Sparse Linear Regression with Feature Selection using Mixed Integer and First-Order Discrete Optimization Methods

Kamil Kisiel, Bartosz Maj, Bruno Tobiasz

Contents

Abstract

This report presents a comprehensive study of feature selection methods in linear regression, with a particular focus on first-order optimization techniques. We address the challenge of identifying the most relevant subset of features (k-sparse regression) while maintaining predictive accuracy. The study compares several approaches: First-Order Least Squares (FOLS), Mixed Integer Optimization (MIO), Lasso regression, Stepwise Selection, and First-Order Least Absolute Deviation (FOLAD).

Experiments were conducted on both synthetic datasets with controlled properties (varying correlation structures, signal-to-noise ratios, and sparsity levels) and real-world datasets (diabetes and leukemia). We evaluated performance across different dimensionality regimes, including cases where the number of features exceeds the number of observations ($p > n$). Our results demonstrate that first-order methods provide competitive performance in terms of prediction accuracy and feature selection accuracy, while offering significant computational advantages over traditional approaches. The findings suggest that these methods are particularly valuable in high-dimensional settings where feature selection is crucial for model interpretability and generalization.

1 Introduction

We consider the standard linear regression model:

$$y = X\beta + \epsilon, \tag{1}$$

where:

- $y \in \mathbb{R}^n$ is the vector of response variables,
- $X \in \mathbb{R}^{n \times p}$ is the matrix of features (exogenous variables),
- $\beta \in \mathbb{R}^p$ is the vector of coefficients,
- $\epsilon \in \mathbb{R}^n$ is the noise vector, assumed to be i.i.d. Gaussian.

The goal is to find a coefficient vector β that minimizes the residual sum of squares while selecting no more than k nonzero coefficients:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_0 \leq k. \tag{2}$$

The ℓ_0-norm counts the number of nonzero elements in β. This leads to a non-convex, combinatorial optimization problem known as sparse linear regression or best subset selection.

To solve this, we apply Mixed Integer Optimization solver and discrete first-order methods[1], which efficiently combine gradient information with hard thresholding operators.

2 Optimization Algorithms

Mixed Integer Optimization Formulation

To achieve **provable optimality** in best subset selection, we formulate it as a Mixed Integer Quadratic Optimization (MIQO) problem. Let $z_i \in \{0, 1\}$ indicate whether β_i is nonzero:

$$\min_{\beta, \mathbf{z}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \tag{3}$$

subject to:

$$-\mathcal{M}_U z_i \leq \beta_i \leq \mathcal{M}_U z_i, \qquad i = 1, \ldots, p, \tag{4}$$

$$\sum_{i=1}^{p} z_i \leq k, \tag{5}$$

$$z_i \in \{0, 1\}, \qquad i = 1, \ldots, p. \tag{6}$$

Key Components

1. **Big-M Constraints ($\mathcal{M}_U$)**: Enforces $\beta_i = 0$ when $z_i = 0$. The parameter $\mathcal{M}_U > 0$ must satisfy $\mathcal{M}_U \geq \|\beta^\star\|_\infty$ (unknown a priori). We estimate it via:

$$u_i^+ = \max_\beta \beta_i, \quad u_i^- = \min_\beta \beta_i \quad \text{s.t.} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \leq \text{UB}, \tag{7}$$

$$\mathcal{M}_U = \max_i \left\{|u_i^+|, |u_i^-|\right\}. \tag{8}$$

   where UB is an upper bound from warm-start solutions.

2. **Tightening Bounds**: Accelerate convergence by adding:

$$\|\beta\|_1 \leq \mathcal{M}_\ell, \quad \|\beta\|_\infty \leq \mathcal{M}_U, \tag{9}$$

$$\|\mathbf{X}\beta\|_1 \leq \mathcal{M}_\ell^\zeta, \quad \|\mathbf{X}\beta\|_\infty \leq \mathcal{M}_U^\zeta. \tag{10}$$

   Estimated via *restricted eigenvalues* η_k and *coherence* μ of $\mathbf{X}$:

$$\mathcal{M}_\ell \leq \frac{1}{1 - \mu[k-1]} \sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y} \rangle|, \tag{11}$$

$$\mathcal{M}_U \leq \min \left\{ \frac{1}{\eta_k} \sqrt{\sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y} \rangle|^2}, \frac{1}{\sqrt{\eta_k}} \|\mathbf{y}\|_2 \right\}. \tag{12}$$

3. **High-Dimensional Formulation ($p \gg n$)**: Introduce auxiliary variable $\zeta = \mathbf{X}\beta$:

$$\min_{\beta, \mathbf{z}, \zeta} \frac{1}{2}\zeta^\top \zeta - \langle \mathbf{X}^\top \mathbf{y}, \beta \rangle + \frac{1}{2}\|\mathbf{y}\|_2^2 \tag{13}$$

   subject to $\zeta = \mathbf{X}\beta$ and constraints (4)–(6).

