Inverse Statistical Problems - Sparse Regime

Luca Raffo
EPFL, Institute of Mathematics - luca.raffo@epfl.ch
Capital Fund Management - luca.raffo@cfm.com

June 2025

Contents

1 Introduction

Statistical models defined on high-dimensional systems often exhibit an underlying graphical structure that encodes conditional independence relations among variables.

Recovering this hidden structure from data is a central problem in statistics, machine learning, and statistical physics. This task, commonly referred

to as an *inverse statistical problem*, consists in reconstructing the parameters of the model (such as the precision matrix in Gaussian graphical models or the interaction couplings in Ising models) given a finite set of observations sampled from the underlying distribution.

In the continuous case, Gaussian graphical models (GGMs) provide a natural framework for modeling multivariate normal distributions with sparse inverse covariance (precision) matrices. The key property is that a zero entry in the precision matrix corresponds to a conditional independence relation: if $(\Theta)_{ij} = 0$, then variables i and j are independent given all the others. Hence, the sparsity pattern of the precision matrix is directly identified with the edge set of the underlying graph.

In the discrete case, the *inverse Ising problem* plays a similarly fundamental role. Here, the goal is to infer the coupling strengths and external fields of an Ising model from empirical spin configurations. As in GGMs, sparsity of the interaction matrix J has a clear structural interpretation: if $J_{ij} = 0$, then spins i and j are conditionally independent given all the remaining spins.

A common feature of both continuous and discrete models is the interest in *sparse regimes*, where only a small fraction of possible edges are present.

The purpose of this work is to present, propose, analyze, and compare different approaches to inverse statistical problems in sparse regimes. We discuss both continuous models (GGMs) and discrete models (Ising). Notice that the two models are linked by maximum entropy principles.

2 Continuous Case - Gaussian Graphical Models

In Gaussian graphical models we assume that our data $\mathcal{D} = \{s^{(1)}, ..., x^{(M)}\}, s^{(i)} \in \mathbb{R}^d$, comes from a multivariate normal $\mathcal{N}(\mu, \Sigma)$, such that the precision $\Theta = \Sigma^{-1}$ is *sparse*. In this scenario, covariates are our spins. Throughout we assume that $\mu = 0$ and $\Sigma \succ 0$.

We will denote our inference on Σ as $\hat{\Sigma}$ (and equivalently our inference on

Θ as $\hat{\Theta}$), and we want to promote inferences with $\hat{\Theta}$ sparse. We start by recalling some properties of gaussians.

Theorem 2.1. If $S \sim \mathcal{N}(0, \Sigma)$, then $\gamma^\top S \sim \mathcal{N}(0, \gamma^\top \Sigma \gamma)$ for any vector γ.

Theorem 2.2. If S and T are gaussians, then $S|T = t$ is gaussian for every vector $t \in \mathbb{R}^d$.

We say that there is an edge between i and $j \neq i$ if and only if $\Theta_{ij} \neq 0$. This defines a graph structure on our spins.

Theorem 2.3. The followings define the same graph structure:

1. There is an edge between i and j if and only if $\Theta_{ij} \neq 0$.
2. There is absence of edge between i and j if and only if i and j are independent when conditioned on everything else.
3. Conditional on its neighbours, i is independent of all other spins.
4. Two subsets of spins are conditionally independent given a different subset of spins that separates them.

Why is it useful to model this conditional dependence structure? First of all, it is interesting in itself. From a computational viewpoint, the multivariate gaussians distributions can be simplified under graphical models, as we will show.

We call the graph structure defined previously as $G = (V, E)$. A *clique* is a fully connected subset of V. A *maximal clique* is a clique that is not a strict subset of another clique. We say that a pdf f_S factorizes over a graph G if

$$f_S(s_1, ..., s_d) = \prod_{C \subseteq V} \psi_C(s_C)$$

for s^d interaction functions $\psi_C > 0$ such that $\psi_C = 1$ unless C is a clique. Under our definiteness assumption of Σ, we have the following.

Theorem 2.4. f_S factorizes over G if and only if the precision of f_S induces the structure of Theorem 2.3. This is known in literature as Hammersley-Clifford.

Now that we have described the properties of the objects we are going to work with, let us move to the inference.

This is done in two steps:

1. *structure estimation*, in which we infer the graph G, or equivalently we choose the support of the matrix $\hat{\Theta}$.
2. *covariance selection*, in which we fit $\hat{\Sigma}$, under the constraint that f_S factorizes over the graph induced by the support of $\hat{\Theta}$.

For *covariance selection*, we will always rely on the following theorem.

Theorem 2.5. Maximising the loglikelihood $l(\Sigma) = -\log|\Sigma| - \mathrm{tr}(\Sigma^{-1}\hat{\Sigma})$ over the set

$$\{\Sigma \succ 0 : e_i^\top(\Sigma^{-1})e_j = 0 \text{ whenever } (i,j) \notin \mathrm{E}\}$$

is equivalent to maximising the entropy $H(\Sigma) \propto \log|\Sigma| + \text{const.}$ over the set

$$\{\Sigma \succ 0 : e_i^\top \Sigma e_j = e_i^\top \hat{\Sigma} e_j \text{ whenever } (i,j) \in E \text{ or } i = j\}.$$

Intuitively: delete all non-adjacent (w.r.t. G) off-diagonal entries of $\hat{\Sigma}$. Then complete the missing entries to maximise the determinant.

