

On information-geometric aspects of graphical models and kernel machines

NIPS 2005 Workshop, Whistler, Canada

Justin Dauwels

www.dauwels.com

Signal and Information Processing Laboratory
ETH Zurich

December 10



Outline

Main result

Information matrices of sophisticated graphical models can be computed efficiently.

Why you may care about information matrices ...

- Statistics: Cramér-Rao-type bounds
- Information geometry: natural-gradient-based algorithms
- Machine learning: Fisher kernels

Aim of this talk

- ① **Introduction** to information matrices and their applications
- ② How to **compute** information matrices?

Information matrices

Notation

- θ : parameter (that we wish to estimate)
- Y : observed random variable
- X : unobserved random variable (that we wish to estimate).

Fisher information matrix of pdf $p(y; \theta)$

$$F(\theta) \triangleq \mathbb{E}_{Y; \theta} \left[\left(\frac{\partial}{\partial \theta} \log p(y; \theta) \right)^2 \right] = -\mathbb{E}_{Y; \theta} \left[\frac{\partial^2}{\partial^2 \theta} \log p(y; \theta) \right].$$

Bayesian information matrix of pdf $p(x, y)$

$$J \triangleq \mathbb{E}_{XY} \left[\left(\frac{\partial}{\partial x} \log p(x, y) \right)^2 \right] = -\mathbb{E}_{XY} \left[\frac{\partial^2}{\partial^2 x} \log p(x, y) \right].$$

Similarly: **hybrid** information matrix of $p(x, y|\theta)$.

Information matrices

Notation

- θ : parameter (that we wish to estimate)
- Y : observed random variable
- X : unobserved random variable (that we wish to estimate).

Fisher information matrix of pdf $p(y; \theta)$

$$F(\theta) \triangleq E_{Y; \theta} \left[\left(\frac{\partial}{\partial \theta} \log p(y; \theta) \right)^2 \right] = -E_{Y; \theta} \left[\frac{\partial^2}{\partial^2 \theta} \log p(y; \theta) \right].$$

Bayesian information matrix of pdf $p(x, y)$

$$J \triangleq E_{XY} \left[\left(\frac{\partial}{\partial x} \log p(x, y) \right)^2 \right] = -E_{XY} \left[\frac{\partial^2}{\partial^2 x} \log p(x, y) \right].$$

We will assume that the matrices are properly defined ...

Outline

Main result

Information matrices of sophisticated graphical models can be computed efficiently.

Why you may care about information matrices ...

- Statistics: Cramér-Rao-type bounds
- Information geometry: natural-gradient-based algorithms
- Machine learning: Fisher kernels

Aim of this talk

- ① **Introduction** to information matrices and their applications
- ② How to **compute** information matrices?

Cramér-Rao-type bounds: introduction

What?

Lower bounds on the mean-squared estimation error (MSEE).

Why?

Assessment of practical estimators.

Cramér-Rao-type bounds: introduction (2)

Example (AR model)

Let X_1, X_2, \dots be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_M X_{k-M} + U_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \dots$ be defined as:

$$Y_k = X_k + W_k, \quad W_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_W^2).$$

Task

Estimate $a_1, \dots, a_M, \sigma_U^2, \sigma_W^2$, and X from observation Y .

Cramér-Rao-type bounds: introduction (2)

Example (AR model)

Let X_1, X_2, \dots be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_M X_{k-M} + U_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \dots$ be defined as:

$$Y_k = X_k + W_k, \quad W_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_W^2).$$

Task

Estimate $a_1, \dots, a_M, \sigma_U^2, \sigma_W^2$, and X from observation Y .

