Sparse Inverse Covariance Estimation with Hierarchical Matrices

Jonas Ballani

MATHICSE-ANCHP
EPF Lausanne

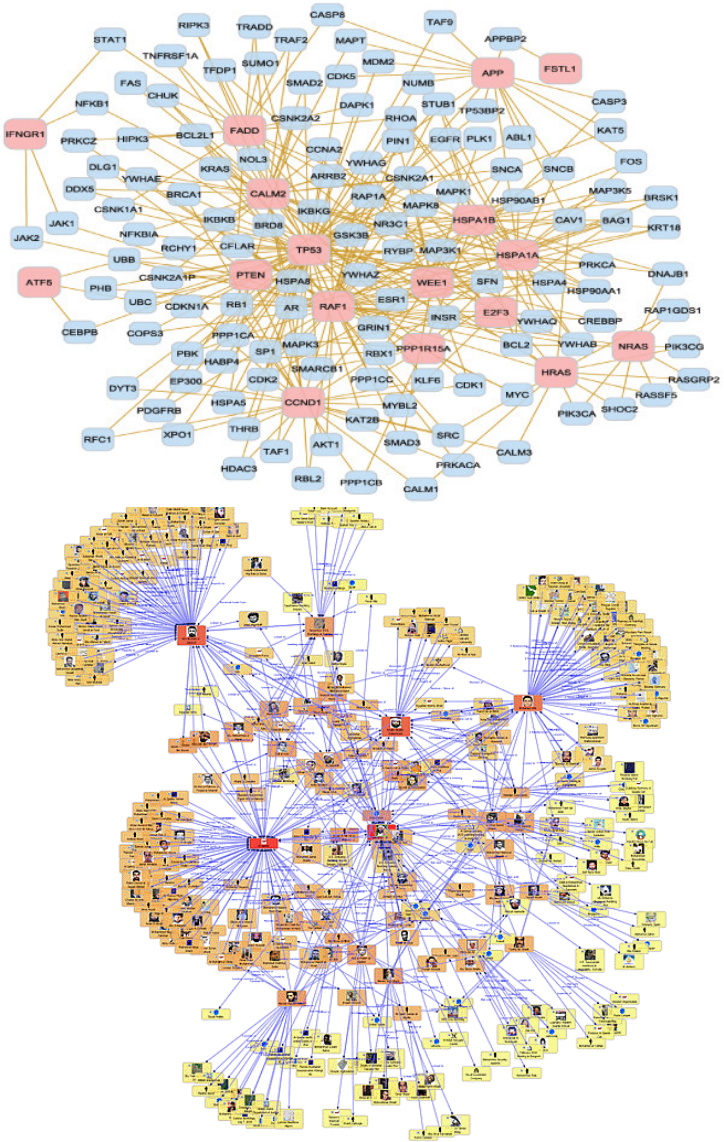
(joint work with Daniel Kressner)