The *structure estimation* is more difficult, and we will rely on different methods.

Thus, in our section on numerical experiments, the proxy will be:

1. use a method to select the graph G
2. use Theorem 2.4 to infer Σ subject to the constraints of G.

and we will compare the performances of different methods both on the ability to reconstruct the real graph G and to infer Σ.

2.1 Naive MLE

The most immediate approach to estimate the covariance structure of a Gaussian graphical model is to rely on the maximum likelihood estimator (MLE) without any sparsity constraints. In practice, this corresponds to computing the empirical covariance matrix

$$\hat{\Sigma} = \frac{1}{M} \sum_{i=1}^{M} s^{(i)} s^{(i)\top},$$

and setting the estimate of the precision as its inverse,

$$\hat{\Theta} = \hat{\Sigma}^{-1}. \tag{1}$$

Then we set a threshold $\gamma > 0$ and if $\hat{\Theta}_{ij} < \gamma$, we put $\hat{\Theta}_{ij} \leftarrow 0$.

This procedure is appealing for its simplicity, as it requires no additional regularization or structural assumptions.

However, it is well known that the naive MLE is problematic in high-dimensional settings: when the number of samples M is smaller than the dimension d, the empirical covariance $\hat{\Sigma}$ is not invertible, and even when $M > d$, the estimator $\hat{\Theta}_{\text{MLE}}$ is typically dense, thus failing to capture the sparse conditional independence structure we are interested in. Moreover, the computation itself can be demanding: inverting a $d \times d$ matrix requires $\mathcal{O}(d^3)$ operations in general.

For this reason, the naive MLE is best regarded as a baseline rather than a viable method in sparse regimes.

2.2 Graphical LASSO

A more effective approach to covariance estimation in sparse regimes is provided by the *Graphical LASSO* (GLASSO). This method addresses the two main limitations of the naive MLE: non-invertibility of the empirical covariance when $M < d$ and lack of sparsity in the estimated precision matrix.

The idea is to directly estimate the precision matrix Θ by solving the ℓ_1-regularized maximum likelihood problem

$$\hat{\Theta} = \arg\max_{\Theta \succ 0} \Big\{ \log\det\Theta - \operatorname{tr}(\hat{\Sigma}\Theta) - \lambda \|\Theta\|_1 \Big\}, \tag{2}$$

where $\|\Theta\|_1 = \sum_{i,j} |\Theta_{ij}|$ denotes the entrywise ℓ_1 norm and $\lambda > 0$ is a regularization parameter controlling the sparsity of the solution.

The penalty term $\lambda\|\Theta\|_1$ encourages many entries of $\hat{\Theta}$ to be exactly zero, thus promoting a sparse graphical structure. Furthrmore we can promote even more sparsity by setting a threshold $\gamma > 0$ and following the same procedure of the previous method.

2.3 CLIME and ACLIME

Another family of methods for sparse precision matrix estimation is based on linear constraints rather than direct penalization of the likelihood. The *Constrained ℓ_1 Minimization Estimator* (CLIME), recasts the estimation of Θ as a set of linear programs.

The idea is to approximate the inverse covariance $\Theta = \Sigma^{-1}$ by solving

$$\hat{\Theta} = \arg\min_{\Theta \in \mathbb{R}^{d \times d}} \|\Theta\|_1 \quad \text{subject to} \quad \|\hat{\Sigma}\Theta - I\|_\infty \leq \lambda, \tag{3}$$

where $\|\Theta\|_1$ denotes the elementwise ℓ_1 norm and $\|\cdot\|_\infty$ is the entrywise maximum norm. The constraint $\|\hat{\Sigma}\Theta - I\|_\infty \leq \lambda$ enforces the approximate inverse relationship, while the ℓ_1 minimization promotes sparsity. Furthrmore we can promote even more sparsity by setting a threshold $\gamma > 0$ and following the same procedure of the previous method.

This formulation decomposes column-wise, making the method computationally tractable via linear programming solvers.

Importantly, CLIME does not require $\hat{\Sigma}$ to be invertible, which is advantageous when $M < d$. While CLIME provides consistency guarantees, it may sometimes lead to asymmetric estimates of Θ; symmetrization (e.g., by averaging with its transpose) is therefore typically applied in practice.

A refinement of CLIME is the *Adaptive* CLIME (ACLIME). The central idea is to replace the single global tuning parameter λ with entry-specific thresholds that adapt to the local variability of the data. In this way, ACLIME avoids the shortcomings of a universal constraint, which tends to over-shrink entries with low variability and under-regularize those with high variability.

The procedure consists of two stages:

1. For each column j, one solves a columnwise linear program with a relatively loose constraint in order to obtain rough estimates of the diagonal entries $\hat{\theta}_{jj}$ of the precision matrix. These diagonals approximate the residual variances associated with column j and will serve as a normalization factor in the next stage.

