Lookahead and Dynamic Beam Search Variants of FOCI for Conditional-Dependence Feature Selection

Arnau Neches-Vilá[1⋆][0009−0007−5727−4562], Carlos Fernández-Llebrez Acedo[1][0009−0002−0280−7441], Pablo Reina-Jiménez[1][0009−0008−8770−3767], and María Martínez-Ballesteros[1,2][0000−0003−3160−7414]

[1] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Av. Reina Mercedes S/N, 41012 Sevilla, Spain
{aneches,cfernandezllebrez,prjimenez,mariamartinez}@us.es
[2] SCORE Lab, I3US Institute, Universidad de Sevilla, Sevilla, Spain

Abstract. Feature selection is a critical preprocessing step for high-dimensional data, where the goal is to identify a minimal subset of variables that maximizes predictive power while removing noise and redundancy. Many existing methods, however, fail to capture the non-linear interactions that arise in real-world problems. The Feature Ordering by Conditional Independence algorithm offers a robust, model-free framework grounded on a rank-based dependence measure, but its greedy nature makes it susceptible to myopia, that is, the tendency to overlook features that are only predictive in combination, as well as to path-dependent redundancy. This paper proposes two improvements to the algorithm: a lookahead variant and a dynamic beam search variant. Both enlarge the search space to detect multi-feature interactions, such as those present in exclusive-or distributions. We evaluate them against the original algorithm, the Hilbert-Schmidt Independence Criterion Lasso, and a Mutual Information filter on six synthetic benchmarks and four real-world datasets. The results show that lookahead strategies substantially improve interaction discovery and predictive accuracy.

Keywords: Feature Selection · FOCI · CODEC · Conditional Dependence · Lookahead Search · Non-linear Interactions.

1 Introduction

The proliferation of high-dimensional data has made feature selection indispensable for maximizing predictive power while eliminating noise. However, traditional methods often rely on linear assumptions that fail to capture complex, non-linear real-world dependencies. A powerful alternative is the Feature Ordering by Conditional Independence (FOCI) framework [1], which leverages the model-free, parameter-free Conditional Dependence Coefficient (CODEC) metric to evaluate feature subsets in efficient $O(n \log n)$ time [8].

⋆ Corresponding author.

Despite these advantages, FOCI's greedy forward selection suffers from structural myopia. By evaluating variables sequentially, it remains blind to multivariable interactions (such as XOR relations) where marginal correlations appear zero. To overcome this, two search-driven extensions are introduced to expand FOCI's search horizon: Lookahead FOCI (FOCI-LA) and Dynamic Beam Search FOCI (FOCI-DB). FOCI-LA addresses this myopia by evaluating candidates on their potential to unlock future gains, while FOCI-DB addresses the related risk of irreversible, path-dependent mistakes by maintaining multiple candidate subsets in parallel. While navigating this expanded state-space incurs higher computational costs, it drastically improves the discovery of hidden interactions. The primary contributions of this work are as follows:

- FOCI-LA and FOCI-DB are introduced expanding the search capabilities of the rank-based conditional dependence framework.
- The structural myopia inherent to strictly greedy forward conditional selection is formalized and documented.
- Empirical evaluation is conducted across six synthetic interaction benchmarks and four real-world datasets.
- The complexity trade-offs is quantified, delineating when expanded search effort is justified by downstream predictive gains.

The remainder of this paper is organized as follows. Section 2 reviews the CODEC metric and the FOCI framework. Section 3 formalizes the proposed search enhancements. Section 4 outlines our experimental setup. Section 5 presents our empirical results and statistical validation and Section 6 discusses the results obtained. Finally, Section 7 concludes the paper and discusses future directions.

2 Background

Feature selection methods are traditionally categorized into filters, wrappers, and embedded methods [2]. Wrappers and embedded techniques are accurate but computationally intensive and model-dependent, whereas filters evaluate relevance through intrinsic data properties, offering model-agnostic scalability. Modern non-parametric filters, such as the Hilbert-Schmidt Independence Criterion (HSIC) Lasso [10] or Mutual Information (MI) approaches [9], focus on non-linear dependencies. However, optimizing for non-linear marginal dependencies is insufficient when a feature appears irrelevant individually but becomes predictive when combined with other variables.

