Signal Processing 242 (2026) 110443

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Short communication

Polynomial order selection for Savitzky-Golay smoothers via N-fold cross-validation

Çağatay Candan

Department of Electrical and Electronics Engineering, Bogazici University, 34342, Bebek Istanbul, Turkey

ARTICLE INFO

Keywords:
Savitzky-Golay filtering
Smoothing
Linear regression
Polynomial fit
Model order selection
Cross-validation.

ABSTRACT

Savitzky-Golay (SG) smoothers are noise suppressing filters operating on the principle of projecting noisy input onto the subspace of polynomials. A poorly selected polynomial order results in over- or under-smoothing which shows as either bias or excessive noise at the output. In this study, we apply the N-fold cross-validation technique (also known as leave-one-out cross-validation) for model order selection and show that the inherent analytical structure of the SG filtering problem, mainly its minimum norm formulation, enables an efficient and effective order selection solution. More specifically, a novel connection between the total prediction error and SG-projection spaces is developed to reduce the implementation complexity of cross-validation method. The suggested solution compares favorably with the state-of-the-art Bayesian Information Criterion (BIC) rule in non-asymptotic signal-to-noise ratio (SNR) and sample size regimes. MATLAB codes reproducing the numerical results are provided.

1. Introduction

Savitzky-Golay (SG) filters are used for smoothing noisy input while preserving local polynomial characteristics such as linear, quadratic or higher order variations [1–3]. SG filters are low-pass filters; however, unlike conventional filters, their spectral properties, such as cut-off frequency, pass-band ripple, side-lobe levels [4], are secondary to the time-domain properties.

SG filters operate on the principle of i) projecting input onto a subspace of polynomials in the least squares sense, ii) mapping the projection result to a scalar output by evaluating the projection result (smoothing or interpolation application [1]) or its derivative (differentiator application [3]) at a point, as discussed in [2, Sec. 8.3.5] and [3]. It is possible to express the cascade of these two operations as a finite impulse response (FIR), linear time-invariant (LTI) filter [3]. In many applications, the local characteristics of the signal are expected to be a polynomial function particularly in the instrumentation and measurement area [5,6].

Fig. 1 illustrates a smooth, piecewise-defined polynomial function that is discretized with a sampling period of $T = 0.5$ sec. The samples shown deviate from the function due to sampling noise. This function can be considered to represent the position of an object, say a satellite being tracked, which is at the origin at $t = 0$ with an initial velocity of 1 m/sec. The object maintains this velocity in the time interval of $t \in (0, 6)$. Over $t \in (6, 14)$, the object decelerates at a rate of $-0.125\,\mathrm{m/sec}^2$ and comes to a stop at 10 meters from its initial position. The signal processing goal in relation to this kinematic experiment is the estimation of velocity, acceleration and other motion parameters from noisy position measurements in Fig. 1. The numerical values above or below each time sample show the best polynomial order fitted to that particular data point via the suggested N-fold cross validation method. Here, the maximum possible polynomial order is chosen as 3. We observe that for $t \in (0, 6)$, the best fit is the 1st degree except at one sample. In the 2nd segment with de-acceleration, the chosen order changes according to the effect of noise on local curvature; but it generally corresponds well to a 2nd degree trend. In the final segment of $(14, 20)$, the order is correctly predicted in the majority of samples.

SG filters have only two design parameters, the processing window length and the maximum polynomial degree. The processing window length, which is also the filter impulse response length, should be selected as long as possible to admit a reliable estimation of the assumed polynomial variation. In Fig. 1, the processing window length is 5 samples, two neighboring samples on the immediate left and right of each target sample. In general, the choice of processing window length depends on the signal characteristics in relation with the signal bandwidth [7–9].

Here we focus on selecting the best polynomial order at a fixed window length, that is the goal is to select the best polynomial order for each segment/sample of the data, as in Fig. 1. The key theoretical contribution is the development of an expression between the total prediction error, which is the cross-validation metric, and the projection spaces of SG-smoothing filters. This key relation not only explains the observed

E-mail address: ccandan@gmail.com

https://doi.org/10.1016/j.sigpro.2025.110443
Received 22 May 2025; Received in revised form 3 December 2025; Accepted 8 December 2025
Available online 11 December 2025