2. Using the diagonal estimates from Stage 1, the universal constraint in (3) is replaced by entry-specific bounds of the form

$$\left|(\hat{\Sigma}\Theta - I)_{ij}\right| \;\le\; \lambda_{ij}, \qquad \lambda_{ij} = \delta \sqrt{\frac{\hat{\sigma}_{ii}\,\hat{\theta}_{jj}}{M}},$$

   where $\hat{\sigma}_{ii}$ is the i-th diagonal entry of the empirical covariance matrix $\hat{\Sigma}$, $\hat{\theta}_{jj}$ is the preliminary estimate from Stage 1, M is the sample size, and $\delta > 0$ is a universal constant (typically selected by cross-validation). The constant δ controls the overall confidence level of the entrywise constraints.

By scaling each constraint proportionally to its own sampling variability, ACLIME yields tighter bounds when both $\hat{\sigma}_{ii}$ and $\hat{\theta}_{jj}$ are small, and looser ones when they are large. This adaptive rescaling improves finite-sample performance, both in estimation accuracy and in support recovery, while preserving minimax optimality in high-dimensional regimes.

2.4 Graphical Score Matching LASSO

Another approach to sparse precision matrix estimation is based on *score matching*, originally introduced by Hyvärinen (2005) as an alternative to maximum likelihood in continuous exponential families.

For a Gaussian model with density

$$p_\Theta(x) \;\propto\; \exp\Big(-\tfrac{1}{2}x^\top \Theta x\Big),$$

the score matching loss can be written explicitly in terms of the precision matrix Θ as

$$\mathcal{L}_{\mathrm{SM}}(\Theta) \;=\; \tfrac{1}{2}\mathrm{tr}(\Theta\hat{\Sigma}\Theta) - \mathrm{tr}(\Theta),$$

where $\hat{\Sigma}$ denotes the empirical covariance matrix.

To encourage sparsity in the estimated precision matrix, we add an ℓ_1 penalty on the off-diagonal entries of Θ, leading to the *Graphical Score Matching LASSO* estimator:

$$\hat{\Theta} = \arg\min_{\Theta \succ 0} \left\{ \tfrac{1}{2}\mathrm{tr}(\Theta\hat{\Sigma}\Theta) - \mathrm{tr}(\Theta) + \lambda\|\Theta\|_1 \right\}, \tag{4}$$

where $\|\Theta\|_1 = \sum_{i,j} |\Theta_{ij}|$ is the entrywise ℓ_1 norm and $\lambda > 0$ is a regularization parameter controlling the sparsity level.

Furthrmore we can promote even more sparsity by setting a threshold $\gamma > 0$ and following the same procedure of the previous method.

Compared with Graphical LASSO, this method avoids the evaluation of log-determinants and is based on a quadratic objective in Θ, which can be more convenient computationally.

2.5 Nodewise Graphical LASSO

An alternative regression-based method for sparse precision matrix estimation is the *Nodewise Graphical LASSO*. This approach is based on the observation that the conditional distribution of each variable X_j given all the others X_{-j} is Gaussian, with linear conditional mean and constant conditional variance:

$$X_j \mid X_{-j} \sim \mathcal{N}\left(-\frac{1}{\Theta_{jj}} \sum_{k \neq j} \Theta_{jk} X_k,\ \frac{1}{\Theta_{jj}} \right).$$

Hence, recovering the neighbourhood of node j reduces to solving a linear regression of X_j against the remaining variables X_{-j}, with coefficients

$$\beta_{jk} = -\frac{\Theta_{jk}}{\Theta_{jj}}.$$

In practice, for each node j, one solves the LASSO problem

$$\hat{\beta}^{(j)} = \arg\min_{\beta \in \mathbb{R}^{d-1}} \left\{ \tfrac{1}{2M}\|X_j - X_{-j}\beta\|_2^2 + \lambda\|\beta\|_1 \right\},$$

which yields sparse estimates of the conditional dependencies of X_j. The nonzero coefficients of $\hat{\beta}^{(j)}$ directly identify the neighbours of node j in the underlying graph.

Once all nodewise regressions are performed, the corresponding estimates of the precision matrix entries are obtained by rescaling with the estimated residual variances. Symmetrization (e.g. averaging $\hat{\Theta}_{jk}$ and $\hat{\Theta}_{kj}$) is then applied to ensure a valid symmetric precision matrix. Finally, a threshold $\gamma > 0$ may be imposed to promote additional sparsity.

2.6 Numerical experiments

In our numerical experiments we set the number of spins at 50. We generate a sparse precision matrix with the following iter, which reflects a Erdos-Renyi random graph structure:

1. We fix a parameter $p \in (0.0, 1.0)$ (in our case we use $p = 0.2$).
2. For each edge, we assign zero with probability $1 - p$.
3. On the edges which are not zero, we assign a random number between 0.5 and 1.0.

For the time being, we set the means vector to zero.

The covariance is given by inverting the precision matrix numerically. When generating data, we further set a parameter $\beta \in (0.1, 1.0)$, so that our precision will be $\beta\Theta$, and our covariance $\frac{1}{\beta}\Sigma$.

We explore the performance of our algorihms at different values of β and given different number of samples. Each algorithm is run until convergence (until each update step is greater than a fixed tolerance threshold).