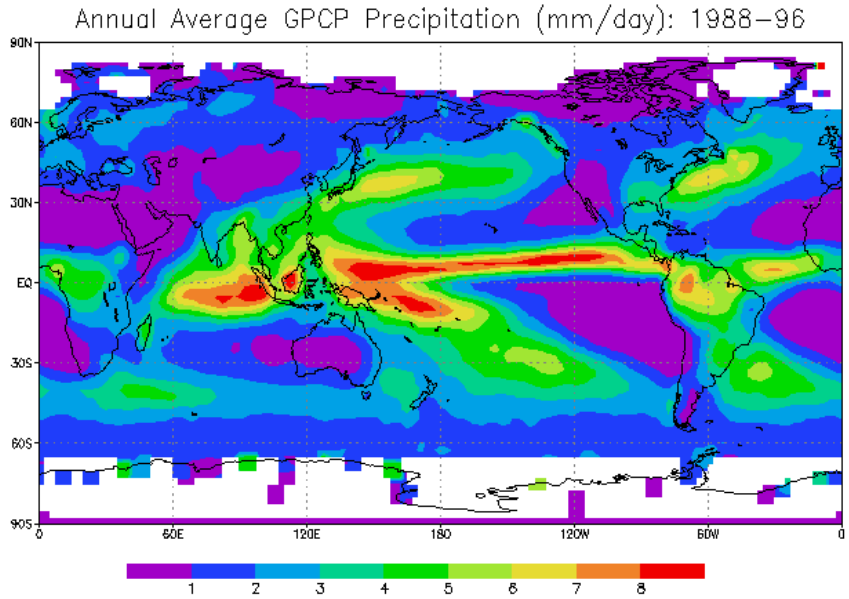
Disentis, 13-15 August 2014

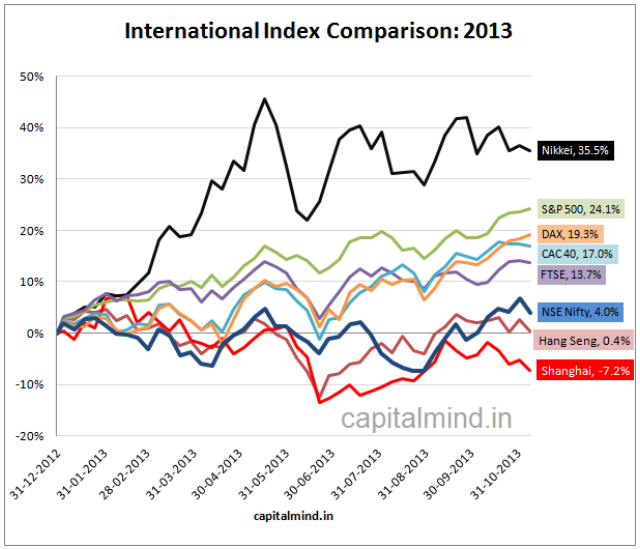
Overview

1. Covariance estimation
2. QUIC algorithm
3. Hierarchical matrices
4. Numerical examples

Covariance Estimation







Problem Setting

- given: n i.i.d. samples $y_1, \ldots, y_n$ drawn from a p-variate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$
- density:

$$f(x) = \frac{1}{\sqrt{(2\pi)^p \det \Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right), \quad x \in \mathbb{R}^p$$

- goal: estimate $\Theta = \Sigma^{-1} \in \mathbb{R}^{p \times p}$
- task: maximise likelihood
- sample mean and sample covariance matrix:

$$\bar{y} := \frac{1}{n}\sum_{j=1}^{n} y_j, \qquad S := \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^\top$$

Maximum Likelihood Estimator

- likelihood:

$$\mathcal{L}(\Sigma, \mu; y) = \prod_{i=1}^{n} f(y_i \mid \Sigma, \mu)$$

$$= \frac{1}{\sqrt{(2\pi)^{pn}(\det \Sigma)^n}} \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^\top \Sigma^{-1}(y_i - \mu)\right)$$

- log-likelihood:

$$\log \mathcal{L}(\Sigma, \mu; y) = \frac{n}{2}\log(\det \Sigma^{-1}) - \frac{n}{2}\operatorname{tr}(\Sigma^{-1}S)$$
$$- \frac{n}{2}(\bar{y} - \mu)^\top \Sigma^{-1}(\bar{y} - \mu) \quad + \text{const.}$$

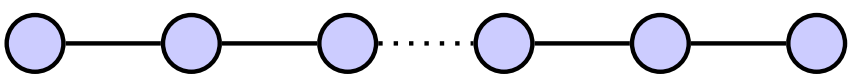
- $\log \mathcal{L}$ is maximal if we estimate μ by $\hat{\mu} = \bar{y}$ and $\Theta = \Sigma^{-1}$ by

$$\hat{\Theta} = \arg\min_{X \succ 0}\{-\log\det X + \operatorname{tr}(SX)\}$$

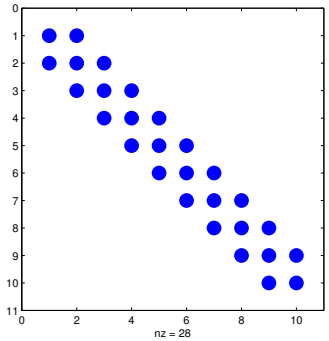
- in high dimensions: $n < p$ and S is singular

Sparsity Prior

- assumption: $\Theta = \Sigma^{-1}$ is sparse
- $\Theta_{ij} = 0 \Leftrightarrow y_i$ and y_j are cond. indep. given all other variables
- structure: Gaussian Markov Random Field represented by graph $G = (V, E)$
- vertices V: variables
- edges E: non-zero entries in Θ
- example: chain graph



$\Theta =$



ℓ_1 regularisation

- add ℓ_1 penalty to promote sparsity
- for $\lambda > 0$, minimise

$$f(X) := -\log\det X + \operatorname{tr}(SX) + \lambda \|X\|_1, \qquad X \succ 0,$$

  where $\|X\|_1 := \sum_{i,j} |X_{ij}|$
- non-smooth but convex problem, p^2 parameters
- first-order methods:
  - block coordinate descent: COVSEL (Banerjee et al 2008), GLASSO (Friedman et al, 2007)
  - alternating linearisation method ALM (Scheinberg et al, 2010)
  - many others: SINCO, PSM, VSM, ...
- second-order methods:
  - projected quasi-Newton method (Schmidt et al, 2009)
  - inexact interior point method IPM (Li et al, 2010)
  - Newton-Lasso-FISTA (Olsen et al., 2012)
  - QUIC, BigQUIC (Dhillon et al., 2011, 2013)