A significant advancement is the Conditional Dependence Coefficient (CODEC), denoted as $T(Y, Z \mid W)$, which provides a robust, rank-based measure of how much a target Y depends on variables Z given a conditioning set W [1]. Unlike symmetric measures, CODEC is asymmetric ($T(Y, Z \mid W) \neq T(Z, Y \mid W)$), reflecting a directional measure of functional dependence. The sample estimate is derived from k-nearest neighbor distances in a rank-transformed space. Crucially, $T \in [0, 1]$, where $T = 1$ if and only if Y is a measurable function of Z given W, and $T = 0$ if and only if Y and Z are conditionally independent given

W. This foundation allows precise evaluation of a feature's contribution to a subset without assuming linear relationships. However, CODEC also has limitations: the convergence rate of its sample estimator degrades with the size of the conditioning set, scaling roughly as $n^{-1/(p+q)}$ up to a logarithmic factor, and no convergence rate is currently known in the fully unconditional case [1].

Feature selection methods also differ in their search strategy over candidate subsets [3]. Forward greedy selection is efficient but sensitive to early mistakes and blind to interactions. While alternatives like beam search and lookahead heuristics are well established in wrappers and decision-tree induction [5], they have rarely been combined with rank-based conditional dependence measures, a gap our work addresses. The FOCI algorithm uses CODEC in a forward-greedy framework to identify a minimal sufficient feature subset [1]. Although effective in domains like bioinformatics [8], its greedy nature leads to "structural myopia" in interaction-based settings like XOR [2]. While alternative meta-heuristics like evolutionary algorithms evaluate subsets jointly, they suffer from high computational costs in high-dimensional spaces [7]. Therefore, enhancing the search strategy around CODEC is necessary to preserve FOCI's non-parametric strengths while maximizing its ability to detect interaction-based dependencies.

3 Methodology

The core of the contribution lies in improving the search strategy of the FOCI algorithm. While the underlying statistical measure remains the rank-based CODEC, we propose shifting from a purely greedy local search to more exhaustive state-space explorations to better capture complex feature interactions.

3.1 Original FOCI Algorithm

The original FOCI algorithm uses a forward-greedy selection process. At each step, it identifies the feature X_{j^*} that maximizes the conditional dependence coefficient given the already selected set, and stops when no remaining feature increases the score. The procedure is summarized in the Algorithm 1.

Algorithm 1: Original FOCI

- **Initialize:** Selected set $S = \emptyset$, Remaining features $R = \{1, \ldots, P\}$.
- **While** not stopped:
  1. For each $j \in R$, compute the conditional dependence: $G_j = T(Y, X_j \mid X_S)$.
  2. Identify j^* that maximizes G_j.
  3. **Stopping Rule:** If $G_{j^*} \leq 0$, terminate the search.
  4. Update $S = S \cup \{j^*\}$ and $R = R \setminus \{j^*\}$.
- **Return:** The selected subset S.

The greedy approach of FOCI suffers from two primary weaknesses:

1. **Structural Myopia:** FOCI is unable to detect k-way interactions where individual features possess zero marginal importance. In a classic XOR relationship, $T(Y, X_i \mid \emptyset) \approx 0$ for all participating features. Consequently, the greedy selector perceives these features as noise and terminates prematurely.
2. **Irreversibility and Redundancy:** FOCI lacks a pruning mechanism. Once a feature is added to S, it remains there. This "path dependency" means that an early suboptimal choice, possibly arising from sampling variability in the CODEC estimator, can permanently corrupt the conditioning set for all subsequent calculations.

3.2 Proposed Enhancements

Two search-based variants are proposed that maintain the non-parametric advantages of the CODEC measure while expanding the search horizon: FOCI-LA extends the planning horizon to address myopia, and FOCI-DB maintains several candidate paths in parallel to address irreversibility. Figure 1 summarizes the differences between the three search strategies.

