# Backward Filtering Forward Deciding in Linear Non-Gaussian State Space Models

**Yun-Peng Li**
ETH Zürich

**Hans-Andrea Loeliger**
ETH Zürich

## Abstract

The paper considers linear state space models with non-Gaussian inputs and/or constraints. As shown previously, NUP representations (normal with unknown parameters) allow to compute MAP estimates in such models by iterating Kalman smoothing recursions. In this paper, we propose to compute such MAP estimates by iterating backward-forward recursions where the forward recursion amounts to coordinatewise input estimation. The advantages of the proposed approach include faster convergence, no "zero-variance stucking", and easier control of constraint satisfaction. The approach is demonstrated with simulation results of exemplary applications including (i) regression with non-Gaussian priors or constraints on $k$-th order differences and (ii) control with linearly constrained inputs.

## 1 INTRODUCTION

Consider the estimation of the input sequence $U_1, \ldots, U_N \in \mathbb{R}^L$ of a linear state space model (SSM) with state sequence $X_1, \ldots, X_{N+1} \in \mathbb{R}^M$ and output sequence $Y_1, \ldots, Y_N \in \mathbb{R}^K$. The SSM evolves according to

$$X_{n+1} = AX_n + BU_n \tag{1}$$

$$Y_n = CX_n \tag{2}$$

for $n = 1, \ldots, N$, where $A$, $B$, and $C$ are known matrices of appropriate dimensions. We observe a noisy version of the outputs

$$\breve{Y}_n = Y_n + Z_n, \tag{3}$$

where the noise $Z_n \in \mathbb{R}^K$ is a sequence of independent zero-mean Gaussian random variables (or random vectors) with known covariance matrices $V_{Z_n}$. We also assume a Gaussian prior on the initial state $X_1$. From $\breve{Y}_n = \breve{y}_n \in \mathbb{R}^K$, $n = 1, \ldots, N$, we wish to estimate the unknown inputs $U_1, \ldots, U_N$. (The estimate of $U_n$ for $n \approx 1$ or $n \approx N$ may be poor if the initial state $X_1$ or the final state $X_N + 1$, respectively, are unknown.)

If the input sequence $U_1, \ldots, U_N$ is Gaussian, then estimating the state sequence $X_1, \ldots, X_{N+1}$ is the standard Kalman smoothing problem (Kalman, 1960). Pertinent Kalman smoothing algorithms can be extended to yield an estimate of the input sequence as well (cf. Glover, 1969; Bruderer et al., 2014; Loeliger et al., 2016; Gakis et al., 2024).

In this paper, we consider the estimation of a non-Gaussian input sequence $U_1, \ldots, U_N$. The motivating applications include (i) regression with non-Gaussian priors or constraints on $(k + 1)$-th order differences (Steidl et al., 2006; Kim et al., 2009; Ramdas et al., 2016; Loeliger et al., 2016; Politsch et al., 2020; Roonizi, 2021) and (ii) control with linear constraints on the control signal (Keusch, 2023; Keusch et al., 2024).

Related prior work considers state estimation with non-Gaussian state noise, which can be used also for input signal estimation (if the input matrix $B$ contains an $L \times L$ diagonal matrix). The algorithm proposed by Aravkin et al. (2014) uses an interior point method (involving both solving a system of linear inequalities and a line search at every step). Roonizi (2022) addresses only Kalman filtering (i.e., state estimation based on past observations), but not smoothing in the usual sense (i.e., trajectory estimation based on all observations).

One way to deal with non-Gaussian priors or constraints are NUP representations (normal with unknown parameters), which are closely related to variational representations of penalty functions (Palmer et al., 2005; Bach et al., 2012), see Loeliger (2023) for a brief review. In prior work (including Loeliger

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

et al., 2016; Keusch et al., 2021, 2024), this approach was used to convert estimation in linear SSMs with non-Gaussian inputs or/and non-Gaussian observation noise into iterations of Kalman smoothers that are augmented with input signal estimation for general $B$ (Loeliger 2016). This approach appears to work very well, but it may suffer from zero-variance stucking (as will be explained in Section 2.3) and convergence may be slow.

In this paper, we use the same NUP representations, but we propose another estimation algorithm, which avoids zero-variance stucking, converges faster, and makes it easier to control the parameters for constraint satisfaction.

The paper is structured as follows. Section 2 introduces the system model with NUP priors and reviews pertinent prior work. The proposed new algorithm is given in Section 3. Some applications and numerical results are given in Section 4.

The following notation will be used: "$k\,{:}\,\ell$" denotes a range of indices, e.g., $U_{k:\ell} = (U_k, \ldots, U_\ell)$; "$\propto$" denotes equality of functions up to a scale factor; $(\xi)_+ \triangleq \max\{\xi, 0\}$; and $\mathcal{N}(\xi;\theta)$ denotes a Gaussian probability density function in $\xi$ with parameter(s) $\theta$.

The factor graph notation follows Loeliger et al. (2007), where variables are represented by edges and factors are represented by nodes/boxes. Forward and backward arrows (e.g., $\overrightarrow{\mu}_{U_n}$ and $\overleftarrow{\mu}_{U_n}$) refer to messages flowing with or against, respectively, the arrows in Figures 1 and 2. In this paper, messages are (possibly degenerate) Gaussian probability density functions, up to a scale factor, and are parameterized either by a mean $\overrightarrow{m}$ and a covariance matrix $\overrightarrow{V}$, or by a precision matrix $\overrightarrow{W} = \overrightarrow{V}^{-1}$ and the vector $\overrightarrow{\xi} = \overrightarrow{W}\overrightarrow{m}$, and likewise with reversed arrows.

## 2 BACKGROUND

### 2.1 System Model with NUP Priors

We will use the system model (1)–(3) with a prior of the (separable) form

$$p(u_1, \ldots, u_N) = \prod_{n=1}^{N} p(u_n) \tag{4}$$

with (possibly improper) $p(u_n)$ that can be written as

$$p(u_n) = \max_{\theta_n} \rho(u_n, \theta_n), \tag{5}$$

where

$$\rho(u_n, \theta_n) \triangleq \mathcal{N}(u_n; \theta_n) g(\theta_n). \tag{6}$$

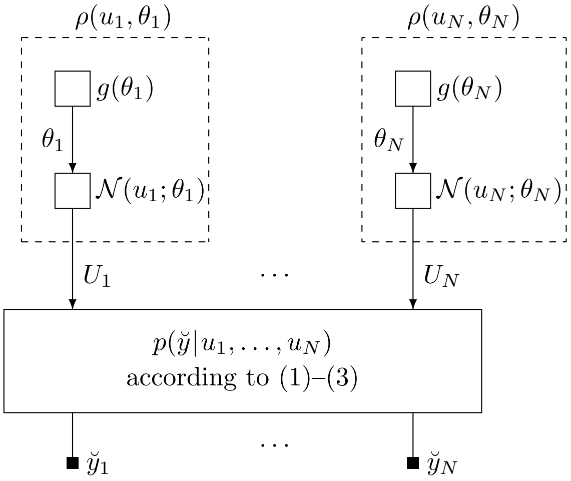

Figure 1: Factor graph of the system model.

The parameter $\theta_n$ in (6) comprises a mean $\overrightarrow{m}_{U_n}$ and a variance (or a covariance matrix) $\overrightarrow{V}_{U_n}$, and $g$ is chosen so that (5) is some desired prior or enforces some constraint on $U_n$.

Note that NUP priors of the form (5) are essentially equivalent to variational representations of regularizers as in Palmer et al. (2005); Bach et al. (2012); Aravkin et al. (2014), which, however, does not exhaust all NUP priors, (cf. Giri et al., 2016; Loeliger, 2023).

With $u \triangleq (u_1, \ldots, u_N) = u_{1:N}$, $\breve{y} \triangleq \breve{y}_{1:N}$, and $\theta \triangleq \theta_{1:N}$, the joint probability density function of $u$ and $\breve{y}$ is

$$p(\breve{y}|u)p(u) = p(\breve{y}|u) \prod_{n=1}^{N} p(u_n) \tag{7}$$

$$= \max_{\theta} f(u, \theta) \tag{8}$$

with

$$f(u, \theta) \triangleq p(\breve{y}|u) \prod_{n=1}^{N} \rho(u_n, \theta_n), \tag{9}$$

which is illustrated in Figure 1.

### 2.2 Iteratively Reweighted Linear Gaussian Estimation (IRLGE)

In (Loeliger et al., 2016, 2018; Loeliger, 2023), a joint MAP estimate of $U$ and $\theta$ (for fixed $\breve{y}$) was computed by alternating maximization, i.e., by iterating the following two steps until convergence:

Table 1: Selected scalar NUP priors and update rules for their mean $\overrightarrow{m}_{U_n}$ and variance $\overrightarrow{V}_{U_n}$

| | $-\log p(u_n)$ | $\overrightarrow{m}_{U_n}$ | $\overrightarrow{V}_{U_n}$ |
|---|---|---|---|
| Laplace/L1 | $\beta \lvert u_n \rvert$ | $0$ | $\lvert \hat{u}_n \rvert / \beta$ |
| Huber loss | $\begin{cases} \frac{u_n^2}{2r^2} + \frac{\beta^2 r^2}{2} & \lvert u_n \rvert \leq \beta r^2 \\ \beta \lvert u_n \rvert & \lvert u_n \rvert > \beta r^2 \end{cases}$ | $0$ | $\begin{cases} r^2 \\ \lvert \hat{u}_n \rvert / \beta \end{cases}$ |
| hinge loss | $\beta(a - u_n)_+$ | $a + \lvert \hat{u}_n - a \rvert$ | $2 \lvert \hat{u}_n - a \rvert / \beta$ |
| Vapnik loss | $\beta \Big( \lvert u_n - a \rvert + \lvert u_n - b \rvert \Big)$ | $\dfrac{a \lvert \hat{u}_n - b \rvert + b \lvert \hat{u}_n - a \rvert}{\lvert \hat{u}_n - a \rvert + \lvert \hat{u}_n - b \rvert}$ | $\dfrac{\lvert \hat{u}_n - a \rvert \lvert \hat{u}_n - b \rvert}{\beta \left[ \lvert \hat{u}_n - a \rvert + \lvert \hat{u}_n - b \rvert \right]}$ |
| plain NUV | $\ln \lvert u_n \rvert$ | $0$ | $\hat{u}_n^2$ |

1. For fixed $\theta = \hat{\theta}$, compute
$$\hat{u} = \operatorname*{argmax}_u p(\breve{y}|u)p(u) \tag{10}$$
$$= \operatorname*{argmax}_u p(\breve{y}|u) \prod_{n=1}^{N} \rho(u_n, \hat{\theta}_n) \tag{11}$$
$$= \operatorname*{argmax}_u p(\breve{y}|u) \prod_{n=1}^{N} \mathcal{N}(u_n; \hat{\theta}_n). \tag{12}$$

2. For fixed $u = \hat{u}$, compute
$$\hat{\theta} = \operatorname*{argmax}_\theta p(\breve{y}|u)p(u) \tag{13}$$
$$= \operatorname*{argmax}_\theta p(\breve{y}|\hat{u}) \prod_{n=1}^{N} \rho(\hat{u}_n, \theta_n), \tag{14}$$
which splits into
$$\hat{\theta}_n = \operatorname*{argmax}_{\theta_n} \rho(\hat{u}_n, \theta_n) \tag{15}$$
for $n = 1, \ldots, N$.