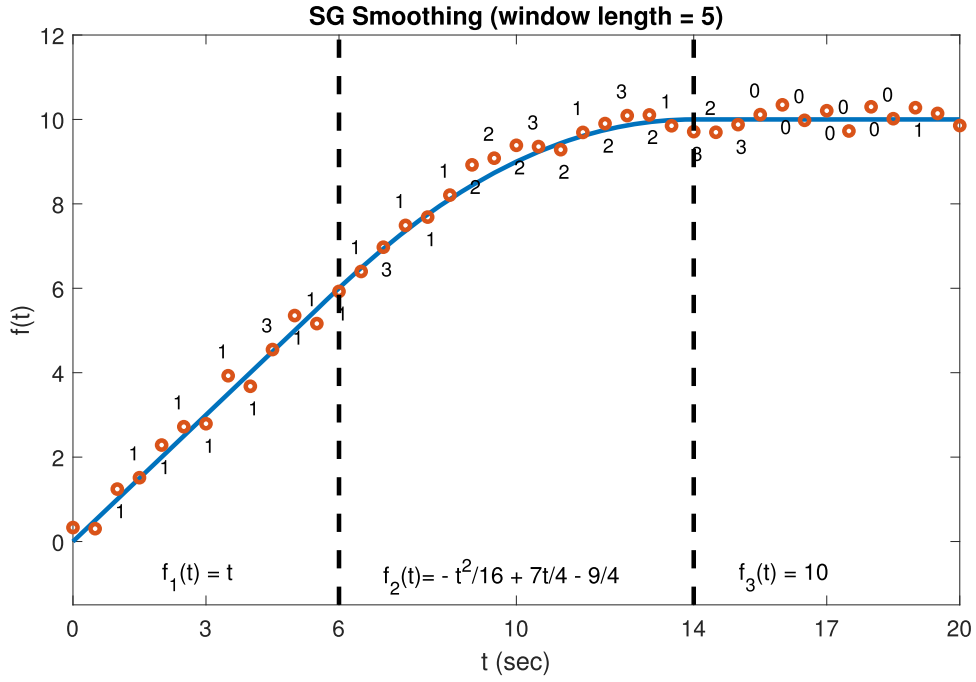

Fig. 1. An SG-smoothing example. The samples (circles) are off the piecewise-defined function due to noise. The numerical values above or below of each sample are the selected polynomial order with the cross-validation technique.

success of CV in this application and also reduces its implementation complexity significantly.

2. Background

SG-smoothing Filters: SG-smoothing filters can be designed as the solution of a minimum Euclidean norm problem [3]. As a concrete example, the SG smoother of length 5 for the 2nd degree polynomials can be formulated as the minimum norm solution of

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1^2 & 2^2 & 3^2 & 4^2 & 5^2 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} h_2 \\ h_1 \\ h_0 \\ h_{-1} \\ h_{-2} \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 3^2 \end{bmatrix}. \tag{1}$$

If we denote the equation system in (1) as $\mathbf{Ah} = \mathbf{b}$, the minimum Euclidean norm solution becomes $\mathbf{h}_{\text{MN}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$. Here $\mathbf{h}_{\text{MN}}$ is the impulse response of an FIR filter of length 5, called SG-smoother, that projects the input onto the 3 dimensional sub-space spanned by the 2nd degree polynomials. In this case, the convolution operation $y_n = \sum_{k=-2}^{2} x_{n-k}h_k$ produces the smoothed output for the central sample $x[n]$ in the processing window of $\{x_{n-2}, x_{n-1}, x_n, x_{n+1}, x_{n+2}\}$.

The rows of the $\mathbf{A}$ matrix in (1) are samples of the 0th, 1st and 2nd order polynomials in $[1, 5]$ with the unit sampling period. Observe that the 3 constraint equations in (1) enforce the mapping of constant ($f_0(t) = 1$), linear ($f_1(t) = t$) and quadratic ($f_2(t) = t^2$) function samples in $t \in [1, 5]$ to the value of the central sample of the function at $t = 3$. By linearity, an arbitrary 2nd degree polynomial $g(t) = \alpha_1 f_1(t) + \alpha_2 f_2(t) + \alpha_3 f_3(t)$ is mapped to $g(3)$ with the same transformation. Observe that any vector $\mathbf{h}$ vector satisfying the equation system in (1) establishes this mapping. What is unique about the minimum norm solution $\mathbf{h}_{\text{MN}}$ is that it is the only solution in the range space of $\mathbf{A}^T$ (i.e., row-space of $\mathbf{A}$ matrix). Since the row space of $\mathbf{A}$ is the sub-space of 2nd degree polynomials, this implies that the vector $\mathbf{h}_{\text{MN}}$ is orthogonal to all 3rd or higher degree polynomials; thus, performing the projection step in the two-step SG filtering procedure described in Section 1 [2, Sec. 8.3.5].

Table 1
5-fold cross-validation example with $x_n = 7n^2 - 3$.

case #	x_{-2}	x_{-1}	x_0	x_1	x_2
❶	?	4	-3	4	25
❷	25	?	-3	4	25
❸	25	4	?	4	25
❹	25	4	-3	?	25
❺	25	4	-3	4	?

