Computing Bayesian Cramér-Rao bounds

Justin Dauwels
Dept. of Information Technology and Electrical Engineering,
ETH, CH-8092 Zürich, Switzerland.
Email: dauwels@isi.ee.ethz.ch.

***Abstract*— An efficient message-passing algorithm for computing the Bayesian Cramér-Rao bound (BCRB) for general estimation problems is presented. The BCRB is a lower bound on the mean squared estimation error. The algorithm operates on a cycle-free factor graph of the system at hand. It can be applied to estimation in (1) general state-space models; (2) coupled state-space models and other systems that are most naturally represented by cyclic factor graphs; (3) coded systems.**

I. Introduction

For many practical estimation problems, popular estimators such as the maximum likelihood estimator (ML), the maximum a posteriori estimator (MAP) or the minimum mean squared error estimator (MMSE) are infeasible. Therefore, one often needs to resort to suboptimal techniques such as expectation maximization, loopy belief propagation, gradient-based algorithms, Markov chain Monte Carlo methods, particle filters, or combinations of those methods.

Suboptimal techniques are typically evaluated by determining the mean squared error (MSE) through simulations and by comparing this error to theoretical performance bounds.

For the estimation of *deterministic* parameters, a commonly used lower bound for the MSE is the Cramér-Rao bound (CRB), given by the inverse of the Fisher information matrix [1] [2]. Van Trees derived an analogous bound to the CRB for *random variables*, referred to as "Bayesian CRB" (BCRB) or "posterior CRB" [1].

The BCRB has been determined for several estimation problems; Tichavský et al. derived the BCRB for filtering in state-space models with *freely* evolving state [3]. A particle method for numerically evaluating that BCRB for the particular case of nonlinear nonstationary dynamical systems is presented in [4]; the method has recently been used for computing the BCRB for various tracking problems (see e.g., [5]).

In this paper, we propose a message-passing algorithm to compute the BCRB for general estimation problems; it operates on a cycle-free factor graph of the system at hand.

When it is applied to a cyclic factor graph, it does not lead to the (exact) BCRB; one needs to convert the cyclic graph into a cycle-free graph before applying the algorithm.

Our method can be applied to several "standard" estimation problems as for example *filtering* and *smoothing* in *general* state-space models. When it is applied to *filtering* in state-space models with *freely* evolving state, one recovers the BCRB of [3]. It can handle coupled state-space models and other systems which are most naturally represented by cyclic factor graphs. It leads to lower bounds on the BCRB of code-aided channel estimation, as for example code-aided phase, frequency and timing estimation.

This paper is structured as follows. In the next section, we review the CRB and BCRB. We present our summary propagation algorithm for computing BCRBs in Section III. In Section IV, we apply our algorithm to filtering and smoothing in state-space models and describe briefly how one can deal with systems represented by cyclic graphs. We elaborate on code-aided channel estimation in Section V. In Section VI, we offer some concluding remarks.

II. Review of the (Bayesian) Cramér-Rao bound

We start by introducing our notation and by reviewing some basic facts. Let $X = (X_1, X_2, \ldots, X_n)^T$ and $Y = (Y_1, Y_2, \ldots, Y_m)^T$, where X_k and Y_k are real random vectors (the extension to complex random vectors is straightforward); the vectors X_k and Y_k do not necessarily all have the same dimensionality. The index k may stand for (discrete) time, i.e., X and Y may be stochastic processes. Suppose $p(x, y)$ is the joint probability density function (pdf) of X and Y. We consider the problem of estimating X from the observation vector $y = (y_1, y_2, \ldots, y_m)^T$. Let the function $\hat{x}(y)$ be an estimator of X based on the observation y. The error matrix $\mathbf{E}$ of the estimator $\hat{x}(y)$ is defined as

$$\mathbf{E} \triangleq \mathrm{E}_{XY}[(\hat{x}(Y) - X)(\hat{x}(Y) - X)^T]. \tag{1}$$