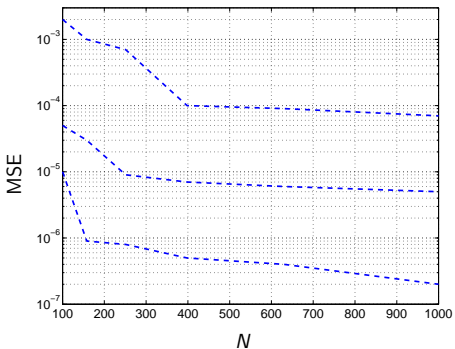
Observation

ML/MAP/MMSE-estimators are **infeasible!**

Neither can their MSE be determined \Rightarrow **lower bounds** on MMSE.

Cramér-Rao-type bounds: introduction (3)

Results for σ_W^2 ($\sigma_U^2 = 0.1$; $\sigma_W^2 = 0.001, 0.01, 0.1$)



Estimation algorithm by Sascha Korl.

Does the algorithm perform well?

Cramér-Rao-type bounds: overview

What is the CRB more precisely?

Inverse of **information matrix**.

Three different types

... as there are three different types of information matrices

- **Standard** Cramér-Rao bounds: parameters
- **Bayesian** Cramér-Rao bounds: random variables
- **Hybrid** Cramér-Rao bounds: parameters and random variables

Cramér-Rao-type bounds: overview

What is the CRB more precisely?

Inverse of **information matrix**.

Three different types

... as there are three different types of information matrices

- Standard Cramér-Rao bounds: parameters
- **Bayesian** Cramér-Rao bounds: random variables
- Hybrid Cramér-Rao bounds: parameters and random variables

Bayesian Cramér-Rao bound

Theorem (Bayesian Cramér-Rao bound)

Let $p(x, y)$ be the joint pdf of $x \in \mathbb{R}$ and $y \triangleq (y_1, \dots, y_N)$.
 If $p(x)$ is **zero** at boundary of its support, then for any **regular** $\hat{x}(y)$:

$$E_{XY} [(x - \hat{x}(y))^2] \geq J^{-1},$$

where the Bayesian information matrix J is defined as:

$$J \triangleq E_{XY} \left[\left(\frac{\partial}{\partial x} \log p(x, y) \right)^2 \right].$$

Properties

- **MAP-estimator** achieves bound as SNR or $N \rightarrow \infty$.
- BCRB holds for **any** regular $\hat{x}(y)$ as SNR or $N \rightarrow \infty$.

Bayesian Cramér-Rao bound: simple example

Example (Mean of a Gaussian random variable)

$Y = X + Z$ with $Z \sim \mathcal{N}(0, \sigma^2)$ with σ^2 **known** and $X \in \mathbb{R}$ **unknown**.

Estimate X from observations y_1, y_2, \dots, y_N with prior $p(X)$ for X .

$$p(x, y_1, y_2, \dots, y_N) = p(x) \prod_{k=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-y_k)^2/2\sigma^2}.$$

$$\begin{aligned} \mathbf{J} &= -\mathbb{E}_{XY} \left[\frac{\partial^2}{\partial x^2} \log p(X, Y) \right] \\ &= \frac{N}{\sigma^2} - \mathbb{E}_X \left[\frac{d^2}{dx^2} \log p(X) \right]. \end{aligned}$$

$$\mathbb{E}_{XY} [(\hat{x}(X) - X)^2] \geq \mathbf{J}^{-1} = \left(\frac{N}{\sigma^2} - \mathbb{E}_X \left[\frac{d^2}{dx^2} \log p(X) \right] \right)^{-1}$$

If $p(x)$ is Gaussian, then BCRB = **minimum** achievable MSE!

Vector case

Bayesian Cramér-Rao bound for component X_k

Given: joint pdf $p(x, y)$ of $x \triangleq (x_1, \dots, x_M)$ and $y \triangleq (y_1, \dots, y_N)$.

Lower bound for the MSE $E_{X_k Y} [(X_k - \hat{x}_k(Y))^2]$?

From marginal

$$E_{X_k Y} [(X_k - \hat{x}_k(Y))^2] \geq J_k^{-1},$$

with $J_k \triangleq E_{X_k Y} \left[\left(\frac{\partial}{\partial x_k} \log p(x_k, y) \right)^2 \right]$.