Cross-Validation: Cross-validation partitions the data into two sets, namely a training and a validation set. The goal is to select the hyper-parameters of a model such as the number of clusters for the k-means algorithm or the polynomial degree for SG-smoothers, by comparing the prediction performance of different models trained (or optimized) on the training set and evaluated on the validation set [10, Ch.13].

Continuing with the running example, let us assume that the samples in the processing window of 5 samples are $x_{-2} = x_2 = 25$, $x_{-1} = x_1 = 4$ and $x_0 = -3$, which corresponds to $x_n = 7n^2 - 3$ sequence. For N-fold cross-validation, the data of length N (5 samples for this case) is partitioned N times by setting aside a single sample as the validation set and using the remaining $N - 1$ samples as the training set. The goal is to predict the validation sample by constructing a model from the training data. Table 1 shows five cases for the partitioning of x_n with 5-fold cross-validation. The samples with the question mark indicate the validation data to be predicted.

Considering the 0th order model and Case-❶, we predict x_{-2} by fitting a 0th order model to $\{x_{-1}, x_0, x_1, x_2\}$. In this case, the prediction is $\hat{x}_{-2} = 15/2$ and is formed by the arithmetic average of the training data. Since $x_{-2} = 25$, the prediction error is $25 - 15/2 = 35/2$. The prediction errors for all cases are computed similarly and the total squared prediction error, i.e., the sum of squared prediction errors for all cases, is recorded. This process is repeated for the 1st, 2nd and higher order models and the total squared prediction error for each model is noted. Once the process finalizes, the model with the smallest total squared prediction error is selected.

The cross-validation procedure does not require any additional parameters such as noise variance, signal power, etc. However, the process is highly repetitive and can be computationally intensive especially for time-series type data where a new model order is selected for each data point, as in Fig. 1.

SG-Prediction Filter: SG-prediction filters are needed to implement the cross-validation procedure. For illustration purposes, we focus on the SG-prediction with 2nd degree polynomials for ❶ in Table 1. The SG-predictor filter for this setting is the minimum norm solution of the following equation:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 4 & 5 \\ 2^2 & 3^2 & 4^2 & 5^2 \end{bmatrix} \begin{bmatrix} h^{\mathrm{p}❶}_{-1} \\ h^{\mathrm{p}❶}_{-2} \\ h^{\mathrm{p}❶}_{-3} \\ h^{\mathrm{p}❶}_{-4} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1^2 \end{bmatrix}. \tag{2}$$

If we compare the equations for the prediction and smoothing operations, we see that the first column of the matrix $\mathbf{A}$ in the smoothing operation given in (1) is missing on the left-hand side of (2); however, the same column appears on the right side of (2). It is worth noting that the constraint equations in (2) map the samples of $x(t) = t^k$ collected at integer valued points in $[2, 5]$ to the value of $x(1)$, which is the error-free prediction output.

The minimum norm solution of (2) is $\mathbf{h}^{\mathrm{p}❶}_{\mathrm{MN}} = \frac{1}{4}[9, -3, -5, 3]^T$ and the corresponding prediction for Case-❶ is $\hat{x}_{-2} = (\mathbf{h}^{\mathrm{p}❶}_{\mathrm{MN}})^T \times [x_{-1}, x_0, x_1, x_2] = 25$, which is identical to x_{-2} value. This result is expected, since x_n satisfies a quadratic relation ($x_n = 7n^2 - 3$) in this example.

3. SG model order selection with N-fold cross-validation

We consider the design of an SG-smoothing filter with a processing window of length N (also the filter impulse response length) projecting the input onto the $P+1$ dimensional subspace of Pth degree polynomials. The constraint equations for the SG-smoother are:

$$\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 & 1 & \dots & 1 \\ 1 & 2 & \dots & k-1 & k & \dots & N \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & 2^P & \dots & (k-1)^P & k^P & \dots & N^P \end{bmatrix}}_{\mathbf{A}} \mathbf{h}^s_k = \underbrace{\begin{bmatrix} 1 \\ k \\ \vdots \\ k^P \end{bmatrix}}_{\mathbf{c}_k}. \tag{3}$$

The equation system in (3) has the vector $\mathbf{c}_k$ on its right side. Here $\mathbf{c}_k$ also refers to the kth column of the matrix $\mathbf{A}$. The index k indicates that the filter is constructed to smooth the kth sample in the processing window. The minimum norm solution of (3), which yields the SG-smoother, is simply $\mathbf{h}^{\mathbf{s}}_{\mathbf{k}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k$. To smooth all N samples in the processing window, from $k=1$ to $k=N$, we can construct $\mathbf{H}^s = [\mathbf{h}^s_1\ \mathbf{h}^s_2\ \dots\ \mathbf{h}^s_N] = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}$ and the output of the smoother is then $(\mathbf{H}^s)^T\mathbf{x}$. Here $\mathbf{x} = [x_1\ x_2\ \dots\ x_N]^T$ is the input data in the processing window.