For each algorithm, for each choice of β and number of samples, we evaluate the performance with the AUC (Area Under the Curve). The AUC is computed from the ROC curve (Receiver Operating Characteristic curve). The ROC curve is obtained by plotting the *True Positive Rate* (TPR) against the *False Positive Rate* (FPR) as the decision threshold t varies.

Concretely, suppose each pair (i, j) is assigned a score s_{ij} by the algorithm, measuring how likely it is that an edge is present. By choosing a threshold $t \in \mathbb{R}$, we predict that an edge is present whenever $s_{ij} \geq t$. As t decreases from $+\infty$ to $-\infty$, more pairs are classified as edges, and we can trace the values of TPR and FPR:

$$\mathrm{TPR}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FN}(t)}, \qquad \mathrm{FPR}(t) = \frac{\mathrm{FP}(t)}{\mathrm{FP}(t) + \mathrm{TN}(t)},$$

where $\mathrm{TP}(t)$, $\mathrm{FP}(t)$, $\mathrm{TN}(t)$, and $\mathrm{FN}(t)$ denote the number of true positives, false positives, true negatives and false negatives obtained at threshold t.

The AUC is then defined as the area under the ROC curve:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(t)\, d\mathrm{FPR}(t),$$

which can be interpreted as the probability that a randomly chosen positive example receives a higher score than a randomly chosen negative one.

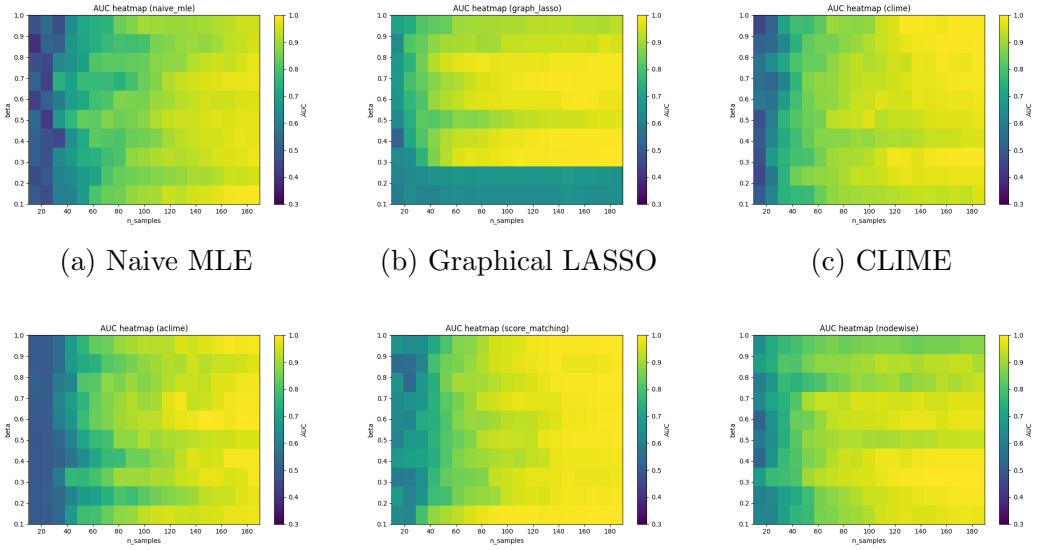
We show below the heatmap of our algoritmhs. On the x axis we let the number of samples used vary from 10 to 200; while on the y axis we let β vary from 0.1 to 1.0.

We show below in the table the time required for each run for a fixed couple $(\beta, \text{number of samples})$.

Naive MLE	Graph LASSO	CLIME	ACLIME	Score Matching	Nodewise
0.18 s	0.63 s	0.06 s	0.08 s	0.63 s	0.42 s

Table 1: Execution times (in seconds) for the six algorithms.

The comparison shows that Naive MLE and ACLIME are overall the worse. Graphical LASSO works badly at low temperature, because the regularization pushes too much towards zero and it cannot differentiate anymore between zeros and low values on the edges. Score matching gives the best results overall, but we highlight that CLIME is much faster since it only requires to solve a linear program.

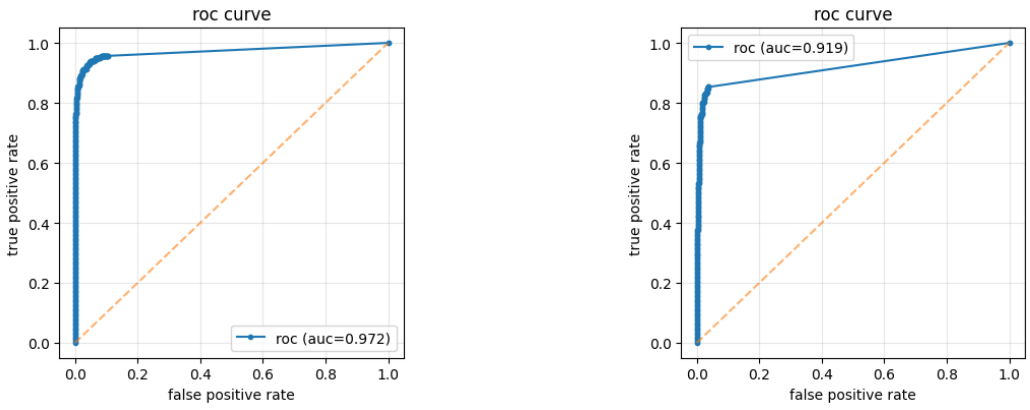


(a) Naive MLE (b) Graphical LASSO (c) CLIME

(d) ACLIME (e) Score Matching (f) Nodewise

Figure 1: Comparison of inverse GGM methods.