Step 1 of this approach is linear Gaussian estimation and can be carried out by Kalman smoothing (involving both a forward recursion and a backward recursion) augmented with input estimation. Step 2 of this approach amounts to simple closed-form updates, some important cases of which are recalled in Table 1 (cf. MacKay, 1992; Tipping et al., 2003; Bach et al., 2012; Keusch, 2023; Loeliger, 2023), where $\beta > 0$ is a scale parameter. The hinge loss and the Vapnik loss will later (in Section 3.4) be used to enforce the constraints $U_n \geq a$ and $a \leq U_n \leq b$.

### 2.3 Zero-Variance Stucking

An issue with the approach of Section 2.2 (IRLGE) is the possibility of getting stuck with $\overrightarrow{V}_{U_n} = 0$ and $\hat{u}_n = \overrightarrow{m}_{U_n}$ for some input $U_n$, at a point $\hat{u} = (\hat{u}_1, \ldots, \hat{u}_N)$ that is not a maximum of $p(\breve{y}|u)p(u)$.

Specifically, the update of $\hat{u}_n$ in (12) can be written as
$$\hat{u}_n = \frac{\overrightarrow{m}_{U_n} \overleftarrow{V}_{U_n} + \overleftarrow{m}_{U_n} \overrightarrow{V}_{U_n}}{\overrightarrow{V}_{U_n} + \overleftarrow{V}_{U_n}} \tag{16}$$
where $\overleftarrow{m}_{U_n}$ and $\overleftarrow{V}_{U_n}$ are the parameters of the (Gaussian) backward message at $U_n$. It can happen that (15) updates $\overrightarrow{V}_{U_n}$ to zero. In this case, (16) will update $\hat{u}_n$ to $\overrightarrow{m}_{U_n}$, which in turn keeps $\overrightarrow{V}_{U_n}$ stuck at zero. In the first line of Table 1, this happens for $\hat{u}_n = 0$; in the third line of Table 1, this happens for $\hat{u}_n = a$; and in the fourth line of Table 1, this happens for $\hat{u}_n \in \{a, b\}$.

The algorithm proposed in the next section does not have this problem (since its update (23) of $\hat{u}_n$ disregards $\overrightarrow{V}_{U_n}$).

# 3 PROPOSED ALGORITHM

### 3.1 Iterated Backward Filtering Forward Deciding (IBFFD)

We now propose to compute the same estimate as in Section 2.2 (i.e., the joint MAP estimate of $U$ and $\theta$) by iterating the following three steps until convergence:

1. Standard backward filtering (BF). For fixed $\theta = \hat{\theta}$, compute
$$\overleftarrow{\mu}_{X_n}(x_n) \propto \max_{u_{n:N}} p(\breve{y}_{n:N}|x_n, u_{n:N}) p(u_{n:N}) \tag{17}$$
$$\propto \max_{u_{n:N}} p(\breve{y}_{n:N}|x_n, u_{n:N}) \prod_{\ell=n}^{\ell=N} \mathcal{N}(u_\ell; \hat{\theta}_\ell) \tag{18}$$
for $n = N, N-1, \ldots, 1$ by the recursion
$$\overleftarrow{\mu}_{X_n}(x_n) \propto \max_{u_n, x_{n+1}} p(x_{n+1}|x_n, u_n) p(\breve{y}_n|x_n) \cdot \mathcal{N}(u_n; \hat{\theta}_n) \overleftarrow{\mu}_{X_{n+1}}(x_{n+1}) \tag{19}$$

beginning with $\overleftarrow{\mu}_{X_{N+1}}(x_{N+1}) = 1$ (or with some Gaussian $\overleftarrow{\mu}_{X_{N+1}}$ expressing some side information about $X_{N+1}$). The function $\overleftarrow{\mu}_{X_n}$ is a (possibly degenerate) Gaussian density, up to an irrelevant scale factor, $p(x_{n+1}|x_n, u_n)$ is deterministic according to (1), and $p(\breve{y}_n|x_n)$ is given by (2) and (3).

The recursion (19) is efficiently computed by the standard backward information filter (Fraser, 1967), or, equivalently, the backward recursion of BIFM (backward information filter, forward with marginals) message passing (Loeliger et al., 2016).

2. Forward deciding (FD). Begin by computing

$$\hat{x}_1 = \underset{x_1}{\operatorname{argmax}}\, p(x_1)\overleftarrow{\mu}_{X_1}(x_1), \tag{20}$$

where $p(x_1)$ is the given (proper or improper) Gaussian prior on $X_1$. Then, for $n = 1, 2, \ldots, N$, compute $\hat{u}_n$ with fixed $X_1 = \hat{x}_1$ and $U_{1:n-1} = \hat{u}_{1:n-1}$ (and thereby fixed $X_{1:n} = \hat{x}_{1:n}$), and fixed $\theta_{n+1:N} = \hat{\theta}_{n+1:N}$ (but ignoring $\hat{\theta}_{1:n}$), by

$$\begin{aligned} \hat{u}_n &= \underset{u_n}{\operatorname{argmax}} \max_{u_{n+1:N}} p(\breve{y}|u)p(u) && (21) \\ &= \underset{u_n}{\operatorname{argmax}} \max_{u_{n+1:N}} p(\breve{y}_{n+1:N}|\hat{x}_n, u_{n:N}) \\ &\qquad \cdot p(u_n) \prod_{\ell=n+1}^{N} \mathcal{N}(u_\ell; \hat{\theta}_\ell) && (22) \\ &= \underset{u_n}{\operatorname{argmax}}\, p(u_n)\overleftarrow{\mu}_{U_n}(u_n), && (23) \end{aligned}$$

where $\overleftarrow{\mu}_{U_n}$ is a (possibly degenerate) Gaussian density, up to an irrelevant scale factor, given by

$$\begin{aligned} \overleftarrow{\mu}_{U_n}(u_n) &\propto \max_{u_{n+1:N}} p(\breve{y}_{n+1:N}|\hat{x}_n, u_{n:N}) \\ &\qquad \cdot \prod_{\ell=n+1}^{N} \mathcal{N}(u_\ell; \hat{\theta}_\ell) && (24) \\ &\propto \max_{x_{n+1}} p(x_{n+1}|\hat{x}_n, u_n)\overleftarrow{\mu}_{X_{n+1}}(x_{n+1}) && (25) \end{aligned}$$

with $\overleftarrow{\mu}_{X_{n+1}}$ from (18).

3. For fixed $u = \hat{u}$, compute

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\breve{y}|u)p(u), \tag{26}$$

which splits as in (15). This step coincides with Step 2 of IRLGE and can be carried out using Table 1.

Table 2: Deciding rule (23)

| | $\hat{u}_n$ | condition |
|---|---|---|
| Laplace/L1 | $\overleftarrow{m}_{U_n} + \beta\overleftarrow{V}_{U_n}$ | $\overleftarrow{m}_{U_n} < -\beta\overleftarrow{V}_{U_n}$ |
| | $0$ | $\lvert\overleftarrow{m}_{U_n}\rvert \leq \beta\overleftarrow{V}_{U_n}$ |
| | $\overleftarrow{m}_{U_n} - \beta\overleftarrow{V}_{U_n}$ | $\overleftarrow{m}_{U_n} > \beta\overleftarrow{V}_{U_n}$ |
| Huber loss | $\overleftarrow{m}_{U_n} + \beta\overleftarrow{V}_{U_n}$ | $\overleftarrow{m}_{U_n} < -\beta(\overleftarrow{V}_{U_n} + r^2)$ |
| | $\frac{r^2\overleftarrow{m}_{U_n}}{\overleftarrow{V}_{U_n}+r^2}$ | $\lvert\overleftarrow{m}_{U_n}\rvert \leq \beta(\overleftarrow{V}_{U_n} + r^2)$ |
| | $\overleftarrow{m}_{U_n} - \beta\overleftarrow{V}_{U_n}$ | $\overleftarrow{m}_{U_n} > \beta(\overleftarrow{V}_{U_n} + r^2)$ |
| hinge loss | $\overleftarrow{m}_{U_n} + \beta\overleftarrow{V}_{U_n}$ | $\overleftarrow{m}_{U_n} < -\beta\overleftarrow{V}_{U_n} + a$ |
| | $a$ | $-\beta\overleftarrow{V}_{U_n} + a \leq \overleftarrow{m}_{U_n} \leq a$ |
| | $\overleftarrow{m}_{U_n}$ | $\overleftarrow{m}_{U_n} > a$ |
| Vapnik loss | $\overleftarrow{m}_{U_n} + 2\beta\overleftarrow{V}_{U_n}$ | $\overleftarrow{m}_{U_n} < -2\beta\overleftarrow{V}_{U_n} + a$ |
| | $a$ | $-2\beta\overleftarrow{V}_{U_n} + a \leq \overleftarrow{m}_{U_n} \leq a$ |
| | $\overleftarrow{m}_{U_n}$ | $a < \overleftarrow{m}_{U_n} < b$ |
| | $b$ | $b \leq \overleftarrow{m}_{U_n} \leq 2\beta\overleftarrow{V}_{U_n} + b$ |
| | $\overleftarrow{m}_{U_n} - 2\beta\overleftarrow{V}_{U_n}$ | $\overleftarrow{m}_{U_n} > 2\beta\overleftarrow{V}_{U_n} + b$ |
| plain NUV | $\overleftarrow{m}_{U_n} - \left[\overleftarrow{V}_{U_n}/\overleftarrow{m}_{U_n}\right]$ | $\overleftarrow{m}_{U_n} < -\overleftarrow{V}_{U_n}^{1/2}$ |
| | $0$ | $\lvert\overleftarrow{m}_{U_n}\rvert \leq \overleftarrow{V}_{U_n}^{1/2}$ |
| | $\overleftarrow{m}_{U_n} - \left[\overleftarrow{V}_{U_n}/\overleftarrow{m}_{U_n}\right]$ | $\overleftarrow{m}_{U_n} > \overleftarrow{V}_{U_n}^{1/2}$ |

For scalar $U_n$, Table 2 gives the update rule (23) (for the same priors $p(u_n)$ as in Table 1) in terms of the mean $\overleftarrow{m}_{U_n}$ and the variance $\overleftarrow{V}_{U_n}$ of $\overleftarrow{\mu}_{U_n}$. Some of these rules coincide with well-known proximal operators (Moreau, 1965; Parikh et al., 2014); e.g., the first line is the soft-thresholding operator of Donoho et al. (1995) and the last line was used by Tipping et al. (2003).