SG-prediction filter for the prediction of the kth sample in the processing window can be similarly expressed as the minimum norm solution of the following system:

$$\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 & 0 & \dots & 1 \\ 1 & 2 & \dots & k-1 & 0 & \dots & N \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & 2^P & \dots & (k-1)^P & 0 & \dots & N^P \end{bmatrix}}_{\mathbf{A_p}} \mathbf{h}^{\mathbf{p}}_{\mathbf{k}} = \underbrace{\begin{bmatrix} 1 \\ k \\ \vdots \\ k^P \end{bmatrix}}_{\mathbf{c}_k}. \tag{4}$$

The equation systems in (3) and (4) are nearly identical, differing only in the kth columns of the matrices $\mathbf{A}$ and $\mathbf{A}_p$. The matrix $\mathbf{A_p}$ in (4) has an all zeros vector in its kth column, in contrast to matrix $\mathbf{A}$ in (3). Note that the kth entry of the minimum norm solution to the equation system in (4) is identically 0, that is $[\mathbf{h}^{\mathbf{p}}_{\mathbf{k}}]_k = 0$. Hence, the kth column of $\mathbf{A_p}$ in (4) can be discarded and $\mathbf{h}^{\mathbf{p}}_{\mathbf{k}}$ can be shortened without any harm. When this column is removed, we retrieve the special case of the SG-prediction filter in (2) for $k=1, N=5, P=2$. In the following, we utilize the redundant $\mathbf{A}_p$ matrix with an all zeros column in the SG-predictor constraint equations to establish a connection with matrix $\mathbf{A}$ in (3). Later, we capitalize on this connection to simply the cross-validation calculations.

Prediction Error Calculation: The prediction error for the kth sample in processing window is defined as $\epsilon^p_k \triangleq x_k - \hat{x}^p_k$ where $\hat{x}^p_k = (\mathbf{h}^p_k)^T\mathbf{x}$ is the predicted value. The prediction error can also be expressed as $\epsilon^p_k = (\mathbf{e}^T_k - (\mathbf{h}^p_k)^T)\mathbf{x}$ where $\mathbf{e}_k = [0\ 0\ \dots\ 1\ \dots\ 0]$ is the kth canonical basis vector in N-dimensional space, with all zero entries except a 1 at the kth entry.

We note that the constraint equations for smoothing and prediction operations are linked with $\mathbf{A}_p = \mathbf{A} - \mathbf{c}_k\mathbf{e}^T_k$; since $\mathbf{A}_p$ in (4) differs from $\mathbf{A}$ in (3) only in the kth column. If we substitute this equality in the expression for the minimum norm solution $\mathbf{h}^p_k = \mathbf{A}^T_p(\mathbf{A}_p\mathbf{A}^T_p)^{-1}\mathbf{c}_k$, we get

$$\begin{aligned} \mathbf{h}^p_k &= (\mathbf{A} - \mathbf{c}_k\mathbf{e}^T_k)^T\left((\mathbf{A} - \mathbf{c}_k\mathbf{e}^T_k)(\mathbf{A} - \mathbf{c}_k\mathbf{e}^T_k)^T\right)^{-1}\mathbf{c}_k \\ &\overset{(a)}{=} (\mathbf{A} - \mathbf{c}_k\mathbf{e}^T_k)^T\left(\mathbf{A}\mathbf{A}^T - \mathbf{c}_k\mathbf{c}^T_k\right)^{-1}\mathbf{c}_k \\ &\overset{(b)}{=} (\mathbf{A} - \mathbf{c}_k\mathbf{e}^T_k)^T\left((\mathbf{A}\mathbf{A}^T)^{-1} + \frac{(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k\mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}}{1 - \mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k}\right)\mathbf{c}_k \\ &\overset{(c)}{=} \left(1 + \frac{\mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k}{1 - \mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k}\right)\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k \\ &\quad - \left(\mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k + \frac{(\mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k)^2}{1 - \mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k}\right)\mathbf{e}_k. \end{aligned} \tag{5}$$

In line-(a), the argument of matrix inverse operation is expanded and simplified by using $\mathbf{c}_k = \mathbf{A}\mathbf{e}_\mathbf{k}$. In line-(b), the matrix inversion lemma (Woodbury formula) is utilized [10, Appendix A]. In line-(c), the product is expanded and the result is observed to be a linear combination of two vectors, namely $\mathbf{h}^{\mathbf{s}}_{\mathbf{k}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k$ (SG-smoother for the kth sample) and the canonical basis vector $\mathbf{e}_k$. For some more simplifications, we note that

$$\mathbf{c}^T_k(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{c}_k = \mathbf{e}^T_k\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{e}_k = \mathbf{e}^T_k\mathbf{P}_{\mathbf{A}^\mathbf{T}}\mathbf{e}_k = [\mathbf{P}_{A^T}]_{kk}. \tag{6}$$

