{z}\|_{s\mathcal{A}} \leq \tau$$

Frank-Wolfe Algorithm

1: Initialize:
$\mathbf{z}^{(0)} \leftarrow \tau \mathbf{a}_0$ for arbitrary $\mathbf{a}_0 \in \mathcal{A}$

2: for $k = 0, 1, 2, \cdots$ do

3: $\mathbf{a} \leftarrow \arg\min_{\mathbf{a}\in\mathcal{A}} \langle \partial f(\mathbf{z}^{(k)}), \mathbf{a} \rangle$

4: $\alpha_k \leftarrow \arg\min_{\alpha\in[0,1]} f(\mathbf{z}^{(k)} + \alpha[\tau\mathbf{a} - \mathbf{z}^{(k)}])$

5: $\mathbf{z}^{(k+1)} \leftarrow \mathbf{z}^{(k)} + \alpha_k[\tau\mathbf{a} - \mathbf{z}^{(k)}]$

6: end for

Converges as O(1/n)

Algorithm

$$\min_{\mathbf{z}} \quad f(\mathbf{z}) \qquad \text{s.t.} \quad \|\mathbf{z}\|_{\mathcal{A}} \leq \tau$$

$$\min_{\mathbf{z}} \quad \|\mathbf{z} - \mathbf{x}_j\|_2 \qquad \text{s.t.} \quad \|\mathbf{z}\|_{s\mathcal{A}} \leq \tau$$

Frank-Wolfe Algorithm

1: Initialize:

$\mathbf{z}^{(0)} \leftarrow \tau \mathbf{a}_0$ for arbitrary $\mathbf{a}_0 \in \mathcal{A}$

2: for $k = 0, 1, 2, \cdots$ do

3: $\mathbf{a} \leftarrow \arg\min_{\mathbf{a} \in \mathcal{A}} \langle \partial f(\mathbf{z}^{(k)}), \mathbf{a} \rangle$

4: $\alpha_k \leftarrow \arg\min_{\alpha \in [0,1]} f(\mathbf{z}^{(k)} + \alpha[\tau \mathbf{a} - \mathbf{z}^{(k)}])$

5: $\mathbf{z}^{(k+1)} \leftarrow \mathbf{z}^{(k)} + \alpha_k [\tau \mathbf{a} - \mathbf{z}^{(k)}]$

6: end for

$$\begin{array}{lcl} L & \leftarrow & \arg\max_l \{\|[\partial f(\mathbf{z}^{(k)})]^T \mathbf{A}_l\|_1\} \\ \mathbf{c} & \leftarrow & -\operatorname{sign}([\partial f(\mathbf{z}^{(k)})]^T \mathbf{A}_L) \\ \mathbf{a} & \leftarrow & \mathcal{A}_L \mathbf{c} \end{array}$$

Algorithm

$$\min_{\mathbf{z}} \quad f(\mathbf{z}) \qquad \text{s.t.} \quad \|\mathbf{z}\|_{\mathcal{A}} \le \tau$$

$$\min_{\mathbf{z}} \quad \|\mathbf{z} - \mathbf{x}_j\|_2 \qquad \text{s.t.} \quad \|\mathbf{z}\|_{s\mathcal{A}} \le \tau$$

Frank-Wolfe Algorithm

1: Initialize:

$\mathbf{z}^{(0)} \leftarrow \tau \mathbf{a}_0$ for arbitrary $\mathbf{a}_0 \in \mathcal{A}$

2: for $k = 0, 1, 2, \cdots$ do

3: $\mathbf{a} \leftarrow \arg\min_{\mathbf{a}\in\mathcal{A}} \langle \partial f(\mathbf{z}^{(k)}), \mathbf{a} \rangle$

4: $\alpha_k \leftarrow \arg\min_{\alpha\in[0,1]} f(\mathbf{z}^{(k)} + \alpha[\tau\mathbf{a} - \mathbf{z}^{(k)}])$

