Dirty Models

Sujay Sanghavi

Electrical and Computer Engg.
University of Texas, Austin

Joint w/ C. Caramanis, Y. Chen, A. Jalali, P. Ravikumar, H. Xu

Outline

High-dimensional Problems

- Background, convex methods
- Why robustness ?
- Why flexibility ?

Our Basic idea

Results

Implications

Current and Future work

High-dimensional Problems

$$y = \mathcal{A}(X) + w$$

$\in \mathbb{R}^n$ $\in \mathbb{R}^p$

# of observations or samples

$n << p$

# of variables to determine or choices to make

Task: given $y, \mathcal{A}$ recover X

Dimensionality reduction : use structure in the data to reduce its effective dimension

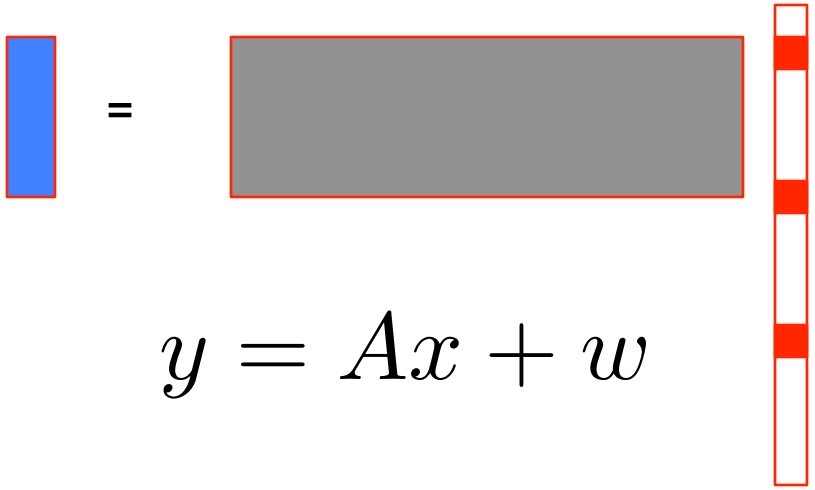
"solve" $y = \mathcal{A}(X) + w$

"s.t." $X \in \mathcal{C}$

e.g. $\mathcal{C}$ = sparse vectors, low rank matrices sparse MRFs,…

Common Structural Assumptions

Sparsity



Most / all of the mass of x in a very small **(but a-priori unknown)** set of coordinates.

$$y = Ax + w$$

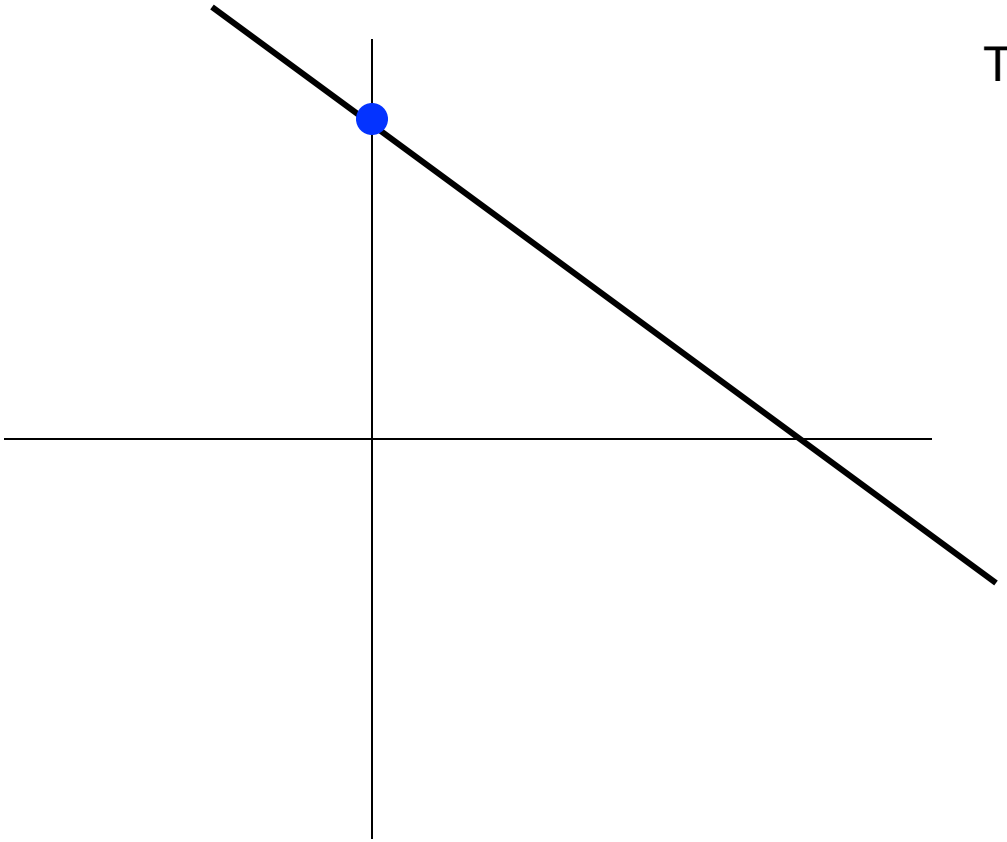
Sample Application Areas:

Natural and medical images – sparse in fourier / wavelet etc. bases

User modeling from website usage data

"Generic" linear regression where
out of many possible causes, only a few are relevant / expressed

Sparsity: Recovery via ℓ_1

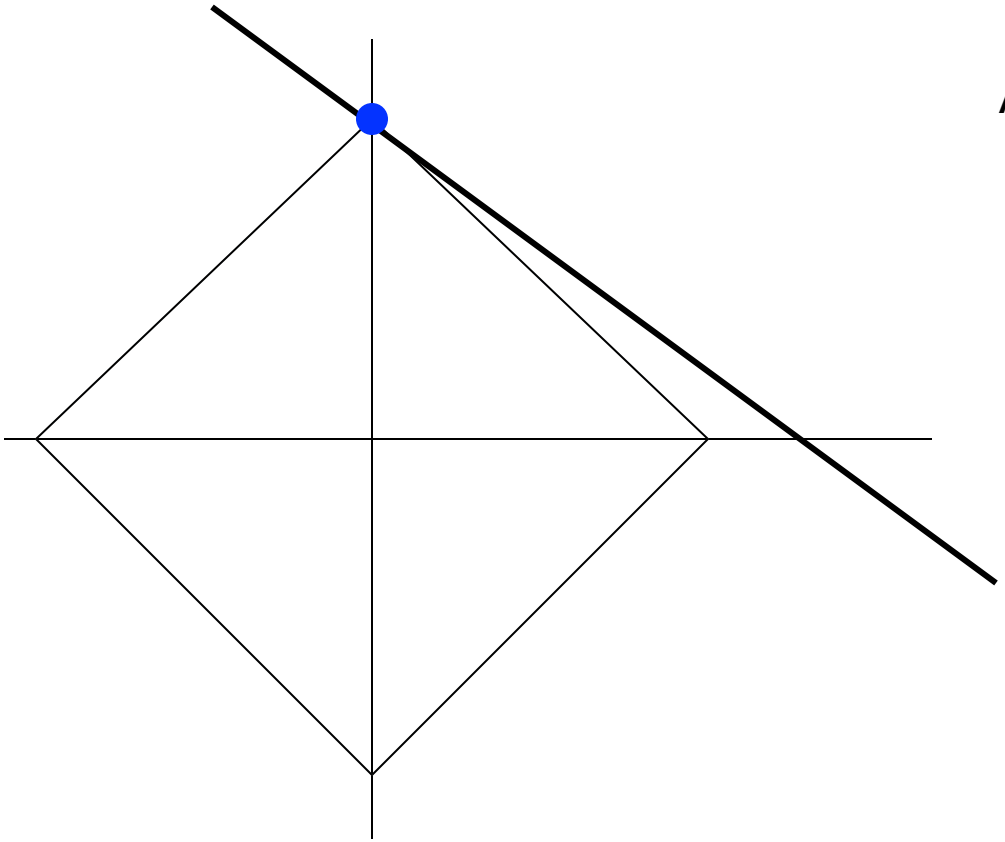


To solve:

$$y = Ax$$

For an x that has “many” zero variables

Recovery via ℓ_1



Algorithm:

$$\min \quad \|x\|_1$$

$$s.t. \quad y = Ax$$

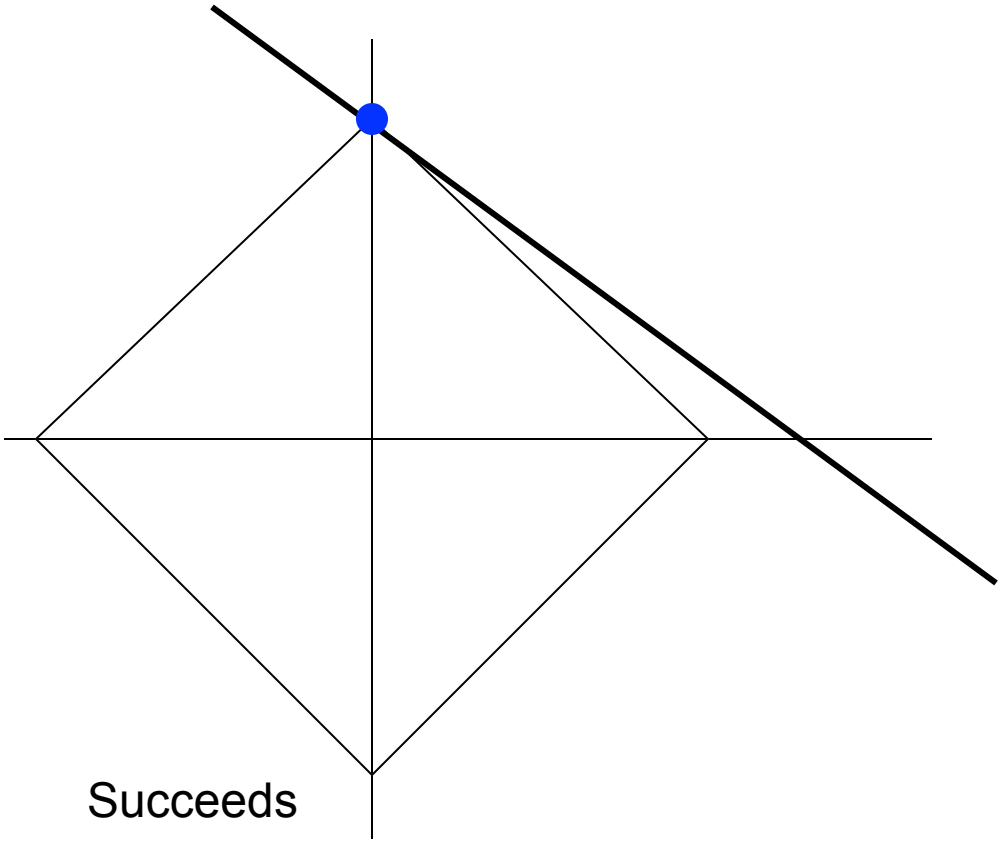
Intuitively:

ℓ_1 norm penalizes non-sparse vectors more

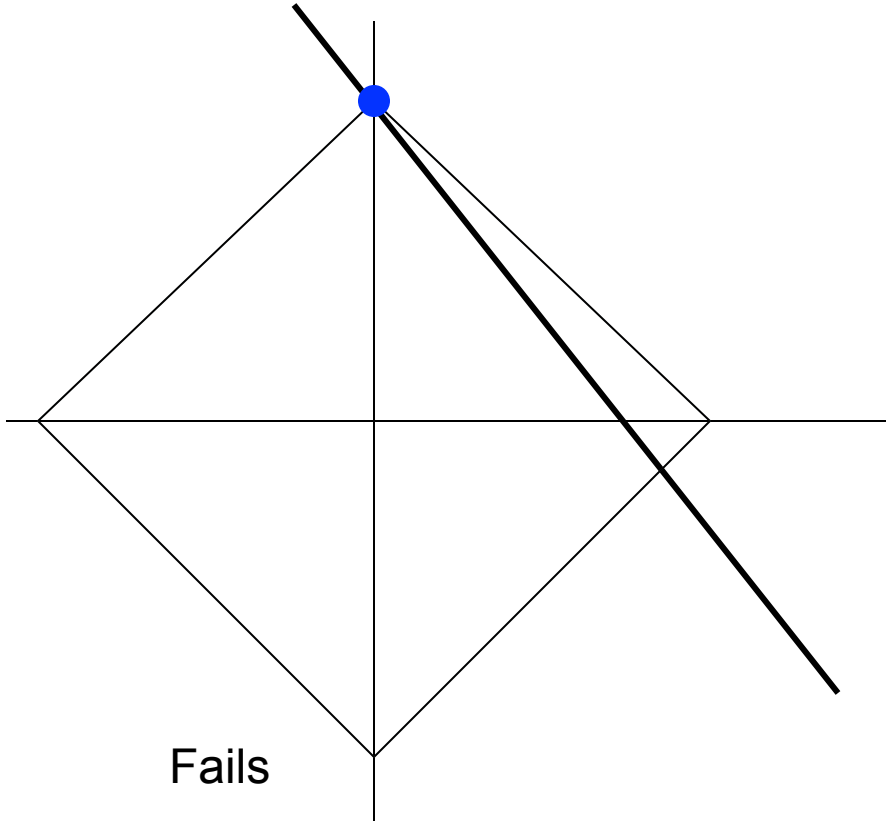
Formally:

It is the atomic norm arising from 1-sparse vectors

Recovery via ℓ_1



Succeeds



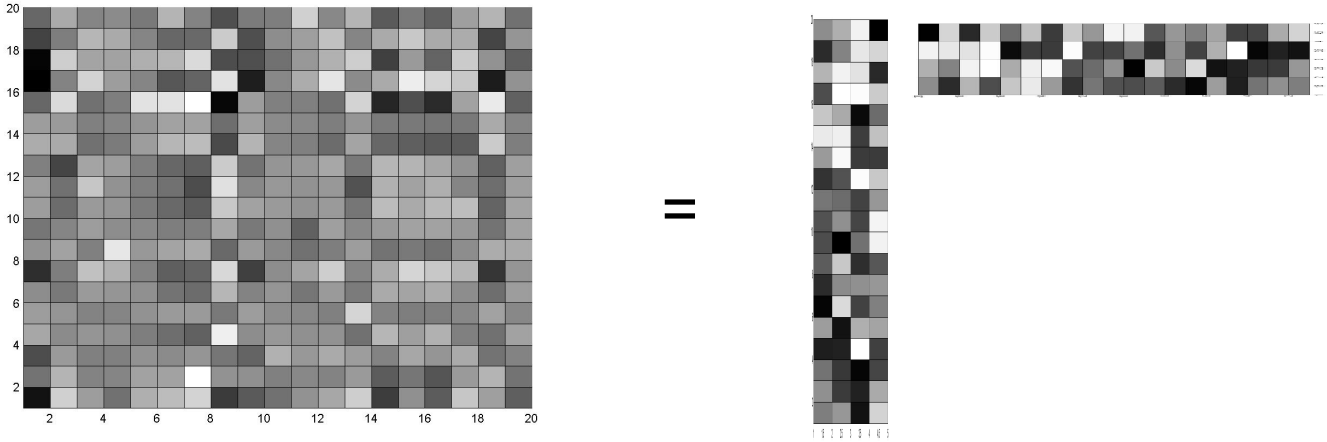
Fails

Recent research (last 6 years): detailed understanding of when we can expect success

Common Structural Assumptions

Low rank

When represented as an appropriate matrix, data is approx / exactly low rank



Sample Application Areas:

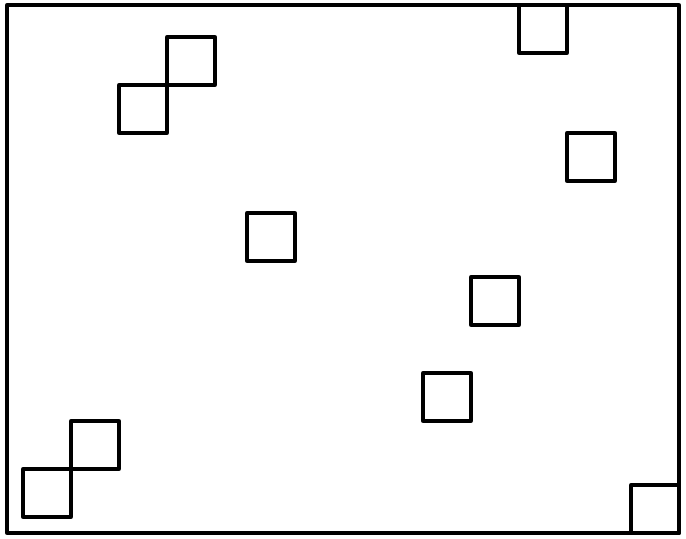
Principal components analysis (PCA) – the most popular dimensionality reduction technique