The Bayesian information matrix $\mathbf{J}$ is given by

$$\mathbf{J}_{ij} \triangleq \mathrm{E}_{XY}\left[\nabla_{x_i} \log p(x, y) \nabla_{x_j}^T \log p(x, y)\right], \tag{2}$$

where $\mathbf{J}_{ij}$ is the (i, j)-th element of $\mathbf{J}$, $\nabla_v \triangleq \left[\frac{\partial}{\partial v_1}, \ldots, \frac{\partial}{\partial v_q}\right]^T$ with $v \in \mathbb{R}^q$, and $v \triangleq (v_1, \ldots, v_q)^T$. Note that $\mathbf{J}_{ij}$ is a matrix, since the components X_k are in general vectors. The Bayesian information matrix $\mathbf{J}$ can be computed in several ways [1, p. 72]

$$\begin{aligned} \mathbf{J}_{ij} &\triangleq \mathrm{E}_{XY}\left[\nabla_{x_i} \log p(x, y) \nabla_{x_j}^T \log p(x, y)\right] && (3) \\ &= -\mathrm{E}_{XY}\left[\nabla_{x_i} \nabla_{x_j}^T \log p(x, y)\right] && (4) \\ &= \mathrm{E}_{XY}\left[\nabla_{x_i} \log p(y|x) \nabla_{x_j}^T \log p(y|x)\right] && (5) \\ &\quad + \mathrm{E}_{X}\left[\nabla_{x_i} \log p(x) \nabla_{x_j}^T \log p(x)\right]. && (6) \end{aligned}$$

The Bayesian information matrix is related to the Fisher information matrix $\mathbf{F}(x)$, which plays an important role in statistics [2], information geometry [6] and machine learning (e.g., [7]), and is defined as

$$\mathbf{F}_{ij}(x) \triangleq \mathrm{E}_{Y|X}\left[\nabla_{x_i} \log p(y|x) \nabla^T_{x_j} \log p(y|x)\right]. \quad (7)$$

Note that the Bayesian information matrix $\mathbf{J}$ is constant, whereas the Fisher $\mathbf{F}(x)$ information matrix depends on x. When X is a deterministic parameter, i.e., $p(x|y) \triangleq \gamma p(y|x)$, where γ is a normalization factor, it follows from (2) and (7):

$$\mathbf{J} = \mathrm{E}_X[\mathbf{F}(X)]. \quad (8)$$

When $\mathbf{F}(x)$ does not depend on x, i.e., $\mathbf{F}(x) \triangleq \mathbf{F}$, then $\mathbf{F}(x) \triangleq \mathbf{F} = \mathbf{J}$.

There exists a well-known relation (see e.g., [1, pp. 66–67], [2, pp. 301–303]) between the error matrix $\mathbf{E}(x)$, defined as

$$\mathbf{E}(x) \triangleq \mathrm{E}_{Y|X}[\hat{x}(Y) - X)(\hat{x}(Y) - X)^T], \quad (9)$$

and the Fisher information matrix $\mathbf{F}(x)$:

$$\mathbf{E}(x) \succeq \mathbf{F}(x)^{-1}. \quad (10)$$

The inequality (10) means that the matrix $\mathbf{D}(x) \triangleq \mathbf{E}(x) - \mathbf{F}(x)^{-1}$ is positive semi-definite. It is known as the Cramér-Rao bound, but was in fact first proposed by Fisher in the early days of statistics [8]. It holds if the so-called "regularity" conditions are fulfilled [2, p. 111], and merely applies to *unbiased* estimators $\hat{x}(y)$. We remind the reader of the fact that the MMSE estimator is *not* necessarily unbiased, which seriously jeopardizes the usefulness of the CRB. This is for example the case when X takes values in an interval $[a, b]$ or $[a, \infty)$ as in phase, frequency, timing and noise variance estimation[1]. Nevertheless, quite some attention has been given to the computation of the CRB for those estimation problems (see e.g., [9]– [12]).

In '68, Van Trees proved a Cramér-Rao-type bound for random variables [1, pp. 72–73]:

$$\mathbf{E} \succeq \mathbf{J}^{-1}. \quad (11)$$

It holds under some regularity conditions in addition to the "weak unbiasedness" condition $\int_x \nabla_{x_j}[p(x)B(x)] = 0$, where $B(x) \triangleq \int_y [\hat{x}(y) - x] p(y|x) dy$. The inequality (11) is often referred to as the "Bayesian Cramér-Rao bound" (BCRB), "posterior CRB" or the "Van Trees bound"[2]. Note that the BCRB also holds for *biased* estimators, in contrast to the CRB. It is however important to realize that the weak unbiasedness condition is not necessarily fulfilled. As for the CRB, this is for example the case when X takes values in an interval $[a, b]$ or $[a, \infty)$. However, one can often obtain a lower bound on the MSE by deriving the BCRB for an estimation problem with the *same* observation model $p(y|x)$, but with a *different* prior $\tilde{p}(x) \triangleq \gamma p(x) w(x)$, where γ is a normalization factor and $w(x)$ is a window function (e.g., Hamming window) that is zero at the edges of the support of $p(x)$. The BCRB is valid for this modified estimation problem; it is moreover a lower bound on the MSE of the *original* problem, since $\tilde{p}(x)$ is more informative than $p(x)$. When n is sufficiently large ($n > 100$) and/or the SNR sufficiently high (SNR > 0dB), the BCRB of the modified problem coincides with the (invalid) BCRB of the original problem.

In practice, one is mostly interested in bounding the estimation error for the *components* X_k, i.e., $\mathbf{E}_{kk} \triangleq \mathrm{E}_{XY}[(\hat{x}_k(Y) - X_k)(\hat{x}_k(Y) - X_k)^T]$, as for example in filtering and smoothing. Of practical relevance is also the average $\sum_k w_k \mathbf{E}_{kk}$, where w_k are nonnegative real numbers (often $w_k = 1/n$). From (11) and the fact that diagonal submatrices of a positive semi-definite block matrix are themselves positive semi-definite, it follows that

$$\mathbf{E}_{kk} \succeq [\mathbf{J}^{-1}]_{kk}, \quad (12)$$

and

$$\sum_S w_k \mathbf{E}_{kk} \succeq \sum_S w_k [\mathbf{J}^{-1}]_{kk}, \quad (13)$$

where S is an arbitrary subset of $\{1, \ldots, n\}$.

III. Summary Propagation Algorithm

Note that in the RHS of (12) and (13), the inverse of the (potentially huge!) matrix $\mathbf{J}$ occurs. However

1) Only the diagonal elements of this inverse are required.
2) The joint probability density $p(x, y)$ has in most practical systems a "nice" structure, i.e., $p(x, y)$ has typically a non-trivial factorization. As a consequence, $\mathbf{J}$ is often sparse.
3) This sparseness can effectively be exploited by applying the matrix inversion lemma.

As a consequence, the elements $[\mathbf{J}^{-1}]_{kk}$ can be determined by local computations involving the inversion of matrices that are much smaller than $\mathbf{J}$. Those computations can be viewed as message passing on a (cycle-free) factor graph of $p(x, y)$. We will use Forney-style factor graphs as in [14], where edges represent variables and nodes represent factors. The summary propagation procedure is similar to the sum-product algorithm [14]; messages (which are in this case matrices) propagate on the factor graph of $p(x, y)$. They are updated at the nodes according to some rules. The expression $[\mathbf{J}^{-1}]_{kk}$ is obtained from the messages along the edge X_k. In the following, we present the procedure in more detail.

At nodes representing differentiable functions, messages are computed according to the following rule

Summary Rule:
The message out of the node $g(x_1, \ldots, x_\ell, y)$ (Fig. 1(a)) along the edge X_ℓ is the matrix

$$\boldsymbol{\mu}_{g \to X_\ell} = \left(\left[(\mathbf{G} + \mathbf{M})^{-1}\right]_{\ell\ell}\right)^{-1}, \quad (14)$$

[1]Moreover, note that the MSE is not a suitable error measure for phase and frequency estimation.

[2]This form of the Bayesian Cramér-Rao lower bound is in the literature also referred to as *unconditional* Bayesian Cramér-Rao lower bound; in contrast, the *conditional* BCRB bounds the MSE conditioned on a particular observation y [13]. Note the conditional BCRB is often difficult to evaluate, since it requires marginals of the *posterior* density, whereas the unconditional BCRB involves marginals of the *prior*.

where

$$\mathbf{M} = \operatorname{diag}(\boldsymbol{\mu}_{X_1\to g}, \ldots, \boldsymbol{\mu}_{X_{\ell-1}\to g}, 0) \tag{15}$$