Convergence of this algorithm will be discussed in Section 3.3.

Note that backward filtering forward deciding in itself is just classical dynamic programming (Bellman, 1954) (with optimal-value function (17)–(19)). The point of this paper is the specific blending of this idea with NUP priors as described above.

### 3.2 Detailed Algorithm

A complete IBFFD algorithm (with Step 3 included in the forward recursion) is given in Algorithm 1, which refers to the system model of Section 1 for the case of multiple inputs, where $U_n = (U_{n,1}, \ldots, U_{n,L})$ consists of independent scalar components $U_{n,\ell}$ and $b_\ell$ denotes column $\ell$ of the matrix $B$, as illustrated in Figure 2.

Algorithm 1 implements IBFFD as Gaussian message passing in Figure 2; its details are easily assembled from the pertinent tables given in Loeliger et al. (2016). The complexity (per iteration) of the algo-

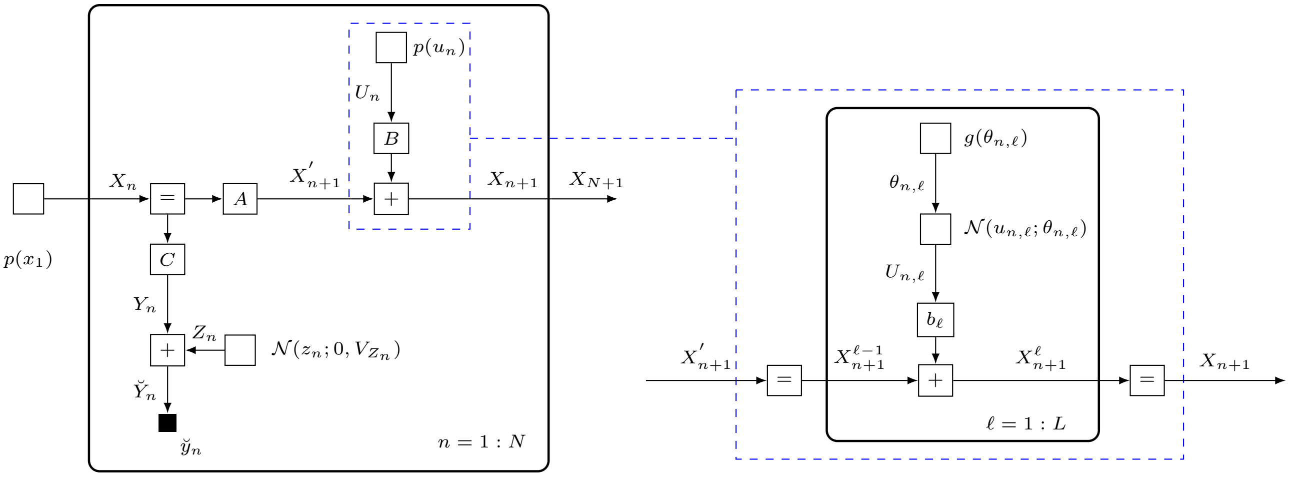

Figure 2: Factor graph of SSM with noisy observations $\breve{y}_1, \breve{y}_2, \cdots, \breve{y}_N$ driven by unknown inputs $U_1, U_2, \cdots, U_N$.

rithm is linear in $N$ and it involves no matrix inversions (or solving linear equations) except in (30).

By default (i.e., unless some extra information about $X_{N+1}$ warrants another choice), the backward message $\overleftarrow{\mu}_{X_{N+1}}$ is neutral (uninformative), i.e., $\overleftarrow{\mu}_{X_{N+1}}(x_{N+1}) = 1$, which is represented by the precision matrix $\overleftarrow{W}_{X_{N+1}} = 0$ and $\overleftarrow{\xi}_{X_{N+1}} = 0$. In this case, no meaningful estimate of $U_N$ is possible, and both $\overrightarrow{m}_{U_N}$ and $\overrightarrow{V}_{U_N}$ can be left unchanged throughout the algorithm.

### 3.3 Convergence and Convexity

Both IRLGE and IBFFD try to maximize (9) by some version of alternating (or cyclic) maximization, which guarantees convergence (to something) except for very exotic situations. However, the NUP representation (5) may introduce spurious maxima (with variance zero) into (9), which underly the problem discussed in Section 2.3. However, the update rule (21)–(23) makes it extremely unlikely to get trapped in such a spurious maximum.

In consequence, IBFFD will "practically almost surely" converge to a local maximum of $p(\breve{y}|u)p(u)$.

In all the applications and numerical experiments in Section 4, $-\log p(\breve{y}|u)p(u)$ is convex, in which case IBFFD will "practically almost surely" find the maximum of $p(\breve{y}|u)p(u)$.

### 3.4 Constraint Satisfaction

Both the hinge loss prior and the Vapnik loss prior can be used to enforce linear constraints on $U_n$. Specifically, with sufficiently large $\beta$, the (improper) hinge loss "prior" $p(u_n) \propto \exp(-\beta(a - u_n)_+)$ can be used to enforces $\hat{u}_n \geq a$, and (improper) Vapnik loss "prior" $p(u_n) \propto \exp(-\beta|u_n - a| - \beta|u_n - b|)$ with $a < b$ can be used to enforce $\hat{u}_n \in [a, b]$.

However, the choice of $\beta$ ("sufficiently large") is not obvious since choosing $\beta$ to be unnecessarily large slows down the convergence of iterative algorithms like IRLGE and IBFFD. An obvious approach (used with IRLGE by Keusch (2023)) is to run the algorithm with fixed $\beta$ until convergence; then check the constraints, increase $\beta$ if necessary (e.g., by a factor of 2), and re-run the algorithm.

However, IBFFD offers an attractive alternative: Algorithm 1 is easily adapted to guarantee constraint satisfaction in every forward recursion by allowing an individual factor $\beta_n$ for each $n$, which is increased, if necessary, during the forward recursion. For each $u_n$, the required minimal value of $\beta_n$ is easily obtained from the update rules in Table 2: if the constraint $\hat{u}_n \geq a$ is violated, increasing $\beta_n$ to

$$\beta_n = \frac{a - \overleftarrow{m}_{Un}}{\overleftarrow{V}_{U_n}} \tag{35}$$

results in $\hat{u}_n \geq a$; if the constraint $\hat{u}_n \in [a, b]$ is violated, increasing $\beta_n$ to

$$\beta_n = \max\left\{ \frac{a - \overleftarrow{m}_{U_n}}{2\overleftarrow{V}_{U_n}}, \frac{\overleftarrow{m}_{U_n} - b}{2\overleftarrow{V}_{U_n}} \right\} \tag{36}$$

results in $\hat{u}_n \in [a, b]$. Consequently, the deciding rules in Table 2 work as proximal operators (Parikh et al., 2014) and project $\hat{u}_n$ on the boundary of the constraints by selecting appropriate $\beta_n$ when necessary.

**Algorithm 1** IBFFD

```
Initialize $\overrightarrow{m}_{U_{n,\ell}}, \overrightarrow{V}_{U_{n,\ell}}$
    for $n = 1, \ldots, N$ and $\ell = 1, \ldots, L$.
while not converged do
    Backward filtering recursion: Begin with $\overleftarrow{\xi}_{X_{N+1}} = 0$ and $\overleftarrow{W}_{X_{N+1}} = 0$ (unless some extra information about $X_{N+1}$ warrants another choice).
    for $n = N$ to 1 do
        $\overleftarrow{W}_{X^L_{n+1}} = \overleftarrow{W}_{X_{n+1}}, \quad \overleftarrow{\xi}_{X^L_{n+1}} = \overleftarrow{\xi}_{X_{n+1}}.$   (27)
        for $\ell = L$ to 1 do
            $H^\ell_{n+1} = \left[\overrightarrow{V}^{-1}_{U_{n,\ell}} + b_\ell^\mathsf{T} \overleftarrow{W}_{X^\ell_{n+1}} b_\ell\right]^{-1},$
            $h^\ell_{n+1} = b_\ell H^\ell_{n+1} \left[\overrightarrow{V}^{-1}_{U_{n,\ell}} \overrightarrow{m}_{U_{n,\ell}} + b_\ell^\mathsf{T} \overleftarrow{\xi}_{X^\ell_{n+1}}\right],$
            $\overleftarrow{\xi}_{X^{\ell-1}_{n+1}} = \overleftarrow{\xi}_{X^\ell_{n+1}} - \overleftarrow{W}_{X^\ell_{n+1}} h^\ell_{n+1},$
            $\overleftarrow{W}_{X^{\ell-1}_{n+1}} = \overleftarrow{W}_{X^\ell_{n+1}} - \overleftarrow{W}_{X^\ell_{n+1}} b_\ell H^\ell_{n+1} b_\ell^\mathsf{T} \overleftarrow{W}_{X^\ell_{n+1}}.$   (28)
        end for
        $\overleftarrow{\xi}_{X_n} = A^\mathsf{T} \overleftarrow{\xi}_{X^0_{n+1}} + C^\mathsf{T} \overleftarrow{V}^{-1}_{Y_n} \overleftarrow{m}_{Y_n},$
        $\overleftarrow{W}_{X_n} = A^\mathsf{T} \overleftarrow{W}_{X^0_{n+1}} A + C^\mathsf{T} \overleftarrow{V}^{-1}_{Y_n} C.$   (29)
    end for
    Forward deciding recursion: beginning with $\overrightarrow{m}_{X_1}, \overrightarrow{V}_{X_1}$, obtain $\hat{x}_1 = m_{X_1}$ by solving the linear equations
        $\left[\overrightarrow{V}^{-1}_{X_1} + \overleftarrow{W}_{X_1}\right] m_{X_1} = \overrightarrow{V}^{-1}_{X_1} \overrightarrow{m}_{X_1} + \overleftarrow{\xi}_{X_1}.$   (30)
    for $n = 1$ to $N$ do
        $\hat{x}^0_{n+1} = A \hat{x}_n.$   (31)
        for $\ell = 1$ to $L$ do
            $\overleftarrow{V}_{U_{n,\ell}} = \left[b_\ell^\mathsf{T} \overleftarrow{W}_{X^\ell_{n+1}} b_\ell\right]^{-1},$
            $\overleftarrow{m}_{U_{n,\ell}} = \overleftarrow{V}_{U_{n,\ell}} \left[b_\ell^\mathsf{T} \overleftarrow{\xi}_{X^\ell_{n+1}} - b_\ell^\mathsf{T} \overleftarrow{W}_{X^\ell_{n+1}} \hat{x}^{\ell-1}_{n+1}\right].$   (32)
            Decide $\hat{u}_{n,\ell}$ according to Table 2.
            $\hat{x}^\ell_{n+1} = \hat{x}^{\ell-1}_{n+1} + b_\ell \hat{u}_{n,\ell}.$   (33)
            Update $\hat{\theta}_{n,\ell} = (\overrightarrow{m}_{U_{n,\ell}}, \overrightarrow{V}_{U_{n,\ell}})$ according to Table 1.
        end for
        $\hat{x}_{n+1} = \hat{x}^L_{n+1}.$   (34)
    end for
end while
```