From joint pdf

$$E_{X_k Y} [(X_k - \hat{x}_k(Y))^2] \geq [J^{-1}]_{kk},$$

with $J_{ij} \triangleq E_{XY} \left[\left(\frac{\partial}{\partial x_i} \log p(x, y) \right)^T \frac{\partial}{\partial x_j} \log p(x, y) \right]$.

BCRB from marginal is **tighter** than from joint pdf, but more difficult to compute.

Algorithms

From joint pdf

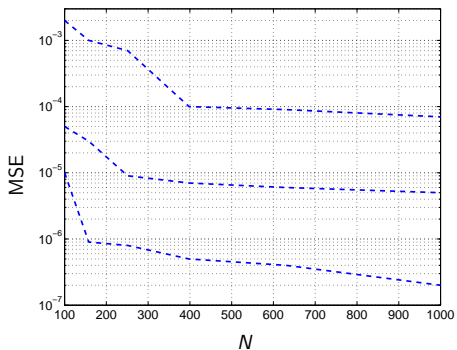
- J is often **sparse**.
- Only **diagonal** elements of inverse required.
- **Local** computations of “small” matrices (message passing).

From marginal

- J_k is usually **dense**.
- Key to J_k : $\frac{\partial}{\partial x_k} \log p(x_k, y) = E_{X \sim k | X_k, Y} \left[\frac{\partial}{\partial x_k} \log p(X, Y) \right]$
- **Expectation** computed by belief propagation (or “sum-product algorithm” or “probability propagation”).

Results

Results for σ_W^2 ($\sigma_U^2 = 0.1$; $\sigma_W^2 = 0.001, 0.01, 0.1$)

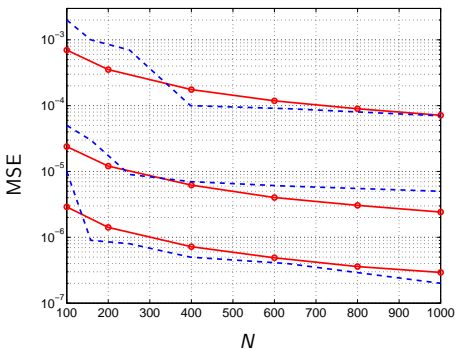


Estimation algorithm by Sascha Korl.

Does the algorithm perform well?

Results

Results for σ_W^2 ($\sigma_U^2 = 0.1$; $\sigma_W^2 = 0.001, 0.01, 0.1$)



Estimation algorithm by Sascha Korl.

A closer look

Example (AR model)

Let X_1, X_2, \dots be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_M X_{k-M} + U_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \dots$ be defined as:

$$Y_k = X_k + W_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_W^2).$$

Reminder

$$F(\theta) \triangleq \mathbf{E}_{Y; \theta} \left[\nabla_{\theta} \log p(Y; \theta) \nabla_{\theta}^T \log p(Y; \theta) \right] \quad \text{with } p(y; \theta) \triangleq \int_x p(x, y; \theta)$$

$$\theta = (\mathbf{a}, \sigma_U^2, \sigma_W^2)$$

$$\nabla_{\theta} \log p(y; \theta) = \mathbf{E}_{X|y} [\nabla_{\theta} \log p(X, y; \theta)].$$

Expectations $\nabla_{\theta} \log p(y; \theta) = E_{X|\theta y} [\nabla_{\theta} \log p(X, y; \theta)]$

$$p(x, y | \mathbf{a}, \sigma_W^2, \sigma_U^2) = \prod_k \underbrace{\mathcal{N}\left(x_k - \sum_{n=1}^M a_n x_{k-n} \mid 0, \sigma_U^2\right)}_{f_1(x_k, \dots, x_{k-M}, \mathbf{a}, \sigma_U^2)} \underbrace{\mathcal{N}(y_k - x_k \mid 0, \sigma_W^2)}_{f_2(x_k, \sigma_W^2, y_k)}.$$

