Robust Inverse Covariance Estimation under Noisy Measurements

Jun-Kun Wang, Shou-De Lin

Intel-NTU, National Taiwan University

ICML 2014
Table of contents

1 Introduction

2 Background
   • Related works: neighborhood estimation

3 This work
   • Robust inverse covariance estimation
   • Generative counterpart of GCRF
   • Positive semi-definiteness Guarantee

4 Experiment
   • Time series prediction
   • Classification

5 Conclusion
What inverse covariance estimation can do?

1) **graphical model/ structure learning.** The non-zeros pattern in a matrix has a one-to-one correspondence of edges in a Markov network.

*Figure:* structure learning of a Markov network
What inverse covariance estimation can do?

2) time series prediction: Gaussian conditional random field (GCRF)


\[
\begin{align*}
\text{minimize} & \quad - \log |\Phi_{xy}| + tr(S_{yy} \Phi_{yy} + 2S_{yx} \Phi_{xy} + \Phi_{yy}^{-1} \Phi_{xy}^T S_{xx} \Phi_{xy}) + \\
& \quad \lambda(\|\Phi_{yy}\|_1 + \|\Phi_{xy}\|_1).
\end{align*}
\]

(1)

where $\Phi_{yy}$ reveals the relations within output variables $y$, and $\Phi_{xy}$ reveals the relations between input and output variables.

After learning the inverse covariance, prediction is made by sampling

\[
y|x \sim N(-\Phi_{yy}^{-1} \Phi_{xy}^T x, \Phi_{yy}^{-1}).
\]

(2)
3) **classification**: Linear discriminant analysis (LDA) (Hastie et al. (2009) and Murphy (2013))

LDA assumes the features conditioned on the class follow multivariate Gaussian distribution.

By maximizing a posterior, the label is assigned by the class that has the maximum linear discriminant score:

\[
\delta(k) = x^T \Phi \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \Phi \hat{\mu}_k + \log \hat{\pi}_k,
\]

where \( \hat{\pi}_k \) is the fraction of class \( k \) in the training set, \( \hat{\mu}_k \) is the mean of features in class \( k \).
1) \(l_1\) penalized negative log-likelihood:

\[
\minimize_{\Phi} = -\log |\Phi| + \text{tr}(S\Phi) + \lambda |\Phi| \quad (4)
\]

Banerjee et al. (2008); d’Aspremont et al. (2008); Rothman et al. (2008); Duchi et al. (2008); Ravikumar et al. (2011); Hsieh et al. (2011, 2013).

2) neighborhood estimation: (The paper belongs to this category.)
Estimate each column by linear regression/programming.
Meinshausen and Buhlmann (2006); Friedman et al. (2008); Yuan (2010); Cai et al. (2011).
Neighborhood estimation

To estimate a column $i \in \{1, \ldots, d\}$ of the inverse covariance.

step 1: running a regression.

$$x_i = c_i + w_{(i)}^T x_{-i} + \epsilon_i.$$  \hspace{1cm} (5)

step 2: forming a column.

$$\Phi_{i,i} = (\text{Var}(\epsilon_i))^{-1}$$

$$\Phi_{-i,i} = -w_{(i)}(\text{Var}(\epsilon_i))^{-1}$$

where $\text{Var}(\epsilon_i) = \frac{1}{m} \|x_i - w_{(i)}^T x_{-i}\|_2^2 = S_{i,i} - 2w_{(i)}^T S_{-i,i} + w_{(i)}^T S_{-i,-i} w_{(i)}.$  \hspace{1cm} (6)

step 3: adjusting the matrix to be symmetric.
Robust inverse covariance estimation
Generative counterpart of GCRF
Positive semi-definiteness Guarantee

Robust optimization:

$$\min_{w(i) \in \mathbb{R}^{d-1}} \left\{ \max_{\Delta \in \mathbb{U}} \| X_i - (X_{-i} + \Delta) w(i) \|_2 \right\},$$