Here $\mathbf{P}_{A^T} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}$ is the projection matrix to the range space of $\mathbf{A}^T$ (or to the row space of $\mathbf{A}$) and $[\mathbf{P}_{A^T}]_{kk}$ is the kth row and kth column entry of this matrix. Given these, we can rewrite $\mathbf{h}^p_k$ in (5) as follows:

$$\mathbf{h}^p_k = \frac{1}{1 - [\mathbf{P}_{A^T}]_{kk}}\mathbf{h}^s_k - \frac{[\mathbf{P}_{A^T}]_{kk}}{1 - [\mathbf{P}_{A^T}]_{k,k}}\mathbf{e}_k. \tag{7}$$

After (7), the prediction error $\epsilon^p_k = x_k - \hat{x}^p_k = (\mathbf{e}_k - \mathbf{h}^p_k)^T\mathbf{x}$ simplifies to

$$\epsilon^p_k = \frac{1}{1 - [\mathbf{P}_{A^T}]_{kk}}(\mathbf{e}_k - \mathbf{h}^s_k)^T\mathbf{x} = \gamma_k(x_k - \hat{x}^s_k) = \gamma_k\epsilon^s_k \tag{8}$$

where $\gamma_k = 1/(1 - [\mathbf{P}_{A^T}]_{kk})$, $\hat{x}^s_k = (\mathbf{h}^s_k)^T\mathbf{x}$ is the SG-smoother output and $\epsilon^s_k \triangleq x_k - \hat{x}^s_k$ is the residual error after smoothing. The residual error vector for all samples can then be written as $\boldsymbol{\epsilon}^s = [\epsilon^s_1\ \epsilon^s_2\ \dots\ \epsilon^s_N]^T = (\mathbf{I} - \mathbf{P}_{A^T})\mathbf{x} = \mathbf{P}^\perp_{A^T}\mathbf{x}$ where $\mathbf{P}^\perp_{A^T} = \mathbf{I} - \mathbf{P}_{A^T}$ is the complementary projection matrix, that is the projector to the null space of $\mathbf{A}$.

From the results obtained, the total squared prediction error, T^p_ϵ, can be written as

$$T^p_\epsilon = \sum_{k=1}^{N}(\epsilon^p_k)^2 = \sum_{k=1}^{N}\gamma^2_k(\epsilon^s_k)^2. \tag{9}$$

where $\gamma_k = 1/[\mathbf{P}^\perp_{A^T}]_{kk}$ and $\boldsymbol{\epsilon}^s = [\epsilon^s_1\ \epsilon^s_2\ \dots\ \epsilon^s_N]^T = \mathbf{P}^\perp_{A^T}\mathbf{x}$.

Cross-validation Procedure: Let $P_{\max} \le N-2$ denote the maximum polynomial order to be tested for SG-filtering. The procedure starts with the 0th order. For this order, the $\mathbf{A}$ matrix is $1 \times N$ dimensional and the residual error vector $\boldsymbol{\epsilon}^s = \mathbf{P}^\perp_{A^T}\mathbf{x}$ is formed by projecting the input $\mathbf{x}$ onto the $N-1$ dimensional null space of this particular $\mathbf{A}$ matrix. The weight $\gamma_k = 1/[\mathbf{P}^\perp_{A^T}]_{kk}$ in (9) is calculated as the reciprocal of the kth diagonal entry of the matrix $\mathbf{P}^\perp_{A^T}$. Note that, we have $1/\gamma_k = [\mathbf{P}^\perp_{A^T}]_{kk} = \mathbf{e}^T_k\mathbf{P}^\perp_{A^T}\mathbf{e}_k = ||\mathbf{P}^\perp_{A^T}\mathbf{e}_k||^2 < ||\mathbf{P}^\perp_{A^T}\mathbf{e}_k||^2 + ||\mathbf{P}_{A^T}\mathbf{e}_k||^2 =$

$||\mathbf{e}_k||^2 = 1$; that is $\gamma_k > 1$, since projection onto a sub-space is a contractive operator reducing the norm of the input. The total squared prediction error T_ϵ^p for the 0th order is calculated from (9). The process is then repeated for higher orders up to the maximum order $P_{\max}$. Once T_ϵ^p is calculated for all orders, the procedure returns the order with the minimum T_ϵ^p value as the selected order.