5: $\mathbf{z}^{(k+1)} \leftarrow \mathbf{z}^{(k)} + \alpha_k[\tau\mathbf{a} - \mathbf{z}^{(k)}]$

6: end for

$$\begin{aligned} L &\leftarrow \arg\max_l \{\|[\partial f(\mathbf{z}^{(k)})]^T \mathbf{A}_l\|_1\} \\ \mathbf{c} &\leftarrow -\operatorname{sign}([\partial f(\mathbf{z}^{(k)})]^T \mathbf{A}_L) \\ \mathbf{a} &\leftarrow \mathcal{A}_L \mathbf{c} \end{aligned}$$

$$\alpha_k \leftarrow \max\{\min\{\frac{[\tau\mathbf{a} - \mathbf{z}^{(k)}]^T[\mathbf{x}_j - \mathbf{z}^{(k)}]}{\|\tau\mathbf{a} - \mathbf{z}^{(k)}\|_2^2}, 1\}, 0\}$$

Algorithm

$$\min_{\mathbf{z}} \quad f(\mathbf{z}) \qquad \text{s.t.} \quad \|\mathbf{z}\|_{\mathcal{A}} \leq \tau$$

$$\min_{\mathbf{z}} \quad \|\mathbf{z} - \mathbf{x}_j\|_2 \qquad \text{s.t.} \quad \|\mathbf{z}\|_{s\mathcal{A}} \leq \tau$$

Frank–Wolfe Algorithm

1: Initialize:

$\mathbf{z}^{(0)} \leftarrow \tau \mathbf{a}_0$ for arbitrary $\mathbf{a}_0 \in \mathcal{A}$

2: for $k = 0, 1, 2, \cdots$ do

3: $\mathbf{a} \leftarrow \arg\min_{\mathbf{a} \in \mathcal{A}} \langle \partial f(\mathbf{z}^{(k)}), \mathbf{a} \rangle$

4: $\alpha_k \leftarrow \arg\min_{\alpha \in [0,1]} f(\mathbf{z}^{(k)} + \alpha[\tau \mathbf{a} - \mathbf{z}^{(k)}])$

5: $\mathbf{z}^{(k+1)} \leftarrow \mathbf{z}^{(k)} + \alpha_k [\tau \mathbf{a} - \mathbf{z}^{(k)}]$