(7)

where $\Delta \in \mathbb{R}^{m \times (d-1)}$ is the measurement errors, and $\mathbb{U}$ is the uncertainty set, or the set of admissible disturbances of the data matrix $X_{-i} \in \mathbb{R}^{m \times (d-1)}$. 
Subproblem about estimating the covariance of each input variable with the others

\[
\minimize_{w(i) \in \mathbb{R}^{d-1}} \maximize_{\Delta \in \mathcal{U}} \|X_i - (X_{-i} + \Delta)w(i)\|_2, \tag{8}
\]

To model the measurement error, we propose to optimize under the following uncertainty set:

\[
\|\Delta_g\|_2 \leq c_g \tag{9}
\]

where \(g\) is the group index and \(\Delta_g\) of which the \(i_{th}\) column is \(\Delta_i\) (which represents the measurement errors for the \(i_{th}\) variable over samples) if the \(i_{th}\) variable belongs to group \(g\), or 0 otherwise.
The robust optimization is equivalent to the following objective, by a proposition in Yang and Xu (2013).

$$\minimize_{w \in \mathbb{R}^d} \| X_{-i} - X_i w \|_2^2 + \sum_{i}^{k} c_{g_i} \| w_{g_i} \|_2^2,$$

which is the non-overlapped group lasso (Yuan and Lin, 2006) and exists efficient methods to solve it (Meier et al., 2008 and Roth and Fischer, 2008).
Generative counterpart of GCRF

Recall the GCRF. The $l_1$ penalized likelihood, Wytock and Kolter (2013).

\[
\text{minimize } - \log \left| \Phi_{xy} \right| + \text{tr} \left( S_{yy} \Phi_{yy} + 2 S_{yx} \Phi_{xy} + \Phi_{yy}^{-1} \Phi_{xy}^{T} S_{xx} \Phi_{xy} \right) + \\
\lambda (\| \Phi_{yy} \|_1 + \| \Phi_{xy} \|_1).
\]  

(11)

We propose a generative counterpart that enables parallelization.

We can view training as the process of estimating the inverse covariance matrices that consists of input and output variables, $\Phi = \begin{pmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{xy}^{T} & \Phi_{yy} \end{pmatrix}$. Thus, the method proposed in the previous subsection can be exploited.
To guarantee positive semi-definiteness

**Concern**

The estimators are not guaranteed to be positive semi-definite. (Meinshausen and Buhlmann (2006); Friedman et al. (2008); Yuan (2010); Cai et al. (2011) also share the same concern).

Most of the sampling methods for multivariate Gaussian distribution require performing the Cholesky factorization of the given estimated covariance. When prediction, predicted values are sampled from (Barr and Slezak (1972); Law and Kelton (1991)).

\[ y|x \sim N(-\Phi_{yy}^{-1}\Phi_{xy}^T x, \Phi_{yy}^{-1}). \quad (12) \]
Algorithm 1 Adjusting the inverse covariance that guarantees positive semi-definiteness

**Input:** $\Phi_{tmp}$ an estimated symmetric inverse covariance by regression (may not be positive semi-definite), $\alpha$ is initialized $\in (0, 1]$, and $\beta \in (0, 1)$.

\[
D = \Phi_{tmp} - I
\]

repeat

Compute the Cholesky factorization of $I + \alpha D$.

if $I + \alpha D$ is not positive definite then

$\alpha = \beta \alpha$

end if

until $I + \alpha D$ is positive definite

---

**Lemma** Hsieh et al. (2011): *For any $X \succ 0$ and a symmetric $D$, there exists an $\alpha' > 0$ such that for all $\alpha \leq \alpha'$: $X + \alpha D \succ 0$*
To estimate a column $i \in \{1, \ldots, d\}$ of the inverse covariance.