Quadratic Approximation

- aim: build quadratic model of $f(X) = -\log\det X + \operatorname{tr}(SX) + \lambda \|X\|_1$ at $X + \Delta$
- second-order expansion of log-determinant function:

$$\log\det(X+\Delta) \approx \log\det X + \operatorname{tr}(X^{-1}\Delta) - \tfrac{1}{2}\operatorname{tr}(X^{-1}\Delta X^{-1}\Delta)$$

- approximation of $g(X) := -\log\det X + \operatorname{tr}(SX)$ at $X + \Delta$:

$$g(X+\Delta) \approx \bar{g}_X(\Delta) := \operatorname{tr}((S-W)\Delta) + \tfrac{1}{2}\operatorname{tr}(W\Delta W\Delta) - \log\det X + \operatorname{tr}(SX)$$

with $W := X^{-1}$

- generalised Newton direction:

$$D = \arg\min_{\Delta} \bar{g}_X(\Delta) + \lambda \|X+\Delta\|_1$$

- for symmetric Δ: $\operatorname{tr}(W\Delta W\Delta) = \operatorname{vec}(\Delta)^\top (W \otimes W)\operatorname{vec}(\Delta)$
- structure of Hessian: $H = W \otimes W$

QUIC [Dhillon et al.]

main ideas:

- approximate Newton direction D by coordinate descent
- a single variable update for $(i,j) \in \mathcal{I} := \{1,\dots,p\}^2$ reads

$$\bar{\mu} = \arg\min_{\mu} \bar{g}_X(D + \mu(e_i e_j^\top + e_j e_i^\top)) + 2\lambda\,|X_{ij} + D_{ij} + \mu|$$

$$D_{ij} := D_{ij} + \mu$$

  and can be performed analytically by soft-thresholding

- exploit sparsity to reduce number of variables to update: split index set into $\mathcal{I} = \mathcal{I}_{\text{active}} \dot{\cup} \mathcal{I}_{\text{inactive}}$

$$(i,j) \in \begin{cases} \mathcal{I}_{\text{inactive}} & \text{if } |\nabla_{ij} g(X)| < \lambda \text{ and } X_{ij} = 0 \\ \mathcal{I}_{\text{active}} & \text{otherwise} \end{cases}$$

- perform updates only on $\mathcal{I}_{\text{active}}$
- observation: $\#\mathcal{I}_{\text{active}} = \mathcal{O}(\|X^*\|_0)$ $\quad (= \mathcal{O}(p)$ for sparse $X^*)$

QUIC (cont.)

- at each Newton step: sparse update

$$X^{(\ell+1)} := X^{(\ell)} + \alpha D^{(\ell+1)}, \quad X^{(0)} := Id$$

- step size α determined by Armijo-type rule s.t. $X^{(\ell+1)}$ remains positive definite
- computational bottlenecks:
  - evaluation of $\log\det X$, check of positive definiteness
  - inversion $W := X^{-1}$
- in BigQUIC (Dhillon): use CG to compute columns of W on demand
- our approach: exploit data-sparse properties of W given a suitable sparsity-structure of X
- examples:
  - if X has bandwidth k, all off-diagonal blocks of W have at most rank k
  - clustered data: if X is block-diagonal, W is block-diagonal
- general framework: hierarchical matrices

Hierarchical Matrices

- split $A \in R^{I \times I}$ hierarchically into subblocks $\sigma \times \tau \subset I \times I$
- represent *admissible* blocks in low-rank format:

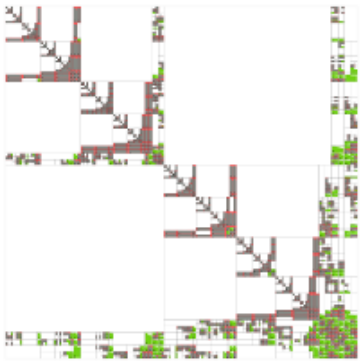
$$A|_{\sigma\times\tau} = UV^\top, \quad U \in \mathbb{R}^{\sigma\times k},\ V \in \mathbb{R}^{\tau\times k}$$