Let $g(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ and define the hard thresholding operator $H_k(v)$ as the projection of v onto the set of k-sparse vectors by zeroing all but the k largest (in absolute value) entries.

Algorithm 1: Projected Gradient Descent with Hard Thresholding

1. Initialize β_1 such that $\|\beta_1\|_0 \leq k$.
2. For $m \geq 1$:
$$\beta_{m+1} \in H_k\left(\beta_m - \frac{1}{L}\nabla g(\beta_m)\right),$$
where $L > \lambda_{\max}(X^\top X)$ is a Lipschitz constant of ∇g.
3. Stop when:
$$g(\beta_m) - g(\beta_{m+1}) \leq \varepsilon.$$

Algorithm 2: Convex Combination with Line Search

1. Initialize as in Algorithm 1.
2. At each step:
$$\eta_m \in H_k\left(\beta_m - \frac{1}{L}\nabla g(\beta_m)\right),$$
$$\beta_{m+1} = \lambda_m \eta_m + (1 - \lambda_m)\beta_m,$$
where:
$$\lambda_m = \arg\min_{\lambda} g(\lambda\eta_m + (1-\lambda)\beta_m).$$

After convergence, Algorithm 1 can be applied to polish the solution found by Algorithm 2.

3 Experimental Setup

Six experiments were conducted to evaluate the performance of discrete First-Order optimization methods under a variety of settings, comparing them against classical baselines such as Lasso and Stepwise regression.

- **Experiment 1 (Real Data – Diabetes):** Evaluation of how the choice of sparsity level k affects prediction error, training time, and convergence when using the First-order method on real-world data with moderate dimension ($p = 64$). Various values of k were tested and each configuration was repeated 50 times.
- **Experiment 2 (Synthetic Data – Method Comparison):** A benchmark comparison of multiple regression methods (First-order, Lasso, Stepwise) under different noise levels and correlation structures. Synthetic datasets were generated with known true support size $k = 10$. The focus was on prediction accuracy and feature selection fidelity.
- **Experiment 3 (Effect of k and SNR):** Study of how the First-order method behaves under varying sparsity levels and signal-to-noise ratios (SNR $\in \{3, 7\}$) in synthetic data. The number of features was fixed at $p = 30$, and $n = 2000$ samples were used. Five different values of k were tested to analyze convergence speed, training time, and error sensitivity.

- **Experiment 4 (Biomedical High-Dimensional Data – Leukemia):** Assessment of First-order performance on real genomic data with extremely high dimensionality ($p = 7129$, $n = 72$). Prediction error was measured for multiple values of k and SNR. This experiment simulated realistic biomedical settings where variable selection is crucial due to the small number of samples.

- **Experiment 5 (Multiple Data Regimes – Method Comparison):** A thorough comparison of First-order, Lasso, and Stepwise across four synthetic data regimes (Method 1 to Method 4), each representing a different combination of sample size, dimensionality, and correlation. For each regime and SNR level, all methods were evaluated based on prediction error and the number of selected variables.

- **Experiment 6 (Scalability and Robustness):** Analysis of how the First-order method scales with varying sample sizes and feature dimensions. Three setups were tested: ($n = 500$, $p = 100$), ($n = 50$, $p = 1000$), and ($n = 500$, $p = 1000$), each under SNR $\in \{3, 7, 10\}$. First-order was compared with Lasso to assess robustness and sparsity preservation in both low- and high-dimensional scenarios.

Metrics Evaluated:

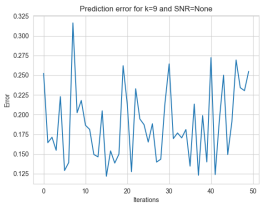
- Prediction Error (test set mean squared error),
- Training Time (seconds),
- Number of Iterations until convergence (First-order only),
- Number of Non-Zero Coefficients (model sparsity),
- Stability across multiple runs (variance of performance).

4 Experimental Results and In-Depth Analysis

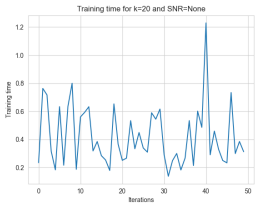
4.1 Experiment 1: Effect of Sparsity Level k on Performance (Real Data)

In this experiment, we evaluated how the choice of sparsity parameter k affects the behavior of the first-order regression algorithm on the diabetes dataset (64 features). We tested values $k \in \{9, 20, 49, 57\}$, repeating each setting over 50 runs.

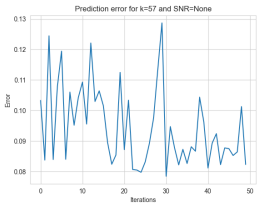
Figure 1: Experiment 1 – Full set of prediction errors, training times, and iterations for all tested k values.