**step 1: running a group lasso.**

$$
\text{minimize} \| X_i - X_{-i} w(i) \|_2 + \sum_i^k c_{g_i} \| w_{g_i} \|_2.
$$

**step 2: forming a column.**

$$
\Phi_{i,i} = (\text{Var}(\epsilon_i))^{-1}
$$

$$
\Phi_{-i,i} = -w(i)(\text{Var}(\epsilon_i))^{-1}
$$

where

$$
\text{Var}(\epsilon_i) = \frac{1}{m} \| x_i - w_{(i)}^T x_{-i} \|_2^2 =
$$

$$
S_{i,i} - 2w_{(i)}^T S_{-i,i} + w_{(i)}^T S_{-i,-i} w(i).
$$

**step 3: adjusting the matrix to be symmetric.**

**step 4: adjusting the matrix to be positive semi-definite by the proposed algorithm.**
Experiment sketch

Robust method of inverse covariance estimation under noisy measurements in sensor data

- Time series prediction: GCRF.
- Classification: LDA.
GCRF Time series prediction: experiment setup

- Preprocessing: We choose time series that have moderate variance ($\sigma \leq 15$). The input features contain the values of time series previous three days (AR 3).

- Datasets:
  1) **Stock**: S&P 100 in year 2012. 60 times series.
  2) **Temperature (medium variable size)**: NOAA 73 time series.
  3) **Temperature (large variable size)**: NOAA 401 time series.

- Baseline: WK13 (ICML 13), an $l_1$ penalized approach for GCRF.
GCRF Time series prediction: experiment setup

To simulate the noisy measurements, we add some artificial noise in the data.

1. Every 10 time series are randomly grouped and the noise in the series of a group will be given the same perturbation bound.

2. Specifying the noise level and noise distribution.
   The average variance of time series over time in a group is first calculated, and for
   **Uniform noise:**
   The range of uniform noise is randomly chosen between $\pm (0.1, 1)$ times the average variance in each group.
   **Gaussian noise:**
   The standard deviation of the noise is set randomly to $k$ times the average variance in each group, where $k$ is a random value between $(0.1, 1)$.
GCRF Time series prediction: experiment setup

Our method only require the **bound**, not the actual value. We provide the information to our method based on the sufficient statistics of the distribution of the noise we add.

1. **Uniform noise**: range of the distribution.
2. **Gaussian noise**: standard deviation of the distribution.

\[
\|\Delta_g\|_2 \leq c_g \quad (15)
\]

\[
\text{minimize} \|X_i - X_{-i}w(i)\|_2 + \sum_{i}^{k} c_{g_i} \|w_{g_i}\|_2. \quad (16)
\]

Denote \(c\) as a vector whose entries are perturbation bound \(c_g\), the regularization vector is searched by \(c\) times \([10^{-8}, 10^{-7}, \ldots, 10^{2}]\) over the grid.
The last 30 trading days are reserved for testing; the second to last 30 days are for validation; and the remaining data are for training.

Note: For clean data, our method use lasso to estimate the inverse convariance.

**Table:** Forecasting results (RMSE) on clean data (no noise is added).

<table>
<thead>
<tr>
<th>Data</th>
<th>WK13</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>stock</td>
<td>2.6314</td>
<td>2.6828</td>
</tr>
<tr>
<td>temp. (medium variable size)</td>
<td>2.3917</td>
<td>2.7966</td>
</tr>
<tr>
<td>temp. (large variable size)</td>
<td>2.4119</td>
<td>1.9832</td>
</tr>
</tbody>
</table>
Table: Forecasting results (RMSE) under uniform perturbation.

<table>
<thead>
<tr>
<th>Data</th>
<th>WK13</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>stock</td>
<td>3.6296</td>
<td>3.1300</td>
</tr>
<tr>
<td></td>
<td>(0.0876)</td>
<td>(0.0613)</td>
</tr>
<tr>
<td>temp. (medium variable size)</td>
<td>3.6697</td>
<td>3.0286</td>
</tr>
<tr>
<td></td>
<td>(0.3561)</td>
<td>(0.0447)</td>
</tr>
<tr>
<td>temp. (large variable size)</td>
<td>4.7019</td>
<td>2.1671</td>
</tr>
<tr>
<td></td>
<td>(0.6669)</td>
<td>(0.0220)</td>
</tr>
</tbody>
</table>
Table: Forecasting results (RMSE) under Gaussian perturbation.