- store *non-admissible* blocks as dense matrices
- for sparse A, admissibility can be derived from purely algebraic information:

$$\min\{\mathrm{diam}(\sigma), \mathrm{diam}(\tau)\} \leq \eta\, \mathrm{dist}(\sigma,\tau), \quad \eta > 0$$

$$\mathrm{diam}(\sigma) := \max_{i,j\in\sigma} d_{ij}, \quad \mathrm{dist}(\sigma,\tau) := \min_{i\in\sigma, j\in\tau} d_{ij}$$

d_{ij} := length of shortest path connecting $i, j \in I$



- complexity: $\mathcal{O}(n \log^\alpha nk^\beta)$
- more details $\rightarrow$ Zürich Summer School 2014

Graph Learning: Evaluation Criteria

- compare learned with true correlations:

edges	true graph $\Theta = \Sigma^{-1}$	learned graph X^*
TruePositive	✓	✓
FalsePositive	✗	✓
FalseNegative	✓	✗

- typical measures:

$$Precision := \frac{TP}{TP + FP}$$

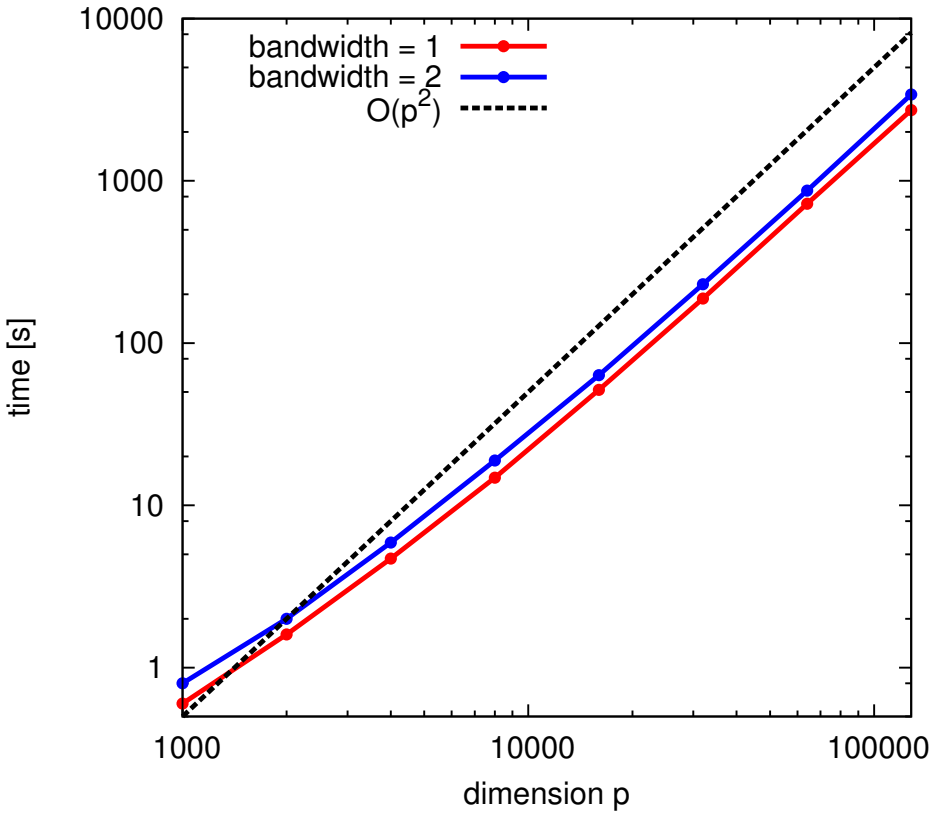
$$Recall := \frac{TP}{TP + FN}$$

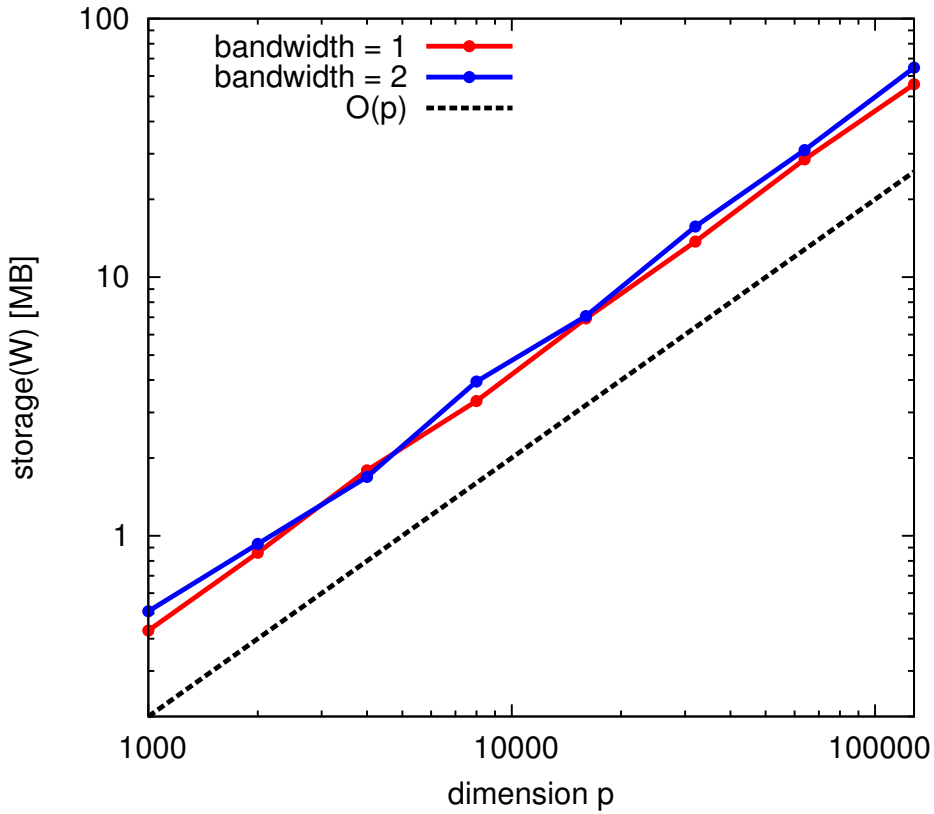
$$F - measure := \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

- *Precision* represents the fraction of correct edges in learned graph
- *Recall* represents the fraction of true edges present in learned graph

Numerical Examples: banded Σ^{-1}

- $n = 500$, $1000 \leq p \leq 128000$
- bandwidth 1: $\lambda = 0.5$, bandwidth $= 2$: $\lambda = 0.3$