Data embedding (MDS / Isomap) and localization

Collbaorative Filtering

Graph clustering / community detection

Recovery via Trace / "Nuclear" Norm



E.g. **matrix completion**: find low-rank matrix from random subset Ω of elements

$$\min \quad \|X\|_*$$

$$s.t. \quad X_{ij} = m_{ij} \quad for\ (i,j) \in \Omega$$

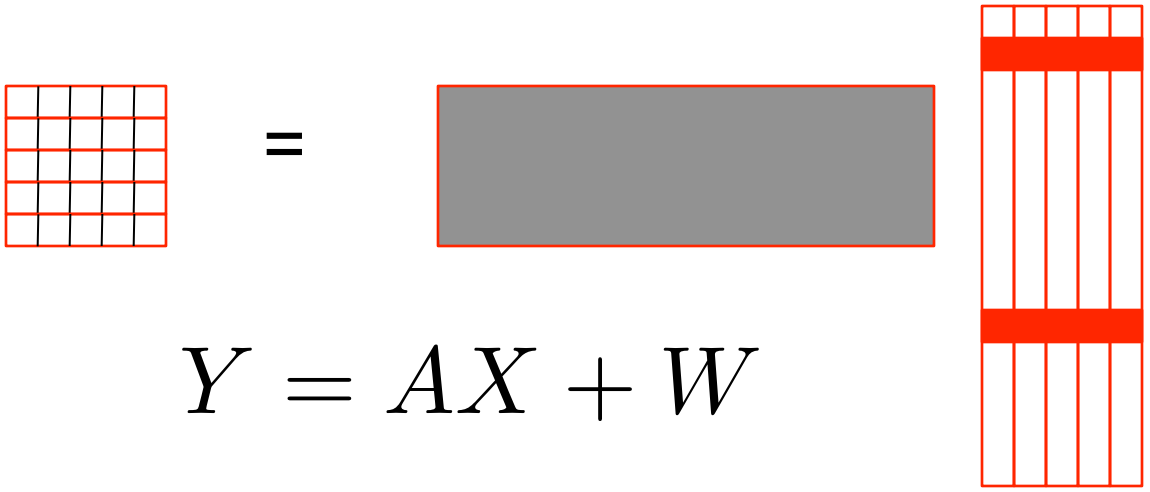
$\|X\|_* =$ Sum of singular values of a matrix

Intuitively: encourages sparsity in the singular values, aka low-rank-ness
(recall: rank = number of non-zero singular values)

Formally: is the atomic norm arising from rank-1 matrices

Common Structural Assumptions

Group Sparsity



Variables organized into **known** groups.

$$Y = AX + W$$

Only very few (a-priori unknown) of the **groups** are non-zero.
Different values possible within a group.

For a set G of groups, the norm that encourages group-sparsity is

$$\|X\|_G = \sum_{g \in G} \|X_g\|_2$$

Intuition: like an ℓ_1 norm at the level of groups

The story so far …

High-dimensional problems: Consistent recovery possible via structural assumptions

solve $y = \mathcal{A}(X) + w$ s.t. $X \in \mathcal{C}$

Convex optimization: a generic framework – penalize with the appropriate atomic norm

$$\min_X \quad \mathcal{L}(y, \mathcal{A}; X) \; + \; \lambda \, r(X)$$

Last decade: huge amount of work on two aspects of this approach

(a) Statistics: under what conditions does this find a good X ?

(b) Algorithms: fast/iterative methods that use structure of resulting convex programs

Our motivation

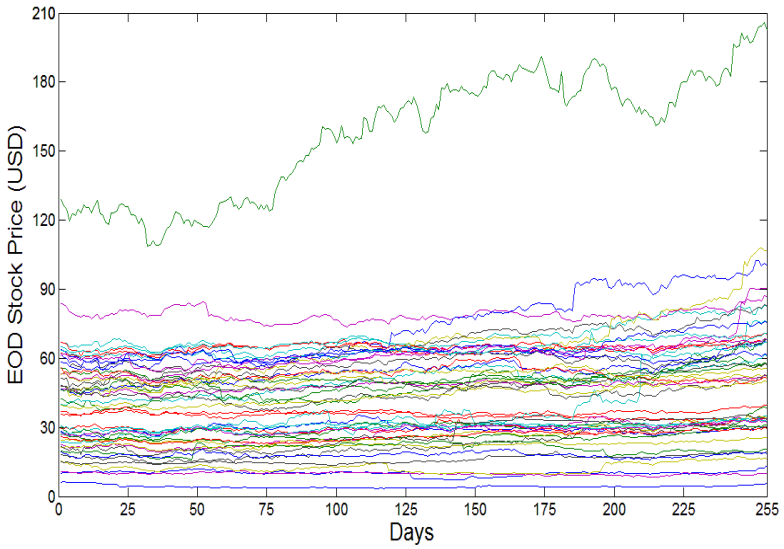
(a) Can we substantially expand the modeling power of such methods ?

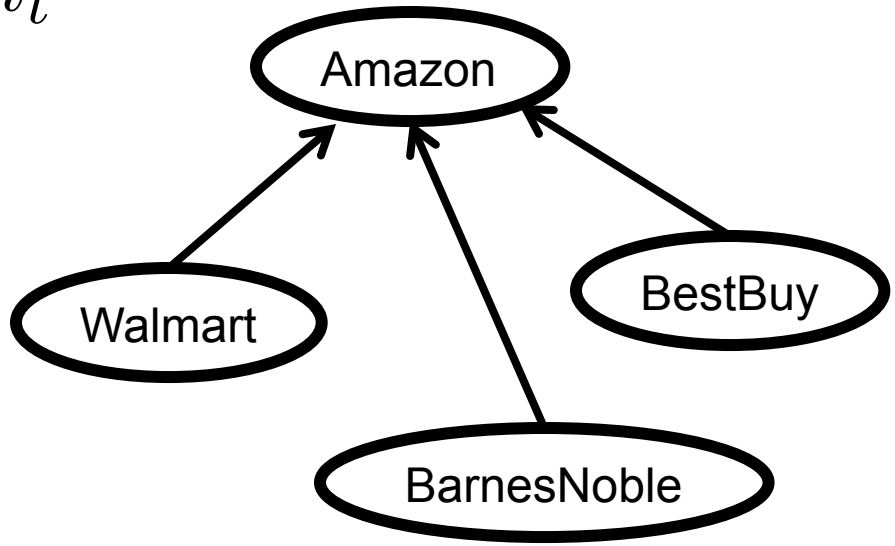
(b) Can we make these methods robust to errors / outliers ?

Example: Modeling

Application: sparse linear predictive modeling of (log of) stock prices

$$x_{t+1} = Ax_t + n_t$$





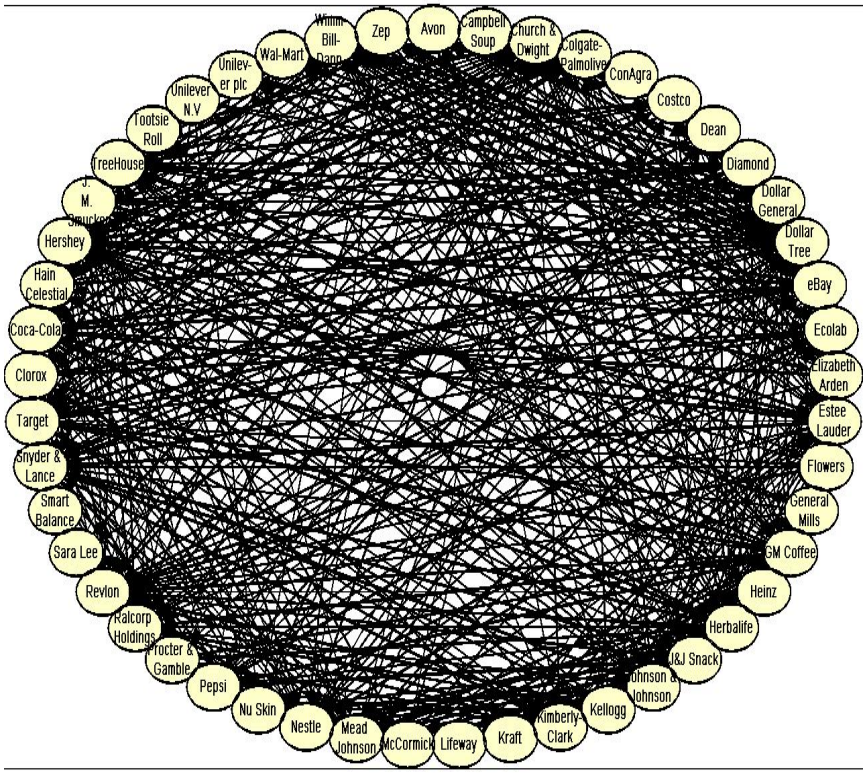
Task: find sparse A given

$$x_1, x_2, \ldots$$

Example: Modeling

Simple ℓ_1 minimization gives a dense model !

(also not very predictive)



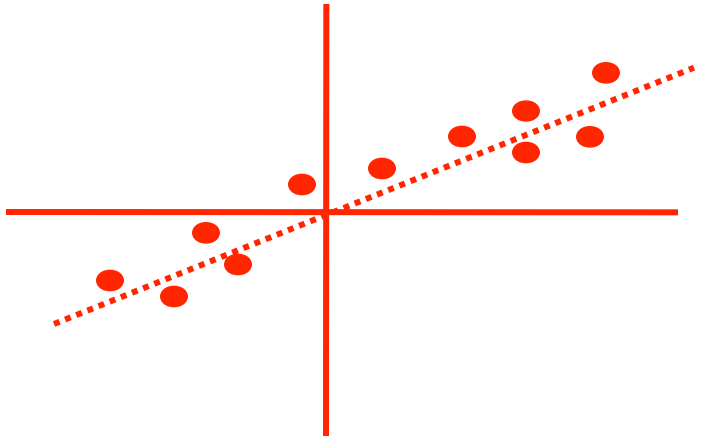
(one possible) Reason:

Each stocks seems to depend on many others because of co-dependence on latent factors

(e.g. US Fed Interest rates, price of gas, etc.)

Question: can we learn both the direct dependence and latent factors from only stock price data ?

Example: Robustness



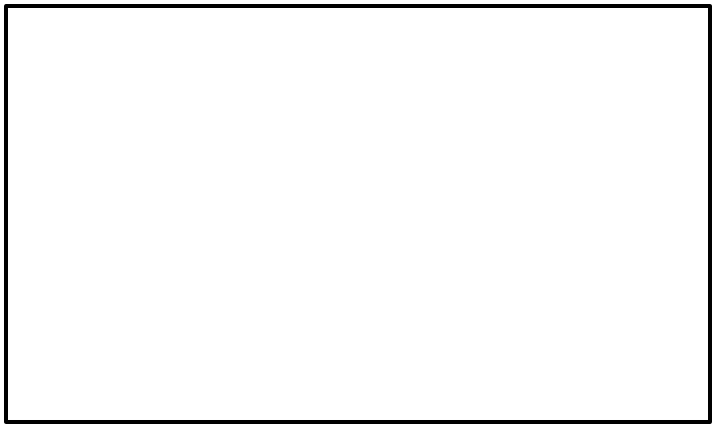
PCA: given n points in p -dim space, find the low-dimensional subspace that best approximates them.

Standard approach:
Organize points into data matrix
Take SVD, and retain significant components

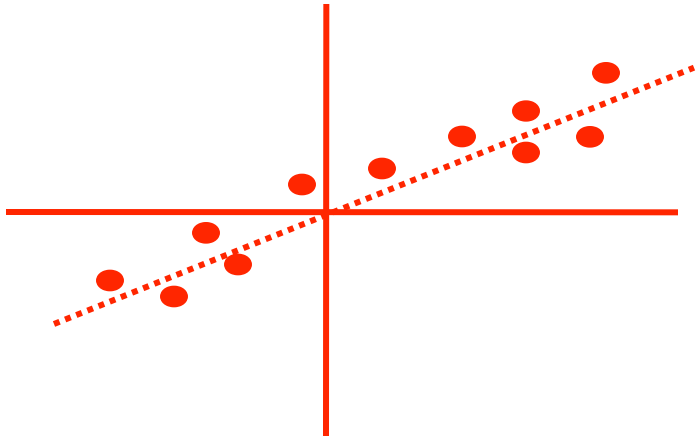
Structural assumption: **low-rank**

points

coordinates



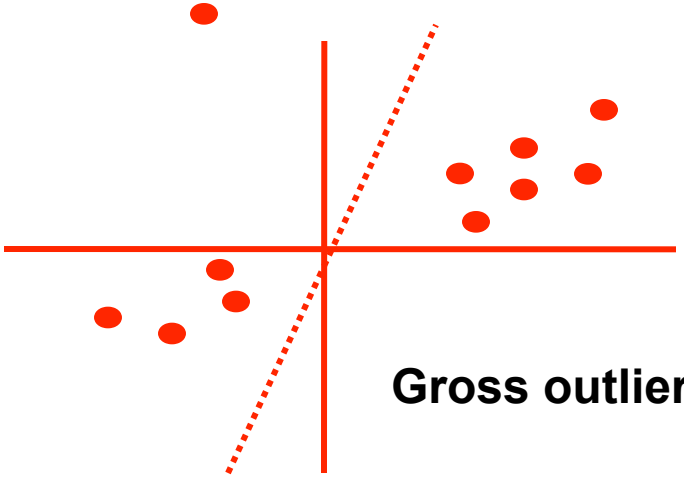
Example: Robustness



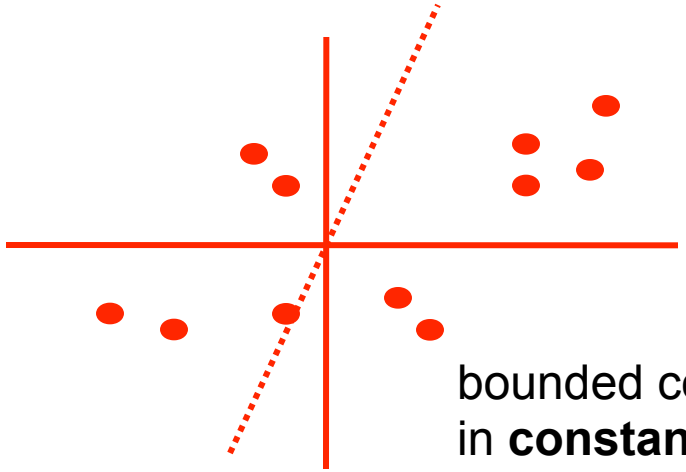
Standard approach:
Organize points into data matrix
Take SVD, and retain significant components

Structural assumption: **low-rank**

Fragile to:



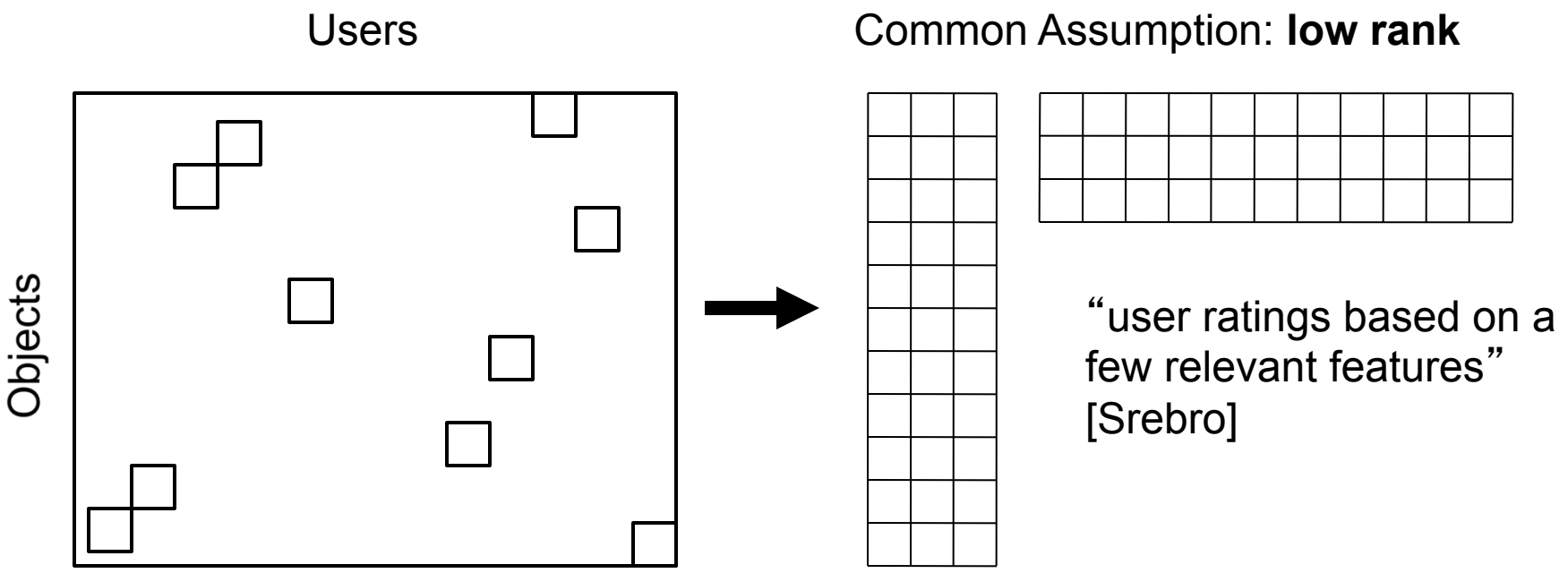
Gross outliers



bounded corruption
in **constant fraction**
of entries

Collaborative Filtering w/ Adversaries

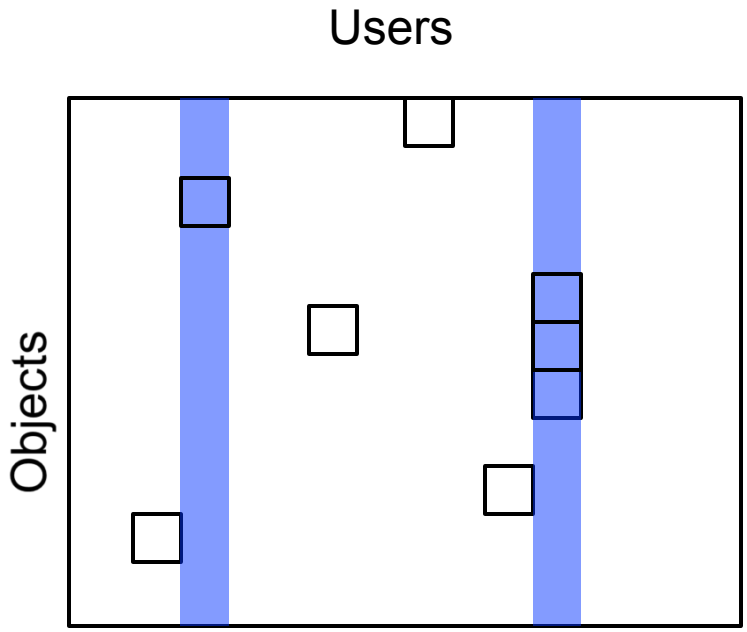
Collaborative Filtering



Popular approach: [Candes-Recht]

$$\min_{X} \quad \sum_{(i,j)\in\Omega} (x_{ij} - m_{ij})^2 \; + \; \lambda \|X\|_*$$

Collaborative Filtering w/ Adversaries

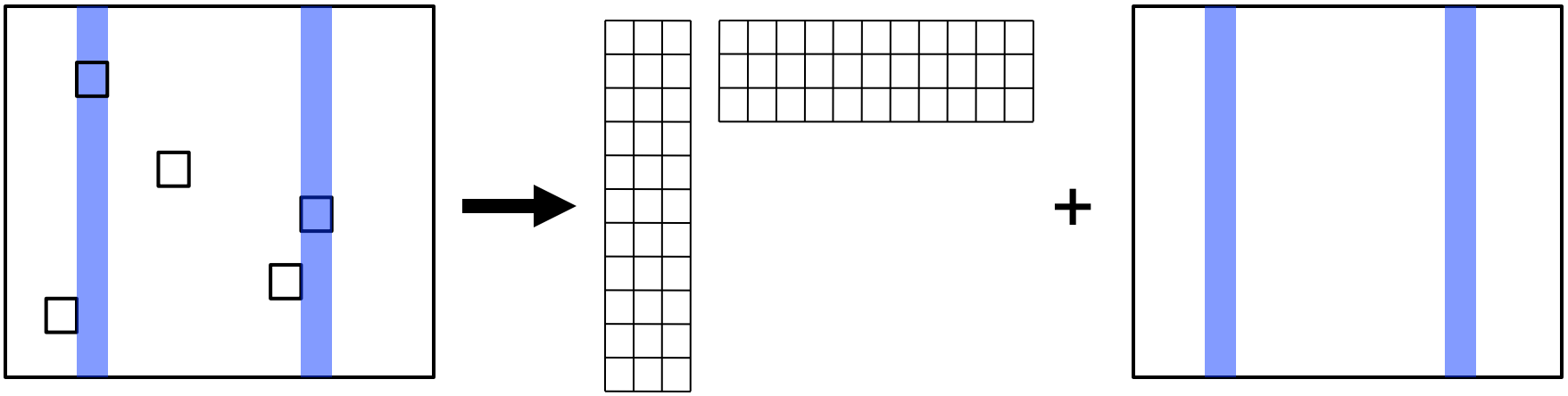


Some users malicious (e.g, “shilling for” or “nuking” a product, or generally disruptive)

If these provide **inconsistent** ratings (which defy the low-rank approx. that works for everyone else), we should be able to negate their effects.

Main challenge: figuring out identities of the adversaries from v. few observations of both their and the true users’ rankings

Our Idea



Given Low-rank Column-sparse

Idea: Represent data as the **superposition** of more than one structural model

(**both** of which now have to be learnt from data)

Basic Idea

Our methods:

"solve" $y = \mathcal{A}(X_1 + X_2) + w$

"s.t." $X_1 \in \mathcal{C}_1 \quad X_2 \in \mathcal{C}_2$

(linear) Superposition of more than one structural model class

- Each individual model class already used in isolation
  - computationally efficient by combining efficient methods for the component classes.
  - But can now be more robust / flexible

Approach: Convex Optimiation

Single structure

$$\min_{X} \quad \mathcal{L}(y, \mathcal{A}; X) \; + \; \lambda \, r(X)$$

Loss function regularizer

Our approach:

$$\min_{X_1, X_2} \quad \mathcal{L}(y, \mathcal{A}; X_1 + X_2) \; + \; \lambda_1 \, r_1(X_1) \; + \; \lambda_2 \, r_2(X_2)$$

(same) Loss function Weighted sum of regularizers

Focus of our analysis: statsitical

when does this approach succeed in recovering the true structure ?

Not our focus today: fast algorithms to solve these convex programs ….

Objectives

Robustness:

X_1 == the target we want to recover

X_2 == "errors" we want to separate out

No hope if errors completely arbitrary

Requires the objects to be separately identifiable / "incoherent"
(need to exclude the possibility of $X_1, X_2 \in \mathcal{C}_1 \cap \mathcal{C}_2$)

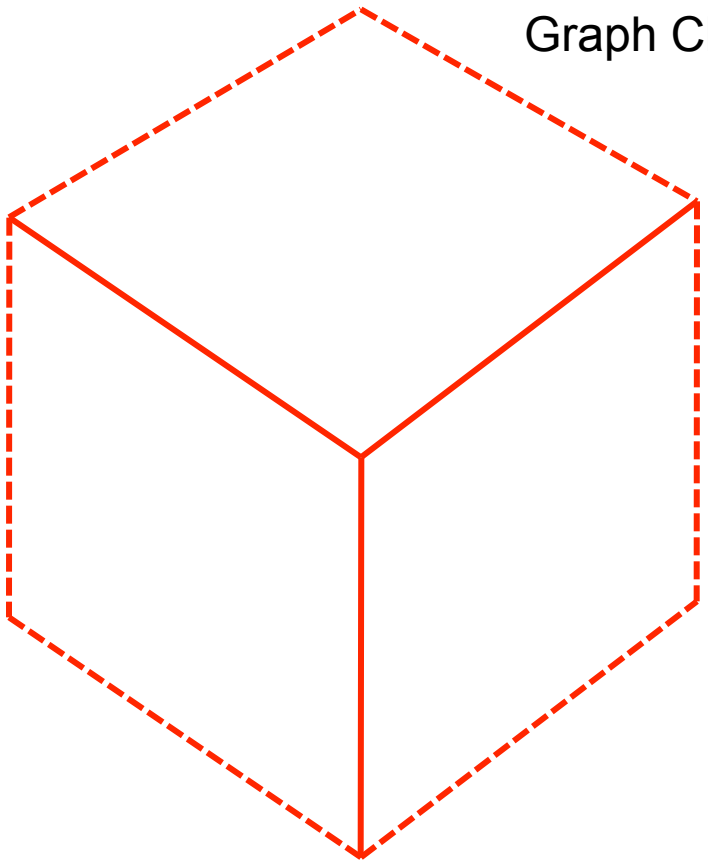
Flexibility:

Recover only the sum $X_1 + X_2$ from few measurements

(faster) Consistency in settings not captured by any one class

Do not need objects to be separately identifiable
because all we anyway care about is their superposition

Our Work



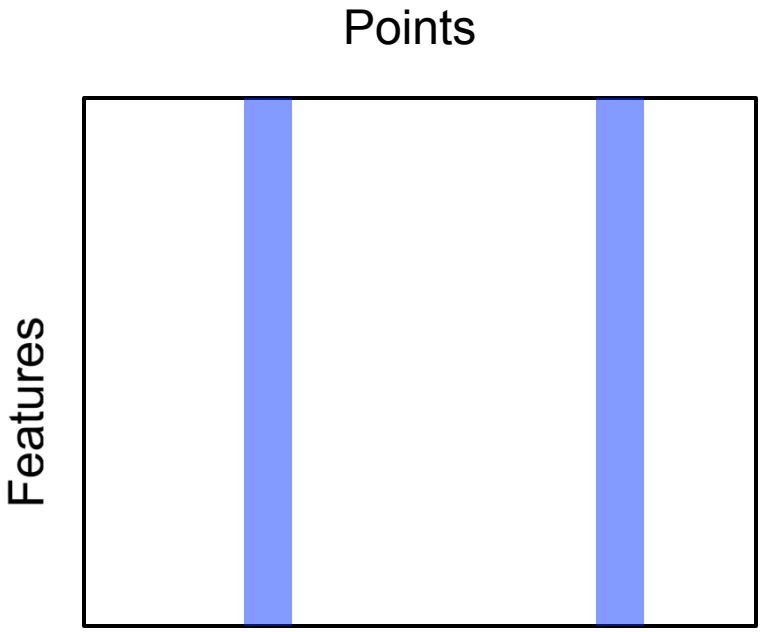
PCA w/ corruptions, **Latent variables in linear time series**, Graph Clustering …

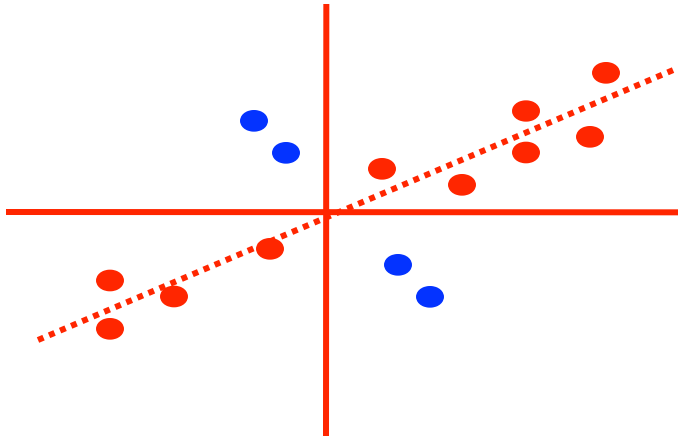
Multiple sparse Model selection

PCA w/ outliers, **Robust Collaborative Filtering**, …

Low-rank + Column-sparse

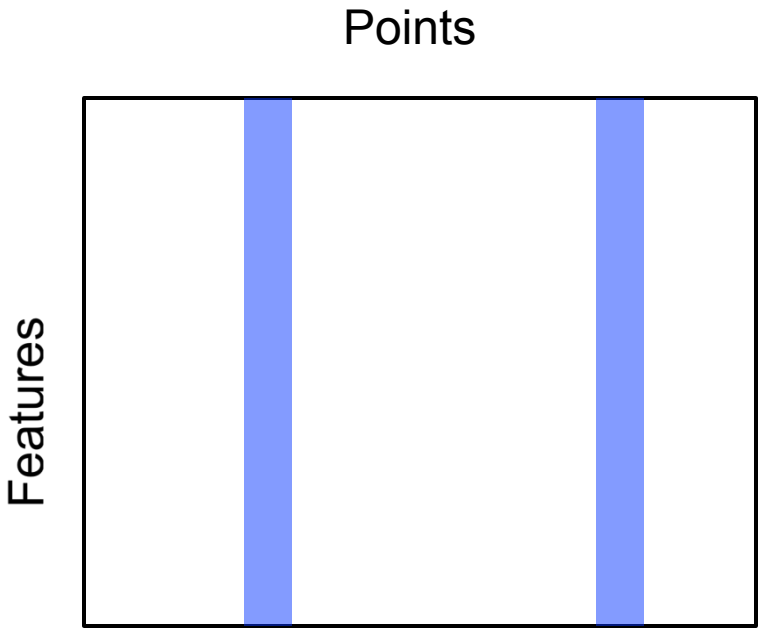
PCA with Outliers

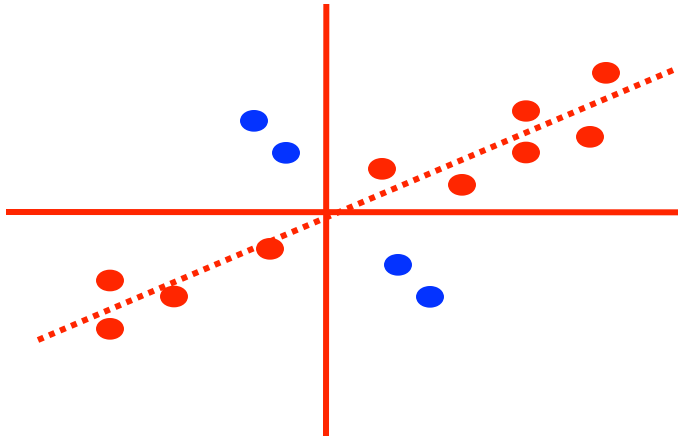




Objective: find identities of outliers
(and hence col. space of true matrix)

Robust PCA





Standard PCA

$$\min_{L} \quad \|M - L\|_F$$

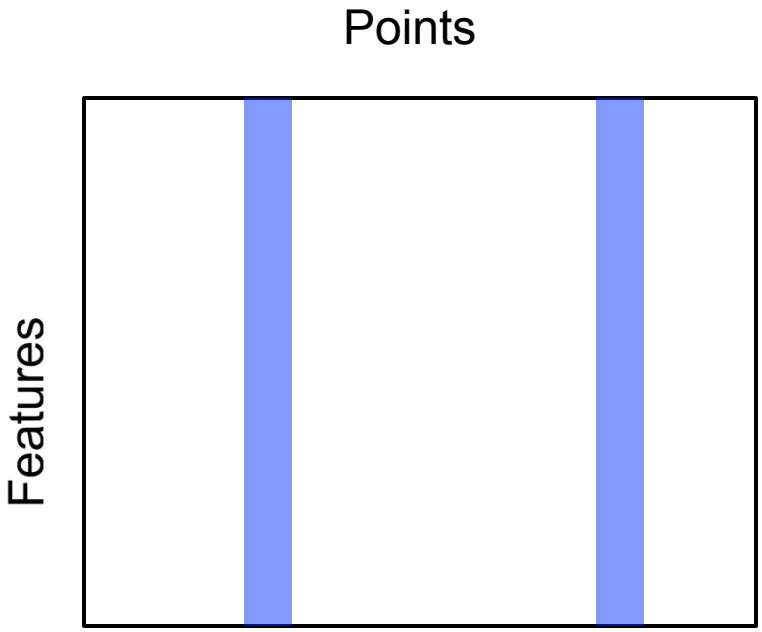
$$s.t. \quad rank(L) = r$$

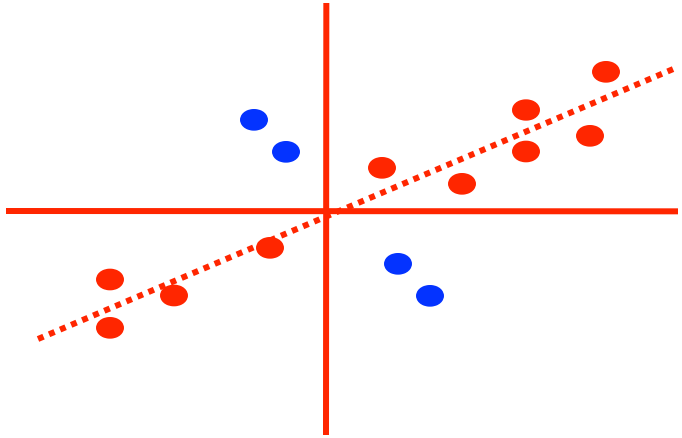
$$\min_{L,C} \quad \|M - L - C\|_F$$

$$s.t. \quad rank(L) = r$$

$$col(C) = c$$

Robust PCA



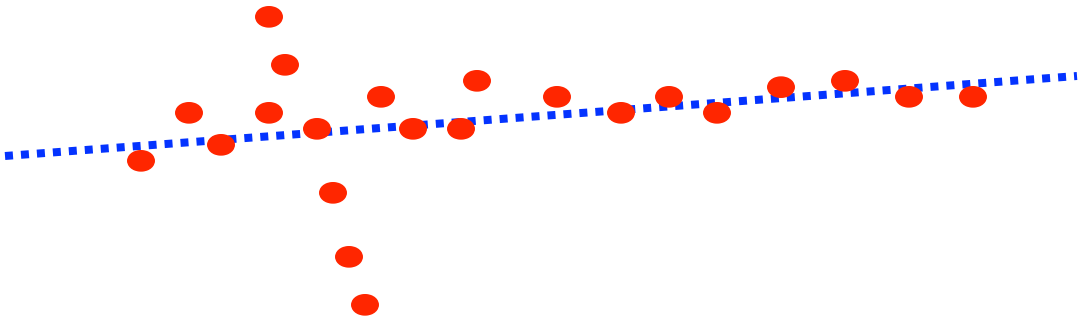


We propose:

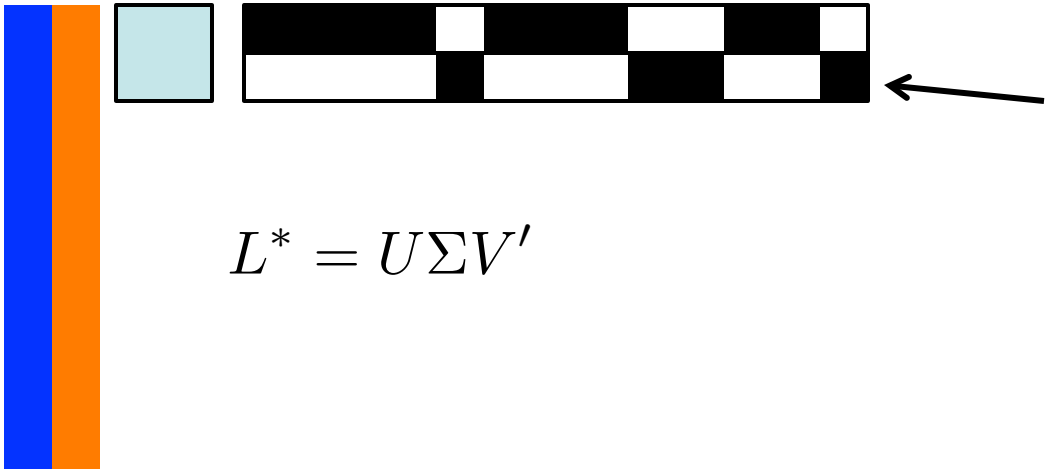
$$\min_{L,C} \quad \|M - L - C\|_F + \lambda_1 \|L\|_* + \lambda_2 \|C\|_{1,2}$$

When does this recover the true (L^*, C^*)

When does it (not) work ?



When certain directions of column space of L^* poorly represented



$$L^* = U\Sigma V'$$

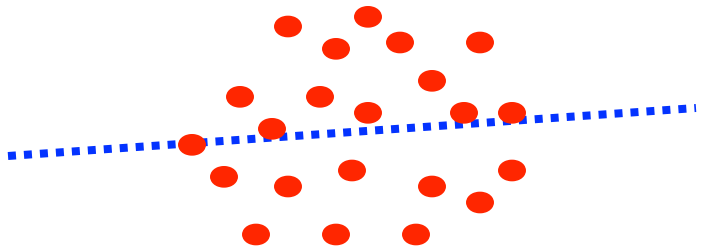
This vector has large inner product with some coordinate axes

$$\max_i \|V' e_i\|$$ is large

Results

Assumption:
Columns of true L^* are **incoherent:**

$$\max_i \|V' e_i\|^2 \leq \frac{\mu r}{n}$$



Note: $r \leq \mu r \leq n$

Theorem: (noiseless case)

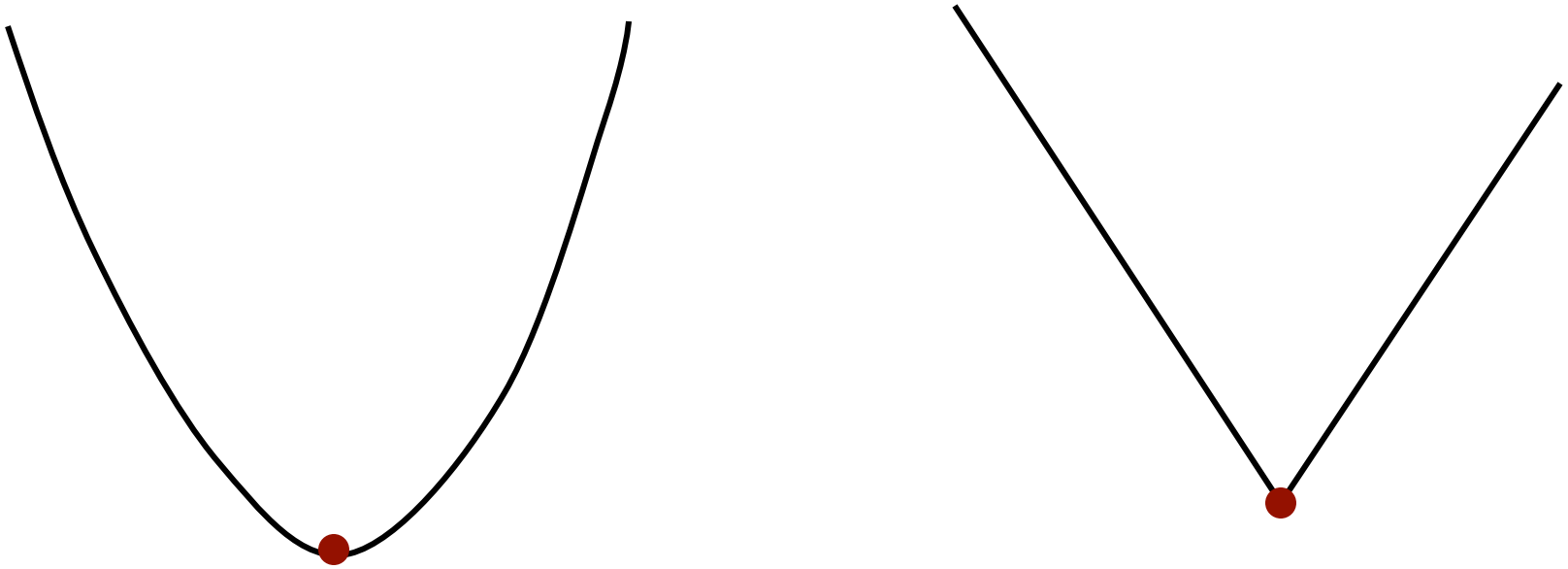
Our convex program can identify upto a fraction γ of outliers as long as

$$\frac{\gamma}{1-\gamma} \leq \frac{c}{\mu r}$$

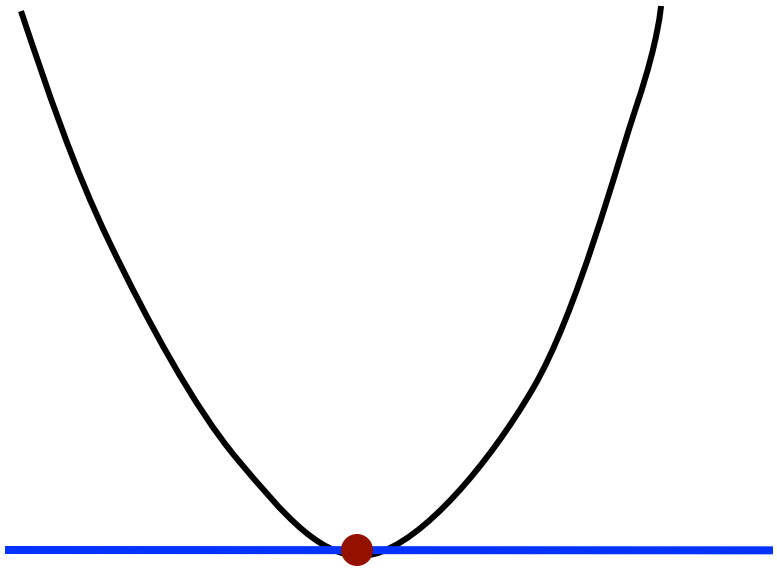
$$\lambda = \frac{3}{7\sqrt{\gamma n}}$$

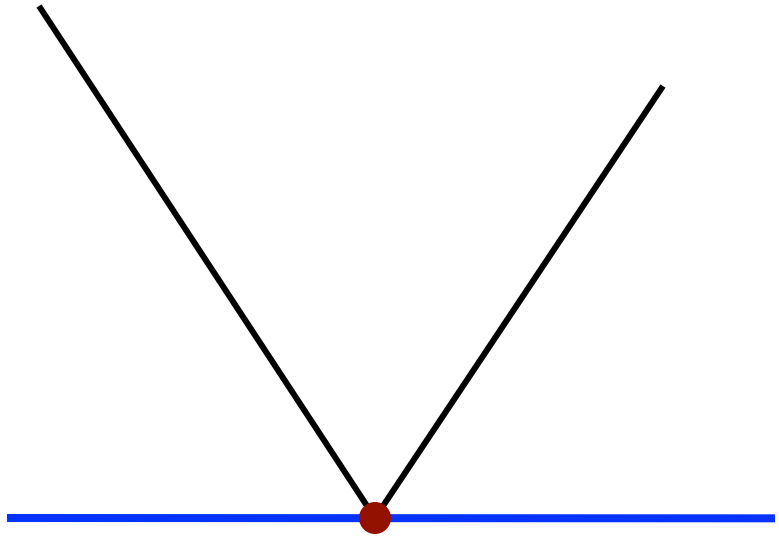
Outer bound: $\gamma > \frac{1}{r+1}$ makes the problem un-identifiable

Proof Technique



Proof Technique



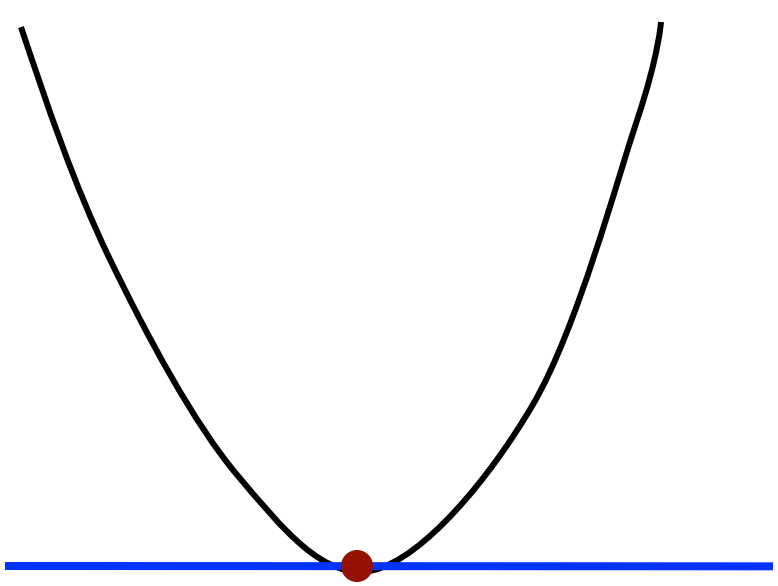


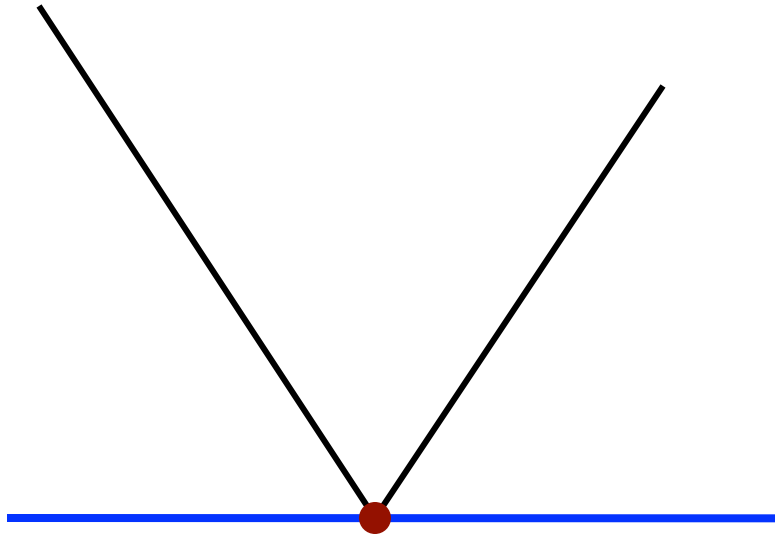
A point x is the optimum of a convex function f

$\Leftrightarrow$

Zero lies in the (sub) gradient $\partial f(x)$ of f at x

Proof Technique





A point x is the optimum of a convex function f

$\Leftrightarrow$

Zero lies in the (sub) gradient $\partial f(x)$ of f at x

Idea: 1. guess a "nice" point, -- **"art"**

2. show it is the optimum by showing zero is in subgradient – **"math"**

Proof Technique

Guessing a "nice" optimum

(Note: in "single structure" problems like matrix completion, compressed sensing etc., this is not an issue)

Oracle Problem:

$$\min_{L,C} \quad \|M - L - C\|_F + \lambda_1 \|L\|_* + \lambda_2 \|C\|_{1,2}$$

$$s.t. \quad ColSupp(C) \subset ColSupp(C^*)$$

$$ColSpace(L) \subset ColSpace(L^*)$$

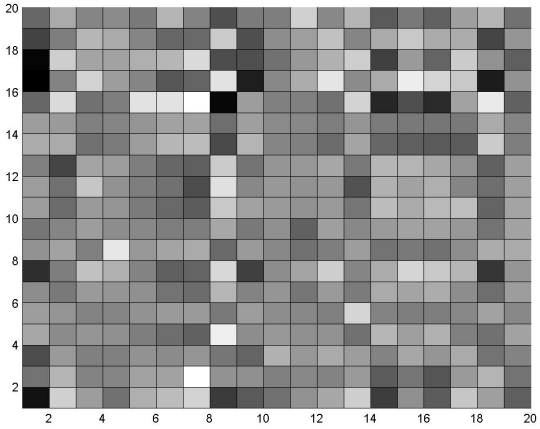
$(\widehat{L}, \widehat{C})$ is, by definition, a nice point.

Rest of proof: showing it is the optimum of original program, under our assumption.

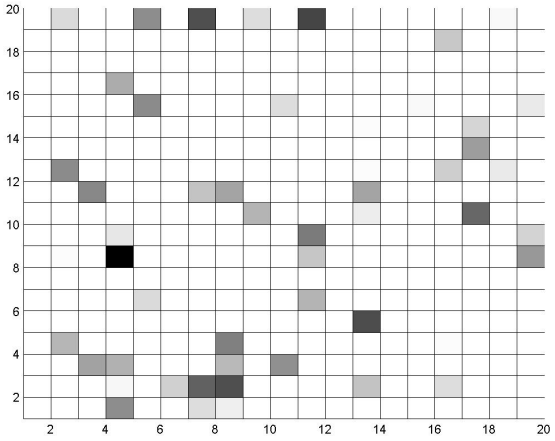
Sparse + Low-Rank

Basic theory

$C = A^* + B^*$

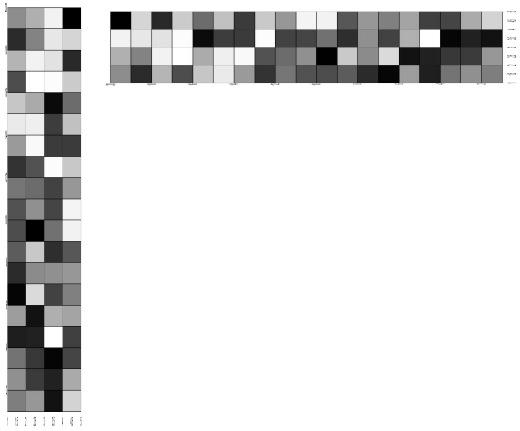


Given Composite matrix



Unknown Sparse Matrix

Unknown support, values

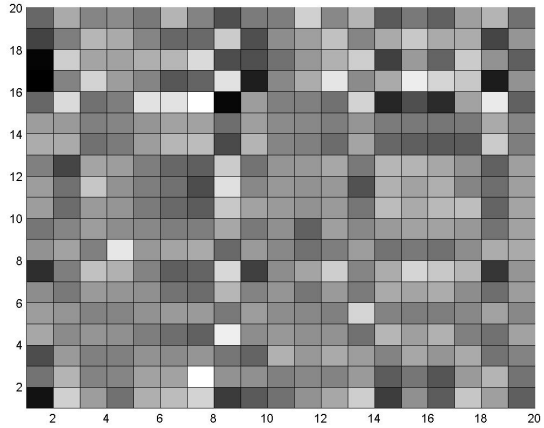


Unknown Low-rank Matrix

Unknown rank, eigenvectors

The Task

$C = A^* + B^*$



Given Composite matrix

→ ?