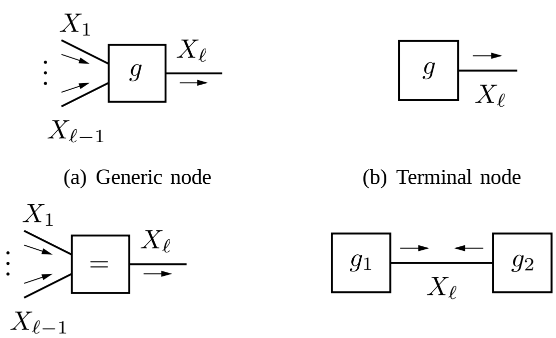
$$\mathbf{G}_{ij} \triangleq -\mathrm{E}_{XY}[\nabla_{x_i}\nabla_{x_j}^T \log g(x_1,\ldots,x_\ell,y)] \tag{16}$$

$$= -\int_{x,y} p(x,y)\nabla_{x_i}\nabla_{x_j}^T \log g(x_1,\ldots,x_\ell,y)dxdy, \tag{17}$$

where it is assumed that the integrals (17) exist $\forall i$ and $j = 1,\ldots,\ell$.

The expression (14) can be written in several ways; one can permute rows and corresponding columns of the matrices $\mathbf{M}$ and $\mathbf{G}$. The expectations (17) can most often be simplified, since the local node g typically does not depend on the *whole* observation vector y, but on a small number of components y_k instead. Despite this fact, a closed-form expression for the expectations (17) may not exist; they may then be evaluated by numerical integration or by Monte Carlo methods.

It is well known that Kalman filters/smoothers are optimal (in terms of MSE) for estimating the state of linear systems perturbed by additive Gaussian noise. For that particular estimation problem, the recursion (14) is identical to the Kalman recursion for the inverse covariance matrix [3].



(a) Generic node (b) Terminal node

(c) Equality constraint node (d) Edge X_ℓ

Fig. 1. Summary propagation

The message out of a terminal node $g(x_\ell, y)$ (Fig. 1(b)) is defined as $\boldsymbol{\mu}_{g\to X_\ell} \triangleq -\mathrm{E}_{XY}[\nabla_{x_\ell}\nabla_{x_\ell}^T \log g(x_\ell, y)]$. Half edges do not carry a message towards the (single) node attached to them; alternatively, they might be thought of as carrying a zero matrix as message. For the equality constraint node (Fig. 1(c)), the integrals (17) do not exist, since the node "function" $f_=(x_1,x_2,\ldots,x_\ell) \triangleq \delta(x_1-x_2)\delta(x_2-x_3)\ldots\delta(x_{\ell-1}-x_\ell)$ is not differentiable; the equality constraint node has its own update rule; the outgoing message $\boldsymbol{\mu}_{\boxminus\to X_\ell}$ is the sum of the incoming messages $\boldsymbol{\mu}_{X_k\to\boxminus}$:

$$\boldsymbol{\mu}_{\boxminus\to X_\ell} = \sum_{k=1}^{\ell-1}\boldsymbol{\mu}_{X_k\to\boxminus}. \tag{18}$$

Eventually, the expression $[\mathbf{J}^{-1}]_{kk}$ (bounding the MSE of X_k) is computed from the two messages $\boldsymbol{\mu}_{g_1\to X_k}$ and $\boldsymbol{\mu}_{g_2\to X_k}$ along the edge X_k (see Fig. 1(d))

$$[\mathbf{J}^{-1}]_{kk} = (\boldsymbol{\mu}_{g_1\to X_k} + \boldsymbol{\mu}_{g_2\to X_k})^{-1}. \tag{19}$$

In the following section, we apply the above procedure to state-space models.

IV. Application to state-space models

We consider a state-space model with freely evolving state X_k. The pdf $p(x,y)$ of such a system is given by

$$p(x,y) = p_0(x_0)\prod_{k=1}^{n} p(x_k|x_{k-1})p(y_k|x_k), \tag{20}$$

its factor graph is shown in Fig. 2(a). Filtering corresponds to forward sum-product message passing through this factor graph. The BCRB for filtering is also computed in a forward sweep, as illustrated in Fig. 2(b) (ignore at this point the backward messages $\tilde{\mu}^B$ and μ^B); by applying the update rules (14) and (18) to the factor graph of Fig. 2(a), one obtains the recursion

$$\tilde{\boldsymbol{\mu}}_{k+1}^F = \left(\left(\left[\begin{array}{cc}\boldsymbol{\mu}_k^F + \mathbf{G}_{k,11} & \mathbf{G}_{k,12}\\ \mathbf{G}_{k,21} & \mathbf{G}_{k,22}\end{array}\right]^{-1}\right)_{22}\right)^{-1} \tag{21}$$