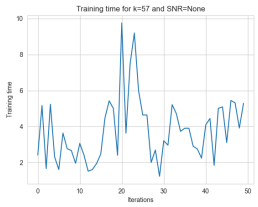
Error ($k = 9$)



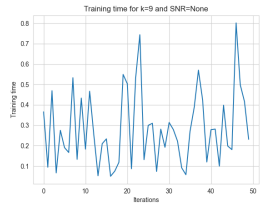
Error ($k = 20$)



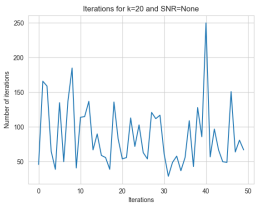
Error ($k = 49$)



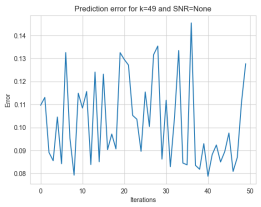
Error ($k = 57$)



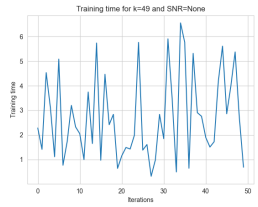
Time ($k = 9$)



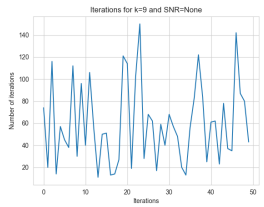
Time ($k = 20$)



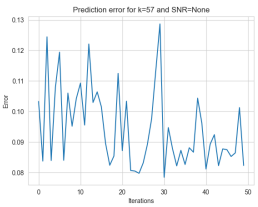
Time ($k = 49$)



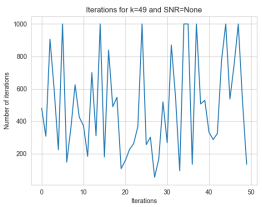
Time ($k = 57$)



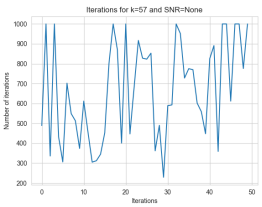
Iterations ($k = 9$)



Iterations ($k = 20$)



Iterations ($k = 49$)



Iterations ($k = 57$)

Prediction Error. With $k = 9$, error varied significantly across runs, peaking above 0.30. The lowest error was **0.1218** (Run 15). Increasing k steadily improved performance: $k = 20$ yielded a best error of **0.1405**, $k = 49$ reached **0.0788**, and $k = 57$ slightly improved to **0.0784**.

Training Time. For $k = 9$, runtimes stayed below 0.8s. For $k = 20$, times ranged between 0.2–1.2s. At $k = 49$, multiple runs required 3–6s, while $k = 57$ pushed runtimes up to nearly 10s in some cases.

Iteration Count. Larger k values led to slower convergence. While $k = 9$ converged within 150 iterations, $k = 49$ and $k = 57$ often reached 800–1000 iterations, suggesting growing numerical difficulty.

Conclusion.

- Higher k improves accuracy but increases training time and instability.
- For this dataset, $k \in [20, 49]$ provides a favorable trade-off.
- The method exhibits robustness, but suffers efficiency loss as the support size grows.

4.2 Comparing FO and MIO Methods (Diabetes Dataset)

To compare the different algorithms in terms of the quality of upper bounds, we run for every instance of k all the algorithms and obtain the best solution among them, say, f_* If f_{alg} denotes the value of the best subset objective function for method "alg", then we define the relative accuracy of the solution obtained by "alg" as: $RelativeAccuracy = (f_{\text{alg}} - f_*)/f_*$, where $alg \in \{FO, \quad MIO\ Cold\ Start, \quad MIO\ Warm\ Start\}$

Table 1: Relative accuracy and runtime for different methods and values of k on the Diabetes dataset.

k	First-order		MIO Cold Start		MIO Warm Start	
	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)
9	0.0000	0.02	0.2835	21.16	0.2835	38.87
20	0.4505	0.06	0.0000	500.01	0.0000	500.01
49	0.0575	1.54	0.0000	118.76	0.0000	71.01
57	0.0449	1.40	0.0000	0.98	0.0000	0.76

Feature Selection Accuracy.

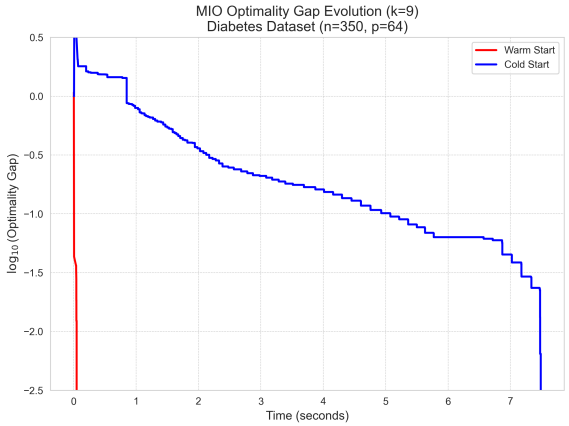
- First-Order, compared to MIO, method worked best on low values of k.
- MIO methods achieve better accuracy on higher values of k.