Bias-variance trade-off: The total squared prediction error T_ϵ^p in (9) depends on $\epsilon^s = \mathbf{P}^\perp_{A^T}\mathbf{x}$ and $1/\gamma_k = ||\mathbf{P}^\perp_{A^T}\mathbf{e}_k||^2$. As the candidate polynomial order P is gradually increased, the number of rows in the matrix $\mathbf{A}$ in (4) increases and this reduces the dimension of its null space (nullity). Since $\mathbf{P}^\perp_{A^T}$ is the projector onto the null space of $\mathbf{A}$, both $||\epsilon^s||^2$ and $1/\gamma_k$ decrease as the model order increases. Hence, the factor γ_k in T_ϵ^p increases monotonically with the model order, while the other factor, the squared residual error decreases monotonically in norm. This results in a trade-off that effectively acts as a penalty function, enabling model order selection. The trade-off is illustrated with a numerical experiment in the next section.

Computational Complexity: The conventional N-fold CV implementation requires $O(N^5)$ operations. The efficient Algorithm 1, based on Gram-Schmidt (GS) orthogonalization via Householder QR, reduces the complexity to $O(N^3)$, or to $O(N^2)$ when the GS step is precomputed. Precomputing the GS step requires storing an $N \times (N-1)$ matrix. In our tests, the precomputed version yielded 36-fold and 400-fold speedups for $N = 10$ and $N = 20$, respectively (27-fold and 310-fold when GS-step is computed on the fly). These values are implementation and hardware dependent, but they indicate the substantial efficiency gain of the proposed method. Further details are available in the appendices of the extended arXiv version in [11] along with MATLAB codes.

Algorithm 1: Model order selection for SG-filtering via cross-validation. See [11] for MATLAB code.

Input : $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$, $P_{\max}$, $\mathbf{Q} = [\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{P_{\max}}]$ ($\mathbf{Q}$ is optional)
Output: best-order
// Gram-Schmidt orthogonalization (if needed)
if $\mathbf{Q}$ *is not available* **then**
  $[Q, R] = \text{qr}(A^T)$ // QR decomposition of $\mathbf{A}^T$, see (3)
end
$\hat{\mathbf{x}} = \text{zeros}(N,1)$; $\mathbf{d} = \text{zeros}(N,1)$ // Initialization
for $k \leftarrow 0$ **to** $P_{\max}$ **do** // Recursion
  $\hat{\mathbf{x}} = \hat{\mathbf{x}} + \mathbf{q}_k\mathbf{q}_k^T\mathbf{x}$ // $\hat{\mathbf{x}}$: Projection of $\mathbf{x}$ to $(k+1)$-dim subspace
  $\epsilon^s = \mathbf{x} - \hat{\mathbf{x}}$ // Smoothing error
  $\mathbf{d} = \mathbf{d} + \mathbf{q}_k .* \mathbf{q}_k$ // $\mathbf{d} = \text{diag}(\mathbf{P}_{A^T})$
  $\epsilon^p = \epsilon^s./(1-\mathbf{d})$ // .* and ./ : element-wise multiplication and division, see (8)
  $T_\epsilon^p(k) = ||\epsilon^p||^2$ // see (9)
end
Return: best-order = $\text{argmin}_{0\le k\le P_{\max}} T_\epsilon^p(k)$

4. Numerical results

We compare the proposed method with the Bayesian Information Criterion (BIC), which is the state-of-the-art model order selection rule for the polynomial fitting problem [12]. BIC approximates the model likelihood by treating the model parameters as random variables and applies either high SNR or large sample size asymptotic theory to develop practical model order estimators [13]. In our performance comparisons, we include both large sample size variant BIC_N, and the low noise variance (high SNR) variant BIC_SNR, as given in [12]:

$$\text{BIC}_\text{N}(P) = (P+1)^2 \ln(N) + N \ln\left(\frac{||\epsilon^s||^2}{N}\right)$$

$$\text{BIC}_\text{SNR}(P) = N \ln\left(\frac{||\epsilon^s||^2}{N}\right) + \max(0, -(P+2)\ln\left(\frac{||\epsilon^s||^2}{N}\right))$$

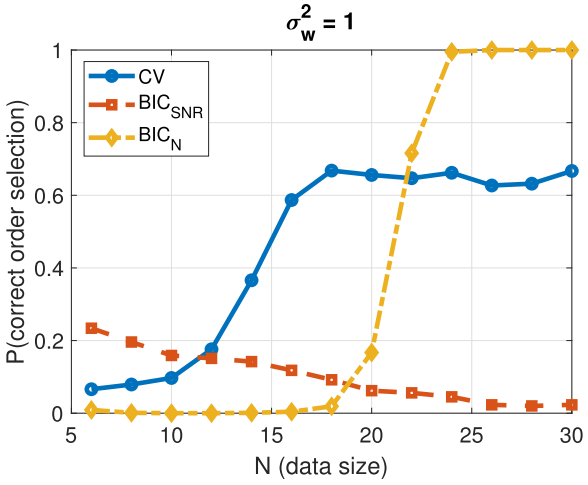

Fig. 2. Probability of correct model order detection at fixed noise variance as the sample size (processing window length) N increases. The suggested method outperforms the asymptotically optimal BIC_N at intermediate, that is non-asymptotic, values of N.