# 4 APPLICATIONS AND SIMULATION RESULTS

## 4.1 Regression with Constraints on $(k+1)$-th Order Differences

For a given time series $\breve{y} \in \mathbb{R}^N$, consider the problem of computing $y \in \mathbb{R}^N$ such that

$$J(y) = \frac{1}{2} \sum_{n=1}^{N} (\breve{y}_n - y_n)^2 + \beta \sum_{n=1}^{N-k-1} \kappa(\Delta^{k+1}_{y_n}) \tag{37}$$

(for fixed $k \geq 0$ and $\beta > 0$) is as small as possible, where

$$\Delta^{k+1}_{y_n} = \begin{cases} y_{n+1} - y_n & k = 0, \\ \Delta^k_{y_{n+1}} - \Delta^k_{y_n} & k = 1, 2, \ldots, \end{cases} \tag{38}$$

is the $(k+1)$-th order forward difference of $y$. For $\kappa(z) = z^2$, (37) is the classical smoothing spline problem (De Boor, 1978). As is well known (Kohn et al., 1992; Durbin et al., 2012), $u_n \triangleq \Delta^{k+1}_{y_n}$ can be viewed as the input of a linear SSM with

$$\begin{aligned} A &= \begin{pmatrix} a_k & a_{k-1} & \cdots & a_1 & a_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \in \mathbb{R}^{(k+1)\times(k+1)}, \\ B &= \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \end{pmatrix}^\mathsf{T} \in \mathbb{R}^{(k+1)\times 1}, \\ C &= \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{1\times(k+1)}, \end{aligned} \tag{39}$$

where coefficients $a_i = (-1)^{i+k}\binom{k+1}{i}$.

We will consider two variations of this problem that have been considered in the literature: in Section 4.1.1, we use $\kappa(u_n) = |u_n|$ for trend filtering (Kim et al., 2009); in Section 4.1.2, we consider the constraint $u_n \geq 0$ for shape-constrained regression (Guntuboyina et al., 2018), and we encourage this constraint with the hinge loss $\kappa(u_n) = (-u_n)_+$. In both cases, there is a finite $\beta_{\mathsf{max}}$ such that the minimizer $y$ of (37) does not change with $\beta$ for $\beta \geq \beta_{\mathsf{max}}$.

### 4.1.1 Trend Filtering with Laplace Prior

In this section, we experimentally compare different algorithms for minimizing (37) for $\kappa(z) = |z|$ and $k \in \{1, 2\}$.

Specifically, we compare IBFFD with the primal-dual interior point method (PDIP) (Kim et al., 2009) and with an alternating direction method of multipliers (ADMM) (Ramdas et al., 2016), which are popular methods for trend filtering with fixed $\beta$. The performance of PDIP depends on its log barrier update as

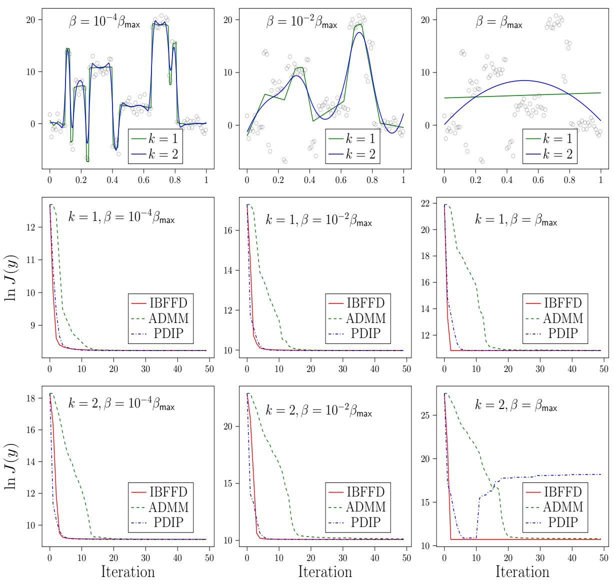

Figure 3: Comparison of different algorithms for trend filtering of the "blocks" data of Donoho et al. (1995) for $k = 1$ and $k = 2$. Top row: raw data $\breve{y}$ (circles) and estimate $y$ from IBFFD. Middle and bottom rows: convergence of the fitting cost $\ln J(y)$ for different values of $\beta$.

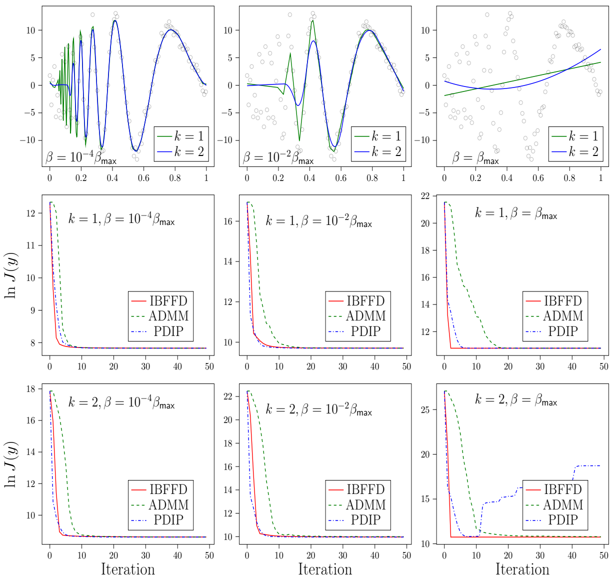

Figure 4: Same as Fig. 3 for the "Doppler" data of Donoho et al. (1995).

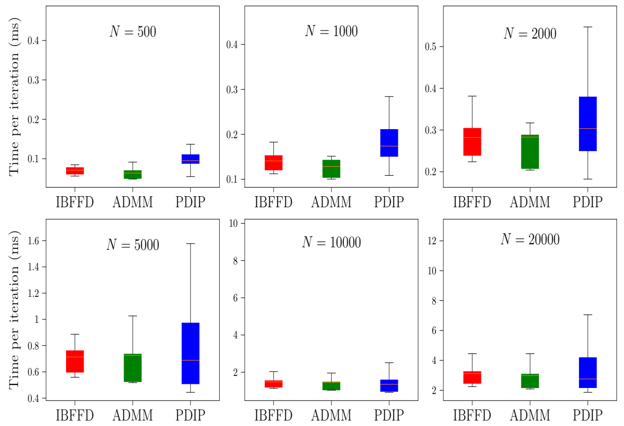

Figure 5: Box plots of running time (ms) per iteration of IBFFD, ADMM, and PDIP. For each N, the time per iteration was averaged over a total of 1000 repetitions: two choices of underlying functions (blocks and Doppler), two choices of $k = 1, 2$, five values of $\beta = 10^{-i}\beta_{\max}(i = -4, \cdots, 0)$, and 50 repetitions of each combination.

well as the backtracking line search; the speed of convergence of ADMM depends on the choice of the augmented Lagrangian parameter.

For experiments in this part, we used the C versions of PDIP and ADMM from Koh et al. (2008) and Arnold et al. (2014), respectively. The IBFFD of this paper was implemented in C as well[1]. The pertinent hyper-parameters of PDIP and ADMM are the default choices suggested by their authors. No hyper-parameters are required by IBFFD (with fixed $\beta$).

Some experimental results are given in Figs. 3–5. Figs. 3 and 4 show the convergence behaviour of these algorithms for the "blocks" and "Doppler" data of Donoho et al. (1995), both with $N = 2048$ and noisy observations with $\text{SNR} = 49$.

In these simulations, IBFFD converges faster than ADMM and never slower than PDIP. Moreover, IBFFD is more robust than PDIP for $k = 2$ and $\beta \approx \beta_{\max}$.

Fig. 5 shows the time per iteration of these algorithms, which are rather similar. (The large variance of the running time of PDIP is due to its unstable backtracking line search.) IRLGE (not shown here) has essentially the same running time per iteration as IBFFD, but it requires more iterations.