Unknown Sparse Matrix

Unknown support, values

?

Unknown Low-rank Matrix

Unknown rank, eigenvectors

Task: given (partially observed) C, recover A^* and B^*

Method

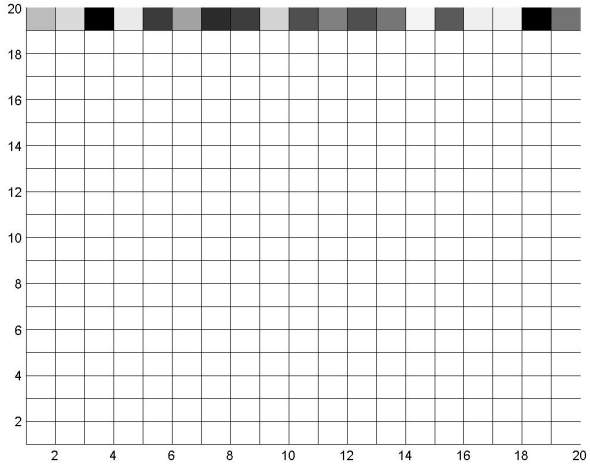
$$\min_{A,B} \quad \gamma \, ||A||_1 + ||B||_*$$

$$\text{s.t.} \quad \mathcal{P}_\Omega(A + B) = \mathcal{P}_\Omega(C)$$

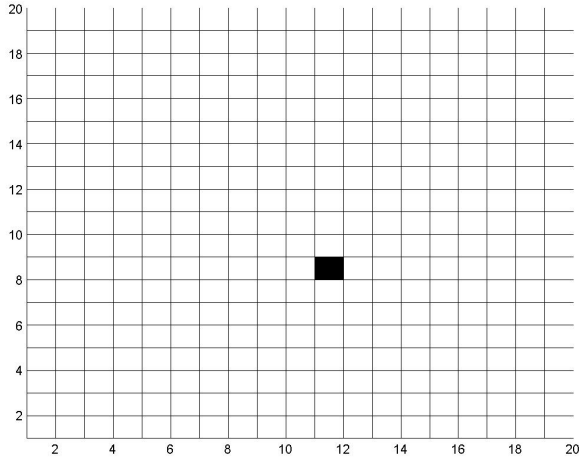
where $||A||_1 = \sum_{i,j} |a_{ij}|$ and $||B||_* = \sum_i \sigma_i(B)$

When is recovery not possible ?

1) sparse matrix is “concentrated”



2) Low-rank matrix is also sparse



3) Too few observations in a given row/column

The setting

Observed set $\Omega = \Omega_d \cup \Omega_r$

Each location in Ω_r w/ prob. p_o

Support of sparse matrix $\Gamma = \Gamma_d \cup \Gamma_r$

Each location in Γ_r w/ prob. τ

Adversarial errors/erasures degree bounded

any row or column of $\Omega_d^c \cup \Gamma_d$ has $\leq d$ elements

Low-rank matrix is μ incoherent: if SVD is $B^* = U\Sigma V'$ then

$$\|U'e_i\|^2 \leq \frac{\mu r}{n} \qquad \|V'e_i\|^2 \leq \frac{\mu r}{n} \qquad \|UV'\|_\infty^2 \leq \frac{\mu r}{n^2}$$

(same as the conditions for pure matrix completion [Candes-Recht])

Main S + L result

Theorem:

Our convex program gives *exact recovery* – i.e. has $\mathcal{P}_\Omega(A^*), B^*$ as its unique optimum if

$$p_o \geq c_1 \max\left\{ \frac{\mu r \log^4 n}{n} , \sqrt{\frac{\mu r d}{n}} \log n \right\}$$

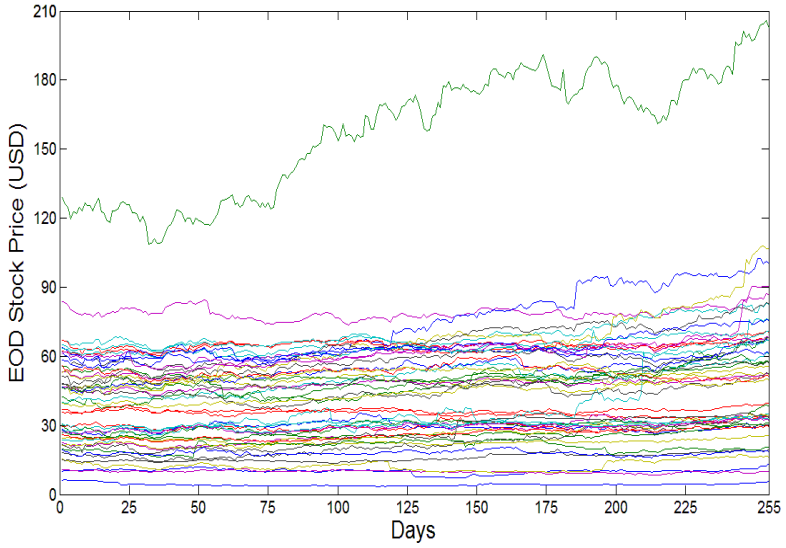
$$\tau \leq c_2$$

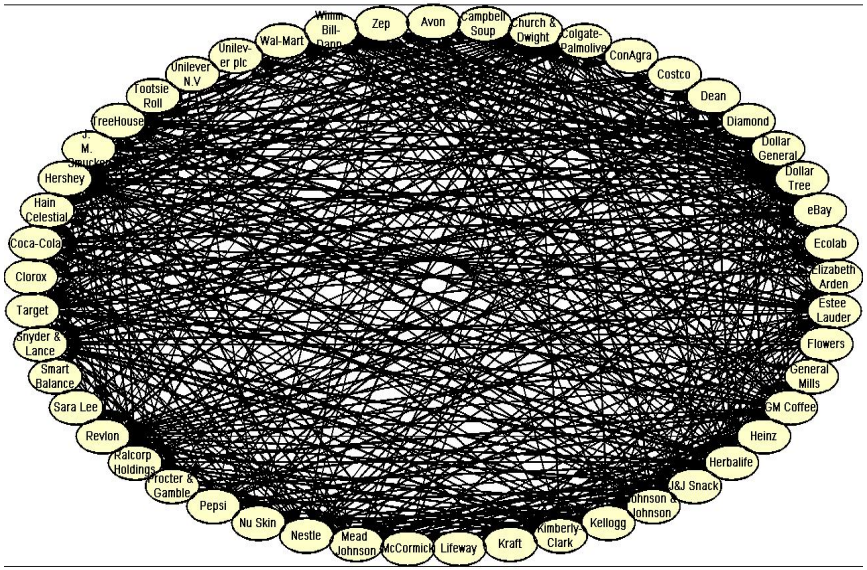
$$d \leq c_3 \frac{n}{\mu r \log^2 n}$$

w/ parameter

$$\gamma = \frac{1}{\sqrt{n(d+1)p_o}}$$

Application: Latent factors in Linear Time Series





Linear Stochastic Dynamical Systems

Vector-valued stochastic process $x(t) = [x_1(t) \dots x_p(t)]^T$

Continuous time:

Standard brownian motion

$$\frac{d}{dt}x(t) \;=\; A\,x(t) \;+\; \frac{d}{dt}w(t)$$

Discrete time:

$\mathcal{N}(0, \eta I)$

$$x(n+1) - x(n) \;= \eta A x(n) + w(n)$$

"sampled version" : as $\eta \to 0$, discrete $\longrightarrow$ continuous with $t \approx n\eta$

Learning / System Identification

Discrete time: Given $x(0:n)$ find A

Maximum Likelihood:

$$\min_{A} \quad \sum_{i=0}^{n-1} \| x(i+1) - x(i) - \eta A x(i) \|_2^2$$

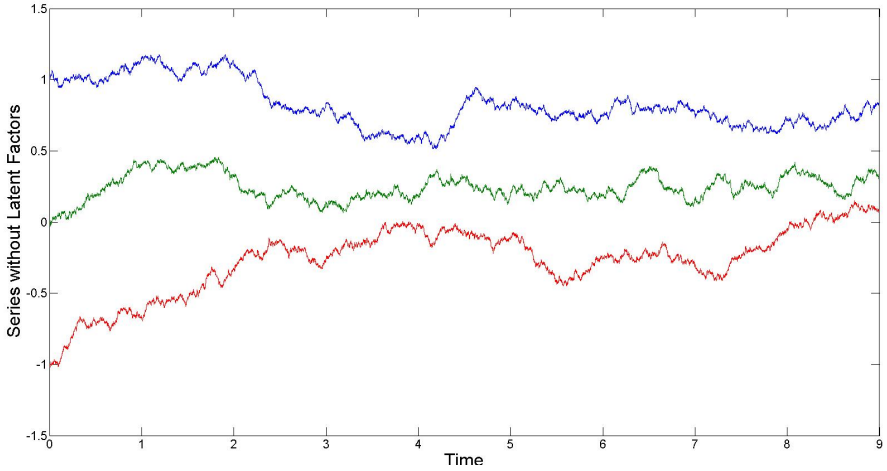
Known: $\widehat{A}_{ML} \to A$ as $n \longrightarrow \infty$

for fixed η

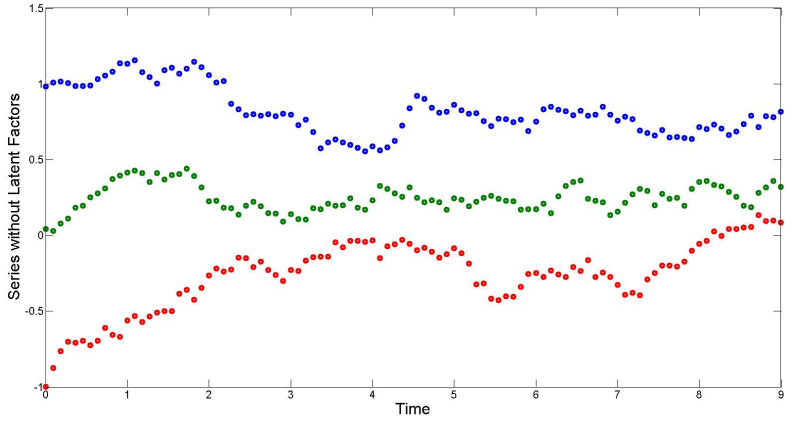
Our focus: **high-dimensional regime**

" fewer samples than the size of A "

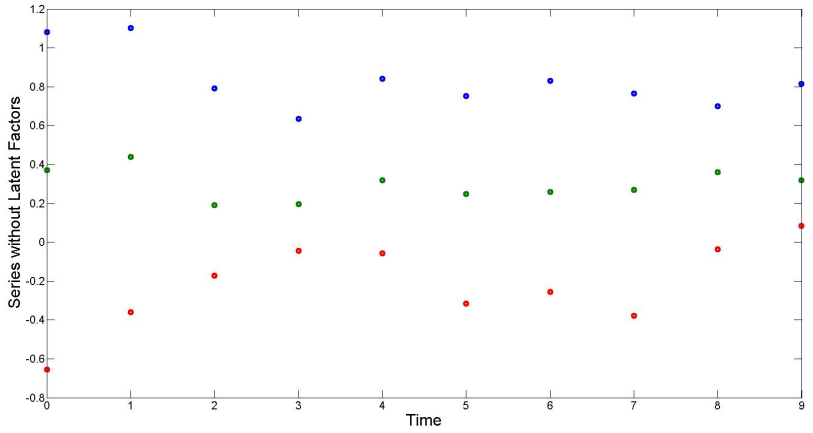
Sampling



What does "sample complexity" and "high-dimensional" mean ?



Many samples, but small innovation per sample



Fewer samples, but more innovation per sample

Latent Variables

x = observed variables u = latent variables – **never observed**

Continuous time:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + \frac{d}{dt} w(t),$$

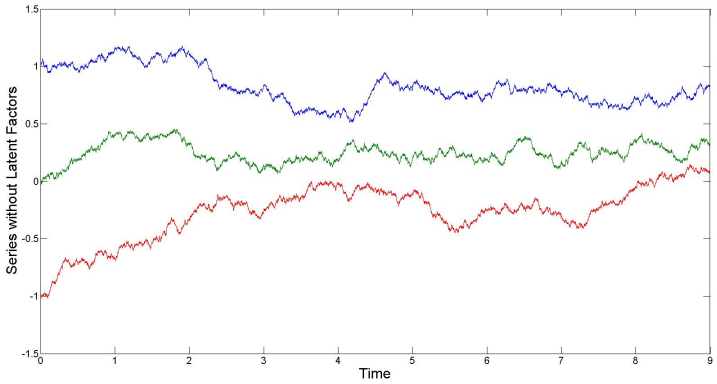
Discrete time:

$$\begin{bmatrix} x(n+1) \\ u(n+1) \end{bmatrix} - \begin{bmatrix} x(n) \\ u(n) \end{bmatrix} = \eta \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(n) \\ u(n) \end{bmatrix} + w(n)$$

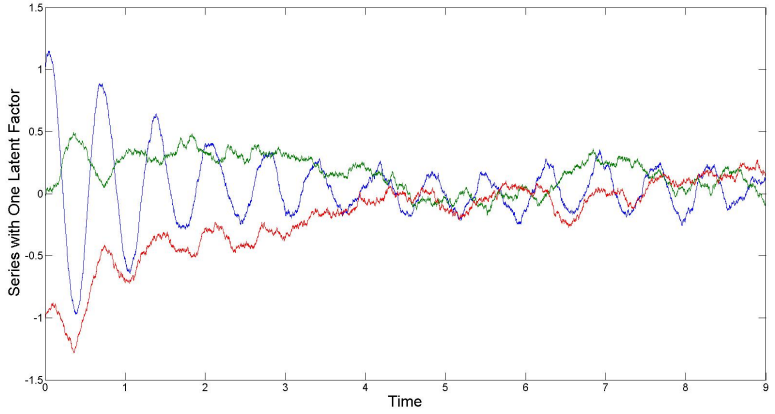
Our SysID problem:Given $x(0:n)$ find A.

SysID with Latent Variables

The problem: (for naïve methods like ML from before),
latent variables $\Rightarrow$ spurious dependence among observed variables



$$\widehat{A}_{ML} = \quad = A$$



$$\widehat{A}_{ML} =$$

But we (still) want to find A

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} =$$

The Effect of Latent Variables

What is the structure of the spurious interactions caused by x ?