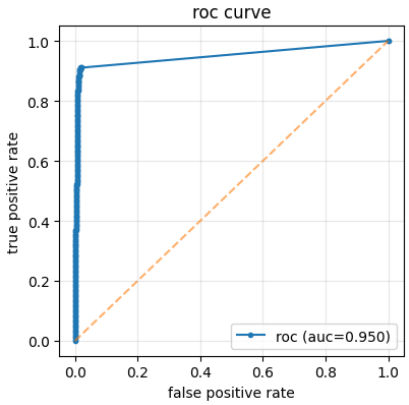
We plot below the ROC curve for the Graphical Score Matching LASSO and the CLIME, at 500 samples and $\beta = 1$.



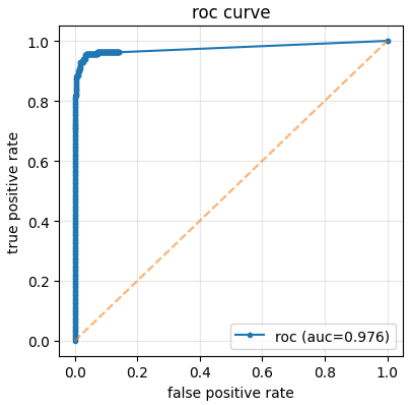
(a) Score Matching (b) CLIME

In order to better understand if they recover the same graph, we fix the graph structure recovered by Graphical Score Matching LASSO when the threshold is zero, and we look at the ROC curve of CLIME on this graph. We then do

the same after inverting the roles.

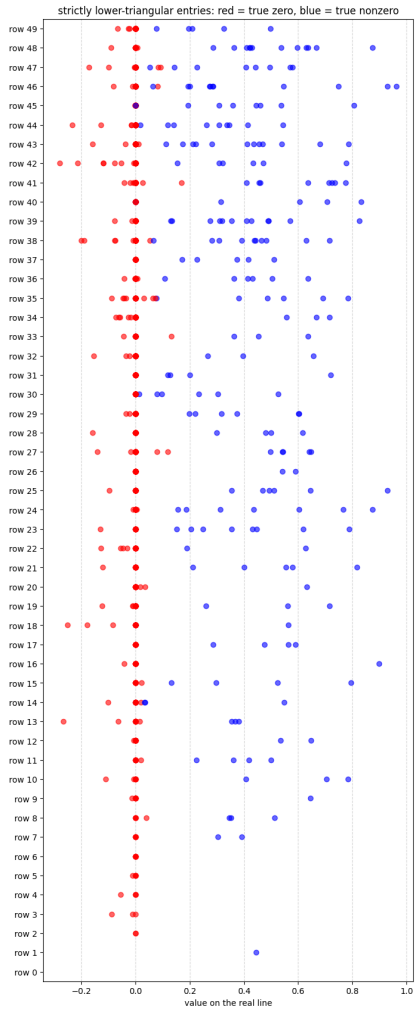


(a) CLIME ROC on Score Matching graph

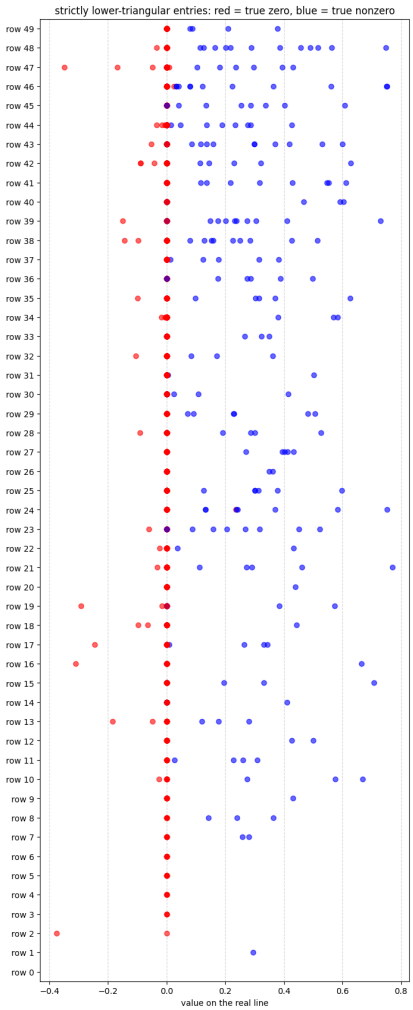


(b) Score Matching ROC on CLIME graph

In order to understand how conservative our reconstructinos are, we take a look at the inferred values of the precision for the Graphical Score Matching LASSO and the CLIME. Each line represents the values of a row of the ifnerred precision, and we color with red the values which should be zero.