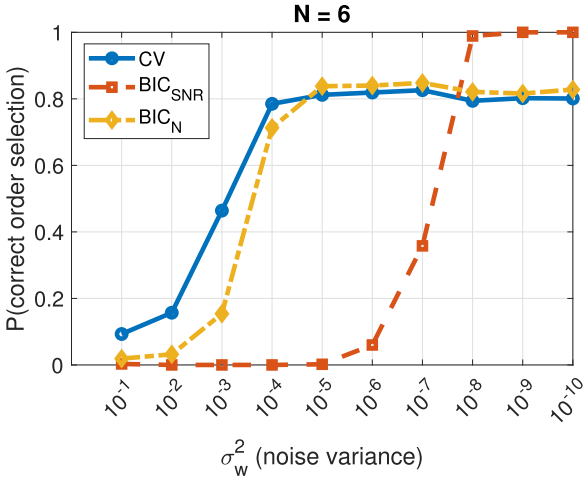

Fig. 3. Probability of correct model order detection at a fixed sample size of $N = 6$ as the noise variance decreases. The suggested method outperforms the asymptotically optimal BIC_SNR at intermediate, that is non-asymptotic, SNR values.

Here P denotes the polynomial order, N is the processing window length, and $\epsilon^s = \mathbf{P}^\perp_{A^T}\mathbf{x}$ is the residual error vector after smoothing, as described in Section 3.

In the experiment conducted, the samples $y[n]$ are noisy observations of the cubic function $x(t) = 0.01t^3 + 1$. The function is sampled with the sampling period of $T_s = 1$, i.e., $x[n] = x(nT_s)$ and the observations $y[n]$ are collected under white Gaussian observation noise $w[n]$, i.e., $y[n] = x[n] + w[n]$. The noise $w[n]$ is zero mean with variance σ_w^2. We assume that $y[n]$ samples are obtained over the interval $[-N/2, N/2-1]$ for even N and over the interval $[-(N-1)/2, (N-1)/2]$ for odd N. The candidate model orders include all polynomial degrees from 0 to $N-2$.

Fig. 2 shows the probability of correctly detecting model order for a fixed noise variance of 1 as the processing window size N increases. We observe that cross-validation (CV) method outperforms both BIC_N and BIC_SNR for intermediate values of N. For asymptotically large N values, BIC_N outperforms all others as expected; since it is designed to be a consistent model estimator in the large sample size regime [12,14]. Therefore, performance superiority over BIC_N is not expected at high sample sizes. Fig. 3 shows the same comparison when the sample size N is fixed at 6 samples, while the noise variance is gradually reduced

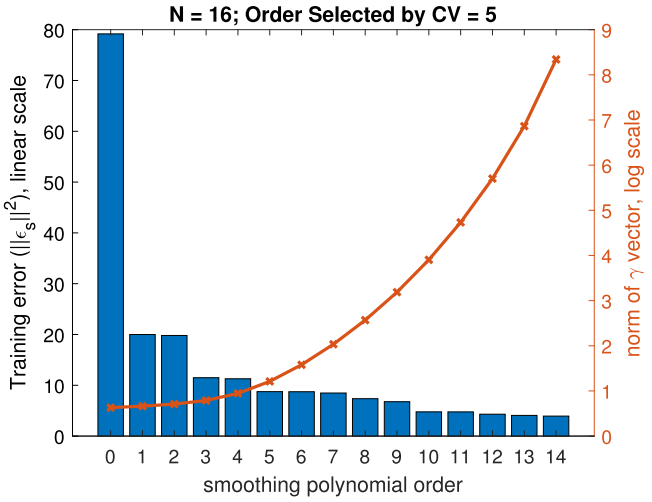

Fig. 4. An illustration for the bias-variance trade-off: Smoothing error decreases with the polynomial order while associated weights increases. This results in a trade-off point between a perfect match to the training data (small bias, large variance) and a preference of simpler models with low ordered polynomials (large bias, small variance).

from 10^{-1} to 10^{-10}. As expected, BIC_{SNR} outperforms the other methods at sufficiently low noise variance levels due to its asymptotic optimality in the high-SNR regime. We observe that the proposed cross-validation based order detector outperforms both BIC_{N} and BIC_{SNR} at moderate sample sizes and noise variance levels, a significant result for the applications limited to non-asymptotic regimes.