Idea: let us solve ML anyway …

$$\min_{A} \quad \sum_{i=0}^{n-1} \|x(i+1) - x(i) - \eta A x(i)\|_2^2$$

but x from $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$

For $n \longrightarrow \infty$ and fixed η

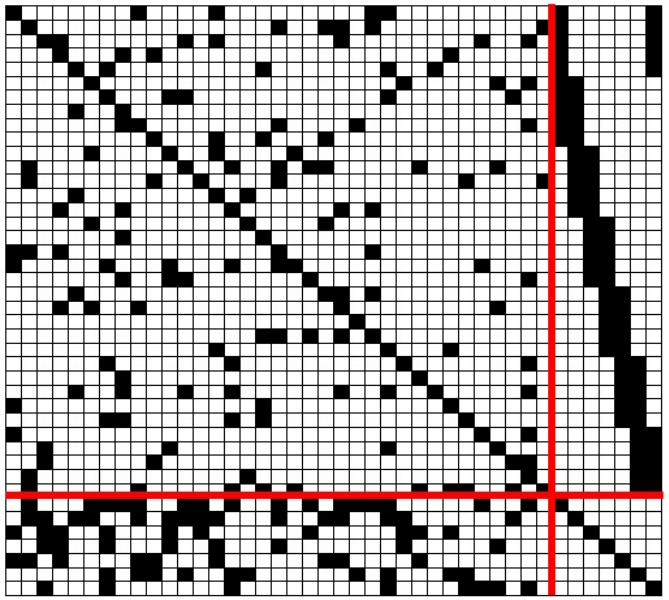
$$\widehat{A}_{ML} \;\rightarrow\; A + BRQ^{-1}$$

where $R = E[ux^T]$ $\qquad$ $Q = E[xx^T]$

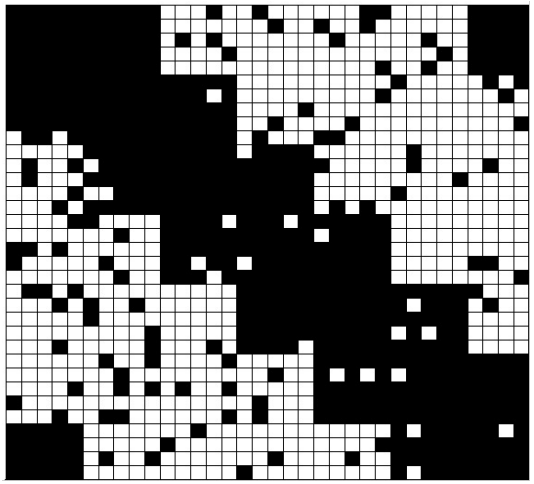
are steady state covariances

The Effect of Latent Variables

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$



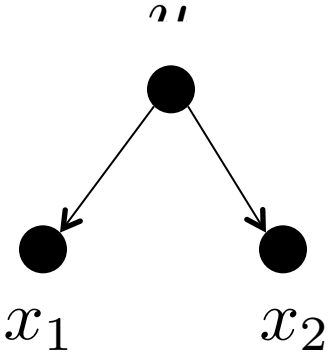
$$A + BRQ^{-1}$$



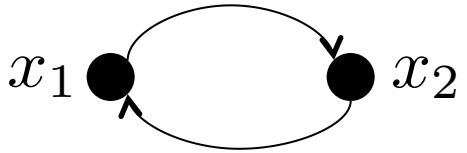
finding A == separating A from BRQ^{-1}

Now: (a) when this is (not) ill-posed ?
(b) And how do we do it ?
(c) sample complexity, high-dimensional scaling etc.

Ill-posed-ness



==



$$\dot{u} = du + \dot{w}$$

$$\dot{x}_1 = a_{11}x_1 + b_1 u + \dot{w}_1$$

$$\dot{x}_2 = a_{22}x_2 + b_2 u + \dot{w}_2$$

$$\dot{x}_1 = \widetilde{a}_{11}x_1 + \widetilde{a}_{21}x_2 + \dot{w}_1$$

$$\dot{x}_2 = \widetilde{a}_{12}x_1 + \widetilde{a}_{22}x_2 + \dot{w}_2$$

Ill-posed problems

(with infinite data / exact statistics),

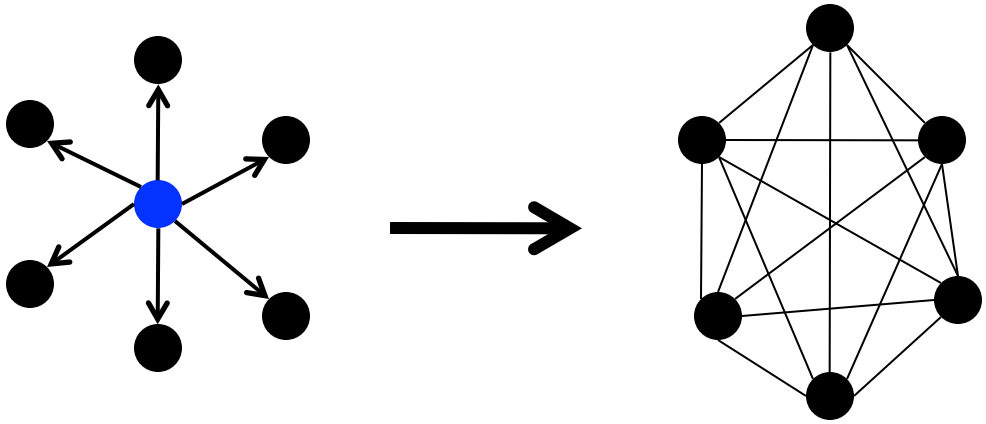
$$\widehat{A}_{ML} \;=\; A + BRQ^{-1}$$

If $card(u) < card(x)$ then $\widehat{A}_{ML}$ is a **low-rank perturbation** of A.

Recovering A == *separating* it from a low-rank matrix

This is the source of ill-posed-ness.

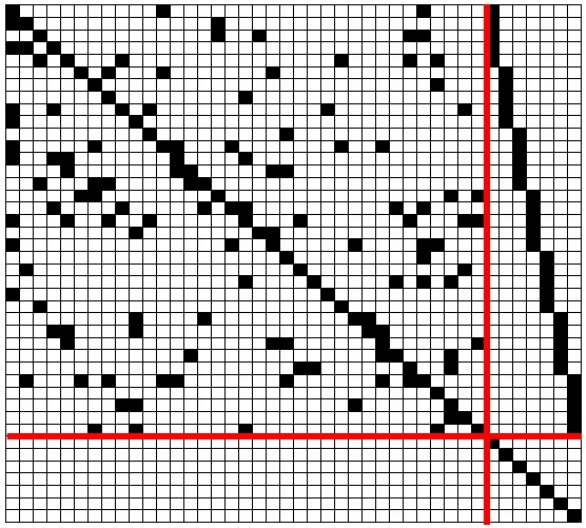
Effect of Latent variables

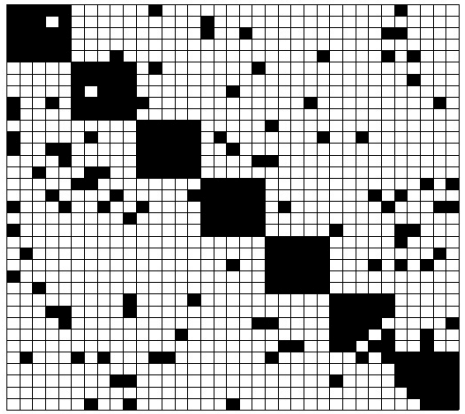


Each latent variable results in a clique of its neighbors.

“Natural” assumption:

***A* sparse**





& each latent var. is “significant” connected to many observed

of course, they can overlap …

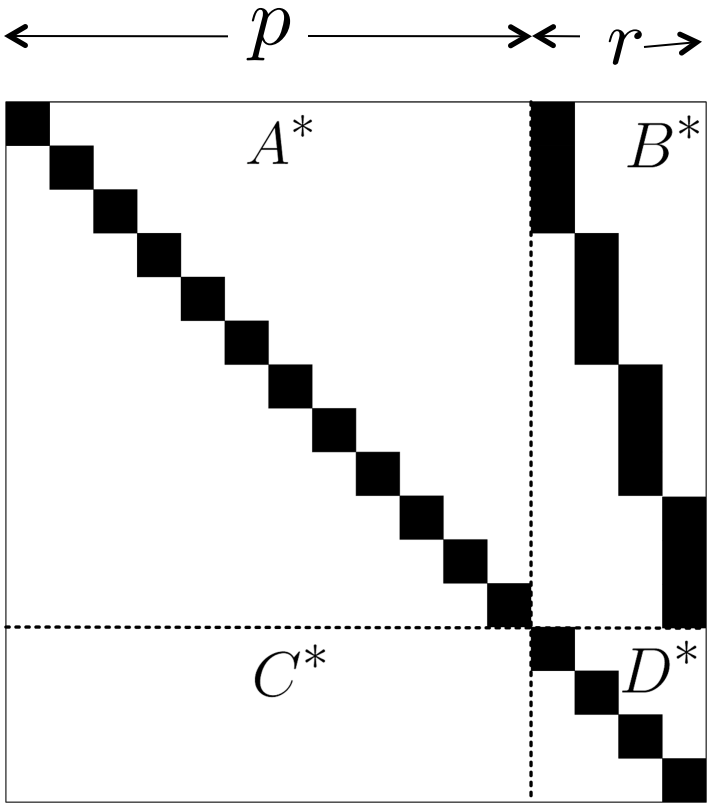
Method

Idea: Look for a superposition of a sparse matrix and a low-rank one

$$\min_{A,L} \quad \frac{1}{2\eta^2 n} \sum_{i=0}^{n-1} \|x(i+1) - x(i) - \eta(A+L)x(i)\|_2^2 + \lambda_A \|A\|_1 + \lambda_L \|L\|_*.$$

Now: the “sample complexity” of identifying *A*

Simple illustrative case



p observed variables, each depends on exactly one latent variable

r latent variables, each evolving independently

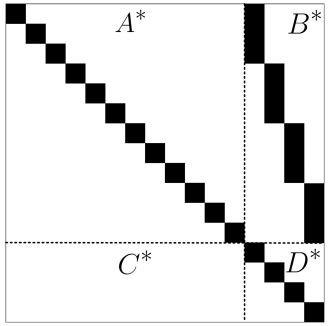
each connected to $\frac{p}{r}$

$$L = BRQ^{-1} = \frac{r}{p+r} BB^T$$

Corollary of our theorem: can uniquely recover *A* when

Simple illustrative case

(Corollary): If $\eta < \dfrac{1}{\sigma_{max}(\mathcal{A})}$



a) $r < \dfrac{\sqrt{p}}{3}$

b) $\eta n \;\geq\; K \log \dfrac{4(1+2r)p + 4r^2}{\delta}$

Then, w.p. at least $1-\delta$, we have that

$$supp(\widehat{A}) = supp(A)$$

and bounds on $\|\widehat{A} - A\|_\infty$ and $\|\widehat{L} - L\|_2$

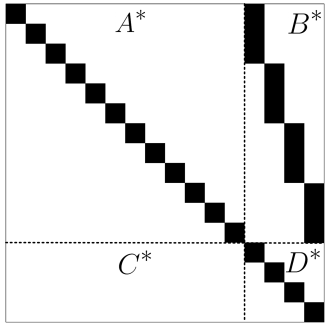
Simple illustrative case

(Corollary): If $\eta < \dfrac{1}{\sigma_{max}(\mathcal{A})}$

Sampling has to be close enough, else "independent" samples

a) $r < \dfrac{\sqrt{p}}{3}$

each latent var. connected to at least $3\sqrt{p}$ observed variables



b) $\eta n \geq K \log \dfrac{4(1+2r)p + 4r^2}{\delta}$

Then, w.p. at least $1-\delta$, we have that

$$supp(\widehat{A}) = supp(A)$$

and bounds on $\|\widehat{A} - A\|_\infty$ and $\|\widehat{L} - L\|_2$

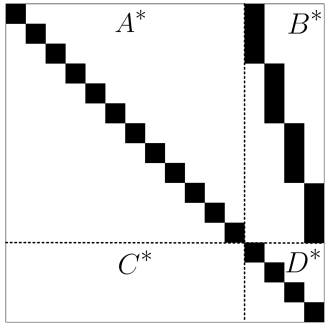
Simple illustrative case

(Corollary): If $\eta < \dfrac{1}{\sigma_{max}(\mathcal{A})}$

Sampling has to be close enough, else "independent" samples

a) $r < \dfrac{\sqrt{p}}{3}$

each latent var. connected to at least $3\sqrt{p}$ observed variables



b) $\eta n \geq K \log \dfrac{4(1+2r)p + 4r^2}{\delta}$

Lower bound on the **total time horizon** of observation

Then, w.p. at least $1-\delta$, we have that

$$supp(\widehat{A}) = supp(A)$$

and bounds on $\|\widehat{A} - A\|_\infty$ and $\|\widehat{L} - L\|_2$

More generally …

Overall system needs to be stable

Q should be a good design matrix.

(assumptions similar to sparse + low-rank decomposition)

Low-rank matrix needs to be incoherent: if $L = U\Sigma V^T$ then

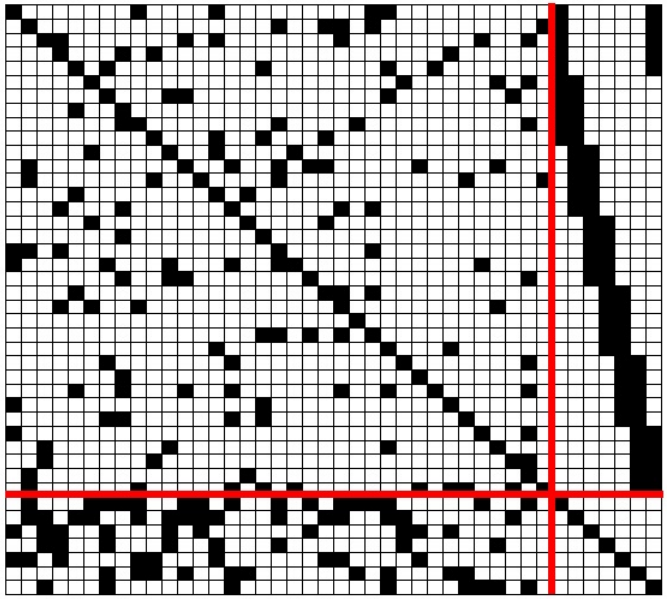
$$\{\max_i \|U^T e_i\|, \max_i \|V^T e_j\|\} \leq \sqrt{\frac{\mu r}{p}} \qquad \|UV^T\|_\infty \leq \sqrt{\frac{\mu r}{p^2}}$$

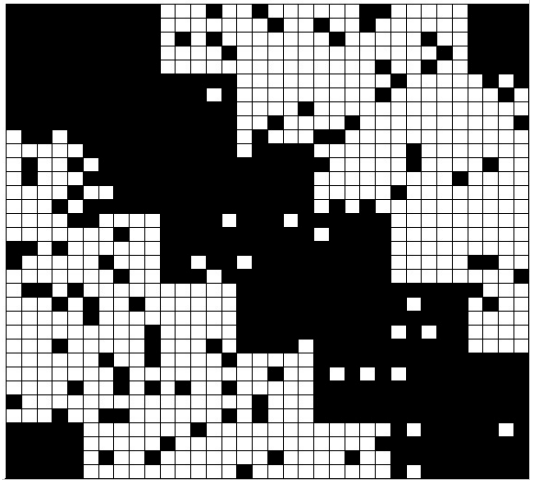
Number of non-zeros in any row or column of *A* is at most *s*