$$= \mathbf{G}_{k,22} - \mathbf{G}_{k,21}(\boldsymbol{\mu}_k^F + \mathbf{G}_{k,11})^{-1}\mathbf{G}_{k,12} \tag{22}$$

$$\boldsymbol{\mu}_{k+1}^F = \tilde{\boldsymbol{\mu}}_{k+1}^F + \boldsymbol{\mu}_{k+1}^Y, \tag{23}$$

where

$$\mathbf{G}_{k,11} \triangleq \mathrm{E}[-\nabla_{x_k}\nabla_{x_k}^T \log p(x_{k+1}|x_k)] \tag{24}$$

$$\mathbf{G}_{k,12} \triangleq [\mathbf{G}_{k,21}]^T = \mathrm{E}[-\nabla_{x_k}\nabla_{x_{k+1}}^T \log p(x_{k+1}|x_k)] \tag{25}$$

$$\mathbf{G}_{k,22} \triangleq \mathrm{E}[-\nabla_{x_{k+1}}\nabla_{x_{k+1}}^T \log p(x_{k+1}|x_k)] \tag{26}$$

$$\boldsymbol{\mu}_k^Y \triangleq \mathrm{E}[-\nabla_{x_k}\nabla_{x_k}^T \log p(y_k|x_k)]. \tag{27}$$

The recursion is initialized as follows

$$\boldsymbol{\mu}_0^F = \mathrm{E}[-\nabla_{x_0}\nabla_{x_0}^T \log p(x_0)]. \tag{28}$$

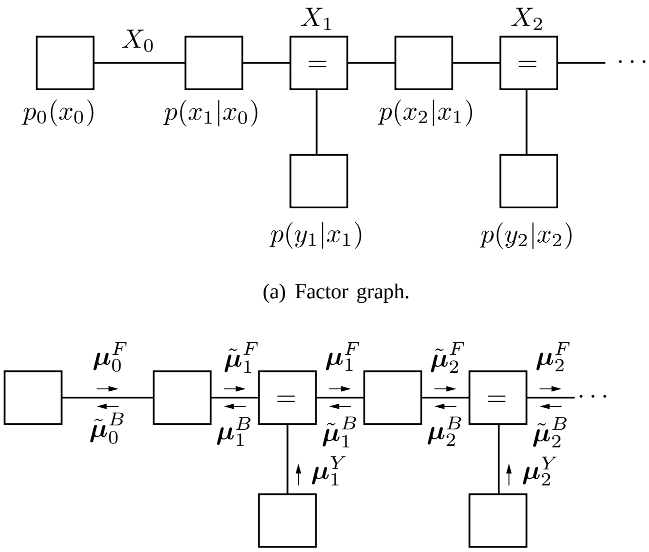
When the expectations (24)–(27) cannot be evaluated analytically, they can easily be evaluated by Monte-Carlo methods; indeed, in most applications, it is easy to sample from $p(x_k, x_{k+1})$ and $p(x_k, y_k)$.

The BCRB for filtering is then

$$\mathrm{E}_{XY}[(\hat{x}_k(Y) - X_k)(\hat{x}_k(Y) - X_k)^T] \triangleq \mathbf{E}_{kk} \succeq \left(\boldsymbol{\mu}_k^F\right)^{-1}, \tag{29}$$

which was derived earlier in [3]. We have thus shown that the recursion of [3] can be viewed as forward-only message passing on the factor graph of (20). In the following, we derive the BCRB for smoothing, which to our knowledge is novel. Smoothing corresponds to updating messages according to the sum-product rule in a forward and a backward sweep [14]. Not surprisingly, the corresponding BCRB is also computed by a forward and backward sweep (Fig. 2(b)); the forward recursion is given by (21)–(27), the backward recursion for $\tilde{\boldsymbol{\mu}}_k^B$ and $\boldsymbol{\mu}_k^B$ is analogous. The BCRB for smoothing is given by

$$\mathbf{E}_k \succeq \left(\tilde{\boldsymbol{\mu}}_k^F + \tilde{\boldsymbol{\mu}}_k^B + \boldsymbol{\mu}_k^Y\right)^{-1}. \tag{30}$$



(a) Factor graph.

(b) Summary propagation.

Fig. 2. State space model with freely evolving state.