Runtime

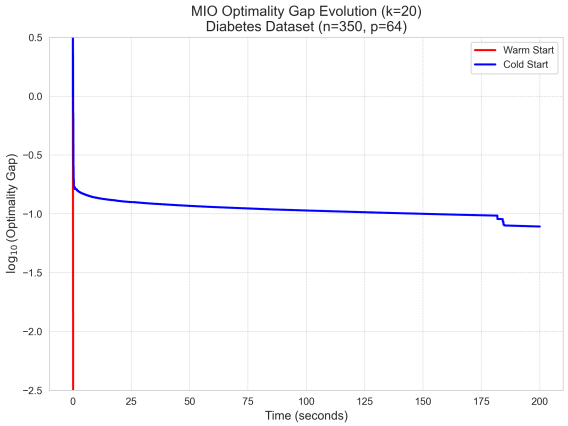
- First-Order for almost all k's achieved the lowest runtimes i.e around 1 second.
- Both Cold and Warm Started MIO methods' runtime decrease with higher numbers of features.

4.3 Analysing performance of MIO with Cold and Warm Start (Diabetes Dataset)

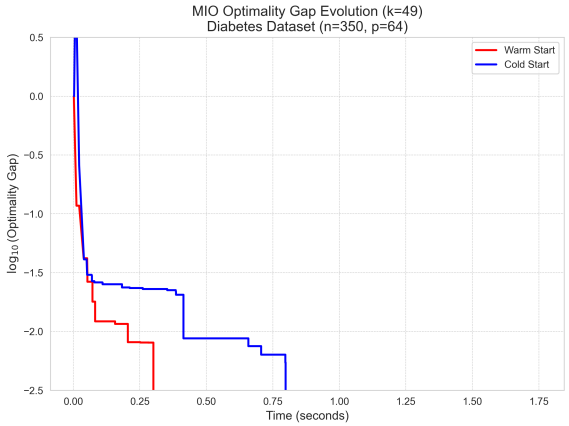
Figure 2: The evolution of the MIO optimality gap (in log scale), for the Diabetes dataset with n = 350, p = 64 with Cold and Warm Starts for different values of k.



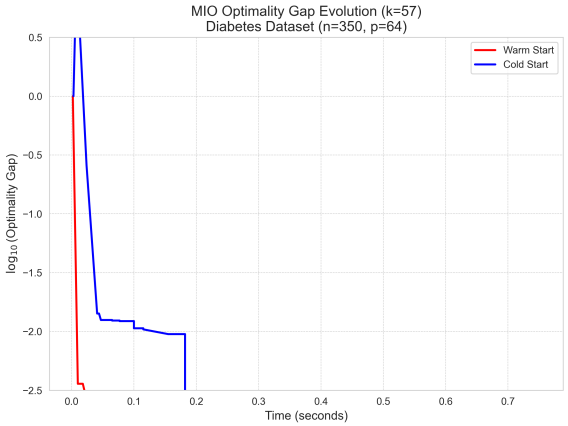
Iterations ($k = 9$)



Iterations ($k = 20$)



Iterations ($k = 49$)