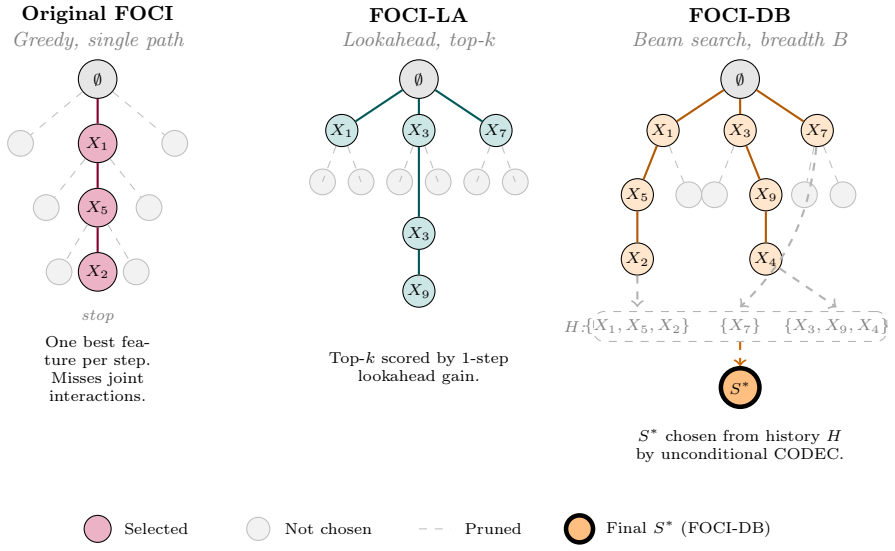


Fig. 1. Search space exploration for the three FOCI variants. Original FOCI commits to a single path, evaluating only the immediate gain at each step. FOCI-LA evaluates the top-k candidates by their combined score after a tentative one-step expansion. FOCI-DB keeps B parallel beams from the root and dynamically prunes them; every beam, whether terminated early by the local threshold or surviving until the last step, is added to a path history H. The returned subset S^* is the one in H that maximizes the unconditional CODEC $T(Y, X_{S^*})$.

FOCI-LA: Lookahead FOCI (Depth Expansion) A 1-step lookahead strategy is applied to address the myopia problem. Instead of evaluating a candidate X_j based solely on its immediate score, its potential to unlock information in the subsequent step is evaluated. The pseudocode is presented in Algorithm 2. The selection criterion is defined as follows:

$$\text{score}(X_j) = T(Y, X_j \mid X_S) + \max_{k \notin S \cup \{j\}} T(Y, X_k \mid X_{S \cup \{j\}}) \tag{1}$$

We restrict ourselves to a 1-step horizon and the top k candidates (e.g., $k = 5$) because the cost already scales as $O(kP)$ per step, making both deeper lookahead and larger k impractical at high dimensionality; larger k broadens the search at a roughly linear computational cost, while smaller k recovers behavior closer to the original greedy algorithm.

Algorithm 2: FOCI-LA

- **Initialize:** Selected set $S = \emptyset$, Remaining features $R = \{1, \ldots, P\}$.
- **While** not stopped:
  1. For each $j \in R$, compute immediate gain: $G_j = T(Y, X_j \mid X_S)$.
  2. Sort R by G_j and select the top k candidates (e.g., $k = 5$);
  3. For each candidate j in the top k:
     (a) Tentatively add to set: $S' = S \cup \{j\}$.
     (b) Find the best "future" feature $l \in R \setminus \{j\}$ that maximizes $T(Y, X_l \mid X_{S'})$.
     (c) Calculate total score: $Score_j = G_j + T(Y, X_l \mid X_{S'})$.
  4. Pick j^* with the highest $Score_j$.
  5. Update $S = S \cup \{j^*\}$ and $R = R \setminus \{j^*\}$.
- **Return:** The selected subset S.