<table>
<thead>
<tr>
<th>Data</th>
<th>WK13</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>stock</td>
<td>5.7316</td>
<td>3.1751</td>
</tr>
<tr>
<td></td>
<td>(0.1611)</td>
<td>(0.0888)</td>
</tr>
<tr>
<td>temp. (medium variable size)</td>
<td>6.2661</td>
<td>3.2863</td>
</tr>
<tr>
<td></td>
<td>(0.4859)</td>
<td>(0.2468)</td>
</tr>
<tr>
<td>temp. (large variable size)</td>
<td>8.0439</td>
<td>2.2704</td>
</tr>
<tr>
<td></td>
<td>(0.8086)</td>
<td>(0.0472)</td>
</tr>
</tbody>
</table>
LDA classification: experiment setup

- Datasets: heart: instances:270 feature dimension:13
breast_cancer: instances:683 feature dimension:10
duke_breast_cancer: instances:44 feature dimension:7129

- Settings: For each dataset, we randomly split data 5 times that 80 percent of data are for cross-validation and the remaining for testing. The noise is added on the 5 replicated data for each dataset.

- Baselines:
  1) QUIC: A $l_1$ penalized likelihood approach (An MRF, Hsieh et al. (2011))
  2) Yuan: A regression based approach (Yuan (2010))
**LDA classification: experiment results**

Table: Classification results (Accuracy) on clean data (no noise is added).

<table>
<thead>
<tr>
<th>Data</th>
<th>QUIC</th>
<th>Yuan</th>
</tr>
</thead>
<tbody>
<tr>
<td>heart</td>
<td>0.8519</td>
<td>0.8630</td>
</tr>
<tr>
<td>breast.</td>
<td>0.9547</td>
<td>0.9635</td>
</tr>
<tr>
<td>duke.</td>
<td>0.9800</td>
<td>0.9600</td>
</tr>
<tr>
<td>colon.</td>
<td>0.8923</td>
<td>0.8923</td>
</tr>
</tbody>
</table>
LDA classification: experiment results

Table: Classification results (Accuracy) under uniform and Gaussian perturbation.

<table>
<thead>
<tr>
<th>Data</th>
<th>QUIC</th>
<th>Yuan</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>heart (uniform noise)</td>
<td>0.8407</td>
<td>0.8333</td>
<td><strong>0.8481</strong></td>
</tr>
<tr>
<td>breast. (uniform noise)</td>
<td>0.9255</td>
<td>0.9197</td>
<td><strong>0.9445</strong></td>
</tr>
<tr>
<td>duke. (uniform noise)</td>
<td>0.8800</td>
<td>0.8400</td>
<td><strong>0.9200</strong></td>
</tr>
<tr>
<td>colon. (uniform noise)</td>
<td>0.8462</td>
<td>0.8308</td>
<td><strong>0.8923</strong></td>
</tr>
<tr>
<td>heart (Gaussian noise)</td>
<td>0.8481</td>
<td>0.8407</td>
<td><strong>0.8556</strong></td>
</tr>
<tr>
<td>breast. (Gaussian noise)</td>
<td>0.9314</td>
<td>0.9343</td>
<td><strong>0.9431</strong></td>
</tr>
<tr>
<td>duke. (Gaussian noise)</td>
<td><strong>0.9000</strong></td>
<td>0.8200</td>
<td>0.8400</td>
</tr>
<tr>
<td>colon. (Gaussian noise)</td>
<td>0.8000</td>
<td>0.8308</td>
<td><strong>0.8769</strong></td>
</tr>
</tbody>
</table>
Contributions

1) Study inverse covariance estimation under the existence of additive noise in the features.

2) Guarantee the estimator to be positive semi-definite.

3) Show the effectiveness of our method in classification and time series prediction.
1. Dealing with noisy measurements and missing value simultaneously.
2. Recovering the Markov network.


