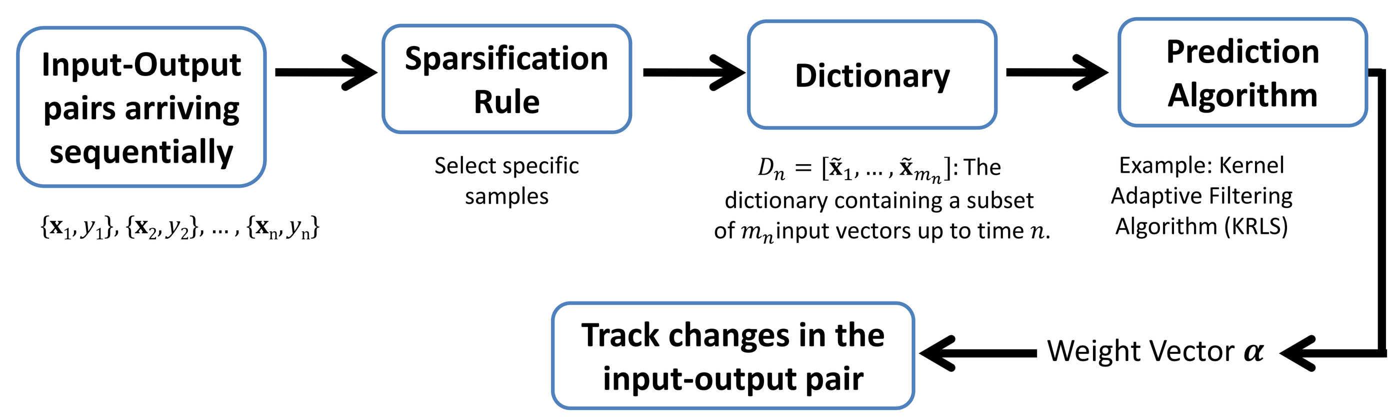
Improving the Tracking Ability of KRLS using Kernel Subspace Pursuit

Jad Kabbara and Ioannis N. Psaromiligkos
jad.kabbara@mail.mcgill.ca,yannis@ece.mcgill.ca

McGill
Department of Electrical and Computer Engineering

Problem Statement

Track time-varying systems using KRLS.



Kernel Adaptive Filtering: KRLS

Kernel Adaptive Filtering:

- Kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ with $\mathcal{X}$: Input space.
- Kernel Trick: $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ where $\phi : \mathcal{X} \mapsto \mathcal{H}$ maps input vectors from $\mathcal{X}$ into a Hilbert space $\mathcal{H}$ (RKHS).
- Adapt estimated output based on error *in the high-dimensional RKHS* $\mathcal{H}$.

KRLS

- KRLS minimization problem:

$$\min_{\boldsymbol{\alpha}} \|\mathbf{K}_n\boldsymbol{\alpha} - \mathbf{y}_n\|^2, \tag{1}$$

where $\mathbf{y}_n = [y_1, \ldots, y_n]^T$, $\mathbf{K}_n$: Gram matrix of the kernel k with entries $[\mathbf{K}_n]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T$: a weight vector.

Sparsification

- Surprise Criterion: Add element to dictionary only its usefulness measured by the surprise $\mathcal{S}_n$ lies within a range specified by thresholds T_1 and T_2.