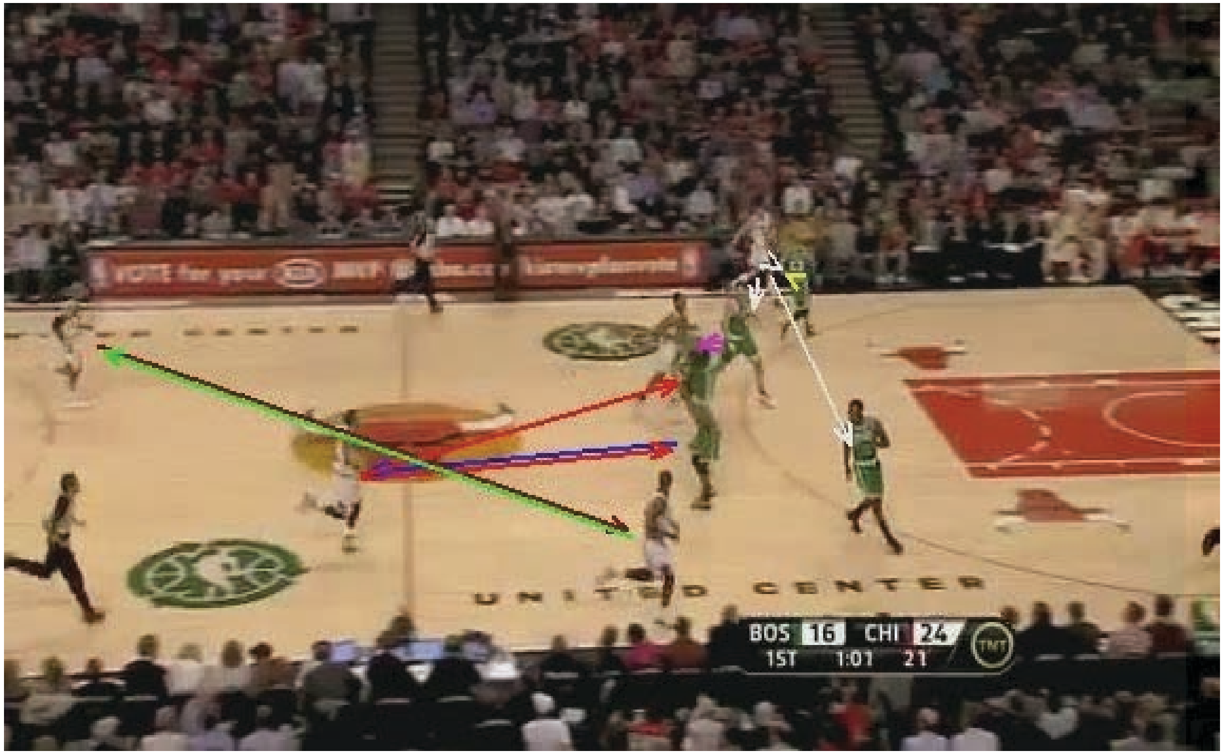
6: end for

$$\begin{array}{lcl} L & \leftarrow & \arg\max_l \{ \| [\partial f(\mathbf{z}^{(k)})]^T \mathbf{A}_l \|_1 \} \\ \mathbf{c} & \leftarrow & -\operatorname{sign}([\partial f(\mathbf{z}^{(k)})]^T \mathbf{A}_L) \\ \mathbf{a} & \leftarrow & \mathcal{A}_L \mathbf{c} \end{array}$$

$$\alpha_k \leftarrow \max\{\min\{\frac{[\tau \mathbf{a} - \mathbf{z}^{(k)}]^T [\mathbf{x}_j - \mathbf{z}^{(k)}]}{\|\tau \mathbf{a} - \mathbf{z}^{(k)}\|_2^2}, 1\}, 0\}$$

Closed form solutions to each step

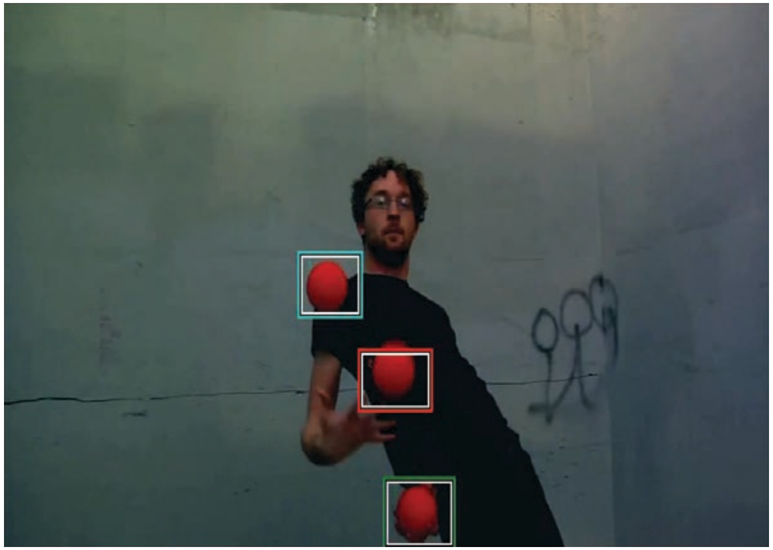
Example



Interactions between human agents

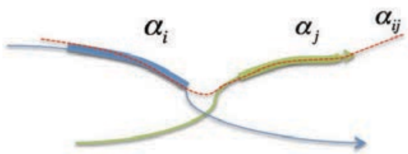
More examples:

Tracking by detection

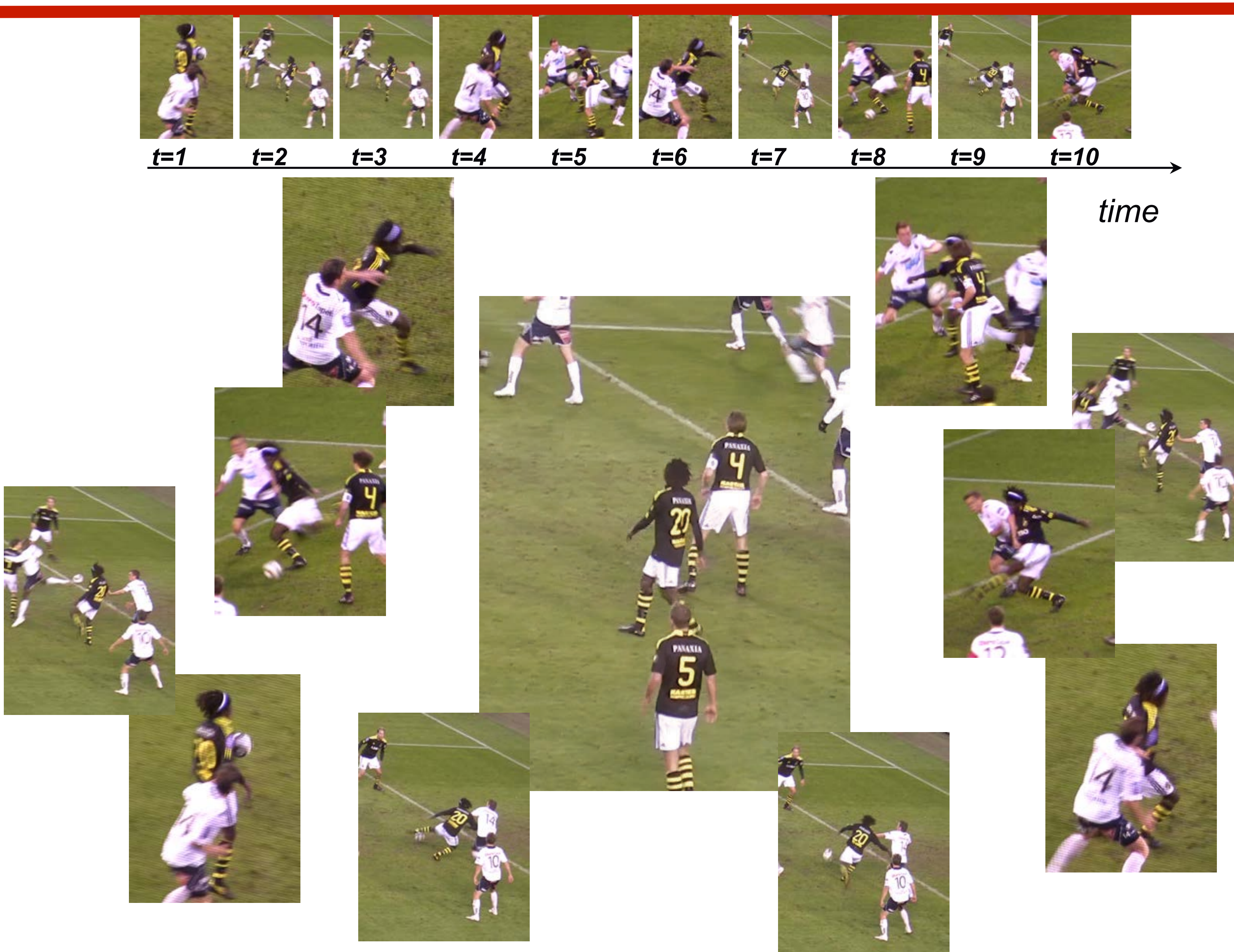


Reduces to an assignment problem with "dynamics- induced" weights

$$\frac{\mathrm{rank}(\mathbf{H}_i)+\mathrm{rank}(\mathbf{H}_j)}{\mathrm{rank}([\mathbf{H}_i\ \mathbf{H}_j])} - 1$$