FOCI-DB: Dynamic Beam Search FOCI (Breadth Expansion) To mitigate the risk of irreversibility, a Beam Search is implemented that maintains a set of B promising candidate subsets $\mathcal{B} = \{S_1, S_2, \ldots, S_B\}$, pruned at each step to those within a relative threshold α of the best score, so the algorithm explores multiple paths when the data is ambiguous. Larger B and higher α increase robustness to early mistakes at a roughly linear increase in computational cost, while smaller B or lower α trade this robustness for speed closer to greedy FOCI. The procedure is summarized in Algorithm 3.

4 Datasets

FOCI-LA and FOCI-DB are evaluating in synthetic benchmarks and real-world datasets. Synthetic environments provide a controlled framework with known ground truth to isolate an algorithm's capability to recover interacting or redundant features under severe noise. Real-world datasets validate how effectively these search enhancements generalize across diverse empirical domains.

Algorithm 3: FOCI-DB

- **Initialize:** Beams $\mathcal{B} = \{(0.0, \emptyset)\}$, Path history $\mathcal{H} = \emptyset$.
- **While** not stopped:
  1. Generate all possible expansions for every $S \in \mathcal{B}$ by adding one remaining feature $j \notin S$.
  2. For each new expansion, calculate the conditional CODEC gain:

$$\Delta = \begin{cases} T(Y, X_j), & S = \emptyset \\ T(Y, X_j \mid X_S), & S \neq \emptyset \end{cases}$$

  3. If $\Delta \leq \gamma$ (local threshold), discard candidate j. If all expansions for S are discarded, add S to $\mathcal{H}$.
  4. Update scores for surviving candidates: $Score_{S \cup \{j\}} = Score_S + \Delta$.
  5. Identify the step's absolute best candidate score $V_{\max}$.
  6. **Dynamic Pruning:** Keep only paths where $Score \geq \alpha \cdot V_{\max}$ (e.g., $\alpha = 0.9$).
  7. Keep only the top B paths from this filtered list to form the new $\mathcal{B}$, and add them to $\mathcal{H}$.
- **Return:** The best unique subset $S^\star \in \mathcal{H}$ that maximizes the unconditional global evaluation metric $T(Y, X_{S^\star})$.

4.1 Synthetic Datasets

Six synthetic environments are generated designed to expose the vulnerabilities of greedy forward selection, specifically targeting structural myopia and functional redundancy. In each dataset, the true target Y is generated using a subset of features; all remaining variables up to P are independent standard normal noise ($\mathcal{N}(0, 1)$). The benchmark formulations follow established methodologies in the literature [2]:

1. **The Sniper Dataset ($N = 1,500, P = 20$):** Contains a linear distraction trap (X_0, X_1) alongside a dominant interaction (X_2, X_3):

$$Y = 5.0 \cdot \text{sign}(X_2 \cdot X_3) + 1.5(X_0 + X_1) + \epsilon \tag{2}$$

   Greedy frameworks preferentially capture the linear components, stalling before discovering the high-magnitude interaction.
2. **Interaction XOR ($N = 2,000, P = 80$):** Designed to test the limits of interaction discovery, where individual features maintain near-zero marginal correlations with Y:

$$Y = 2(X_0 \oplus X_1) + 2(X_2 \wedge X_3) + 2\neg(X_4 \oplus X_5) + 3\sin(\pi X_6)\sin(\pi X_7) + \epsilon \tag{3}$$

   where $\epsilon \sim \mathcal{N}(0, 1)$. This represents a classic landscape where greedy search completely fails.

3. **Nonlinear Hard ($N = 2,000, P = 80$):** Evaluates the robustness of the rank-based CODEC estimator across highly heterogeneous, non-linear functional architectures:

$$Y = 3\sin(X_0)+2X_1^2+1.5|X_2|+\exp(X_3/2)+2X_4X_5+\cos(X_6X_7)+\mathbb{I}(X_8 > 0)\cdot X_9+\epsilon \tag{4}$$

4. **Linear Redundant ($N = 2,000, P = 80$):** Verifies metric behavior under severe linear collinearity, where $X_{10} \dots X_{19}$ act as noisy duplicated traps ($X_{10+i} = X_i + \mathcal{N}(0, 0.2^2)$):

$$Y = \sum_{i=0}^{9} b_i X_i + \epsilon \tag{5}$$

where $b_i \sim \mathcal{U}(-3, 3)$ and $\epsilon \sim \mathcal{N}(0, 0.5^2)$.