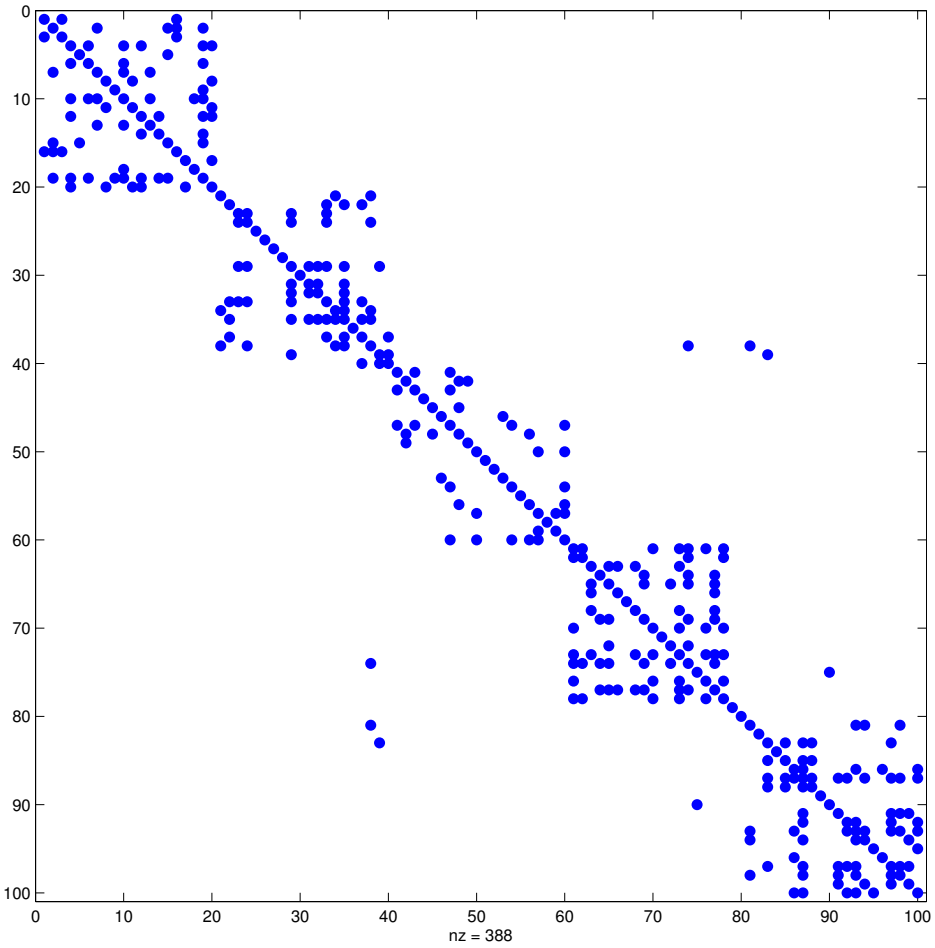




- F-measure for all p:
  ≈ 0.997 for bandwidth 1
  ≈ 0.978 for bandwidth 2

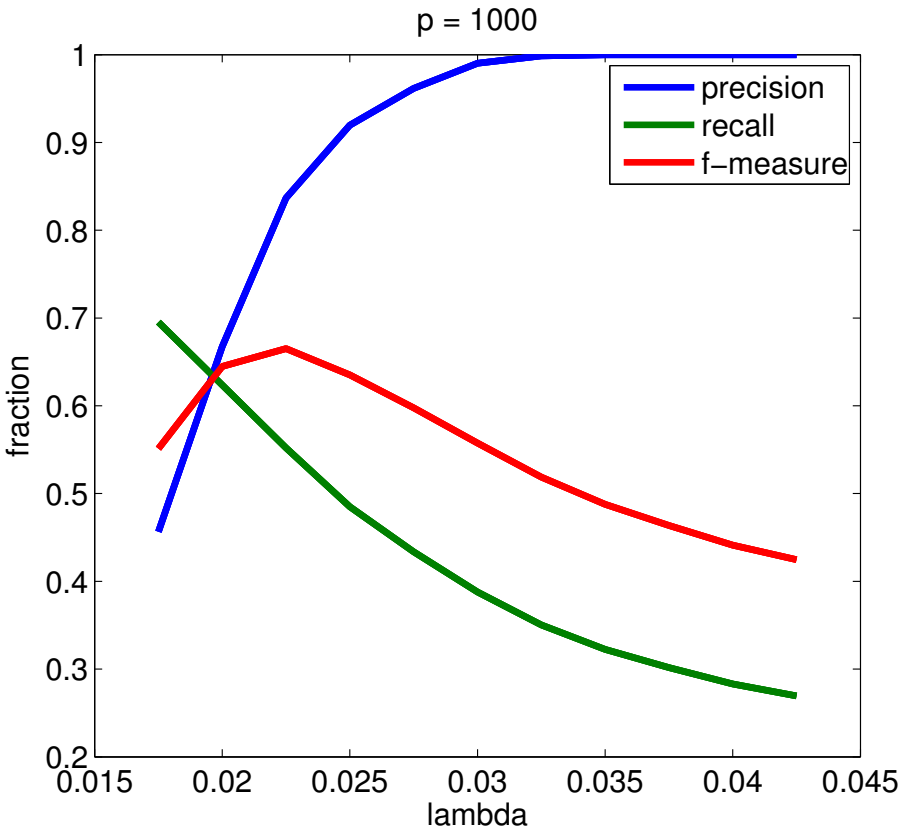