Theorem: recover support of *A* with prob. at least $1 - \delta$ if

$$T = n\eta \geq K\, s^3 \log\left(\frac{4((s+2r)p + r^2)}{\delta}\right)$$

Synthetic Experiments





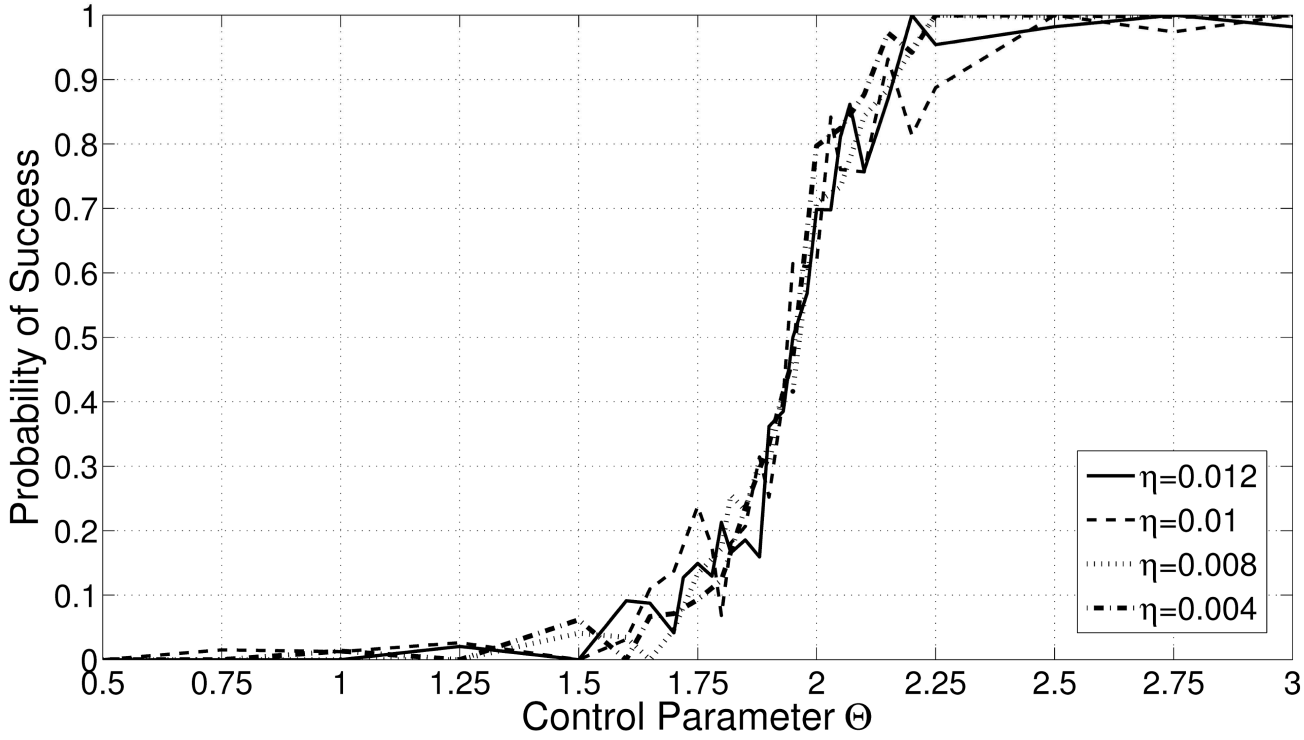
See where **phase transition for support recovery** happens as a function of

$$\Theta = \frac{\eta n}{s^3 \log\left((s+2r)p + r^2\right)}$$

for different values of

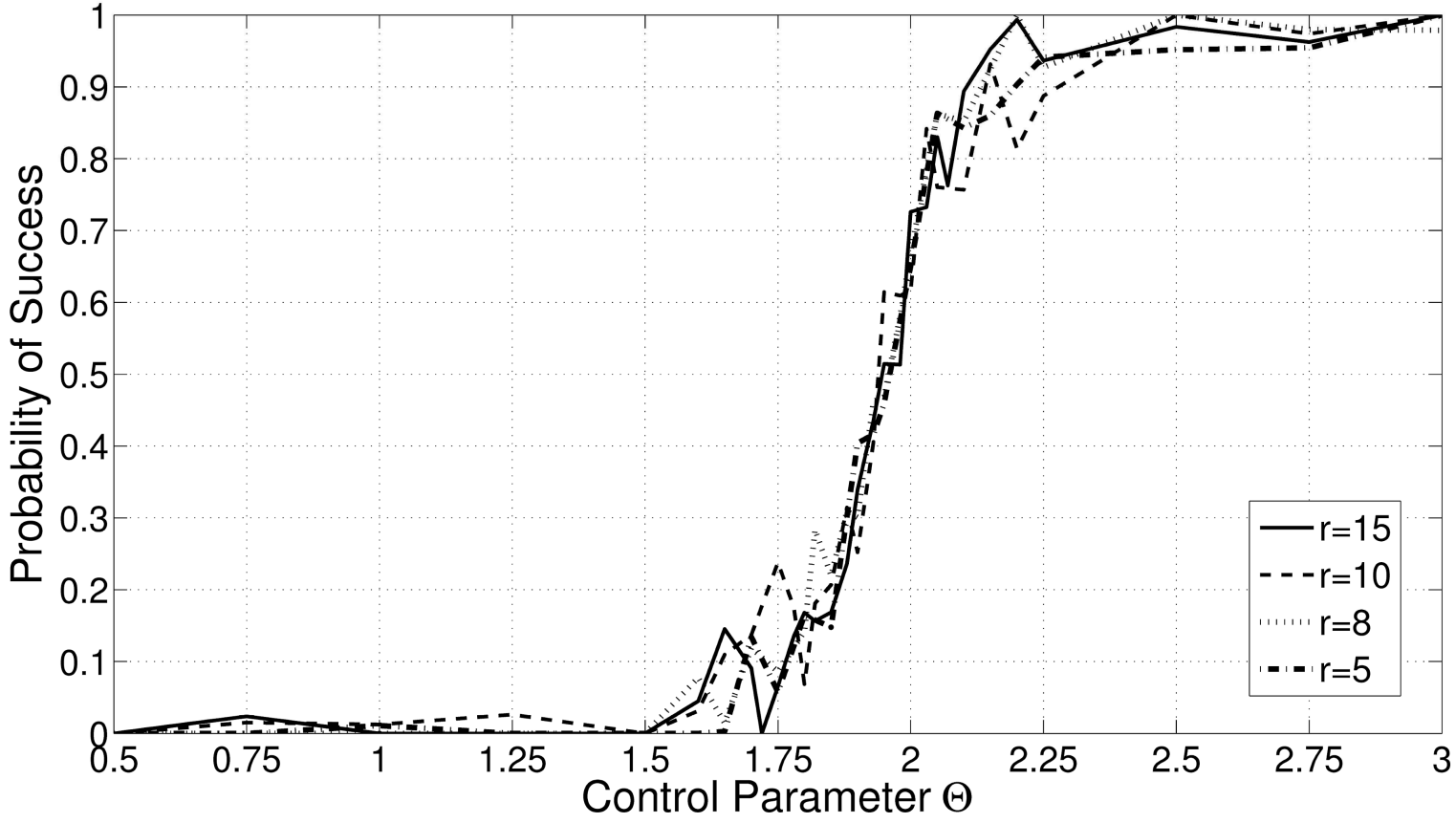
$$s, \eta, r, p$$

Effect of sampling rate η

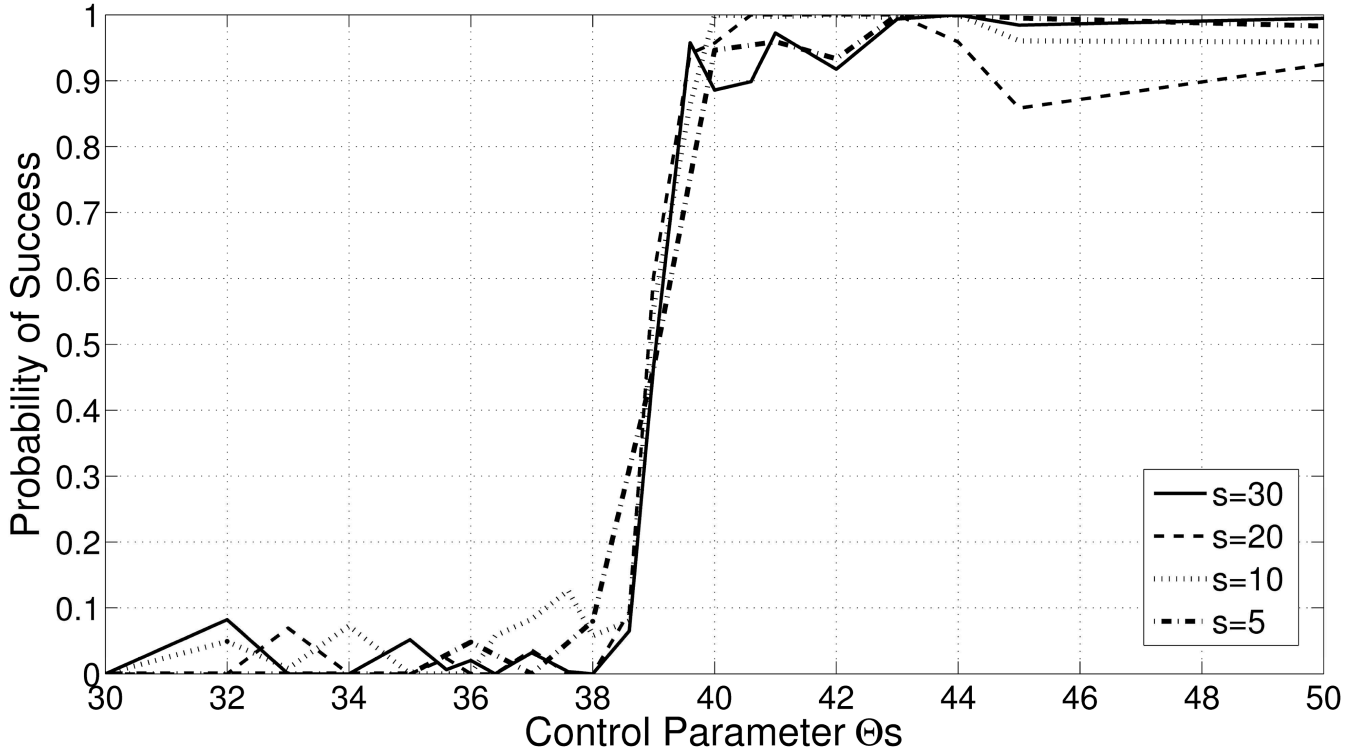


Fine-ness of sampling has no effect, once $\eta < \dfrac{1}{\sigma_{max}(\mathcal{A})}$

Effect of number of latent variables r



Effect of Sparsity of *A*



Empirically seems to be $T \approx s^2 \log p$

instead of $s^3 \log p$

Recovery of A vs recovery of L

[Chandrasekaran, Parrilo, Willsky] : learning hidden variables in Gaussian Graphical models

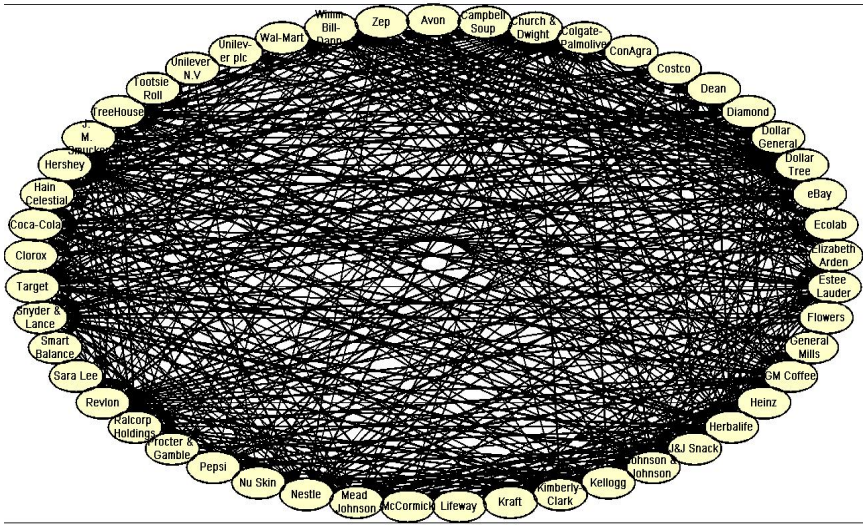
== inferring from samples an inverse covariance matrix that is S + L

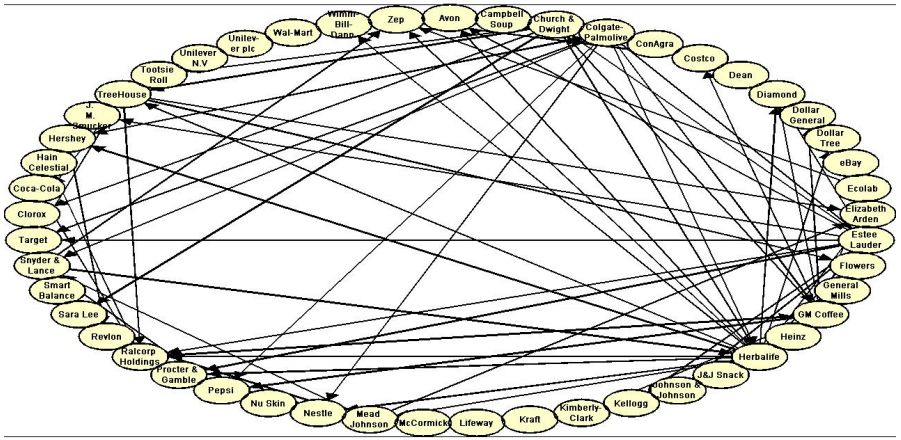
showed that $\Theta(p)$ sample complexity for recovery of rank(*L*)

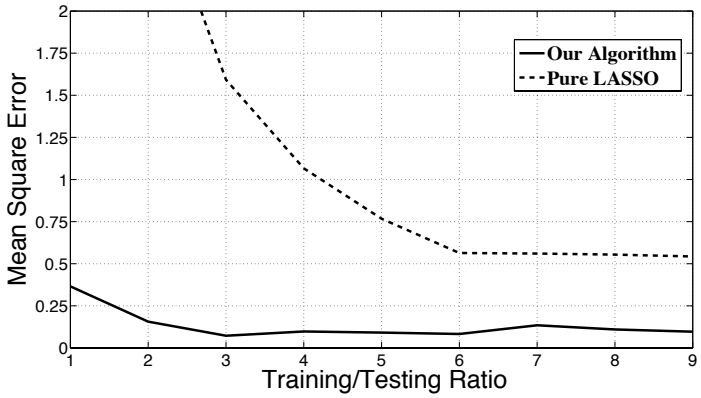
Our work: only $\Theta(s^3 \log p)$ sample complexity for support recovery of *A*

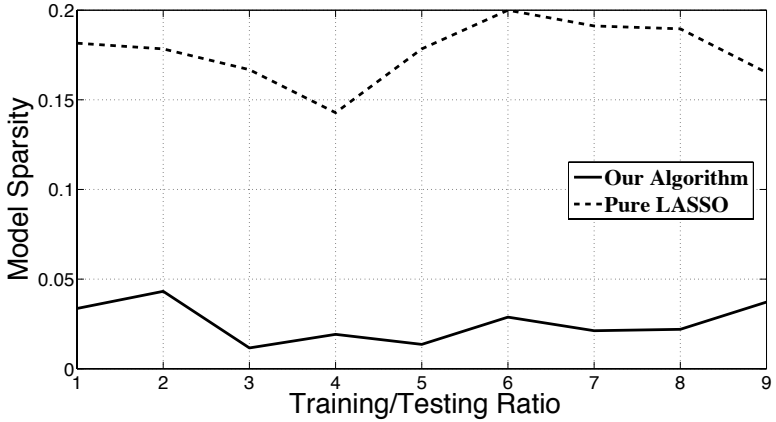
Penalty of "not knowing the basis" of the low-rank matrix ?

Stock Market Data

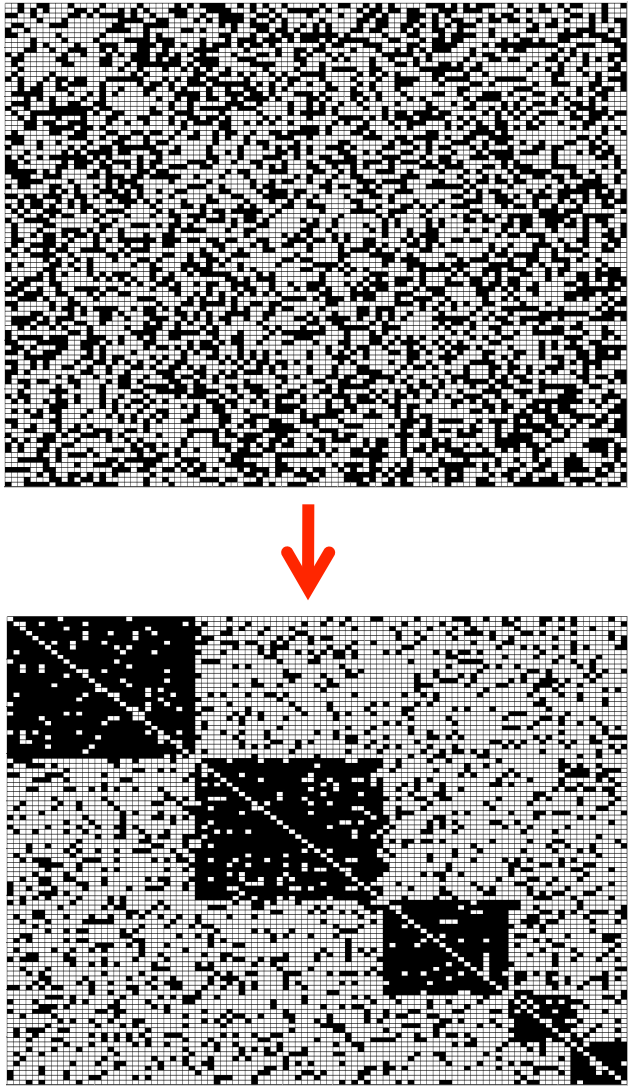








Clustering Partially-observed Graphs



Given : **partially observed graph**

(i.e. for most node pairs i , j, do not know if they are connected or not)

(combinatorial) Objective: Find clustering that minimizes number of **disagreements**

More on this in the next talk …

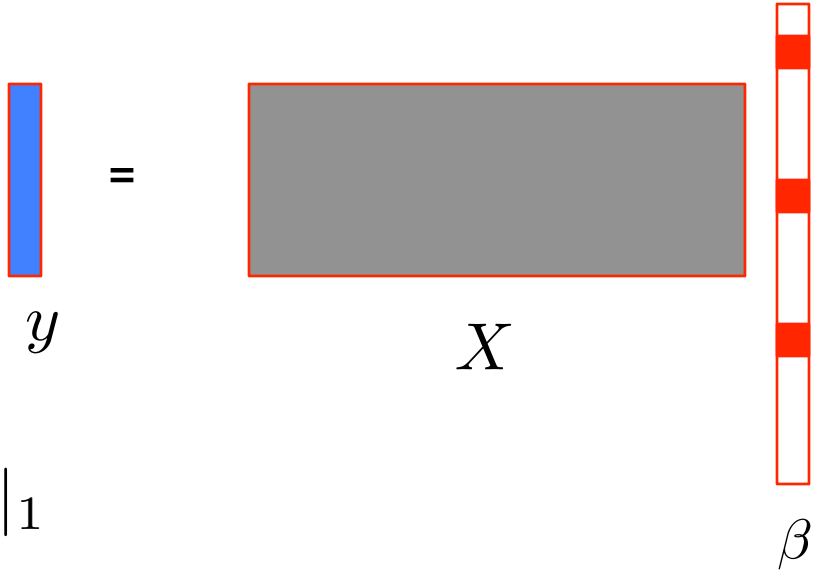
Sparse + Block-sparse

A Dirty Model for Multi-task Learning
in **NIPS 2010**
w/ A. Jalali, P. Ravikumar , C. Ruan

(Linear) Sparse Model Selection

Task: Find a (sparse) vector from a small number of linear measurements

Convex Optimization approach:
LASSO / Compressed Sensing



$$\min_x \|y - Ax\|^2 + \lambda\|x\|_1$$

Useful for a very broad and fast-growing range of applications, e.g.