Similarly, the BCRB can also be derived for *general* state-space models, i.e., input-driven state-space models. The pdf $p(u,x,y)$ of such a system is given by

$$p(u,x,y) = p_0(x_0)\prod_{k=1}^{n} p(u_k,x_k,y_k|x_{k-1}), \qquad (31)$$

where U is the input process. The factor graph of (31) is shown in Fig. 3(a). Applying the update rule (14) to this factor graph amounts to the forward recursion (see Fig. 3(b))

$$\boldsymbol{\mu}_{k+1}^F = \left(\left(\begin{bmatrix} \boldsymbol{\mu}_k^F + \mathbf{G}_{k,11} & \mathbf{G}_{k,12} & \mathbf{G}_{k,13} \\ \mathbf{G}_{k,21} & \mathbf{G}_{k,22} & \mathbf{G}_{k,23} \\ \mathbf{G}_{k,31} & \mathbf{G}_{k,32} & \mathbf{G}_{k,33} \end{bmatrix}^{-1}\right)_{22}\right)^{-1} (32)$$

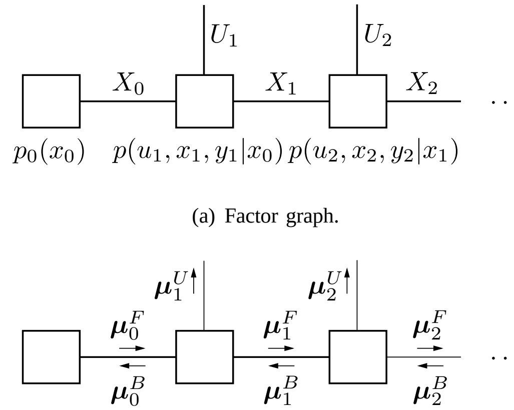
and a similar backward recursion

$$\boldsymbol{\mu}_k^B = \left(\left(\begin{bmatrix} \mathbf{G}_{k,11} & \mathbf{G}_{k,12} & \mathbf{G}_{k,13} \\ \mathbf{G}_{k,21} & \mathbf{G}_{k,22} + \boldsymbol{\mu}_{k+1}^B & \mathbf{G}_{k,23} \\ \mathbf{G}_{k,31} & \mathbf{G}_{k,32} & \mathbf{G}_{k,33} \end{bmatrix}^{-1}\right)_{11}\right)^{-1}, (33)$$

where

$$\mathbf{G}_{k,11} \triangleq \mathrm{E}[-\nabla_{x_k}\nabla_{x_k}^T \log p(u_{k+1},y_{k+1},x_{k+1}|x_k)] \qquad (34)$$
$$\mathbf{G}_{k,12} \triangleq \mathrm{E}[-\nabla_{x_k}\nabla_{x_{k+1}}^T \log p(u_{k+1},y_{k+1},x_{k+1}|x_k)] \qquad (35)$$
$$\mathbf{G}_{k,13} \triangleq \mathrm{E}[-\nabla_{x_k}\nabla_{u_{k+1}}^T \log p(u_{k+1},y_{k+1},x_{k+1}|x_k)] \qquad (36)$$
$$\mathbf{G}_{k,22} \triangleq \mathrm{E}[-\nabla_{x_{k+1}}\nabla_{x_{k+1}}^T \log p(u_{k+1},y_{k+1},x_{k+1}|x_k)] \qquad (37)$$
$$\mathbf{G}_{k,23} \triangleq \mathrm{E}[-\nabla_{x_{k+1}}\nabla_{u_{k+1}}^T \log p(u_{k+1},y_{k+1},x_{k+1}|x_k)] \qquad (38)$$
$$\mathbf{G}_{k,33} \triangleq \mathrm{E}[-\nabla_{u_{k+1}}\nabla_{u_{k+1}}^T \log p(u_{k+1},y_{k+1},x_{k+1}|x_k)], \qquad (39)$$

and $\mathbf{G}_{k,ij} = [\mathbf{G}_{k,ji}]^T$ for $i,j = 1,2$ and 3. To compute the expectations (34)–(39), the joint pdf $p(u_{k+1},x_k,x_{k+1},y_{k+1})$ is required. In most applications, it straightforward to sample from $p(u_{k+1},x_k,x_{k+1},y_{k+1})$. Therefore, when a closed-form expression for the expectations (34)–(39) does not exist, they may be evaluated by Monte Carlo methods, as in state-space models with freely evolving state.



(a) Factor graph.

(b) Summary propagation.