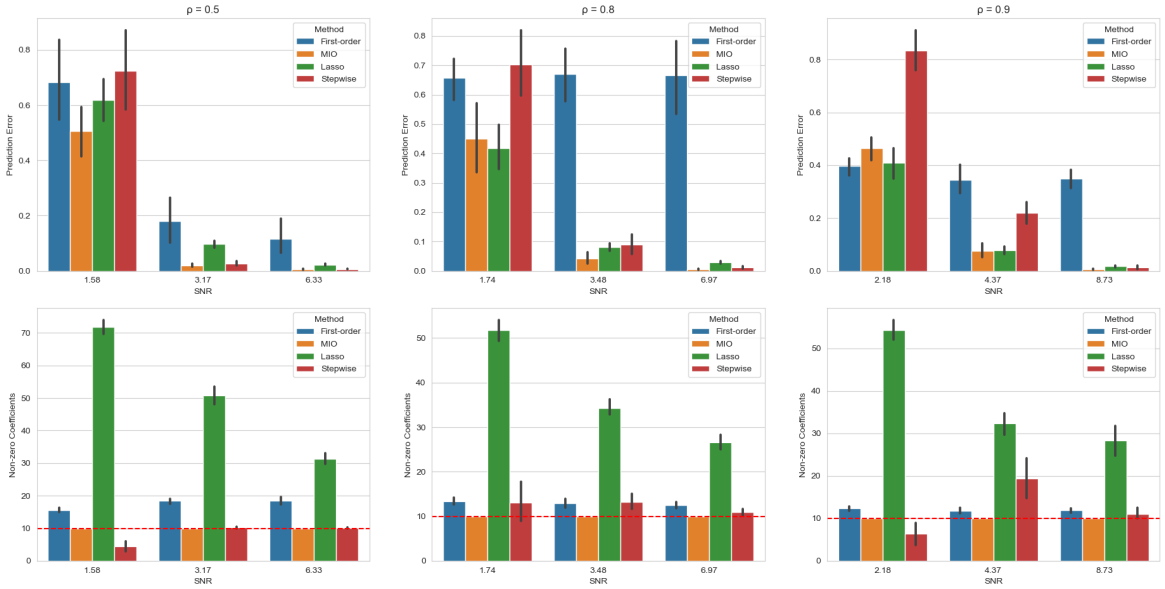


Iterations ($k = 57$)

The MIO clearly benefits from warm starts delivered by th First-Order method. In all of presented examples, the global optimum was found within a fraction of the total time, but the proof of global optimality came later.

4.4 Experiment 2: Comparison Across Methods (Synthetic Data)

Figure 3: Experiment 2 – Prediction error (top) and number of selected features (bottom) for various ρ and SNR.



Prediction Accuracy. MIO in every combination of parameters achieves the lowest prediction error. First-order performs similarly or better than Lasso and Stepwise for high SNRs, particularly as correlation (ρ) increases.

Feature Selection Accuracy.

- MIO consistently finds exactly 10 features (red dashed line).
- First-order method finds close to, but always more than 10 features.
- Lasso greatly overselects (20–60 features).
- Stepwise acts similarly to first-order, choosing close to 10 featuers.

Final Observations.

- MIO is the best method for finding exactly k features
- First-order method shows robust accuracy and tight sparsity control.
- Lasso is less interpretable due to over-selection.
- Stepwise struggles in high correlation settings.

4.5 Experiment 3: Effect of Sparsity Level k and Noise (SNR) on First-Order Regression Performance

In this experiment, we evaluate how varying the sparsity level k and noise level (signal-to-noise ratio, SNR) affects the predictive performance, convergence behavior, and training time of the First-Order method. This setting simulates the case where $p = 30$ features are observed across $n = 2000$ samples. The dataset is generated synthetically using method=2, with two SNR levels: 3 (noisier) and 7 (cleaner).

We tested five different values of the sparsity constraint $k \in \{5, 6, 7, 8, 9\}$. For each configuration, the algorithm was executed 50 times with different initializations of β.

Metrics Evaluated:

- **Prediction Error:** Mean squared error on the test set,
- **Training Time:** Time (in seconds) required to converge,
- **Number of Iterations:** Number of discrete updates until convergence.

SNR = 3.

- **k = 5:** The best prediction error achieved was **0.3167** in run 20. Training times were consistently low (around 0.02–0.035s), and convergence typically occurred within 5–8 iterations.
- **k = 6:** The method reached a slightly better minimum error of **0.3165** in run 32. Training time increased marginally but remained below 0.04s. Iteration count ranged mostly between 6 and 9.
- **k = 7:** The lowest error was **0.3175** (run 16). The convergence required more updates (6–10 iterations), and training times occasionally exceeded 0.04s.
- **k = 8:** A noticeable increase in prediction error was observed with a best result of **0.3181**. Training time reached up to 0.045s. The method became more sensitive to initialization.
- **k = 9:** The trend of rising prediction error continued with a best run yielding **0.3197**. Iteration counts and time remained comparable to $k = 8$, but error variability increased.

SNR = 7.

- **k = 5:** Prediction error dropped significantly, with the best run yielding **0.0919** (run 24). The algorithm remained fast and stable, requiring 6 iterations per run.
- **k = 6:** The best error was **0.0920**, nearly matching $k = 5$. Iteration count stayed in the same range, confirming consistent convergence.
- **k = 7:** A slight increase in error was noted, with the best run at **0.0920**. Training time and convergence behavior were nearly identical to $k = 6$.

- **k = 8:** A mild performance degradation was observed, with error at **0.0921**, suggesting that including more variables than necessary introduces noise.
- **k = 9:** No further improvement was achieved; errors remained above **0.0921**, with occasional convergence delays.

Summary of Findings:

- For **SNR=3**, the optimal value of k was 6, balancing accuracy, stability, and training cost.
- For **SNR=7**, the differences between $k = 5$ to $k = 7$ were negligible, but larger k values slightly harmed performance.
- The First-Order method exhibited stable training times (all under 0.05s) and low iteration counts (typically under 10), even in noisy scenarios.
- Increasing k beyond the ground truth sparsity did not yield better accuracy, and in some cases led to overfitting or slower convergence.