As a consequence:

$$\begin{aligned} \nabla_{\theta} \log p(x, y | \mathbf{a}, \sigma_W^2, \sigma_U^2) &= \sum_k \nabla_{\theta} \log f_1(x_k, \dots, x_{k-M}, \mathbf{a}, \sigma_U^2) \\ &\quad + \sum_k \nabla_{\theta} \log f_2(x_k, \sigma_W^2, y_k). \end{aligned}$$

Expectations $\nabla_{\theta} \log p(y; \theta) = E_{X|y} [\nabla_{\theta} \log p(X, y; \theta)]$ (2)

$$E_{X|a, \sigma_W^2, \sigma_U^2} \left[\nabla_{a_i} \log f_1(X_k, \dots, X_{k-M}, \mathbf{a}, \sigma_U^2) \right]$$

$$= \frac{1}{\sigma_U^2} \left(E_{X|a, \sigma_W^2, \sigma_U^2} [X_k - i X_k] - \sum_{\ell=1}^M a_{\ell} E_{X|a, \sigma_W^2, \sigma_U^2} [X_{k-i} X_{k-\ell}] \right)$$

$$E_{X|a, \sigma_W^2, \sigma_U^2} \left[\nabla_{\sigma_U^2} \log f_1(X_k, \dots, X_{k-M}, \mathbf{a}, \sigma_U^2) \right]$$

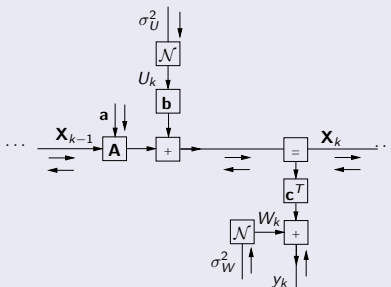
$$= -\frac{1}{2\sigma_U^2} + \frac{1}{2\sigma_U^4} \left(E_{X|a, \sigma_W^2, \sigma_U^2} [X_k^2] - 2 \sum_{\ell=1}^M a_{\ell} E_{X|a, \sigma_W^2, \sigma_U^2} [X_k X_{k-\ell}] \right)$$

$$+ \sum_{\ell=1}^M \sum_{m=1}^M a_{\ell} a_m E_{X|a, \sigma_W^2, \sigma_U^2} [X_{k-\ell} X_{k-m}]$$

$$E_{X|a, \sigma_W^2, \sigma_U^2} \left[\nabla_{\sigma_W^2} \log f_2(X_k, \sigma_W^2, y_k) \right]$$

$$= -\frac{1}{2\sigma_W^2} + \frac{1}{2\sigma_W^4} \left(y_k^2 - 2y_k E_{X|a, \sigma_W^2, \sigma_U^2} [X_k] + E_{X|a, \sigma_W^2, \sigma_U^2} [X_k^2] \right).$$

Expectations $\nabla_{\theta} \log p(y; \theta) = E_{X|\theta y} [\nabla_{\theta} \log p(X, y; \theta)]$ (3)



$E_{X|a\sigma_W^2\sigma_U^2} Y$ computed by forward/backward **Kalman recursions**.

(= instance of belief propagation)

Computing the Fisher information matrix of AR model

- 1 Generate a list of samples $\{\hat{y}^{(j)}\}_{j=1}^N$ from $p(y|\theta)$.
- 2 For $j = 1, \dots, N$:
 - Forward and backward Kalman recursion with $y = \hat{y}^{(j)}$.
 - Evaluate the expression:

$$E_{X|\theta, \hat{y}^{(j)}} \left[\nabla_{\theta} \log p(X, \hat{y}^{(j)}; \theta) \right].$$

- 3 Compute the estimate $\hat{\mathbf{F}}(\theta)$ for $\mathbf{F}(\theta)$:

$$\hat{\mathbf{F}}(\theta) \triangleq \frac{1}{N} \sum_{j=1}^N \left[E_{X|\theta, \hat{y}^{(j)}} \left[\nabla_{\theta} \log p(X, \hat{y}^{(j)}; \theta) \right] E_{X|\theta, \hat{y}^{(j)}}^T \left[\nabla_{\theta} \log p(X, \hat{y}^{(j)}; \theta) \right] \right].$$

Outline

Main result

Information matrices of sophisticated graphical models can be computed efficiently.