Fig. 3. General state-space model.

The BCRB for filtering X_k is again given by (29), where $\boldsymbol{\mu}_k^F$ is now updated according to (32); the BCRB for smoothing X_k is of the form:

$$\mathrm{E}_{XY}[(\hat{x}_k(Y)-X_k)(\hat{x}_k(Y)-X_k)^T] \succeq \left(\boldsymbol{\mu}_k^F + \boldsymbol{\mu}_k^B\right)^{-1}. \quad (40)$$

The messages $\boldsymbol{\mu}_k^U$ (see Fig. 3(b)) are computed from the messages $\boldsymbol{\mu}_k^F$ and $\boldsymbol{\mu}_k^B$ as follows

$$\boldsymbol{\mu}_{k+1}^U = \left(\left(\begin{bmatrix} \mathbf{G}_{k,11} + \boldsymbol{\mu}_k^F & \mathbf{G}_{k,12} & \mathbf{G}_{k,13} \\ \mathbf{G}_{k,21} & \mathbf{G}_{k,22} + \boldsymbol{\mu}_{k+1}^B & \mathbf{G}_{k,23} \\ \mathbf{G}_{k,31} & \mathbf{G}_{k,32} & \mathbf{G}_{k,33} \end{bmatrix}^{-1}\right)_{33}\right)^{-1} (41)$$

and the BCRB for filtering and smoothing the input U_k is given by

$$\mathrm{E}_{XY}[(\hat{u}_k(Y)-U_k)(\hat{u}_k(Y)-U_k)^T] \succeq \left[\boldsymbol{\mu}_k^U\right]^{-1}. \qquad (42)$$

The message $\boldsymbol{\mu}_{k+1}^B$ in (41) is a zero matrix in the case of filtering; for smoothing, $\boldsymbol{\mu}_{k+1}^B$ is computed by the recursion (33).

Coupled state-space models and other systems which are most often represented by *cyclic* factor graphs can be treated along similar lines. One obtains BCRBs for estimation in such models by transforming the cyclic factor graphs into *cycle-free* graphs, for example by clustering nodes and edges (see e.g., [15]). Computing BCRBs by applying the proposed algorithm on the resulting graphs is most often feasible, since the computational complexity scales cubically (not exponentially!) with the cluster size.

V. Code-aided channel estimation

We now consider the problem of code-aided channel estimation; symbols U_k, protected by an error-correcting code (with indicator function $I(u)$), are transmitted over a channel with memory; the state of the channel is denoted by X_k, whereas Y_k

stands for the channel observation at time k. The probability function of this system is given by

$$p(u, x, y) = p_0(x_0) \prod_{k=1}^{n} p(x_k, y_k | u_k, x_{k-1}) I(u), \quad (43)$$

as depicted in Fig. 4. The box at the top represents the indicator function $I(u)$, the row of nodes at the bottom stands for the factors $p_0(x_0)$ and $p(x_k, y_k|u_k, x_{k-1})$. We wish to compute the BCRB for the estimation of X_k.

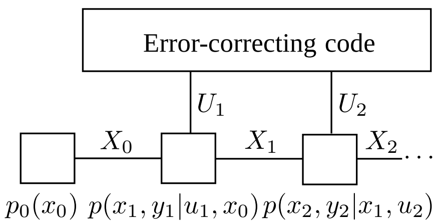


Fig. 4. Code-aided channel estimation.

Computing the (exact) BCRB for this estimation problem is in practice often infeasible since

1) the (exact) marginals $p(u_k|x, y)$ are required; for many practical probabilistic codes, those marginals can only be computed approximately.
2) the Bayesian information matrix $\mathbf{J}$ is *not* sparse due to the correlation between the symbols U_k; therefore, the elements $[\mathbf{J}^{-1}]_{kk}$ can only be obtained by inverting the (typically large) matrix $\mathbf{J}$.

Note that replacing the exact marginals by *approximate* ones delivered by an iterative decoder, as suggested in [16] for deriving code-based CRBs, does not amount to the exact BCRBs. Whereas it is difficult to compute the exact BCRBs for this setup, it is often feasible to derive upper and lower bounds on the BCRB. Upper bounds are obtained by assuming that the symbols are uniformly distributed (uncoded). On the other hand, if one assumes that the symbols are known, one obtains lower bounds, referred to as "modified" BCRBs (MBCRB). The latter are computed by the techniques described in Section IV. The modified BCRBs are often tight lower bounds on the BCRB, since one typically operates communications systems in a regime in which the (bit and frame) error rates are low, and the symbols can therefore be considered as (almost) "known". We computed MBCRBs for the code-aided phase estimation problem described in [17]; the phase estimators we proposed in [17] achieve this MBCRB for SNR values corresponding to frame error rates below 10^{-2}.