$$\mathcal{S}_n = -\ln p(\mathbf{x}_n, y_n | \mathcal{D}_{n-1}). \tag{2}$$

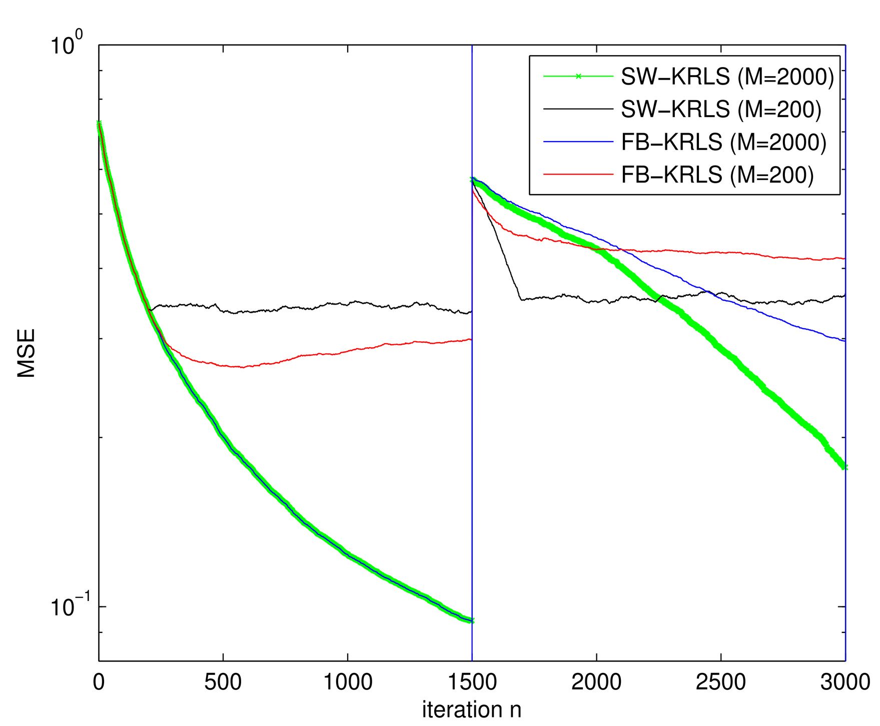
Pruning

- Remove the Oldest, Minimal Introduced Error criterion, etc.

Not good enough for tracking time-varying systems!

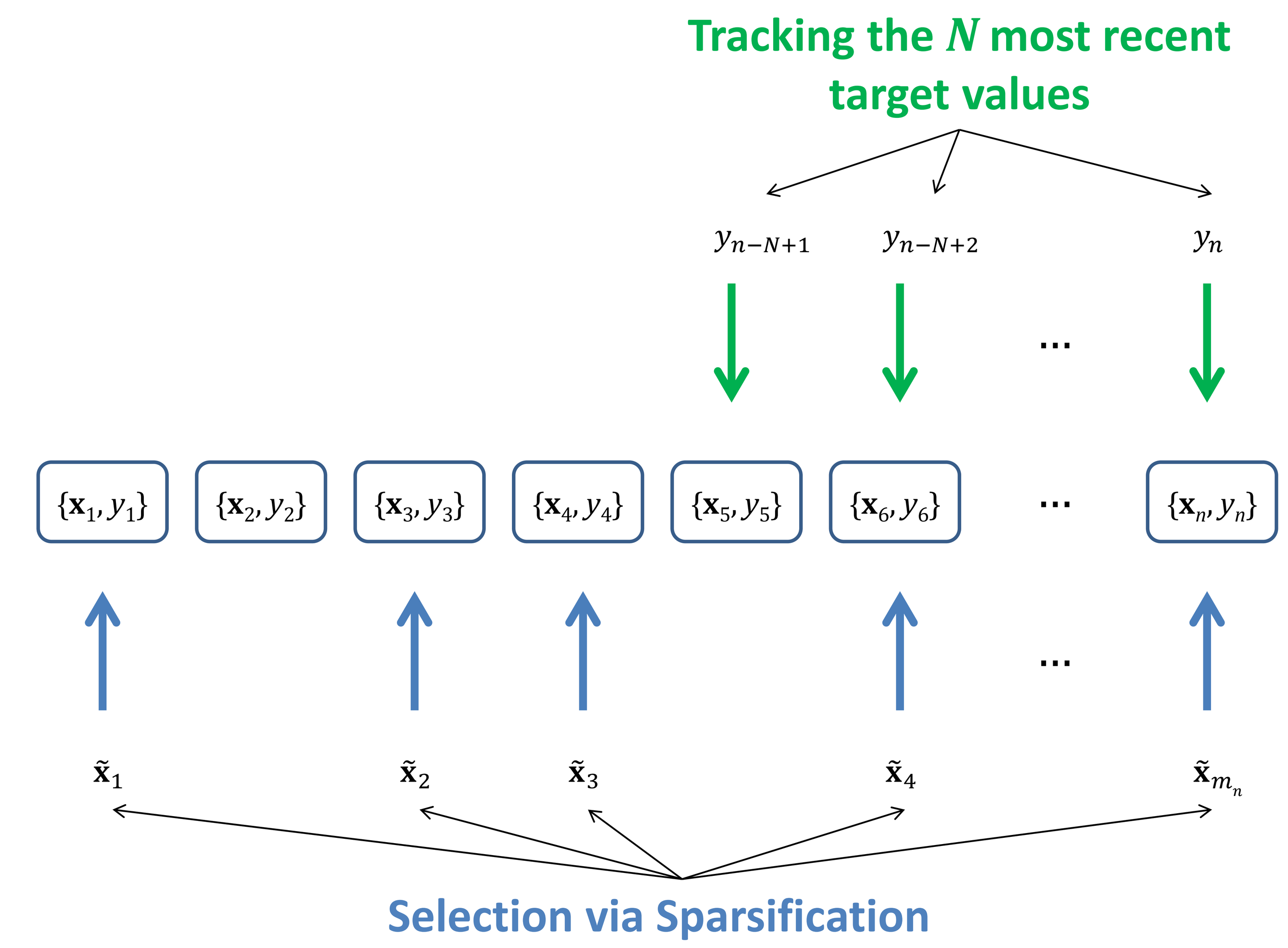
Motivation

Motivation: Decouple the size of the of the weight vector from the dictionary size.