[1] https://github.com/yunpli2sp/NUP4SSM/

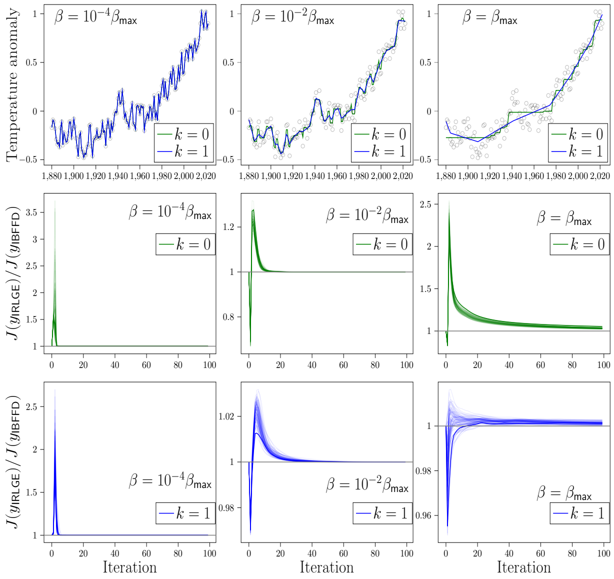

Figure 6: Shape-constrained regression of the global warming dataset. Top row: raw data $\breve{y}$ (circles) and estimated $y$ from IBFFD. Middle and bottom rows: the ratio $J(y_{\mathsf{IRLGE}})/J(y_{\mathsf{IBFFD}})$ of the fitting costs of IRLGE and IBFFD.

### 4.1.2 Encouraging Shape Constraints with Hinge Loss Prior

In this section, we demonstrate the use of the hinge-loss $\kappa(u_n) = (-u_n)_+$, and we compare IBFFD (with fixed $\beta$) with IRLGE as in Loeliger et al. (2016). For $k = 0$, $\kappa(u_n) = (-y_{n+1}+y_n)_+$ encourages the estimate $y$ to be monotonously increasing; for $k = 1$, $\kappa(u_n) = (-y_{n+2} + 2y_{n+1} - y_n)_+$ encourages the estimate $y$ to be convex.

Following Tibshirani et al. (2011), we here use the global warming datasets containing the annual temperature anomalies from 1880 to 2022 (NOAA, 2023).

Some pertinent simulation results are shown in Fig. 6. In the top row, we see the progression from interpolation (for small $\beta$) to constraint satisfaction (for $\beta = \beta_{\max}$). The middle and bottom rows of Fig. 6 summarize 100 experiments for each choice of $\beta$, with random initial variances $\overrightarrow{V}_{U_n}$; for each such experiment, the cost ratio $J(y_{\mathsf{IRLGE}})/J(y_{\mathsf{IBFFD}})$ is recorded for every iteration.

We observe that $J(y_{\mathsf{IRLGE}})/J(y_{\mathsf{IBFFD}}) \geq 1$ holds almost always, and sometimes $J(y_{\mathsf{IRLGE}})/J(y_{\mathsf{IBFFD}}) > 1$ even after convergence. We therefore conclude that (i) IBFFD converges faster than IRLGE and (ii) IRLGE may fail to converge to the actual minimum; the latter issue is due to zero-variance stucking (Section 2.3) and avoided by IBFFD.

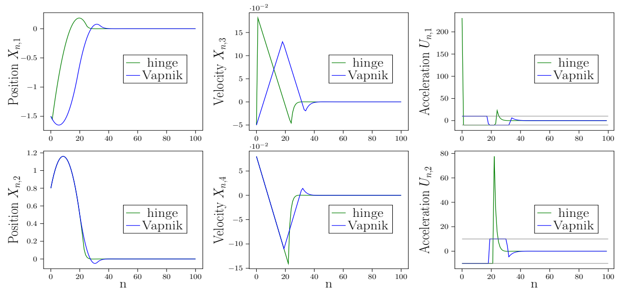

Figure 7: States $X_n$ and inputs $U_n$ of the example in Section 4.2, for two different constraints on the inputs. Green: $U_{n,\ell} \geq -10$; blue: $U_{n,\ell} \in [-10, 10]$.

### 4.2 Control with Linearly Constrained Inputs

Given a dynamical system (1), a desired final state $x_f \in \mathbb{R}^M$, and a cost function

$$J(u_{1:N}) \triangleq (x_{N+1} - x_f)^{\mathsf{T}} Q_f \, (x_{N+1} - x_f) + \sum_{n=1}^{N} (x_n - x_f)^{\mathsf{T}} Q \, (x_n - x_f) \qquad (40)$$

with positive definite matrices $Q$ and $Q_f$, consider the problem of computing an input signal $u_{1:N}$ that minimizes (40) subject to $u_{n,\ell} \geq a$, or subject to $a \leq u_{n,\ell} \leq b$, for all $n$ and $\ell$.

The hinge loss prior and the Vapnik prior can be used to solve such problems using IRLGE as in Keusch (2023). However, as mentioned in Section 3.4, IBFFD offers an attractive alternative by allowing an individual factor $\beta_{n,\ell}$ for each scalar input $u_{n,\ell}$, and increasing it, if necessary, during the forward recursion according to (35) or (36).

For this application, we modify Algorithm 1 as follows:

1. Since $x_1$ is given, the matrix computation (30) is omitted.
2. The backward filtering begins with $\overleftarrow{W}_{X_{N+1}} = Q_f$ and $\overleftarrow{\xi}_{X_{N+1}} = Q_f x_f$.
3. Equation (29) is changed to

$$\begin{aligned} \overleftarrow{\xi}_{X_n} &= A^{\mathsf{T}} \overleftarrow{\xi}_{X^0_{n+1}} + Q x_f, \\ \overleftarrow{W}_{X_n} &= A^{\mathsf{T}} \overleftarrow{W}_{X^0_{n+1}} A + Q. \end{aligned} \qquad (41)$$

The proposed algorithm works very well, as illustrated by the following example. Consider a object moving

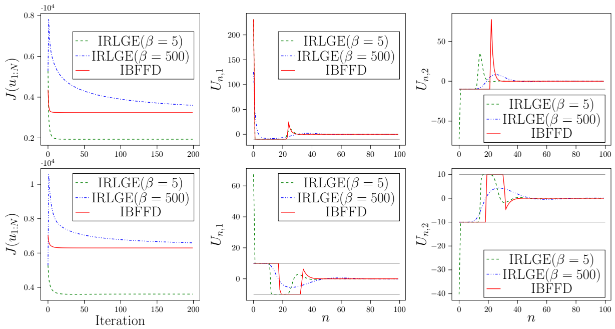

Figure 8: Comparing IBFFD and IRLGE in linearly constrained control. Top row: $U_{n,\ell} \geq -10$ with hinge loss; bottom row: $U_{n,\ell} \in [-10, 10]$ with Vapnik loss.

in 2D according to

$$\underbrace{\begin{pmatrix} X_{n+1,1} \\ X_{n+1,2} \\ X_{n+1,3} \\ X_{n+1,4} \end{pmatrix}}_{X_{n+1}} = \underbrace{\begin{pmatrix} 1 & 0 & \tau & 0 \\ 0 & 1 & 0 & \tau \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{A} X_n + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \alpha & 0 \\ 0 & \alpha \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} U_{n,1} \\ U_{n,2} \end{pmatrix}}_{U_n}, \tag{42}$$

where $(X_{n+1,1}, X_{n+1,2}) \in \mathbb{R}^2$ is the position, $(X_{n+1,3}, X_{n+1,4}) \in \mathbb{R}^2$ is the velocity, and the input $(U_{n,1}, U_{n,2}) \in \mathbb{R}^2$ is the acceleration.

The results of a pertinent numerical example are shown in Figure 7. The details are as follows: the given initial state is $x_1 = (-1.5, 0.8, -0.05, 0.08)^\mathsf{T}$, the desired final state is $x_f = (0, 0, 0, 0)^\mathsf{T}$, $\tau = 1, \alpha = 10^{-3}, N = 100$, $Q = 10^2 I$, and $Q_f = 10^8 I$ (where $I$ is the identity matrix).

As shown in Figure 7, the latent state $X_n$ is successfully steered from $x_1$ to $x_f$ using duly constrained control inputs.

A comparison of IBFFD with IRLGE (in the same setting as Figure 7) is shown in Figure 8. IRLGE normally works with a fixed global $\beta$ that needs to be adjusted: if $\beta$ is too small, the constraints are not satisfied, and if $\beta$ is large, convergence is slow. By contrast, IBFFD (as discussed in Section 3.4) guarantees constraint satisfaction and converges faster.

# 5 CONCLUSION

We proposed an iterated BFFD (backward filtering forward deciding) algorithm for MAP estimation in linear state space models with NUP priors, with a focus on non-Gaussian input estimation. Compared with prior work, the proposed approach avoids zero-variance stucking, converges faster, and makes constraint satisfaction much easier. These advantages were demonstrated with several examples, all with convex cost functions; for nonconvex cost functions (e.g., for a binarizing prior $p(u_n)$), the deciding rule (21) can be too agressive and the method of (Keusch et al., 2021, 2024) may still perform better.

# References

A. Y. Aravkin, J. V. Burke, and G. Pillonetto, “Optimization viewpoint on Kalman smoothing with applications to robust and sparse estimation,” *Compressed Sensing & Sparse Filtering*, pp. 237–280, 2014.

T. B. Arnold, V. Sadhanala, and R. J. Tibshirani, `glmgen`: Fast algorithms for generalized lasso problems, September 2014. URL `https://github.com/glmgen`.

F. Bach, R. Jenatton, J. Mairal, G. Obozinski, “Optimization with Sparsity-Inducing Penalties,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

L. Bruderer and H.-A. Loeliger. “Estimation of sensor input signals that are neither bandlimited nor sparse,” in *2014 Information Theory and Applications Workshop (ITA)*, pp. 1–5, 2014.

R. Bellman, “The theory of dynamic programming,” *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954.

C. De Boor, *A Practical Guide to Splines*, vol. 27. Springer-Verlag, New York, 1978.

D. L. Donoho and I. M. Johnstone. “Adapting to unknown smoothness via wavelet shrinkage,” *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*, OUP Oxford, 2012.

R. C. Fraser, *A New Technique For the Optimal Smoothing of Data.* PhD thesis, Massachusetts Institute of Technology, 1967.

A. Guntuboyina and B. Sen, “Nonparametric shape-restricted regression,” *Statistical Science*, vol. 33, no. 4, pp. 568–594, 2018.

G. Gakis and M. C. Smith, “A limit Kalman filter and smoother for systems with unknown inputs,” *Int. Journal of Control*, vol. 97, no. 3, pp. 532–542, 2024.

J. Glover, “The linear estimation of completely unknown signals,” *IEEE Trans. on Automatic Control*, vol. 14, no. 6, pp. 766–767, 1969.

R. Giri and B. Rao, “Type I and Type II Bayesian methods for sparse signal recovery using scale mixtures,” *IEEE Trans. on Signal Processing*, vol. 64, no. 13, pp. 3418–3428, 2016.