5. **Nonlinear Redundant ($N = 2,000, P = 80$):** Evaluates whether a minimal sufficient subset is maintained when facing non-linear functional redundancy, featuring seven noisy transformed traps ($X_{10} \dots X_{16}$):

$$Y = 2\sin(X_0)+1.5X_1^2+\exp(X_2/2)+3|X_3|+2X_4X_5+\mathbb{I}(X_6 > 0)\cdot X_7+X_8-X_9+\epsilon \tag{6}$$

6. **Stress Test ($N = 5,000, P = 100$):** A larger scale mixture of linear, unary non-linear, and multi-variable interaction terms embedded within 91 noise channels to evaluate precision and scalability:

$$Y = 2X_0-1.5X_1+X_{10}+3\sin(X_{11})+2X_{12}^2+6\text{sign}(X_2\cdot X_3)+3X_4\cdot X_5+\epsilon \tag{7}$$

4.2 Real-World Datasets

The proposed methods are tested across four real-world classification benchmarks chosen for their varied feature layouts, three of which are sourced from the UCI repository [4]:

- **MADELON:** A classic NIPS 2003 challenge dataset synthesized to benchmark multivariate interaction discovery, featuring 500 dimensions with only 20 informative attributes [6].
- **Cardiotocography:** Comprises 2,126 fetal heart rate and uterine contraction diagnostic recordings across 21 continuous attributes optimized for a 3-class fetal health state classification.
- **Mice Protein Expression:** Consists of 1,080 instances measuring expression levels of 75 signaling proteins across the mouse cerebral cortex to identify distinct genotype, treatment, and behavioral cohorts.
- **Spambase:** A cybersecurity dataset consisting of 4,601 email instances quantified via 57 continuous text-frequency attributes for binary spam classification.

5 Experiments and Results

This section presents the empirical evaluation of proposed variants and the experimental setup. The variants are verified on synthetic data and validated on real-world classification tasks.

5.1 Experimental Setup and Metrics

To evaluate the proposed algorithms, the CODEC metric and selection logic are implemented in Python using `NumPy` and `Scikit-Learn`. For all algorithms, the number of nearest neighbors is fixed at $k = 3$ for the CODEC estimation. The FOCI-LA variant was configured to evaluate the top $k = 5$ candidates at each step. For FOCI-DB, the beam width was set to $B_{min} = 1$ and $B_{max} = 5$, with a dynamic threshold of 0.9 relative to the top-scoring path. These values were not tuned via grid search or sensitivity analysis; they were fixed a priori to reasonable defaults, and a systematic study of their effect on the accuracy-cost trade-off is left for future work.

The evaluation pipeline utilized three distinct classifiers to ensure the selected features were model-agnostic: Random Forest (RF), k-Nearest Neighbors (k-NN), and a Multi-Layer Perceptron (MLP). The proposed algorithms are compared against three baselines: the original greedy FOCI, HSIC Lasso [10], and a Mutual Information (MI) filter [9]. In the synthetic environment, performance is measured using Feature Recall (percentage of ground-truth features recovered). For real-world datasets, we used Classification Accuracy and Subset Size (S), utilizing a 5-fold cross-validation pipeline. The experiments were conducted on a workstation equipped with an Intel(R) Core(TM) i9-10900 CPU running at 3.70 GHz, 64 GB of RAM, and an NVIDIA GeForce RTX 2080-ti GPU. Each synthetic experiment was repeated 5 times with different random seeds, and the reported recall values are averages over those runs. The implementation is available at GitHub repository [3].