Fig. 4 illustrates the bias-variance tradeoff. Here, the results for a single Monte-Carlo run with the data length of $N = 16$ are shown. We observe that training error gets smaller with the smoothing polynomial order; while the associated weight for higher orders increases. The trade-off between two enables the selection of medium ordered polynomials, which is a 5th degree polynomial in this case.

5. Summary and conclusions

A simple model order (polynomial order) selection procedure with N-fold cross-validation is derived for SG filtering. The expression is based on the minimum norm formulation of SG filters and can be computed efficiently. An order-recursive implementation of the proposed technique, utilizing Gram-Schmidt orthogonalization, is given in Algorithm 1. A ready-to-use MATLAB implementation is also given in an online code repository [11]. The absence of tuning parameters and the strong performance in the non-asymptotic regimes makes the proposed method attractive.

From technical viewpoint, the performance of BIC-based order estimators is known to degrade when either N (sample size) or signal-to-noise-ratio (SNR) is not sufficiently high. This limitation arises from the asymptotic approximations used in model likelihood calculations. The numerical results demonstrate that in regimes where BIC-based estimators falter, namely moderate N or moderate SNR, the proposed cross-validation technique can be adopted for improved performance. Although cross-validation is often regarded as an empirical method with loose theoretical grounding, it is interesting to note that its application to the SG filtering problem forms a rare case in which the method admits a theoretically justifiable trade-off explanation for its good empirical performance.

For future work, the robustness of the CV method (see [11] for initial findings under impulsive noise), case of irregular sampling and extensions to other types of SG-filters can be investigated.

CRediT authorship contribution statement

Çağatay Candan: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization.

Data availability

No data was used for the research described in the article.

References

[1] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (8) (1964) 1627–1639. https://doi.org/10.1021/ac60214a047
[2] S.J. Orfanidis, Introduction to Signal Processing, Prentice Hall, 1995.
[3] C. Candan, H. Inan, A unified framework for derivation and implementation of Savitzky-Golay filters, Signal Process. 104 (2014) 203–211. https://doi.org/10.1016/j.sigpro.2014.04.016
[4] R. Schafer, What is a Savitzky-Golay filter?, IEEE Signal Process. Mag. 28 (4) (2011) 111–117. https://doi.org/10.1109/MSP.2011.941097
[5] K. Hasan, S.T. Meraj, M.M. Othman, M.S.H. Lipu, M.A. Hannan, K.M. Muttaqi, Savitzky-Golay filter-based PLL: modeling and performance validation, IEEE Trans. Instrum. Meas. 71 (2022) 1–6. https://doi.org/10.1109/TIM.2022.3196946
[6] A. John, J. Sadasivan, C.S. Seelamantula, Adaptive Savitzky-Golay filtering in non-Gaussian noise, IEEE Trans. Signal Process. 69 (2021) 5021–5036. https://doi.org/10.1109/TSP.2021.3106450
[7] P.-O. Persson, G. Strang, Smoothing by Savitzky-Golay and Legendre filters, in: Mathematical Systems Theory in Biology, Comm., Computation, Finance, NY, Springer, 2003, pp. 301–315.
[8] M. Sadeghi, F. Behnia, R. Amiri, Window selection of the Savitzky-Golay filters for signal recovery from noisy measurements, IEEE Trans. Instrum. Meas. 69 (8) (2020) 5418–5427. https://doi.org/10.1109/TIM.2020.2966310
[9] S. Krishnan, C. Seelamantula, On the selection of optimum Savitzky-Golay filters, IEEE Trans. Signal Process. 61 (2) (2013) 380–391. https://doi.org/10.1109/TSP.2012.2225055
[10] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge Univ. Press, 2012.
[11] C. Candan, Polynomial order selection for Savitzky-Golay smoothers via N-fold cross-validation (extended version), in: arXiv:2512.00171, 2025. (MATLAB source codes at codeocean: https://doi.org/10.24433/CO.1732394.v2)
[12] P. Stoica, P. Babu, On the proper forms of BIC for model order selection, IEEE Trans. Signal Process. 60 (9) (2012) 4956–4961. https://doi.org/10.1109/TSP.2012.2203128
[13] P. Stoica, Y. Selen, Model-order selection: a review of information criterion rules, IEEE Signal Process. Mag. 21 (4) (2004) 36–47. https://doi.org/10.1109/MSP.2004.1311138
[14] Q. Ding, S. Kay, Inconsistency of the MDL: on the performance of model order selection criteria with increasing signal-to-noise ratio, IEEE Trans. Signal Process. 59 (5) (2011) 1959–1969. https://doi.org/10.1109/TSP.2011.2108293