- MRIs
- interpreting DNA microarrays,
- wideband spectrum monitoring,
- …

Multiple Sparse Model Selection

Linear measurements of **several** unknown sparse vectors

Arises in:

- DNA analysis of related individuals

- whenever there is motion during measurement

Q: Can one use overlap to improve estimate (i.e. use fewer samples) ?

A: Prior work: depends on (unknown) level of sharing

Method: block-regularization : $\min_X \|Y - AX\|_2^2 + \lambda \|X\|_{1,\infty}$

where $\|X\|_{1,\infty} = \sum_i \|X_i\|_\infty$

[Negahban-Wainwright]

Our Method

$$X = B + S$$

Block-sparse Sparse

Solve $$\min_{B,S} \|Y - A(B+S)\|_2 + \gamma\|B\|_{1,\infty} + \lambda\|S\|_1$$

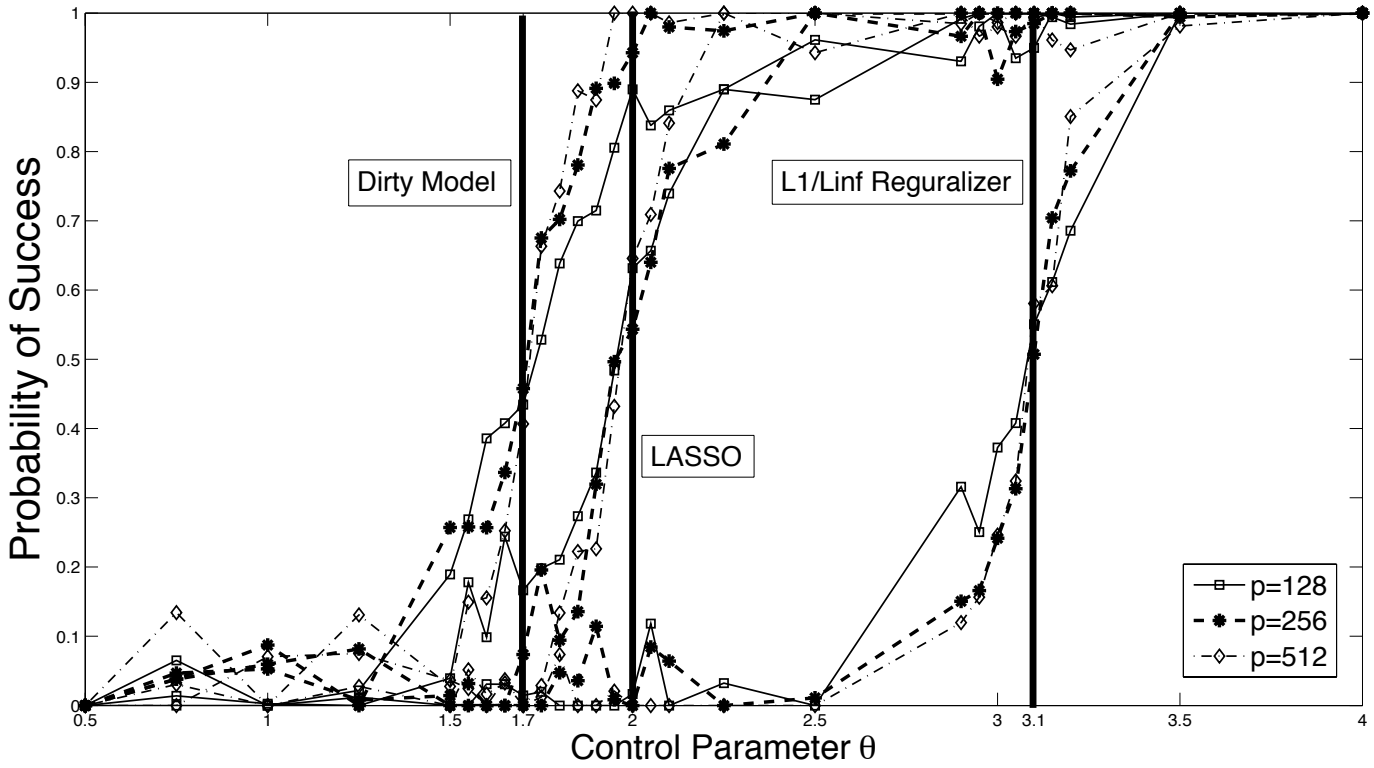
(where parameters chosen by cross-validation)

Success == $supp(X^*) = supp(\hat{B} + \hat{S})$ (and also close in values)

Performance : Gaussian case

Success = correct identification of support for all tasks

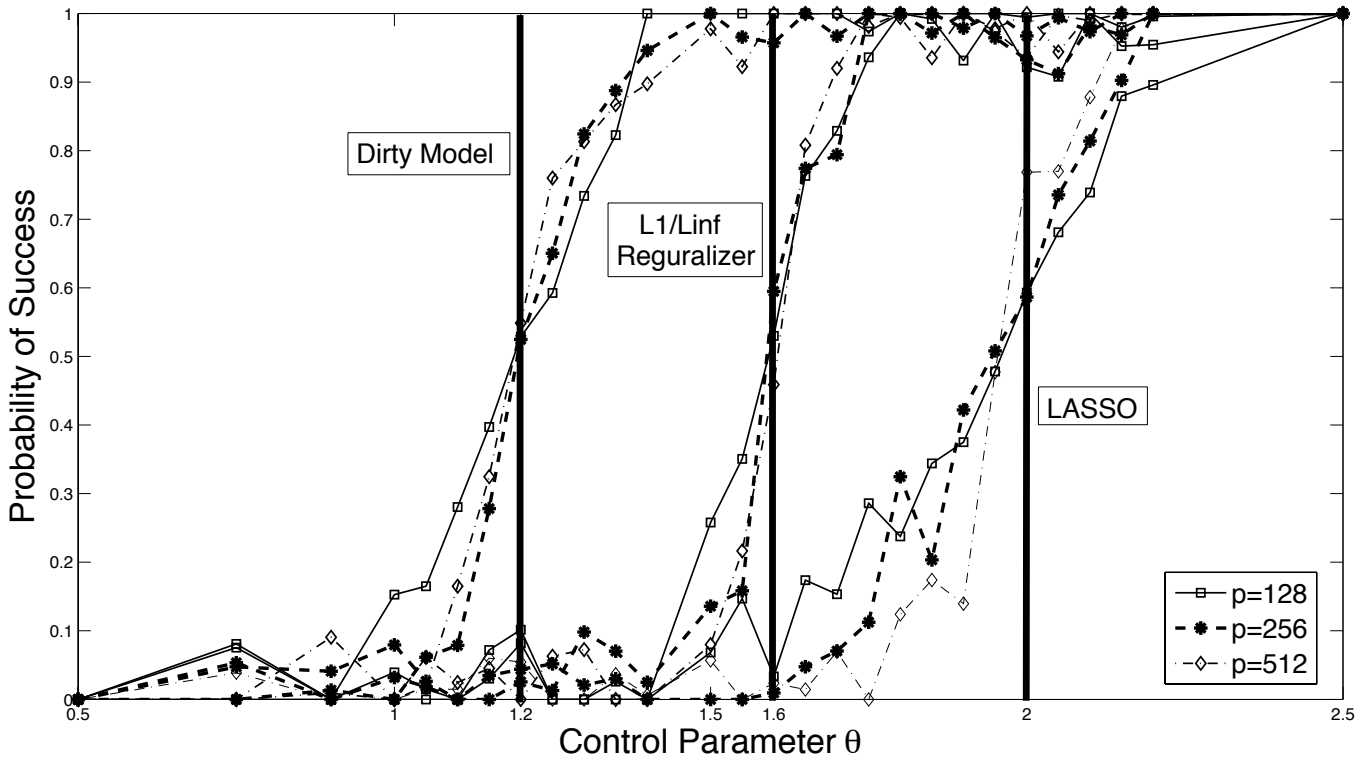
When tasks share a fraction α = 1/3 of their supports



(rescaled) # of Samples $\frac{n}{s \log p}$

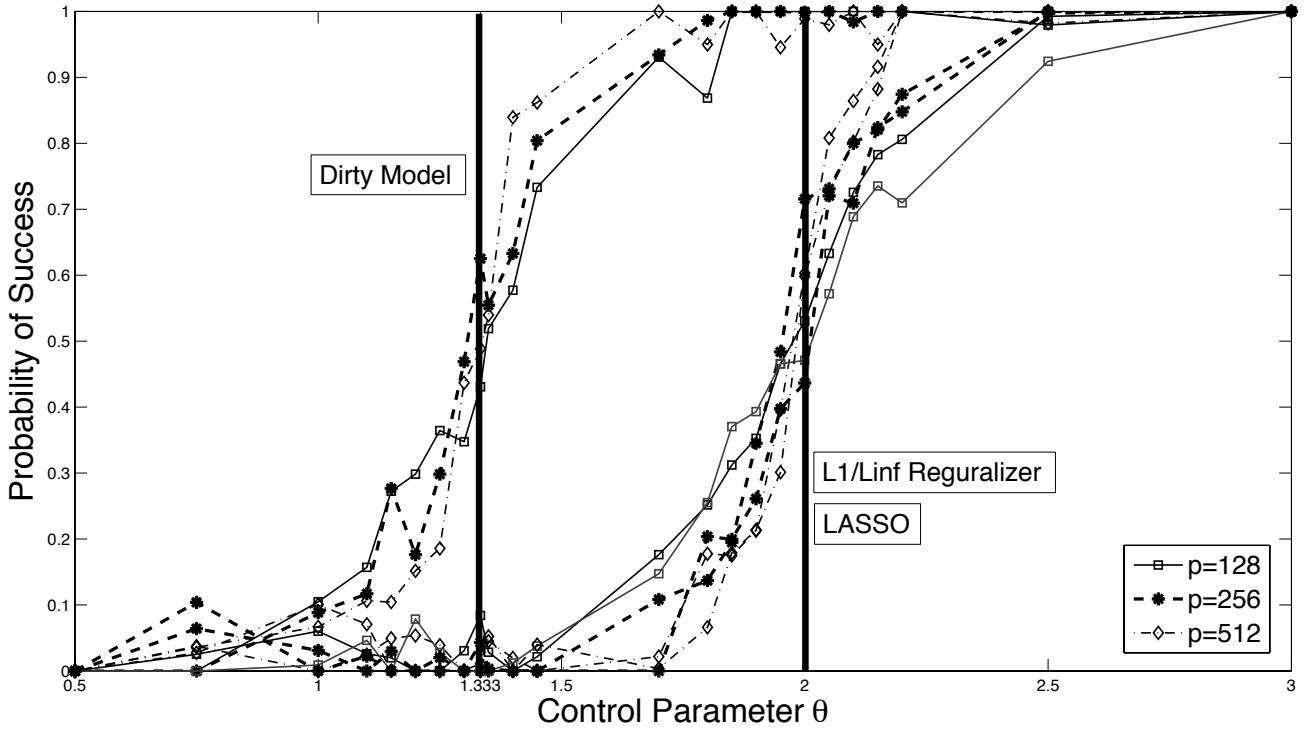
Performance

.. when each pair shares fraction α = 4/5 of their supports …



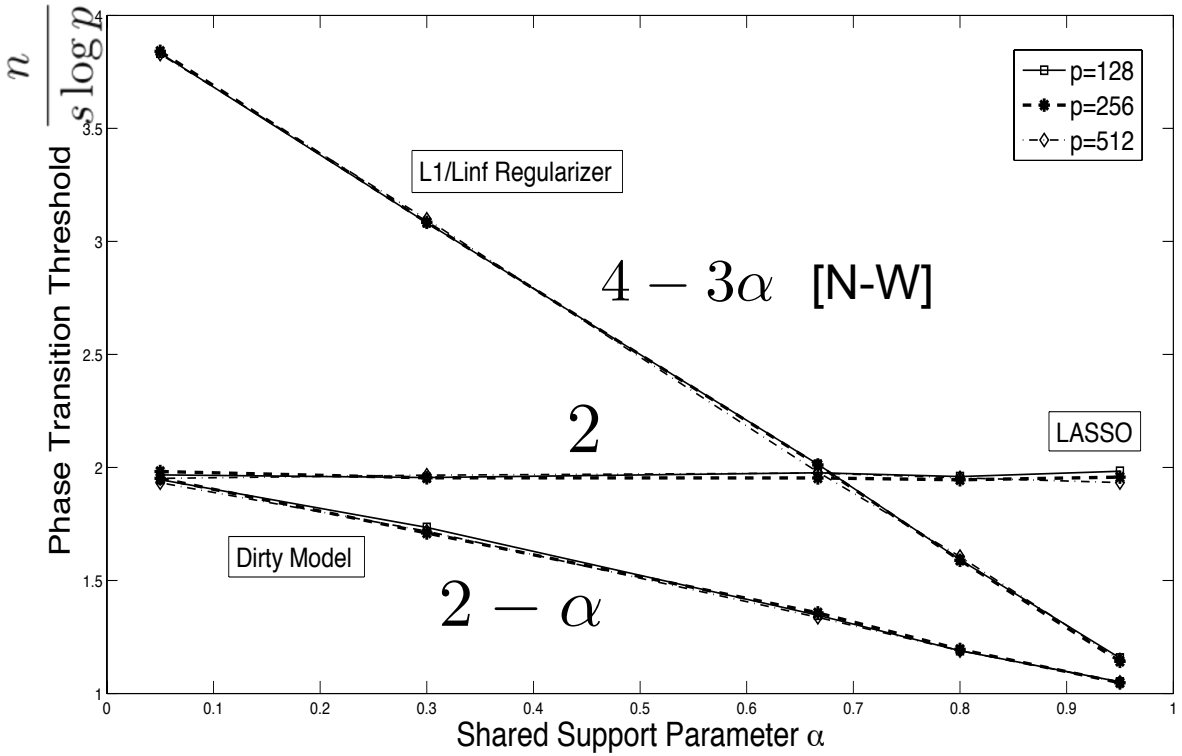
Performance

… and when they share α = 2/3 of their supports.



Performance: Theory + practice

In fact, our method *provably* outperforms both LASSO and ℓ_1/ℓ_∞ every time.



Furthermore, this matches **exactly** with theory

(down to exact constants)

Similar gains for real datasets in classification, prediction etc.

Conclusion

Usage of more than one structural model allows for increased flexibility and robustness

- often without too much extra computational overhead

- leveraging the computational tractability of the component models

Robustness = separation of one structure from another

- needs additional incoherence assumptions between objects

Flexibility = better recovery of an overall superposed structure

- does not need incoherence

Huge range of applications

Ongoing

1. Extensive empirical validation using more real datasets
2. Lower-complexity methods (e.g. alternating projections)
3. General theory
4. Applications (and interesting modifications they motivate)

Papers at

www.ece.utexas.edu/~sanghavi

(and NIPS ‘10, ICML ‘11, ISIT ’11, Arxiv)

Also:

we are looking for postdocs

Thanks !