(a) Graphical Score Matching precision



(b) CLIME precision

Finally, for both we calculate the frobenius norm of the reconstruction error of the covariance, after setting a threshold of $\gamma = 0.1$, and using the maximum entropy matrix completion theorem.

Score Matching	CLIME
0.31	0.29

Table 2: Reconstruction errors on the covariance.

3 Inverse Ising Problem

The inverse Ising problem consists in recovering the interaction network of a system of binary variables ("spins") from empirical observations. Formally, given samples $\{\sigma^{(k)}\}_{k=1}^{N} \subset \{-1,+1\}^n$ drawn from a Boltzmann distribution

$$\mathbb{P}_J(\sigma) \;\propto\; \exp\Big(\sum_{1\le i<j\le n} J_{ij}\,\sigma_i\sigma_j + \sum_{i=1}^{n} h_i\sigma_i\Big),$$

the goal is to reconstruct the unknown parameters $J = (J_{ij})$ (the coupling matrix) and possibly the external fields $h = (h_i)$. In this manuscript we assume that $h = 0$ is known.

Because of the exponential complexity of the partition function, direct maximum likelihood estimation is computationally intractable except for very small systems. Over the last decades, a variety of approximation strategies have been proposed, ranging from mean-field methods to convex relaxations and more recent variational approaches. A recurrent theme is the introduction of *regularization*, which enforces sparsity and improves statistical robustness in high-dimensional regimes.

In this section we review and compare several regularized estimators for the inverse Ising problem.

1. In Section 2.1 we discuss *Regularized Minimum Probability Flow* (RMPF), which builds upon the principle of minimizing the instantaneous flow of probability under a perturbation of the empirical distribution.

2. In Section 2.2 we recall the *Regularized Pseudolikelihood Estimator* (RPLE), a classical and widely used method based on logistic regressions of each spin against its neighborhood.

3. Section 2.3 is devoted to the more recent *Regularized Erasure Machine* (REM), which exploits a high-temperature expansion to linearize the moment-matching conditions.

4. Finally, Section 2.5 presents numerical experiments comparing the performance of these methods across different regimes of sample size, system size and interaction strength.

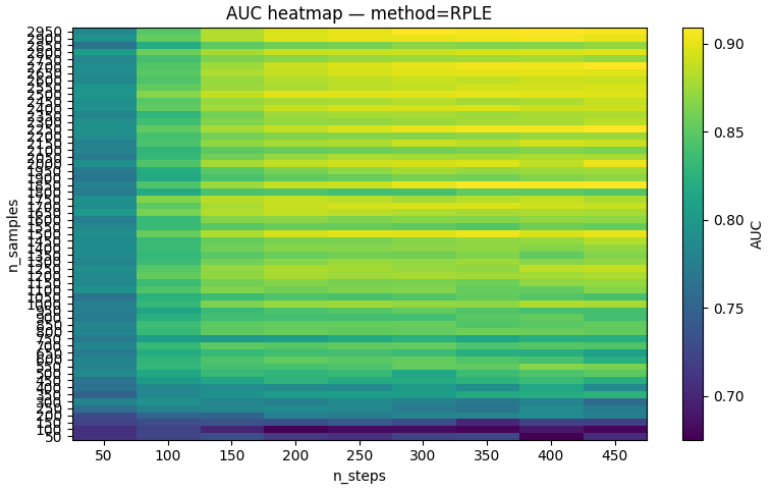


Figure 5: pseudolikelihood 11.1s

Our aim is to provide a unified framework and a systematic empirical comparison, highlighting strengths and limitations of each approach.

References

A Appendix

Here we collect additional derivations, proofs, and technical details that complement the main text.