5.2 Synthetic Data: Algorithmic Verification

The synthetic experiments verify that search enhancements successfully address the "myopia" of the original greedy approach. Table 1 provides recall results across all six synthetic benchmarks. Key findings include:

- **Interaction Discovery:** In the **Sniper** dataset, original FOCI, HSIC Lasso, and Mutual Info all fall into the linear traps, whereas FOCI-LA and FOCI-DB perfectly recover the interaction. However, on the Interaction XOR dataset, recall is universally low ($< 30\%$), suggesting that pure interactions in high-noise environments remain a significant challenge even for expanded search strategies. By contrast, perfect recall on the Sniper and Linear Redundant datasets is attainable because both have a small, fixed number of

[3] `https://github.com/ArnauNeches/pyfoci`

Table 1. Feature Recall (%) across all Synthetic Datasets. Best result per row shown in **bold**.

Dataset	Original FOCI	FOCI-LA	FOCI-DB	HSIC Lasso	Mutual Info	Characteristic
The Sniper	50%	**100%**	**100%**	50%	50%	Linear Traps
Interaction XOR	0%	22%	18%	5%	**27%**	Pure Interaction
Nonlinear Hard	10%	34%	**36%**	30%	12%	Heterogeneous
Linear Redundant	80%	90%	**100%**	70%	64%	Redundancy
Nonlinear Redundant	65%	**85%**	80%	40%	42%	Redundant Traps
Stress Test	40%	**60%**	55%	35%	38%	High-Dim/Noise

truly informative features (4 and 10, respectively) embedded among noise or redundant traps, so once the search horizon is wide enough to escape the traps, identifying the remaining informative features is straightforward.

- **Search Breadth vs. Depth:** FOCI-DB slightly outperforms on the Nonlinear Hard dataset (36% vs 34% recall). This indicates that in environments with complex functional forms, expanding the search breadth can be as effective as depth-first strategies in validating dependencies.
- **Redundancy Pruning:** FOCI-DB proved most effective in the **Linear Redundant** case, achieving 100% recall by maintaining alternative paths that are often discarded by greedy selectors.
- **Overall pattern:** Across all six benchmarks, FOCI-LA and FOCI-DB recover more ground-truth features than the original greedy algorithm in five of six cases, with the sole exception (Interaction XOR) underscoring that pure, high-order interactions remain difficult to detect from rank statistics alone regardless of search strategy.

5.3 Real-World Data: Predictive Performance

On real data the relevant question is whether the selected subset translates into better downstream predictions. Table 2 summarizes the classification accuracy and subset size obtained by each selector across the four real-world datasets. "No Selector" represents the performance using the full feature set. The results highlight several critical aspects of the proposed search enhancements:

1. **Synergy Detection in MADELON:** On this highly interactive dataset, greedy FOCI stalls at 9 features ($\approx$ 80% accuracy), while FOCI-LA and FOCI-DB reach $\approx 87 - 88\%$, confirming that expanding the search horizon is essential under high interaction density.
2. **Redundancy pruning in FOCI-DB:** On Mice Protein, FOCI-DB nearly matches FOCI-LA (97.4% vs 98.8% RF accuracy) with a much smaller subset (25 vs 57 features), as beam-search pruning avoids the redundant features a pure lookahead may add.
3. **Robustness of CODEC:** Even on simpler datasets like Spambase and Cardiotocography, all FOCI-based methods stay competitive while using a fraction of the features, showing that rank-based CODEC remains a powerful primitive under more exhaustive search.

Table 2. Classification accuracy and selected subset size (S) across real-world datasets. Best result per column shown in **bold** (No Selector excluded).