Crowd photography sequencing

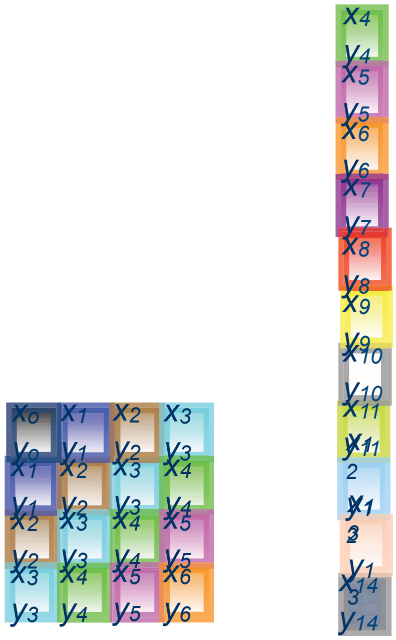


More examples where sparsity & self similarity help

- **Semi-supervised SysId**
- **Wiener systems identification**
- **Identification with outliers**
- **Identification of PWA systems**
- **(In)validation of PWA systems**
- **Sparse network Id.**
- **Optimal sensor placement**
- **Controller design subject to sparsity constraints**

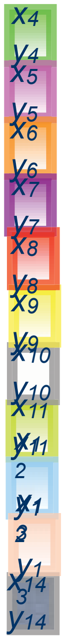
All of these are known to be NP-hard, yet often solvable in polynomial time using sparsity based convex relaxations

What is Big Data?

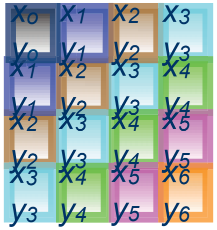


What is Big (Dynamic) Data?

Computational complexity is related to data interconnectivity, not data size!!



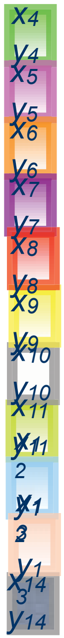
Easy



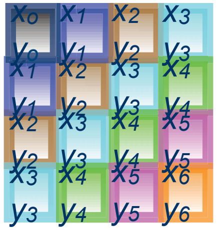
Hard

What is Big (Dynamic) Data?

Computational complexity is related to data interconnectivity, not data size!!



Easy



Hard

Related to max clique size of an underlying graph

Big Data & Sparsity:

Sparsity can provide a way around the curse of dimensionality

- **Challenge: how to build in and exploit the "right" sparsity**
  - **Graphs with small tree width (network design)**
  - **Low order models**
- **Submodularity also helps**
  - **what other properties can we exploit?**
- **An interesting connection between several communities:**
  - **Control, semi-algebraic optimization, machine learning,….**

Big Data & Sparsity:

> ***Sparsity can provide a way around the curse of dimensiona***

- **Challenge: how to build in and exp ne "right" sparsity**
  - **Graphs with small tree width (n rk design)**
  - **Low order models**
- **Submodularity also help**
  - **what other propertie we exploit?**
- **An interesting co on between several communities:**
  - **Control, sem braic optimization, machine learning,….**

If interested, consider joining the IEEE CSS CACSD TC

SoS for "real sized" problems :

- **Many promising advances towards making SoS/Moments practical:**
  - **ADMM, Frank-Wolfe, Factorizations**

- **Empirical experience: force M to be rank 1**

  In many practical problems (e.g. subspace clustering) forcing a small matrix (much smaller than the running intersection) to have rank 1 guarantees rank(M)=1

- **New developments covered elsewhere in this workshop**
  - **Ahmadi & Hall: DSoS and SDSoS**
  - **Lasserre: Krivine+Putinar P-satz (LP+fixed size SDC)**

- **Getting there, but more work needed. Keep tuned for more**

Acknowledgements:

- **Many thanks to:**
  - **Audience**
  - **Students: Y. Cheng, Y. Wang, X. Zhang**
  - **Colleagues: O. Camps. C. Lagoa, N. Ozay**
  - **Workshop organizers**
  - **Funding agencies (AFOSR, DHS, NSF)**

More information as http://robustsystems.coe.neu.edu