Approach: Select the M dictionary elements that best approximate the N most recent target values.

Proposed Method



Subspace Pursuit (SP)-KRLS

$$\min_{\boldsymbol{\alpha}} \|\mathbf{K}_n\boldsymbol{\alpha} - \mathbf{y}_n\|^2 \text{ s.t. } \boldsymbol{\alpha} \text{ is } M-\text{sparse}. \tag{3}$$

- Maximum size M of weight vector: fixed and independent from the size of the dictionary.
- If sample is admitted to the dictionary and dictionary size exceeds M, KSP is used to select M elements to form the LS regressor.
- KSP selects out of the αM ($\alpha > 1$) most recent entries in the dictionary $\mathcal{D}_n = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_{m_n}]$ the M elements that lead to the best approximation of the most recent N received target values.
- KSP gram matrix $\mathbf{G}_n$: obtained by evaluating $k(\cdot, \tilde{\mathbf{x}}_i)$, $i = m_n - \alpha M + 1, \ldots, m_n$ at the N most recent inputs $\mathbf{x}_{n-N+1}, \ldots, \mathbf{x}_n$.
- KSP vector $\mathbf{y}_n$: vector consisting of the most recent N target values, i.e., $\mathbf{y}_n = [y_{n-N+1}, \ldots, y_n]^T$.
- KSP does not run at every iteration.

Summary of SP-KRLS

When the system receives a new pair $\{\mathbf{x}_n, y_n\}$, it is checked against the Surprise Criterion (SC) test.

Two possible scenarios:

1. If it does not pass the SC test $\rightarrow$ input vector not added to dictionary, weight vector updated appropriately via the KRLS recursions.
2. If input vector admitted to the dictionary $\rightarrow$ two cases:

(a) If dictionary size $\leq M$, weight vector updated via regular KRLS recursions.

(b) If dictionary size $> M$, use KSP to identify the M input vectors that will be used by the KRLS algorithm.

Kernel Subspace Pursuit

- $\mathbf{G}_n$: The $m_n \times n$ Gram matrix obtained by evaluating the functions $k(\cdot, \tilde{x}_i)$, $i = 1, \ldots, m_n$ at the input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, i.e.,

$$[\mathbf{G}_n]_{i,j} = k(\tilde{\mathbf{x}}_i, \mathbf{x}_j). \tag{4}$$

- T: A set of column indices.
- $\mathbf{G}_T$: The matrix consisting of columns of $\mathbf{G}_n$ with indices in T.
- Projection of $\mathbf{y}_n$ onto $\mathbf{G}_T$: proj$(\mathbf{y}_n, \mathbf{G}_T) := \mathbf{G}_T\mathbf{G}_T^{\dagger}\mathbf{y}_n$

Algorithm 1 Kernel Subspace Pursuit Algorithm

Input: M, $\mathbf{y}_n$, $\mathcal{D}$.
Initialization:
Compute $\mathbf{G}$ using (4).
$T_0 = \{$Indices of the M largest magnitude entries in the vector $\mathbf{Gy}\}$
$\mathbf{r}_0 = \mathbf{y}_n -$ proj$(\mathbf{y}_n, \mathbf{G}_{T_0})$
Loop: At iteration ℓ ($1 \leq \ell \leq 5$), execute the following:
$\tilde{T}_\ell = T_{\ell-1} \cup \{$Indices of the M largest magnitude entries in the vector $\mathbf{G}^T\mathbf{r}_{\ell-1}\}$
$T_\ell = \{$Indices of the M largest magnitude entries in $\mathbf{G}_{\tilde{T}_\ell}^{\dagger}\mathbf{y}_n\}$
$\mathbf{r}_\ell = \mathbf{y}_n -$ proj$(\mathbf{y}_n, \mathbf{G}_{T_\ell})$
If $||\mathbf{r}_\ell||_2 > ||\mathbf{r}_{\ell-1}||_2$, let $T_\ell = T_{\ell-1}$ and exit the loop.
Output: T_ℓ

Simulation Results

Compare the performance of SP-KRLS to that of ALD-KRLS, FB-KRLS and SW-KRLS.

Tracking of a Time-Varying Wiener System

- Training set: 3200 sample points, test set: 400 sample points.
- SP-KRLS thresholds for learnable data: $T_1 = 3$ and $T_2 = -3$.
- SP-KRLS sparsity level: $M = 200$.
- KSP number of recent target values: $N = 10$.
- Gaussian kernel with a width $\sigma = 0.8$.