Numerical Examples: clustered Σ^{-1}

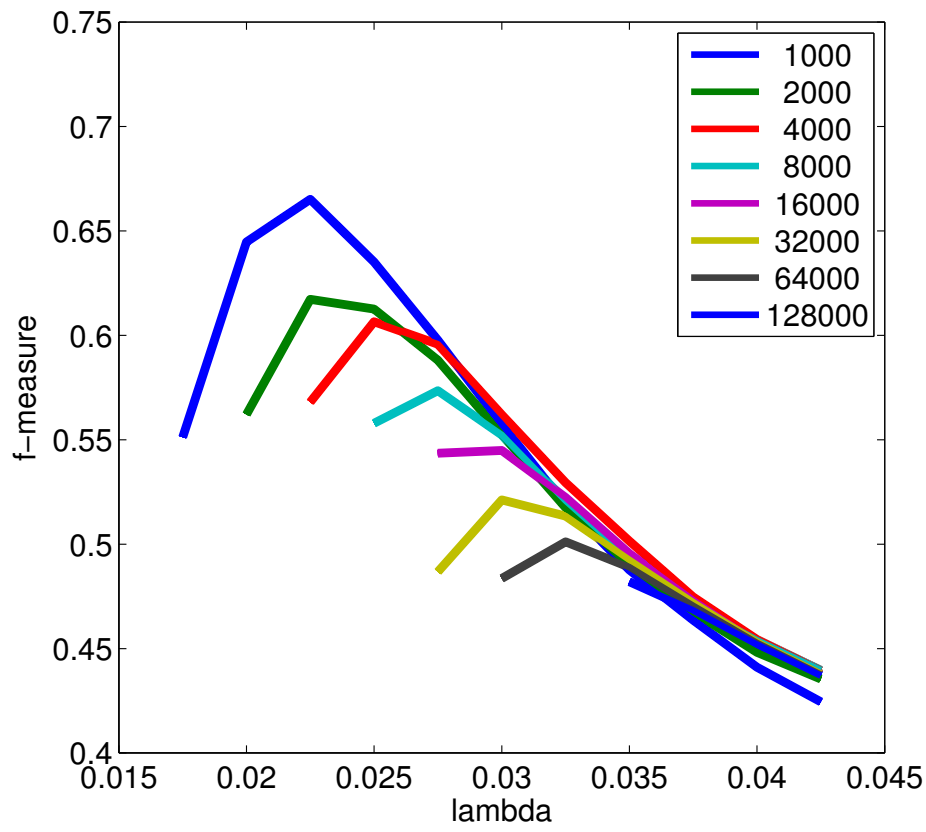
- $n = 500$, $1000 \leq p \leq 128000$
- clusters of size 20, average degree 4, 99% of edges within clusters