VI. Conclusions

We have presented an efficient message passing algorithm for computing Bayesian Cramér-Rao bounds (BCRB) for general estimation problems. Whereas the standard Cramér-Rao bound (CRB) is only applicable to the estimation of *deterministic* parameters, the BCRB holds for *random* variables (or processes) with non-trivial priors. In addition, the CRB is only valid for *unbiased* estimators, whereas the BCRB is valid under significantly weaker conditions. We demonstrated how the BCRB for filtering and smoothing in general state-space models can be computed and outlined how coupled state-space models can be treated. The method seems also attractive for evaluating iterative code-aided channel estimators, such as turbo-synchronizers. It might also be useful as a tool for designing pilot sequences for realistic channel models.

VII. Acknowledgments

This project was supported in part by the Swiss National Science Foundation grant 200021-101955. The author thanks Shun-ichi Amari, Sascha Korl, Frank Kschischang, Amos Lapidoth, Hans-Andrea Loeliger, and Marc Moeneclaey for helpful comments.

References

[1] H. L. Van Trees, *Detection, estimation, and modulation theory: Part I,* Wiley, New York, 1968.
[2] M. J. Schervish, *Theory of statistics,* Springer Verlag, 1995.
[3] P. Tichavský, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing,* vol. 46, May 1998, pp. 1386–1396.
[4] R. M. Taylor Jr., B. P. Flanagan and J. A. Uber, "Computing the recursive posterior Cramér-Rao bound for a nonlinear nonstationary system", *Proc. of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 6, April 6–10, 2003, pp. 673–676.
[5] M. G. S. Bruno and A. Pavlov, "Improved particle filters for ballistic target tracking," *Proc. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Montreal, Quebec, Canada, May 17–21, 2004, vol. 2, pp. 705–708.
[6] S. Amari and H. Nagaoka, *Methods of Information Geometry, AMS Translations of Mathematical Monographs,* vol. 191, American Mathematical Society (AMS) and Oxford University Press, 2000.
[7] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *In Advances in Neural Information Processing Systems,* vol. 11, 1998.
[8] R. Fisher, "The logic of inductive inference," *Journal of the Royal Society of Statistics,* vol. 98, pp. 39–54, 1935.
[9] M. Moeneclaey, "On the true and the modified Cramer-Rao bounds for the estimation of a scalar parameter in the presence of nuisance parameters," *IEEE Transactions on Communications,* vol. 46, Nov. 1998, pp. 1536–1544.
[10] N. Noels, H. Steendam, M. Moeneclaey, "The true Cramér-Rao bound for carrier frequency estimation from a PSK signal", *IEEE Transactions on Communications,* vol. 52, May 2004, pp. 834–844.
[11] J. Zhang, R. A. Kennedy, T. D. Abhayapala, "Cramér-Rao lower bounds for the time delay estimation of UWB signals", *Proc. 2004 IEEE International Conference on Communications,* Paris, France, June 20–24, 2004, vol. 6, pp. 3424–3428.
[12] B. Friedlander and B. Porat, "The exact Cramer-Rao bound for Gaussian autoregressive processes", *IEEE Transactions on Aerospace and Electronic Systems,* vol. 25, Jan. 1989, pp. 3–7.
[13] B. Bobrovsky, E. Mayer-Wolf, and M. Zakai, "Some classes of global Cramér-Rao bounds," *A. Statistics,* vol. 15, pp. 1421-1438, 1987.
[14] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine,* Jan. 2004, pp. 28–41.
[15] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory,* vol. 47, pp. 498–519, Feb. 2001.
[16] N. Noels, H. Steendam, and M. Moeneclaey, "On the Cramér-Rao lower bound and the performance of synchronizers for (turbo) encoded systems", *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, Lisboa, Portugal, july 11–14, 2004.
[17] J. Dauwels and H. -A. Loeliger, "Phase estimation by message passing," *IEEE International Conference on Communications*, Paris, France, June 20–24, 2004, pp. 523–527.