R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

R. Keusch, H. Malmberg, and H.-A. Loeliger, "Binary control and digital-to-analog conversion using composite NUV priors and iterative Gaussian message passing," in *2021 IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5330–5334, IEEE, 2021.

R. Keusch, *Composite NUV Priors and Applications.* PhD thesis, ETH Zurich, 2023.

R. Keusch, H.-A. Loeliger, and T. Geyer, "Long-horizon direct model predictive control for power converters with state constraints," *IEEE Trans. on Control Systems Technology*, vol. 32, pp. 340–350, 2024.

S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "$\ell_1$ Trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

K. Koh, S.-J. Kim, and S. Boyd, `l1_tf`: Software for l1 Trend Filtering, May 2008. URL `http://stanford.edu/~boyd/l1_tf/`.

R. Kohn, C. F. Ansley, and C.-M. Wong, "Nonparametric spline regression with autoregressive moving average errors," *Biometrika*, vol. 79, no. 2, pp. 335–346, 1992.

H.-A. Loeliger, J. Dauwels, Junli Hu, S. Korl, Li Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE,* vol. 95, no. 6, pp. 1295–1322, June 2007.

H.-A. Loeliger, L. Bruderer, H. Malmberg, F. Wadehn, and N. Zalmai, "On sparsity by NUP-EM, Gaussian message passing, and Kalman smoothing," *2016 Information Theory and Applications Workshop*, pp. 1–10, 2016.

H.-A. Loeliger, "On NUP priors and Gaussian message passing," *IEEE Int. Workshop on Machine Learning for Signal Processing*, 2023.

H.-A. Loeliger, B. Ma, H. Malmberg, and F. Wadehn, "Factor graphs with NUP priors and iteratively reweighted descent for sparse least squares and more," *2018 IEEE 10th Int. Symposium on Turbo Codes & Iterative Information Processing*, pp. 1–5, 2018.

D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société mathématique de France*, vol. 93, pp. 273–299, 1965.

NOAA National Centers for Environmental information, "Climate at a Glance: Global Time Series, published October 2023," retrieved on October 16, 2023 from `https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/global/time-series`, 2023.

C. A. Politsch, J. Cisewski–Kehe, R. A. Croft, and L. Wasserman, "Trend filtering – I. A modern statistical tool for time-domain astronomy and astronomical spectroscopy," *Monthly Notices of the Royal Astronomical Society*, vol. 492, no. 3, pp. 4005–4018, 2020.

J. Palmer, K. Kreutz–Delgado, B. Rao, and D. Wipf, "Variational EM algorithms for non-Gaussian latent variable models," in *Advances in Neural Information Processing Systems*, 2006.

N. Parikh, S. Boyd, "Proximal Algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

A. Ramdas and R. J. Tibshirani, "Fast and flexible ADMM algorithms for trend filtering," *Journal of Computational and Graphical Statistics*, vol. 25, no. 3, pp. 839–858, 2016.

A. K. Roonizi, "$\ell_2$ and $\ell_1$ Trend filtering: a Kalman filter approach [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 38, no. 6, pp. 137–145, 2021.

A. K. Roonizi, "Kalman filtering in non-Gaussian model errors: a new perspective," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 105–114, 2022.

G. Steidl, S. Didas, and J. Neumann, "Splines in higher order TV regularization," *Int. Journal of Computer Vision*, vol. 70, no. 3, pp. 241–255, 2006.

M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Int. Workshop on Artificial Intelligence and Statistics*, pp. 276–283, 2003.

R. J. Tibshirani, H. Hoefling, and R. Tibshirani, "Nearly-Isotonic Regression," *Technometrics*, vol. 53, no. 1, pp. 54–61, 2011.

## Checklist

1. For all models and algorithms presented, check if you include:
   - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Not Applicable
2. For any theoretical claim, check if you include:
   - (a) Statements of the full set of assumptions of all theoretical results. Not Applicable
   - (b) Complete proofs of all theoretical results. Not Applicable
   - (c) Clear explanations of any assumptions. Not Applicable
3. For all figures and tables that present empirical results, check if you include:
   - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). No: the code is not presently included or publicly available
   - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Not Applicable
   - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes
   - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). No: not relevant
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   - (a) Citations of the creator If your work uses existing assets. Yes
   - (b) The license information of the assets, if applicable. No: we use only publicly available data and code
   - (c) New assets either in the supplemental material or as a URL, if applicable. No
   - (d) Information about consent from data providers/curators. Not Applicable
   - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   - (a) The full text of instructions given to participants and screenshots. Not Applicable
   - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

# Backward Filtering Forward Deciding in Linear Non-Gaussian State Space Models: Supplementary Materials

## A MORE ABOUT NUP PRIORS

For selected NUP priors in Table 1, we discuss their variation representations and pertinent updating rules for unknown parameter $\theta_n$ in this Section. Their densities $p(u_n)$ (up to a scalar factor) and corresponding penalty functions $-\ln p(u_n)$ (up to a additive constant) are shown in Figure 9.

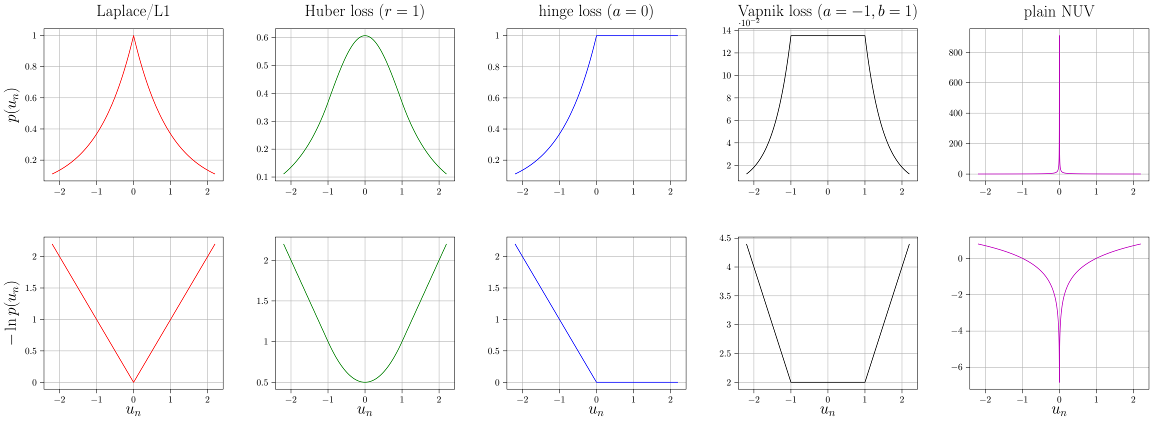

Figure 9: The densities $p(u_n)$ and penalty functions $-\ln p(u_n)$ of selected scalar NUP priors.

### A.1 Laplace/L1

Considering the variational representation of Laplace/L1 density in

$$\begin{aligned} p(u_n) &\propto \exp\left[-\beta|u_n|\right] \\ &= \max_{\sigma_n^2} \exp\left[-\frac{u_n^2}{2\sigma_n^2} - \frac{\beta^2\sigma_n^2}{2}\right] \\ &= \max_{\sigma_n^2} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{u_n^2}{2\sigma_n^2}\right] \sqrt{2\pi\sigma_n^2} \exp\left[-\frac{\beta^2\sigma_n^2}{2}\right] \\ &= \max_{\theta_n} \mathcal{N}(u_n;\theta_n) g(\theta_n), \end{aligned} \tag{43}$$

where unkown parameter $\theta_n$ contains mean $\overrightarrow{m}_{U_n} = 0$, variance $\overrightarrow{V}_{U_n} = \sigma_n^2$. Given $u_n = \hat{u}_n$, closed-form updates for (15) become

$$\overrightarrow{m}_{U_n} = 0, \quad \overrightarrow{V}_{U_n} = |\hat{u}_n|/\beta. \tag{44}$$

### A.2 Huber loss

When $|u_n| > \beta r^2$, Huber loss shares a same variation representation with Laplace/L1, and its closed-form updates for (15) is (44).

When $|u_n| \leq \beta r^2$, the pertinent density of Huber loss is

$$\begin{aligned} p(u_n) &\propto \exp\left[-\frac{u_n^2}{2r^2} - \frac{\beta^2 r^2}{2}\right] \\ &= \frac{1}{\sqrt{2\pi r^2}} \exp\left[-\frac{u_n^2}{2r^2}\right] \sqrt{2\pi r^2} \exp\left[-\frac{\beta^2 r^2}{2}\right] \\ &= \max_{\theta_n} \mathcal{N}(u_n; \theta_n) g(\theta_n), \end{aligned} \tag{45}$$

where unknown parameter $\theta_n$ contains fixed mean $\overrightarrow{m}_{U_n} = 0$, fixed variance $\overrightarrow{V}_{U_n} = r^2$, and they are closed-form updates for (15). Given $u_n = \hat{u}_n$, thus the update rules for Huber loss are

$$\begin{cases} \overrightarrow{m}_{U_n} = 0, \quad \overrightarrow{V}_{U_n} = r^2, & |\hat{u}_n| \leq \beta r^2, \\ \overrightarrow{m}_{U_n} = 0, \quad \overrightarrow{V}_{U_n} = |\hat{u}_n|/\beta, & |\hat{u}_n| > \beta r^2. \end{cases} \tag{46}$$

### A.3 Hinge loss

For hinge loss, we consider its variational representation

$$\begin{aligned} p(u_n) &\propto \exp\left[-\beta(a - u_n)_+\right] \\ &= \exp\left[-\frac{\beta}{2}(|u_n - a| + a - u_n)\right] \\ &= \max_{\sigma_n^2} \exp\left[-\frac{(u_n - a)^2}{2\sigma_n^2} - \frac{\beta^2 \sigma_n^2}{8} - \frac{\beta}{2}(a - u_n)\right] \\ &= \max_{\sigma_n^2} \exp\left[-(u_n - a - \frac{\beta\sigma_n^2}{2})^2/(2\sigma_n^2)\right] \\ &= \max_{\sigma_n^2} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-(u_n - a - \frac{\beta\sigma_n^2}{2})^2/(2\sigma_n^2)\right] \sqrt{2\pi\sigma_n^2} \\ &= \max_{\theta_n} \mathcal{N}(u_n; \theta_n) g(\theta_n), \end{aligned} \tag{47}$$

where the mean $\overrightarrow{m}_{U_n}$ and the variance $\overrightarrow{V}_{U_n}$ contained in the unknown parameter $\theta_n$ are

$$\overrightarrow{m}_{U_n} = a + \frac{\beta\sigma_n^2}{2}, \quad \overrightarrow{V}_{U_n} = \sigma_n^2. \tag{48}$$