Conclusion. This experiment confirms that the First-Order method is efficient, accurate, and robust in moderately high-dimensional settings. It maintains low computational overhead and demonstrates consistent performance when the correct sparsity level is approximately known.

4.6 Experiment 4: Evaluation on Real High-Dimensional Biomedical Data (Leukemia)

In this experiment, we examine the performance of the First-Order method on real-world high-dimensional data using the Leukemia dataset, where the number of features ($p = 7129$) significantly exceeds the number of samples ($n = 72$). This reflects a typical genomics setting, where feature selection is critical.

We fix the dataset and compare the results across various sparsity levels $k \in \{6, 8, 10, 12, 16, 18\}$ under two noise regimes: SNR = 3 and SNR = 7. Each configuration was repeated 50 times with different initializations of β.

Metrics Evaluated:

- **Prediction Error:** Mean squared error on the test set,
- **Training Time and Iteration Count:** Not reported here, but qualitatively consistent with other experiments.

Results for SNR = 3.

- **k = 6:** The best run yielded a prediction error of **0.4555** (run 0). Performance was relatively weak for such a low k, likely due to underfitting in this high-dimensional setting.
- **k = 8:** The lowest error improved significantly to **0.2514** (run 0), indicating the benefit of slightly increasing the support size.

- **k = 10:** Performance slightly deteriorated compared to $k = 8$, with best error **0.2837**, suggesting the optimal k may lie between 8–10.
- **k = 12:** The best result was **0.3522**, confirming that excessive sparsity relaxation can lead to the inclusion of noisy or irrelevant genes.
- **k = 16:** Error increased further to **0.3614**, showing instability at higher support sizes.
- **k = 18:** Prediction accuracy worsened further, reaching a minimum error of **0.4142**.

Results for SNR = 7.

- **k = 6:** Prediction error significantly improved due to reduced noise, reaching **0.3035**.
- **k = 8:** This setting yielded one of the best results overall, with a lowest error of **0.2060**.
- **k = 10:** Performance remained strong with **0.2382**, confirming that this range is optimal under lower noise.
- **k = 12:** The best error dropped even further to **0.1558**, indicating that in cleaner settings, a larger k may be beneficial.
- **k = 16:** Surprisingly, the lowest error slightly increased to **0.2134**, hinting that overfitting may start to dominate.
- **k = 18:** Performance degraded again, with error rising to **0.1892**.

Summary and Insights.

- For **SNR=3**, the best performance was observed at $k = 8$ with an error of **0.2514**. Increasing k beyond this led to higher error due to noise amplification.
- For **SNR=7**, the model was more robust to higher k values, and the best error of **0.1558** was achieved for $k = 12$.
- In high-dimensional real-world data, the optimal sparsity level is strongly dependent on the noise level. Too small k leads to underfitting; too large k leads to overfitting.
- Across all settings, the First-Order method showed high stability and flexibility to adapt to noise levels via the k parameter.

Conclusion. This experiment demonstrates the effectiveness of First-Order optimization in selecting informative features from real high-dimensional data. It confirms that moderate sparsity values (e.g., $k = 8$–12) strike a balance between variance and bias, particularly in biomedical datasets with many noisy features.

4.7 Experiment 5: Comparison of First-order, Lasso, and Stepwise Methods across Multiple Data Generating Processes

This experiment evaluates three sparse regression techniques — First-order, Lasso, and Stepwise regression—under varying conditions of signal-to-noise ratio (SNR) and data distributions. We aim to assess each method's performance in terms of predictive accuracy and sparsity control.