Numerical Examples: clustered Σ^{-1}

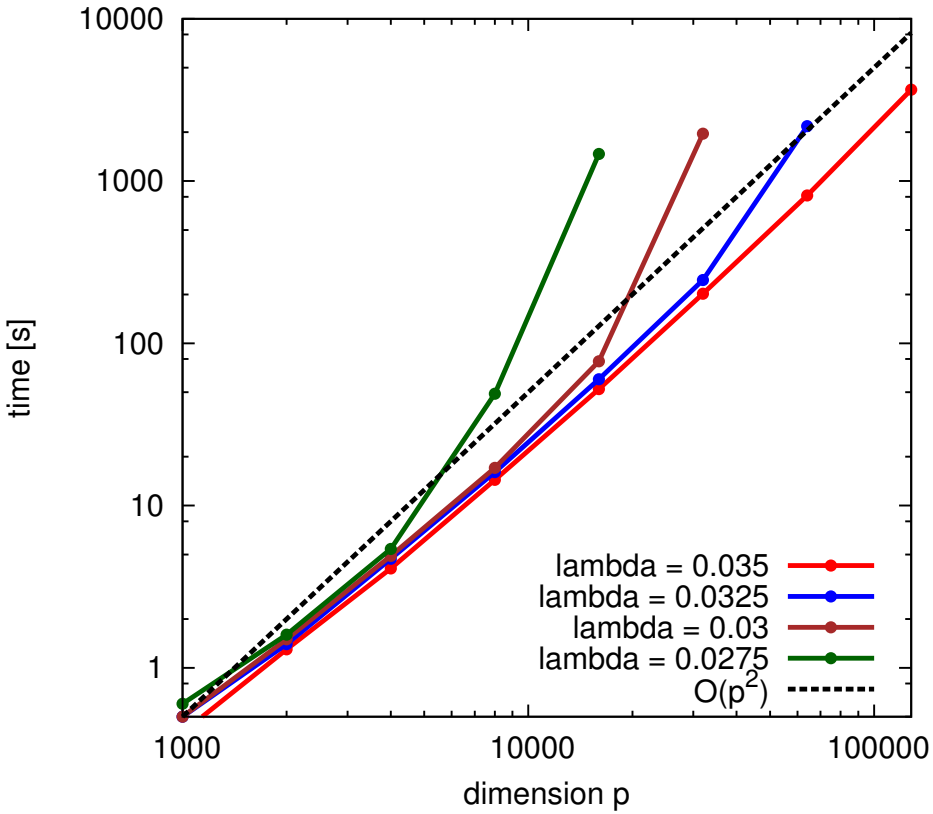
- $n = 500$, $1000 \leq p \leq 128000$
- clusters of size 20, average degree 4, 99% of edges within clusters
- graph statistics:

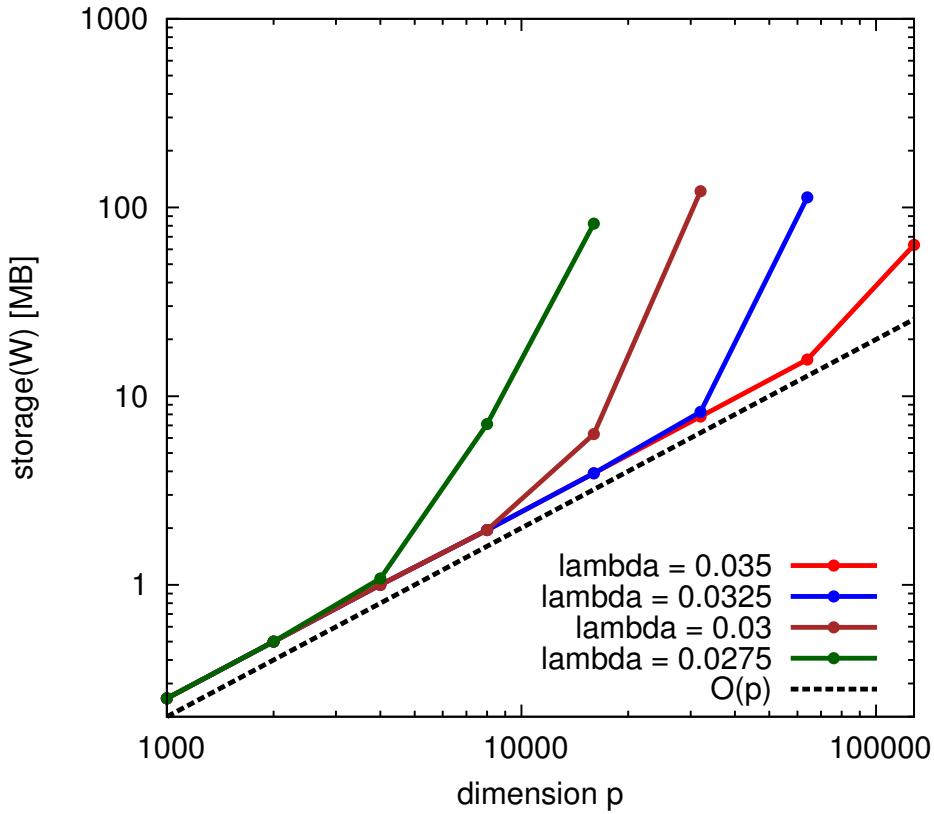




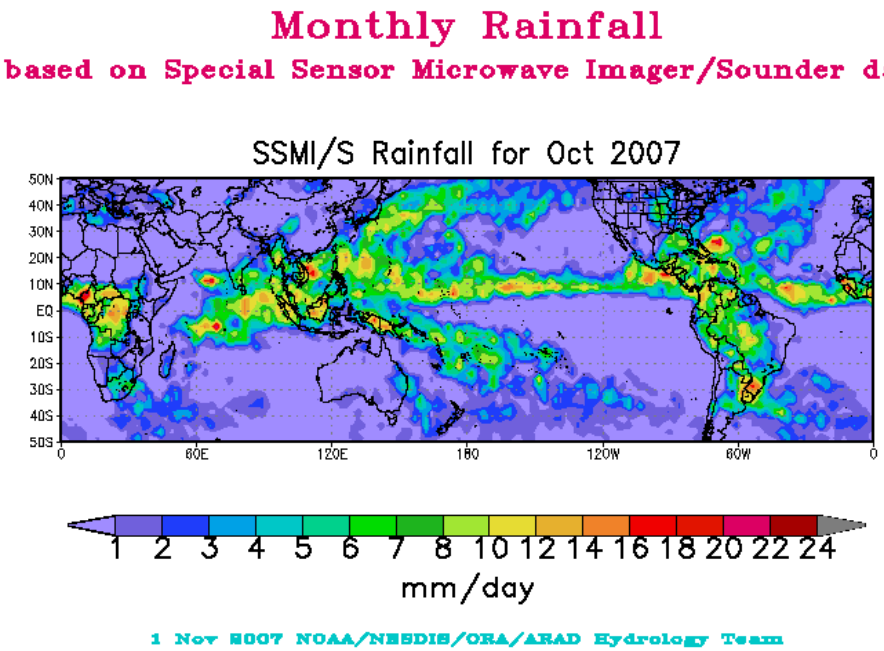
Numerical Examples: clustered Σ^{-1}

- $n = 500$, $1000 \leq p \leq 128000$
- clusters of size 20, average degree 4, 99% of edges within clusters
- performance:





Numerical Examples: Precipitation



monthly rainfall data in October of $n = 26$ years between 1987 and 2013,
grid: $p = 144 \times 72 = 10368$ points
size(full(W)) $\approx$ 820 MB

λ	iter	time [s]	size(W) [MB]	nnz(X)
0.05	8	17.4	2.44	10406
0.04	9	19.5	2.44	10548
0.03	9	20.2	2.59	11372
0.025	10	25.2	3.08	13202
0.02	16	62.8	4.54	19800
0.015	34	426.8	8.44	116818

Summary

- state of the art: combine second-order methods with sparsity
- our contribution: exploit (structured) sparsity by low-rank techniques

References:


C.-J. Hsieh, M. Susik, I. S. Dhillon, and P. Ravikumar
Sparse inverse covariance matrix estimation using quadratic approximation
In *Advances in Neural Information Processing Systems 24*, pages 2330–2338, 2011


C.-J. Hsieh, M. Susik, I. S. Dhillon, P. Ravikumar, and R. Poldrack
BIG & QUIC: Sparse inverse covariance estimation for a million variables.
In *Advances in Neural Information Processing Systems 26*, pages 3165–3173, 2013.