Dataset	Selector	S	RF	k-NN	MLP
Cardiotocography	No Selector	21	94.49 ± 0.73	90.35 ± 1.30	89.48 ± 1.24
	Original FOCI	7	93.19 ± 1.46	90.97 ± 1.38	87.72 ± 2.30
	FOCI-LA	7	93.20 ± 1.13	**91.53 ± 0.86**	86.85 ± 2.43
	FOCI-DB	7	93.11 ± 1.17	91.31 ± 1.34	87.18 ± 2.10
	HSIC Lasso	10	**94.59 ± 0.94**	91.38 ± 1.24	**89.59 ± 1.34**
	Mutual Info	10	94.34 ± 0.79	90.78 ± 0.87	88.40 ± 1.85
MADELON	No Selector	500	70.54 ± 1.23	54.49 ± 2.31	55.51 ± 2.25
	Original FOCI	9	79.78 ± 1.89	80.61 ± 1.17	75.77 ± 1.13
	FOCI-LA	11	**87.29 ± 1.09**	87.83 ± 2.63	83.46 ± 2.92
	FOCI-DB	11	**87.31 ± 0.81**	**88.45 ± 2.26**	**83.79 ± 3.61**
	HSIC Lasso	10	68.30 ± 4.15	60.56 ± 3.24	62.82 ± 2.73
	Mutual Info	10	68.40 ± 7.14	60.31 ± 4.96	62.28 ± 3.06
Mice Protein	No Selector	75	99.41 ± 0.56	97.74 ± 1.21	98.78 ± 0.83
	Original FOCI	23	97.52 ± 1.47	94.43 ± 3.00	88.41 ± 9.93
	FOCI-LA	57	**98.83 ± 1.05**	**97.19 ± 1.76**	**96.56 ± 5.66**
	FOCI-DB	25	97.43 ± 1.49	95.35 ± 2.89	91.20 ± 8.60
	HSIC Lasso	10	93.31 ± 5.51	94.20 ± 6.92	88.02 ± 8.74
	Mutual Info	10	95.43 ± 1.72	91.20 ± 2.25	85.35 ± 6.15
Spambase	No Selector	57	95.36 ± 0.61	90.88 ± 0.83	93.54 ± 0.92
	Original FOCI	17	93.75 ± 0.91	90.29 ± 1.42	**91.50 ± 1.31**
	FOCI-LA	22	94.08 ± 0.78	90.05 ± 1.29	91.22 ± 1.30
	FOCI-DB	23	**94.10 ± 0.73**	90.03 ± 1.44	**91.52 ± 1.30**
	HSIC Lasso	10	93.34 ± 0.74	90.13 ± 0.72	91.12 ± 0.84
	Mutual Info	10	93.17 ± 0.73	**90.39 ± 0.88**	90.49 ± 1.13

4. **Bayesian Superiority Analysis:** A Bayesian signed-rank analysis across the downstream tasks (mean F1-Macro per dataset, averaged over folds, repeats, and models; ROPE of 1 percentage point) confirms these findings. Both FOCI-LA and FOCI-DB achieve high posterior probabilities of outperforming greedy *Original FOCI* ($P(\text{LA} > \text{FOCI}) = 0.843$, $P(\text{DB} > \text{FOCI}) = 0.674$) and baselines like *Mutual Info* ($P > 0.88$) and *HSIC Lasso* ($P \approx 0.80$). The full *No Selector* baseline yields higher nominal accuracy ($P \approx 0.65$–0.74) since it retains all information, but offers no dimensionality reduction.

5.4 Computational Cost and Complexity Trade-offs

Expanding the search space shifts the theoretical complexity from the original FOCI's $O(PN \log N)$ toward approximately $O(kP^2 N \log N)$ for the FOCI-LA and FOCI-DB variants. To quantify this impact, the average wall-clock execution time across representative datasets is shown in Table 3.

Table 3. Average Selection Time (Seconds).

Dataset	Original FOCI	FOCI-LA	FOCI-DB
MADELON ($P = 500$)	175.9 ± 70.05	709.58 ± 521.27	903.1 ± 459.66
Interaction XOR ($P = 80$)	4.94 ± 4.23	36.33 ± 18.36	7.43 ± 4.85
Nonlinear Hard ($P = 80$)	0.97 ± 0.01	12.68 ± 4.26	5.0 ± 3.57

The results demonstrate that while the original FOCI is efficient, the search-based variants require more time to explore feature interactions. In the MADELON dataset, the extra processing time for FOCI-LA is a justified trade-off for the substantial improvement in accuracy. However, in synthetic benchmarks such as Interaction XOR, we observe that FOCI-DB can be significantly faster than FOCI-LA due to its ability to prune unpromising paths early.