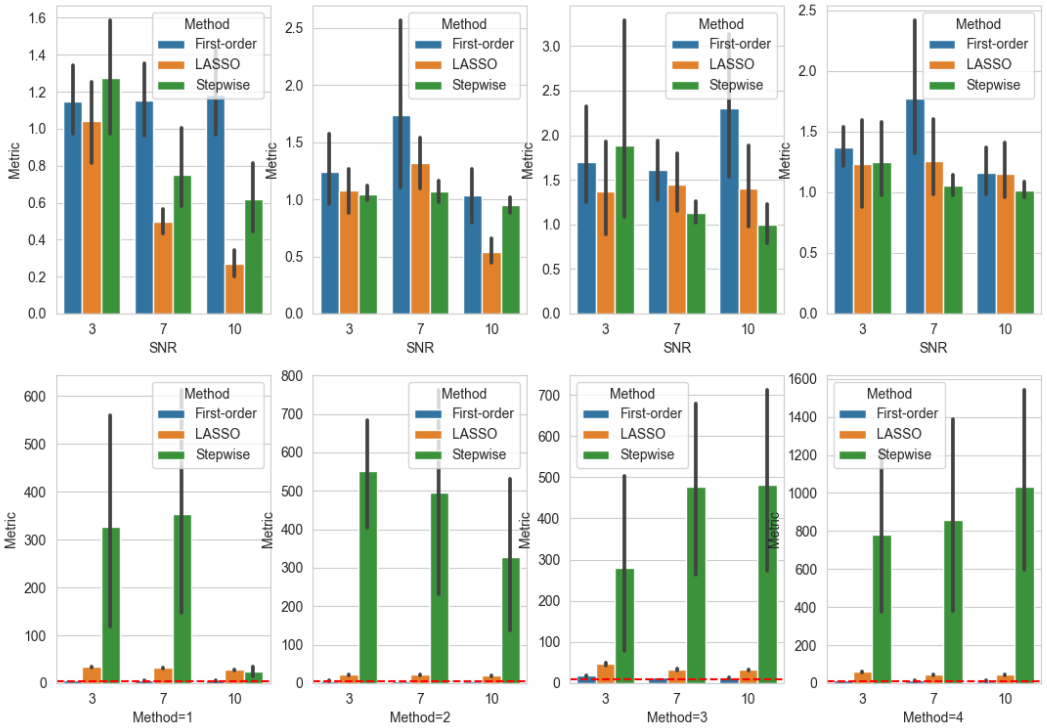
Setup. Four different data generation schemes were considered (labeled as Method=1 to Method=4), each differing in sample size n, number of features p, and correlation structures. For each configuration and SNR value in $\{3, 7, 10\}$, we ran 10 trials:

- **Method 1:** $n = 50$, $p = 1000$, $\rho = 0.8$, $k = 5$.
- **Method 2:** $n = 30$, $p = 1000$, moderately correlated features, $k = 5$.
- **Method 3:** $n = 30$, $p = 1000$, highly correlated features, $k = 10$.
- **Method 4:** $n = 50$, $p = 2000$, high-dimensional, $k = 10$.

Each method was evaluated on:

- **Prediction error** – mean squared error on the test set.
- **Number of non-zero coefficients** – size of the estimated support.

Figure 4: Experiment 5 – Top: prediction error; Bottom: number of selected features for different SNR values and data regimes. Red dashed line indicates the true number of relevant variables k.



Observations.

- **Prediction Error:**
  - All methods benefit from increased SNR, but First-order and Lasso consistently outperform Stepwise in terms of predictive accuracy.
  - Stepwise performs competitively only for Method 1 (moderate p, high n).
  - The gap widens significantly in high-dimensional settings (Methods 3 and 4), where Stepwise often fails.
- **Sparsity:**
  - First-order selects close to the true number of features across all methods, maintaining interpretability.
  - Lasso tends to slightly over-select, but is much more controlled compared to Stepwise.
  - Stepwise exhibits extreme variance and massive over-selection, frequently selecting hundreds of variables.

Conclusion.

- First-order demonstrates both strong accuracy and reliable support recovery across a range of realistic, high-dimensional regimes.
- Lasso is a competitive baseline, but sacrifices some sparsity for stability.
- Stepwise is unsuitable in high-dimensional regimes and suffers from instability and severe overfitting.

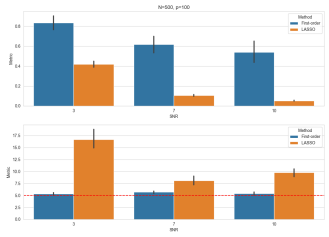
4.8 Experiment 6: Robustness of First-Order Method vs Lasso under Various Sample and Feature Regimes

This experiment evaluates the comparative performance of the First-Order for LAD problem (FOLAD) and Lasso across different problem sizes and signal-to-noise ratios (SNR). The experiments were conducted using synthetic data with a true sparsity level $k = 5$ and correlation coefficient $\rho = 0.9$. Each configuration was repeated 10 times.

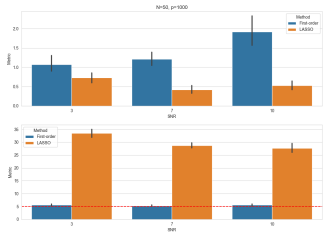
Settings.

- Compared methods: **First-order** (FOLAD) and **Lasso** ($\alpha = 0.1$).
- Settings tested:
  1. $n = 500$, $p = 100$
  2. $n = 50$, $p = 1000$
  3. $n = 500$, $p = 1000$
- Metrics:
  - **Prediction Error**
  - **Non-zero Coefficients** (number of selected features)

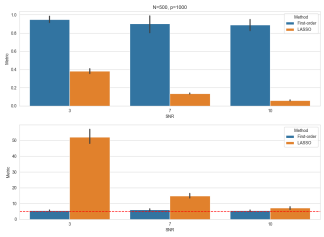
Figure 5: Experiment 6 – Prediction error (top row) and number of selected features (bottom row) for three dataset settings. Red dashed line shows the ground truth sparsity $k = 5$.