Given $u_n = \hat{u}_n$, closed-form updates for (15) become

$$\overrightarrow{m}_{U_n} = a + |\hat{u}_n - a|, \quad \overrightarrow{V}_{U_n} = 2|\hat{u}_n - a|/\beta. \tag{49}$$

### A.4 Vapnik loss

For the density of Vapnik loss, we consider its variational representation as below

$$\begin{aligned} p(u_n) &\propto \exp\left[-\beta(|u_n - a| + |u_n - b|)\right] \\ &= \max_{\sigma_{n,a}^2} \exp\left[-\frac{(u_n - a)^2}{2\sigma_{n,a}^2} - \frac{\beta^2\sigma_{n,a}^2}{2}\right] \cdot \max_{\sigma_{n,b}^2} \exp\left[-\frac{(u_n - b)^2}{2\sigma_{n,b}^2} - \frac{\beta^2\sigma_{n,b}^2}{2}\right] \\ &= \max_{\sigma_{n,a}^2} \max_{\sigma_{n,b}^2} \frac{1}{\sqrt{2\pi\sigma_{n,a}^2}} \exp\left[-\frac{(u_n - a)^2}{2\sigma_{n,a}^2}\right] \frac{1}{\sqrt{2\pi\sigma_{n,b}^2}} \exp\left[-\frac{(u_n - b)^2}{2\sigma_{n,b}^2}\right] \cdot (2\pi\sqrt{\sigma_{n,a}^2\sigma_{n,b}^2}) \exp\left[-\beta\frac{(\sigma_{n,a}^2 + \sigma_{n,b}^2)}{2}\right] \\ &= \max_{\theta_n} \mathcal{N}(u_n; \theta_n) g(\theta_n), \end{aligned} \tag{50}$$

where the mean $\overrightarrow{m}_{U_n}$, variance $\overrightarrow{V}_{U_n}$ in $\theta_n$ satisfy

$$\frac{\overrightarrow{m}_{U_n}}{\overrightarrow{V}_{U_n}} = \frac{a}{\sigma_{n,a}^2} + \frac{b}{\sigma_{n,b}^2}, \quad \frac{1}{\overrightarrow{V}_{U_n}} = \frac{1}{\sigma_{n,a}^2} + \frac{1}{\sigma_{n,b}^2}, \tag{51}$$

and the unknown parameter $\theta_n$ contains

$$\overrightarrow{m}_{U_n} = \frac{a\sigma_{n,b}^2 + b\sigma_{n,a}^2}{\sigma_{n,a}^2 + \sigma_{n,b}^2}, \quad \overrightarrow{V}_{U_n} = \frac{\sigma_{n,a}^2\sigma_{n,b}^2}{\sigma_{n,a}^2 + \sigma_{n,b}^2}. \tag{52}$$

Given $u_n = \hat{u}_n$, we maximize $p(u_n)$ by selecting

$$\sigma_{n,a}^2 = |\hat{u}_n - a|/\beta, \quad \sigma_{n,b}^2 = |\hat{u}_n - b|/\beta, \tag{53}$$

and plug them into (52) to obtain

$$\overrightarrow{m}_{U_n} = \frac{a|\hat{u}_n - b| + b|\hat{u}_n - a|}{|\hat{u}_n - a| + |\hat{u}_n - b|}, \quad \overrightarrow{V}_{U_n} = \frac{|\hat{u}_n - a||\hat{u}_n - b|}{\beta\left[|\hat{u}_n - a| + |\hat{u}_n - b|\right]} \tag{54}$$

for (15).

### A.5 Plain NUV

For plain NUV, its desnity can be expressed as

$$\begin{aligned} p(u_n) &\propto \exp\left[-\ln|u_n|\right] \\ &\propto \max_{\sigma_n^2} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{u_n^2}{2\sigma_n^2}\right] \\ &= \mathcal{N}(u_n;\theta_n)g(\theta_n), \end{aligned} \tag{55}$$

where unknown parameter contains mean $\overrightarrow{m}_{U_n} = 0$, variance $\overrightarrow{V}_{U_n} = \sigma_n^2$. Given $u_n = \hat{u}_n$, closed-form updates for (15) become

$$\overrightarrow{m}_{U_n} = 0, \quad \overrightarrow{V}_{U_n} = \hat{u}_n^2. \tag{56}$$

## B INPUT ESTIMATION IN IRLGE

To apply message passing involving $U_n$ in Figure 1, we consider its local part shown in Figure 10, where backward mean $\overleftarrow{m}_{U_n}$, backward variacne $\overleftarrow{V}_{U_n}$ come from $\overleftarrow{\mu}_{U_n}(u_n)$, and forward mean $\overrightarrow{m}_{U_n}$, forward variance $\overrightarrow{V}_{U_n}$ come from NUP prior $p(u_n)$.

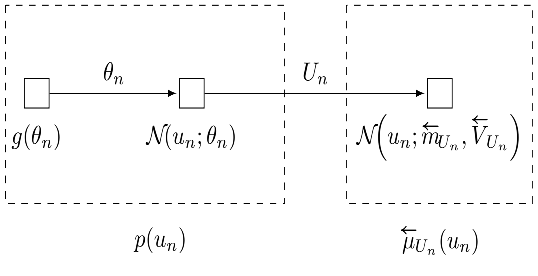
$\theta_n$
$U_n$
$g(\theta_n)$
$\mathcal{N}(u_n;\theta_n)$
$\mathcal{N}\left(u_n;\overleftarrow{m}_{U_n},\overleftarrow{V}_{U_n}\right)$
$p(u_n)$
$\overleftarrow{\mu}_{U_n}(u_n)$

Figure 10: Local part of system model involving $U_n$. $p(u_n)$ is the NUP prior, and $\overleftarrow{\mu}_{U_n}(u_n)$ is the Gaussian message collected from the rest of Figure 1.

For single $U_n$, we compute

$$\hat{u}_n = \operatorname*{argmax}_{u_n} p(u_n)\overleftarrow{\mu}_{U_n}(u_n) \tag{57}$$

via IRLGE, and the iterative two steps are:

1. For fixed $\theta_n = \hat{\theta}_n = (\overrightarrow{m}_{U_n}, \overrightarrow{V}_{U_n})$, computing

$$\begin{aligned} \hat{u}_n &= \underset{u_n}{\operatorname{argmax}}\, p(u_n)\overleftarrow{\mu}_{U_n}(u_n) \\ &= \underset{u_n}{\operatorname{argmax}}\, \mathcal{N}(u_n; \hat{\theta}_n) g(\hat{\theta}_n))\overleftarrow{\mu}_{U_n}(u_n) \\ &= \underset{u_n}{\operatorname{argmax}}\, \mathcal{N}(u_n; \hat{\theta}_n)\overleftarrow{\mu}_{U_n}(u_n) \\ &= \underset{u_n}{\operatorname{argmax}}\, \mathcal{N}(u_n; \overrightarrow{m}_{U_n}, \overrightarrow{V}_{U_n})\mathcal{N}(u_n; \overleftarrow{m}_{U_n}, \overleftarrow{V}_{U_n}) \end{aligned} \tag{58}$$

   yields

$$\hat{u}_n = \frac{\overrightarrow{m}_{U_n}\overleftarrow{V}_{U_n} + \overleftarrow{m}_{U_n}\overrightarrow{V}_{U_n}}{\overrightarrow{V}_{U_n} + \overleftarrow{V}_{U_n}}. \tag{59}$$

2. For fixed $u_n = \hat{u}_n$, update $\hat{\theta}_n$ same to equation (15).

According to (59), the description for zero-variance stucking becomes much clear: once $\overrightarrow{V}_{U_n}$ becomes zero owing to certain $\hat{u}_n^{(k)}$ at $k$-th iteration, the likelihood in $\mathcal{N}(u_n; \hat{\theta}_n)$ become unbounded, and $\hat{u}_n$ gets stuck with $\overrightarrow{m}_{U_n} = \hat{u}_n^{(k)}$ in subsequent iterations.

## C DERIVATIONS OF DECIDING RULES

To avoid zero-variance stucking, the deciding rule in Table 2 can derived by considering (59)(15) simultaneously, and the obtained fixed point $\hat{u}_n$ is the solution to (57). Given fixed $\overleftarrow{m}_{U_n}, \overleftarrow{V}_{U_n}$, deciding rule in (57) selects the best $\hat{u}_n$ to maximize the joint MAP probability compared with the limited increase caused by (59) used in IRLGE,

We next use the $\overrightarrow{m}_{U_n}, \overrightarrow{V}_{U_n}$ in Table 1 and (59) to derive deciding rules for the selected NUP priors.