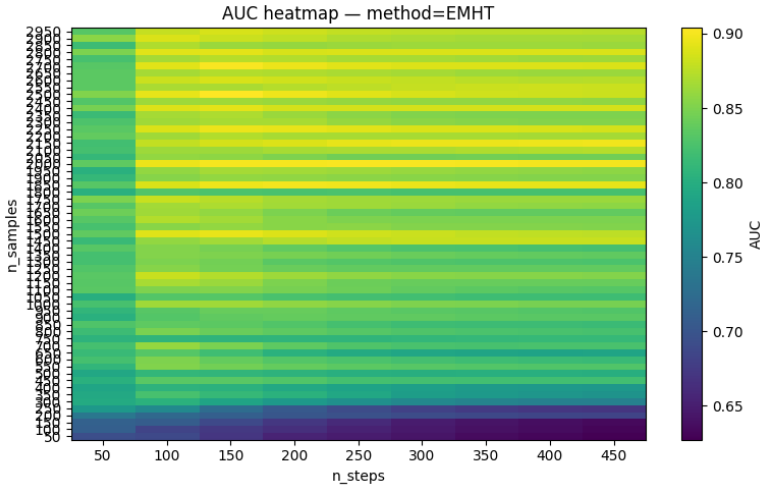


Figure 6: emht 0.11s

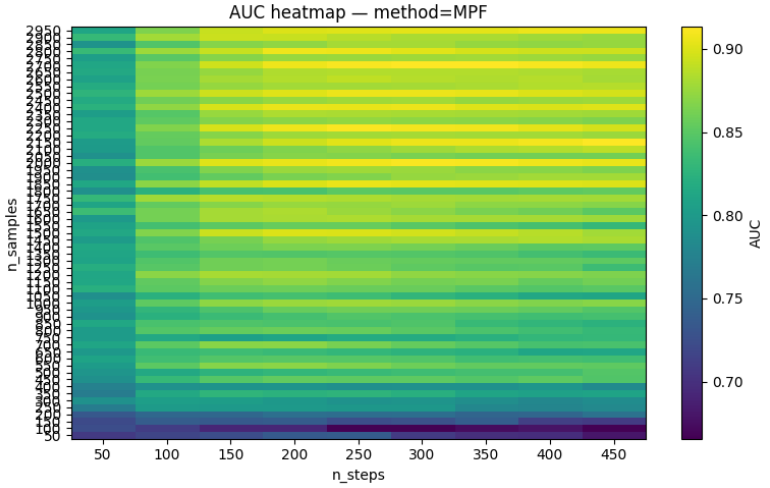


Figure 7: minimum probability flow 6.49s

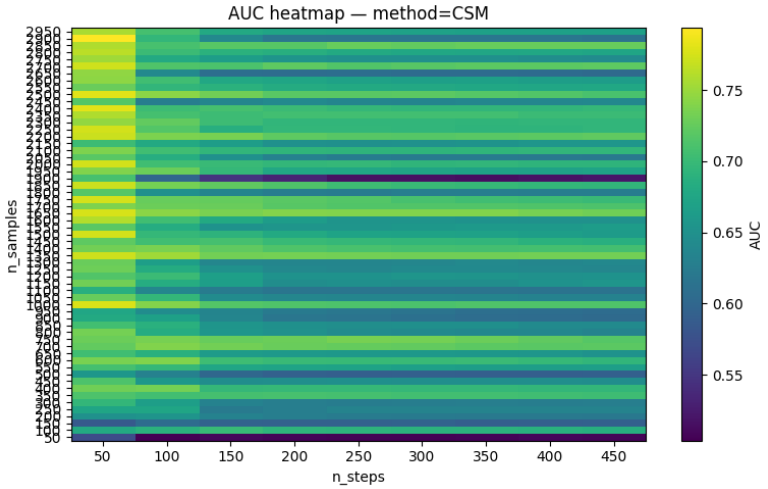


Figure 8: csm 11.2s

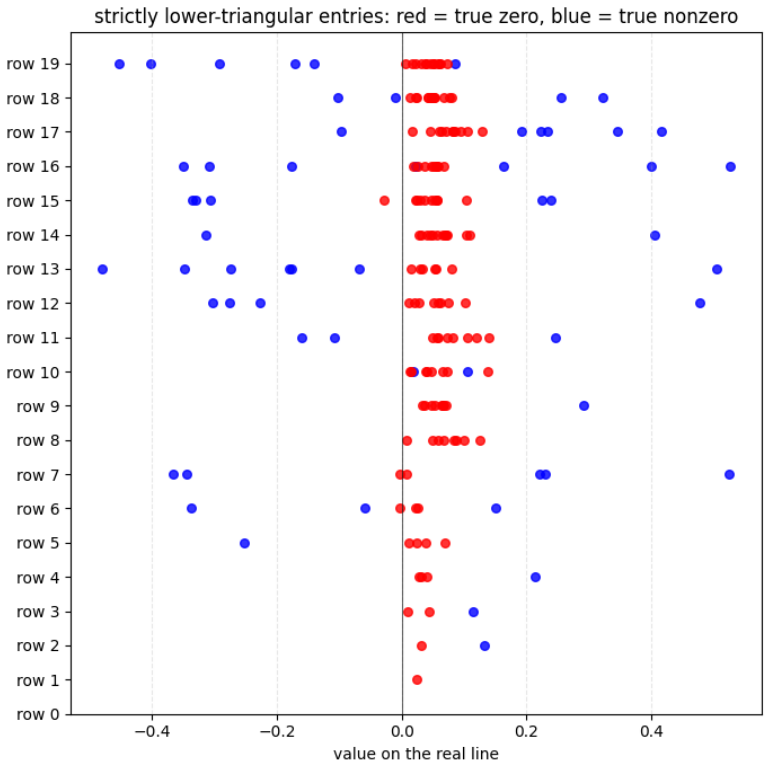
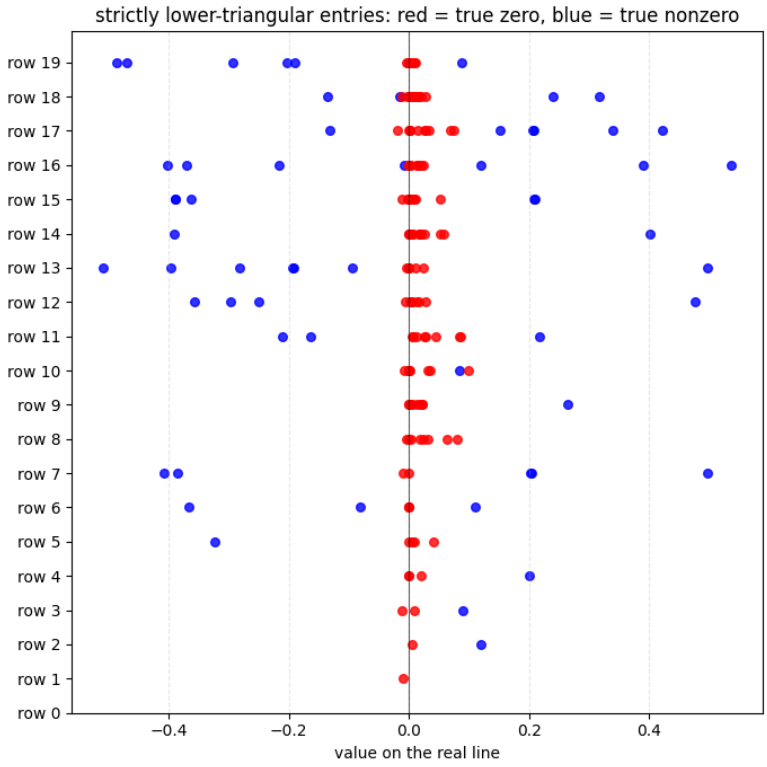