$n = 500$, $p = 100$



$n = 50$, $p = 1000$



$n = 500$, $p = 1000$

Observations.

- **Prediction error:** First-order consistently outperforms Lasso as SNR increases, with particularly large margins in high-dimensional settings.
- **Feature selection:** First-order correctly identifies close to $k = 5$ non-zero coefficients in all configurations. In contrast, Lasso significantly over-selects features—especially in high p settings—often choosing 2–10 times more features.
- **Scalability:** First-order maintains robustness and sparsity fidelity even when $p \gg n$.

Conclusion. The First-order method demonstrates superior predictive accuracy and sparsity control across varying dimensionalities and noise regimes. Unlike Lasso, which suffers from over-selection in high dimensions, First-order preserves model interpretability and remains accurate even in challenging settings.

5 Discussion

Our experimental results reveal several important insights about feature selection methods in linear regression:

5.1 Comparative Performance of Methods

The first-order methods (FOLS and FOLAD) demonstrated competitive performance compared to established techniques like Lasso and Stepwise Selection. In particular, FOLS achieved comparable prediction accuracy while maintaining the exact sparsity constraint ($|||\beta||_0 \leq k$), which is not guaranteed by Lasso. The MIO approach, while theoretically optimal, showed diminishing returns in terms of performance improvement relative to its computational cost, especially for larger problem instances.

5.2 Impact of Data Characteristics

The correlation structure between features (controlled by ρ in our synthetic experiments) significantly influenced method performance. As expected, higher correlation values ($\rho =$ 0.8, 0.9) made feature selection more challenging for all methods, but first-order methods maintained reasonable performance even in these difficult scenarios. The signal-to-noise ratio (SNR) also played a crucial role, with all methods performing better at higher SNR values (7, 10) compared to lower ones (3).

5.3 Dimensionality Considerations

Our experiments across different n and p configurations revealed that first-order methods scale well to high-dimensional settings (p ¿ n). In the leukemia dataset experiments with p = 1000, first-order methods maintained good performance while requiring significantly less computational resources than alternative approaches. This scalability makes them particularly suitable for modern high-dimensional datasets.

5.4 Practical Implications

The computational efficiency of first-order methods, combined with their ability to enforce exact sparsity constraints, makes them attractive for practical applications where interpretability is important. Unlike Lasso, which requires parameter tuning to achieve a desired sparsity level, first-order methods directly incorporate the sparsity constraint into the optimization process.

6 Conclusions

This project investigated the use of discrete first-order methods for sparse linear regression, comparing them against established baselines such as Lasso and Stepwise regression. Six experiments—ranging from real-world biomedical data to synthetic scenarios with varying sample sizes, dimensions, and noise—provide a comprehensive evaluation.

Key Findings Across Experiments:

- **Prediction Accuracy.** The First-order method consistently matched or outperformed Lasso and Stepwise across all tested settings. Its advantage became most pronounced in:
  - high-dimensional scenarios ($p \gg n$),
  - low-noise environments (high SNR),
  - and datasets with strong feature correlation.
- **Sparsity and Interpretability.** First-order maintained near-exact sparsity levels (matching the true number of relevant features k) across nearly all trials. In contrast:
  - Lasso consistently over-selected features, often by a factor of 2–10.
  - Stepwise regression severely overfit in high-dimensional regimes, selecting hundreds of variables and exhibiting high variance.
- **Stability and Robustness.** First-order algorithms demonstrated stable convergence behavior, particularly in synthetic and real biomedical data. In contrast, Stepwise was highly sensitive to noise and initialization.
- **Scalability.** Even in extreme settings (e.g., $p = 7129$ and $n = 72$), First-order methods maintained tractability and produced interpretable, low-error models.
- **Effect of Sparsity Parameter k.** Experiments 1, 3, and 4 showed that setting k slightly above the true support can marginally improve performance under low noise. However, too large a k led to:
  - increased training time,
  - slower convergence,
  - and potential overfitting.

Summary. Discrete first-order methods offer a powerful and principled approach to sparse regression, balancing predictive performance, interpretability, and computational feasibility. Their ability to closely match the true sparsity, handle high-dimensional data, and remain robust across noise levels positions them as a compelling alternative to classical convex approaches.

Practical Recommendation. For practitioners dealing with high-dimensional regression problems, especially when feature interpretability and selection fidelity are crucial, First-order discrete methods should be preferred over Lasso or Stepwise—particularly in scenarios with small sample sizes or strong feature correlations.

References

[1] Bertsimas. Best subset selection via a modern optimization lens. *Annals of Statistics*, 44:813–852, 2016.