### C.1 Laplace/L1

(59) becomes

$$\hat{u}_n = \frac{|\hat{u}_n|\overleftarrow{m}_{U_n}}{\beta\overleftarrow{V}_{U_n} + |\hat{u}_n|}, \tag{60}$$

therefore, the fixed point $\hat{u}_n$ is the solution of

$$\hat{u}_n|\hat{u}_n| + \beta\overleftarrow{V}_{U_n}\hat{u}_n - \overleftarrow{m}_{U_n}|\hat{u}_n| = 0, \tag{61}$$

which idicates that

$$\begin{aligned} &\hat{u}_n < 0, -\hat{u}_n^2 + \beta\overleftarrow{V}_{U_n}\hat{u}_n + \overleftarrow{m}_{U_n}\hat{u}_n = 0 &&\Longleftrightarrow \hat{u}_n = \overleftarrow{m}_{U_n} + \beta\overleftarrow{V}_{U_n} < 0 &&\Rightarrow \overleftarrow{m}_{U_n} < -\beta\overleftarrow{V}_{U_n}, \\ &\hat{u}_n > 0, \hat{u}_n^2 + \beta\overleftarrow{V}_{U_n}\hat{u}_n - \overleftarrow{m}_{U_n}\hat{u}_n = 0 &&\Longleftrightarrow \hat{u}_n = \overleftarrow{m}_{U_n} - \beta\overleftarrow{V}_{U_n} > 0 &&\Rightarrow \overleftarrow{m}_{U_n} > \beta\overleftarrow{V}_{U_n}. \end{aligned} \tag{62}$$

When $\hat{u}_n = 0$, we have

$$\hat{u}_n = 0 \Rightarrow |\overleftarrow{m}_{U_n}| \leq \beta\overleftarrow{V}_{U_n}. \tag{63}$$

### C.2 Huber loss

(59) becomes

$$\hat{u}_n = \begin{cases} \frac{r^2\overleftarrow{m}_{U_n}}{\overleftarrow{V}_{U_n} + r^2}, & |\hat{u}_n| \leq \beta r^2, \\ \frac{|\hat{u}_n|\overleftarrow{m}_{U_n}}{\beta\overleftarrow{V}_{U_n} + |\hat{u}_n|}, & |\hat{u}_n| > \beta r^2. \end{cases} \tag{64}$$

Similar to Laplace/L1, when $|\hat{u}_n| > \beta r^2$, the fixed point $\hat{u}_n$ satisifies following rules

$$\begin{aligned} \hat{u}_n < -\beta r^2, -\hat{u}_n^2 + \beta\overleftarrow{V}_{U_n}\hat{u}_n + \overleftarrow{m}_{U_n}\hat{u}_n = 0 \quad &\iff \hat{u}_n = \overleftarrow{m}_{U_n} + \beta\overleftarrow{V}_{U_n} < -\beta r^2 \quad \Rightarrow \overleftarrow{m}_{U_n} < -\beta(\overleftarrow{V}_{U_n} + r^2), \\ \hat{u}_n > \beta r^2, \hat{u}_n^2 + \beta\overleftarrow{V}_{U_n}\hat{u}_n - \overleftarrow{m}_{U_n}\hat{u}_n = 0 \quad &\iff \hat{u}_n = \overleftarrow{m}_{U_n} - \beta\overleftarrow{V}_{U_n} > \beta r^2 \quad \Rightarrow \overleftarrow{m}_{U_n} > \beta(\overleftarrow{V}_{U_n} + r^2). \end{aligned} \tag{65}$$

Additionally, for $|\hat{u}_n| \leq \beta r^2$, we have

$$\hat{u}_n = \frac{r^2\overleftarrow{m}_{U_n}}{\overleftarrow{V}_{U_n} + r^2} \Rightarrow |\overleftarrow{m}_{U_n}| \leq \beta(\overleftarrow{V}_{U_n} + r^2). \tag{66}$$

### C.3 Hinge loss

(59) becomes

$$\hat{u}_n = \frac{\beta(a + |\hat{u}_n - a|)\overleftarrow{V}_{U_n} + 2|\hat{u}_n - a|\overleftarrow{m}_{U_n}}{\beta\overleftarrow{V}_{U_n} + 2|\hat{u}_n - a|}, \tag{67}$$

therefore, the fixed point $\hat{u}_n$ is the solution of

$$2|\hat{u}_n - a|\hat{u}_n + \beta\overleftarrow{V}_{U_n}\hat{u}_n - \beta(a + |\hat{u}_n - a|)\overleftarrow{V}_{U_n} - 2|\hat{u}_n - a|\overleftarrow{m}_{U_n} = 0, \tag{68}$$

which indicates that

$$\begin{aligned} \hat{u}_n < a, 2|\hat{u}_n - a|\hat{u}_n + \beta\overleftarrow{V}_{U_n}\hat{u}_n - \beta(a + |\hat{u}_n - a|)\overleftarrow{V}_{U_n} - 2|\hat{u}_n - a|\overleftarrow{m}_{U_n} = 0 \iff \\ \hat{u}_n < a, (\hat{u}_n - a)(\hat{u}_n - \beta\overleftarrow{V}_{U_n} - \overleftarrow{m}_{U_n}) = 0 \iff \\ \hat{u}_n = \overleftarrow{m}_{U_n} + \beta\overleftarrow{V}_{U_n} < a \Rightarrow \overleftarrow{m}_{U_n} < -\beta\overleftarrow{V}_{U_n} + a; \\ \hat{u}_n > a, 2|\hat{u}_n - a|\hat{u}_n + \beta\overleftarrow{V}_{U_n}\hat{u}_n - \beta(a + |\hat{u}_n - a|)\overleftarrow{V}_{U_n} - 2|\hat{u}_n - a|\overleftarrow{m}_{U_n} = 0 \iff \\ \hat{u}_n > a, (\hat{u}_n - a)(\hat{u}_n - \overleftarrow{m}_{U_n}) = 0 \iff \\ \hat{u}_n = \overleftarrow{m}_{U_n} > a \Rightarrow \overleftarrow{m}_{U_n} > a. \end{aligned} \tag{69}$$

When $\hat{u}_n = a$, we have

$$\hat{u}_n = a \Rightarrow -\beta\overleftarrow{V}_{U_n} + a \leq \overleftarrow{m}_{U_n} \leq a. \tag{70}$$

### C.4 Vapnik loss

(59) becomes

$$\hat{u}_n = \frac{\beta a\overleftarrow{V}_{U_n}|\hat{u}_n - b| + \beta b\overleftarrow{V}_{U_n}|\hat{u}_n - a| + \overleftarrow{m}_{U_n}|\hat{u}_n - a||\hat{u}_n - b|}{|\hat{u}_n - a||\hat{u}_n - b| + \beta\overleftarrow{V}_{U_n}|\hat{u}_n - b| + \beta\overleftarrow{V}_{U_n}|\hat{u}_n - a|}, \tag{71}$$

therefore, the fixed point $\hat{u}_n$ is the solution of

$$(\hat{u}_n - \overleftarrow{m}_{U_n})|\hat{u}_n - a||\hat{u}_n - b| + \beta\overleftarrow{V}_{U_n}(\hat{u}_n - a)|\hat{u}_n - b| + \beta\overleftarrow{V}_{U_n}|\hat{u}_n - a|(\hat{u}_n - b) = 0. \tag{72}$$

If $\hat{u}_n < a$, (72) tells us

$$\hat{u}_n < a, (\hat{u}_n - \overleftarrow{m}_{U_n} - 2\beta\overleftarrow{V}_{U_n})(\hat{u}_n - a)(\hat{u}_n - b) = 0 \iff \hat{u}_n = \overleftarrow{m}_{U_n} + 2\beta\overleftarrow{V}_{U_n} < a \Rightarrow \overleftarrow{m}_{U_n} < -2\beta\overleftarrow{V}_{U_n} + a. \tag{73}$$

if $\hat{u}_n > b$, (72) tells us

$$\hat{u}_n > b, (\hat{u}_n - \overleftarrow{m}_{U_n} + 2\beta\overleftarrow{V}_{U_n})(\hat{u}_n - a)(\hat{u}_n - b) = 0 \iff \hat{u}_n = \overleftarrow{m}_{U_n} - 2\beta\overleftarrow{V}_{U_n} > b \Rightarrow \overleftarrow{m}_{U_n} > 2\beta\overleftarrow{V}_{U_n} + b. \tag{74}$$

if $a < \hat{u}_n < b$, (72) tells us

$$a < \hat{u}_n < b, (\hat{u}_n - \overleftarrow{m}_{U_n})(\hat{u}_n - a)(\hat{u}_n - b) = 0 \iff a < \hat{u}_n = \overleftarrow{m}_{U_n} < b \Rightarrow a < \overleftarrow{m}_{U_n} < b. \tag{75}$$

In addition, when $\hat{u}_n = a$ or $\hat{u}_n = b$, we have

$$\hat{u}_n = a \Rightarrow -2\beta\overleftarrow{V}_{U_n} + a \leq \overleftarrow{m}_{U_n} \leq a, \quad \hat{u}_n = b \Rightarrow b \leq \overleftarrow{m}_{U_n} \leq 2\beta\overleftarrow{V}_{U_n} + b. \tag{76}$$

### C.5 Plain NUV

Different from the joint MAP estimation used in previous derivations, we use Type-II estimation to form a decision $\hat{u}_n$ for plain NUV.

Considering the margin density $p(\sigma_n^2)$ in Figure 10

$$\begin{aligned} p(\sigma_n^2) &\propto \int_{-\infty}^{+\infty} \mathcal{N}(u_n;\theta_n) g(\theta_n) \overleftarrow{\mu}_{U_n}(u_n) du_n \\ &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{u_n^2}{2\sigma_n^2}\right] \frac{1}{\sqrt{2\pi\overleftarrow{V}_{U_n}}} \exp\left[-\frac{(u_n-\overleftarrow{m}_{U_n})^2}{2\overleftarrow{V}_{U_n}}\right] du_n \\ &\propto \frac{1}{\sqrt{2\pi(\sigma_n^2+\overleftarrow{V}_{U_n})}} \exp\left[-\frac{\overleftarrow{m}_{U_n}^2}{2(\sigma_n^2+\overleftarrow{V}_{U_n})}\right]. \end{aligned} \tag{77}$$

Maximizing (77) over $\sigma_n^2$ provides

$$\overrightarrow{V}_{U_n} = \hat{\sigma}_n^2 = \begin{cases} \overleftarrow{m}_{U_n}^2 - \overleftarrow{V}_{U_n} & \overleftarrow{m}_{U_n} < -\overleftarrow{V}_{U_n}^{\frac{1}{2}} \\ 0 & |\overleftarrow{m}_{U_n}| \leq \overleftarrow{V}_{U_n}^{\frac{1}{2}} \\ \overleftarrow{m}_{U_n}^2 - \overleftarrow{V}_{U_n} & \overleftarrow{m}_{U_n} > \overleftarrow{V}_{U_n}^{\frac{1}{2}} \end{cases}. \tag{78}$$

We next plug (78) to (59), and obtain

$$\hat{u}_n = \begin{cases} \overleftarrow{m}_{U_n} - \left[\overleftarrow{V}_{U_n}/\overleftarrow{m}_{U_n}\right] & \overleftarrow{m}_{U_n} < -\overleftarrow{V}_{U_n}^{1/2} \\ 0 & |\overleftarrow{m}_{U_n}| \leq \overleftarrow{V}_{U_n}^{1/2} \\ \overleftarrow{m}_{U_n} - \left[\overleftarrow{V}_{U_n}/\overleftarrow{m}_{U_n}\right] & \overleftarrow{m}_{U_n} > \overleftarrow{V}_{U_n}^{1/2} \end{cases}. \tag{79}$$