Figure 9: emht



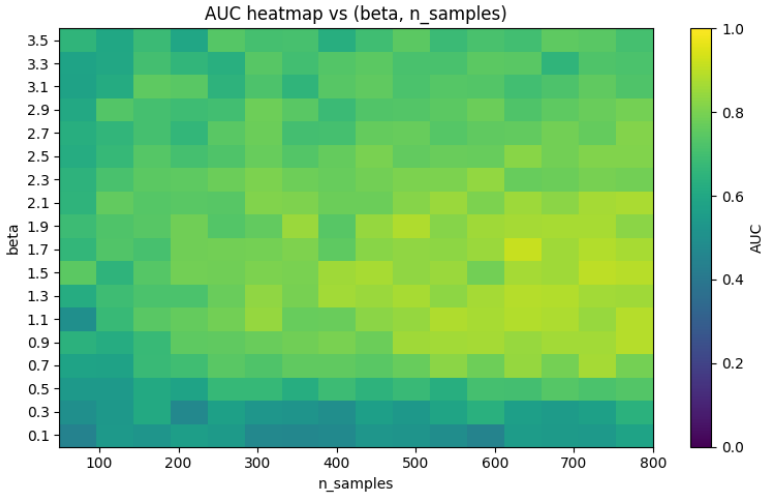


Figure 10: emht

Figure 11: ple

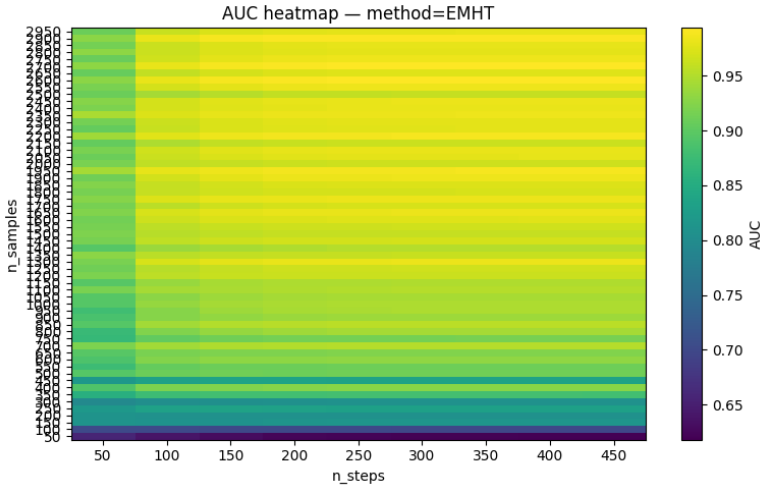


Figure 12: emht 0.5 beta

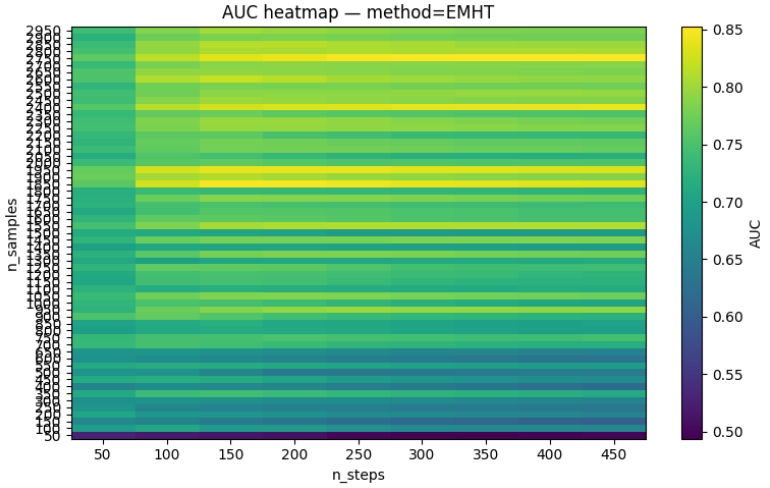


Figure 13: emht 1.5 beta