Why you may care about information matrices ...

- Statistics: Cramér-Rao-type bounds
- Information geometry: natural-gradient-based algorithms
- Machine learning: Fisher kernels

Aim of this talk

- ① **Introduction** to information matrices and their applications
- ② How to **compute** information matrices?

Information geometry

Information geometry [Amari]

- Given: family of pdfs $p(y|\theta)$.
- Pdfs $p(y|\theta)$ as points in a space.
- Unique invariant metric is Fisher information matrix.

Natural gradient-based algorithms [Amari]

- Task: estimate θ from observations $y = (y_1, y_2, \dots)$.

- Standard gradient methods:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \lambda_k \nabla_{\theta} \log p(y|\theta)|_{\hat{\theta}_{k-1}}$$

- **Natural gradient** method:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \lambda_k F^{-1}(\theta) \nabla_{\theta} \log p(y|\theta)|_{\hat{\theta}_{k-1}}.$$

- Natural-gradient based algorithms often **converge faster** ...
- ... but require $F^{-1}(\theta)$.

Natural gradient-based estimation

Reminder

Information matrix computed by belief propagation.
(or “sum-product algorithm” or “probability propagation”).

Approximations

Approximate information matrices by **iterative** message passing.
Approximate but **efficient** inversion by imposing **structure**.

Applications

Novel algorithms for estimation (e.g., AR model).

Framework

Conveniently derived as **message passing** on factor graphs.

Outline

Main result

Information matrices of sophisticated graphical models can be computed efficiently.

Why you may care about information matrices ...

- Statistics: Cramér-Rao-type bounds
- Information geometry: natural-gradient-based algorithms
- Machine learning: Fisher kernels

Aim of this talk

- ① **Introduction** to information matrices and their applications
- ② How to **compute** information matrices?

Machine learning

Fisher kernels [Jaakkola et al., 1998]

- Given: pdf $p(y|\hat{\theta})$ trained by means of dataset \mathcal{Y} .
- For example:

$$\hat{\theta} = \hat{\theta}_{\text{ML}} \triangleq \underset{\theta}{\operatorname{argmax}} \prod_{y_i \in \mathcal{Y}} p(y_i|\theta).$$

- Fisher kernel:

$$\kappa(y, y') \triangleq \nabla_{\theta} p(y|\hat{\theta}) F^{-1}(\hat{\theta}) \nabla_{\theta} p(y'|\hat{\theta}).$$

- So far, in practice: $F^{-1}(\hat{\theta}) \approx I$.

Outlook

Potentially better results with

- **true** inverse information matrix
- **approximation** (loopy BP/approximate inversion).

Summary

In this talk...

- Introduction to information matrices
- Applications:
 - Statistics: Cramér-Rao-type bounds
 - Information geometry: natural gradient-based algorithms
 - Machine learning: Fisher kernels.
- Information matrices computed by **message-passing algorithms** operating on factor graphs.

Take-home message

Information matrices of sophisticated graphical models can be computed efficiently.

Want to know more?

Information matrices and applications

- PhD thesis (ETH Zurich): www.dauwels.com/PhD.htm
On Graphical Models for Communications and Machine Learning: Algorithms, Bounds, and Analog Implementation
- Questions/Comments: justin@dauwels.com

Estimation in AR models

Sascha Korl, PhD. Thesis (ETH Zurich),
A Factor Graph Approach to Signal Modelling, System Identification, and Filtering.