6 Discussion

Choosing between greedy FOCI and the deeper search variants mainly depends on the data structure. Synthetic experiments show that FOCI-LA and FOCI-DB recover truly relevant features better when the signal is spread across groups of variables, especially under misleading marginal effects, redundancy, or interactions that cause greedy selection to stop too early. In real-world datasets, the benefits are more dataset-dependent. For Cardiotocography and Spambase, the gains are marginal: the added search co st yields at most a one-point accuracy gain, making the original FOCI the more practical choice, and Spambase even shows that feature selection is not always beneficial, since the full feature set achieved the best Random Forest performance. In contrast, MADELON is a clear case where expanding the search horizon helps: a roughly 4–5$\times$ increase in selection time (Table 3) buys a 7–8 percentage point accuracy gain, consistent with the interaction-driven nature of this dataset.

Overall, the proposed methods should not be seen as universally better selectors in accuracy. Their main contribution is to reduce the myopia of greedy conditional-dependence selection and improve the recovery of interaction-relevant variables. The additional cost is most justified when feature interactions are expected, or when recovering variables that reflect the underlying data-generating mechanism matters as much as predictive performance. The results also support CODEC as a robust dependence criterion whose effectiveness depends not only on the measure itself but on the search strategy to explore the feature space.

7 Conclusions and Future Work

In this paper is proposed Lookahead FOCI and Dynamic Beam Search FOCI to overcome the structural myopia and path-dependent redundancy of greedy conditional-dependence selection. Both variants recover jointly informative features that the original algorithm often discards. Across six synthetic benchmarks

and four real-world datasets, the proposed variants improved interaction discovery and, on interaction-heavy datasets, downstream classification accuracy, especially on multivariate non-linear benchmarks such as MADELON, where accuracy rose from approximately 80% to 88%. Although both variants incur additional computational cost, the accuracy gains justify their use when interaction density is high.

Future work will focus on improving the scalability of these search variants: a natural first step is to parallelize the evaluation of candidate features, since both variants are embarrassingly parallel at the scoring level, and another is to develop an adaptive "hybrid search" that triggers deeper lookahead steps only when the local conditional dependence gain plateaus, preserving the speed of greedy FOCI on simple datasets while automatically escalating to a more powerful search on complex, interaction-heavy data.

Acknowledgments. This research has been supported by grant PID2023-146037OB-C21 funded by MICIU/AEI/10.13039/501100011033. The authors would like to thank the Spanish Ministry of Universities for the FPU predoctoral fellowship (FPU23/03390).

References

1. Azadkia, M., Chatterjee, S.: A simple measure of conditional dependence. Annals of Statistics **49**(6), 3070–3102 (2021)
2. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. Knowledge and Information Systems **34**(3), 483–519 (2013)
3. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Computers & Electrical Engineering **40**(1), 16–28 (2014)
4. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
5. Esmeir, S., Markovitch, S.: Lookahead-based algorithms for anytime induction of decision trees. In: Proceedings of the 21st International Conference on Machine Learning (ICML). pp. 33–40. ACM (2004)
6. Guyon, I., Gunn, S., Ben-Hur, A., Dror, G.: Result analysis of the NIPS 2003 feature selection challenge. In: Advances in Neural Information Processing Systems. vol. 17. MIT Press (2004)
7. Linares-Barrera, M.L., Jiménez-Navarro, M.J., Brito, I.S., Riquelme, J.C., Martínez-Ballesteros, M.: Evolutionary feature selection for time-series forecasting. In: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing. pp. 395–399. ACM (2024)
8. Linares-Barrera, M.L., Martínez-Ballesteros, M., García-Heredia, J.M., Riquelme, J.C.: A feature selection and association rule approach to identify genes associated with metastasis and low survival in sarcoma. In: International Conference on Hybrid Artificial Intelligence Systems. pp. 731–742. Springer (2023)
9. Vergara, J.R., Estévez, P.A.: A review of feature selection methods based on mutual information. Neural Computing and Applications **24**(1), 175–186 (2014)
10. Yamada, M., Jitkrittum, W., Sigal, L., Xing, E.P., Sugiyama, M.: High-dimensional feature selection by feature-wise kernelized lasso. Neural Computation **26**(1), 185–207 (2014)