Contents

Conclusion: Achieved Results

In this chapter, we present the results obtained on various datasets, comparing the SVR with Level Bundle Method (LBM) against a classical SVR implementation, referred to as the *Oracle*. The hyperparameters used in each experiment are reported, and efforts were made to keep them as similar as possible to ensure a fair comparison.

As stated in the introductory chapter, the goal of this project is **not** to achieve the best possible MSE through extensive hyperparameter tuning. Instead, the objective is to demonstrate that the Level Bundle Method can deliver **comparable—if not superior—performance** to that of a classical SVR.

Introduction

The dataset used are: Abalone, Airfoil, Red Whine and Friedman function

The parameters for the Level Bundle Method used are:

- **tol**

  Tolerance for stopping criterion.

- **theta**

  Controls the trade-off between cutting plane approximation and descent direction.

- **lr** *(learning rate)*

  Step size for the gradient-based update.

- **momentum**

  Momentum term to accelerate convergence and avoid local minima.

- **scale_factor**

  Scaling coefficient applied to linear and quadratic terms.

- **max_constraints**

  Maximum number of cutting planes (constraints) maintained in the bundle.

Common Parameters

This parameters are used by both SVR (Oracle and SVR with LBM).

Parameter	Value
Kernel	RBF(sigma=0.5)
C	1
Epsilon	0.05

Abalone

Oracle

Iterations	MSE	Time (s)
60	4.2186	292.2166
90	4.2186	796.9473

SVR with Level Bundle Method (LBM)

- **tol**: `1e-2`
- **theta**: `0.5`
- **lr** (learning rate): `1e-07`
- **momentum**: `0.3`
- **scale_factor**: `1e-05`
- **max_constraints**: `60`

Iterations	MSE	Time (s)
60	4.2819	28.2933
90	4.1963	88.0550

As observed, the *Oracle* achieves excellent results in just a few iterations, which suggests that `fmincon` is a highly effective solver for this specific type of problem. We also experimented with other alternatives, such as `quadprog`, but the resulting Hessian matrix proved to be unmanageable for large-scale problems like this one, making its use impractical.

An interesting observation is that once the minimum is reached, `fmincon` tends to plateau without further improvements. However, the major drawback is the training time: the average time between iterations is around **6 seconds**, resulting in very high total runtimes.

Our SVR implementation with LBM, on the other hand, performs slightly worse with the same number of iterations, but when increasing the iteration count moderately, it achieves an **even lower MSE** than the Oracle.

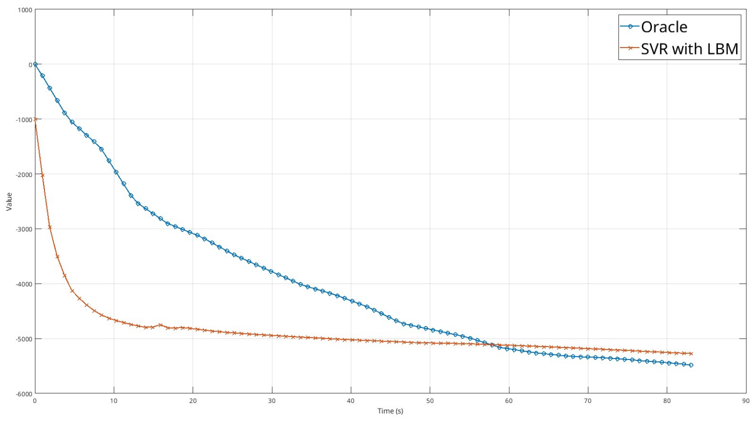

Figure 1: Plot of function values at same time t

This graph shows the behavior of the function at the same time t, using the best parameters for both SVRs. Since the times for the 2 SVRs are markedly different, the values have been interpolated to the values at the common time, so it is easier to understand what happens at time t.

One particularly interesting aspect, evident from this time-based plot, is that up to around **60 seconds**, our LBM-SVR approaches the function minimum much faster, while the Oracle lags significantly behind. Notably, there exists a time `t` at which both models reach **similar performance levels**.

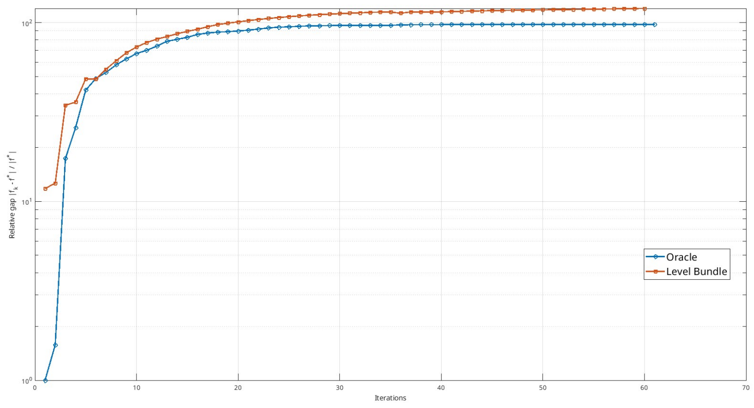

Figure 2: Relative gap between the Oracle and LBM function evaluations

Another key observation comes from the **relative gap** plot: despite the differences in runtime, the learning behavior is remarkably comparable. This indicates that the introduction of regularization had a **stabilizing effect** on the optimization dynamics.

Airfoil

Parameter	Value
Kernel	RBF(sigma=0.6)

Oracle

Iterations	MSE	Time (s)
60	10.7398	54.05
90	10.7397	94.32

SVR with Level Bundle Method (LBM)

- **tol**: `1e-2`
- **theta**: ‘0.57
- **lr** (learning rate): `1e-07`
- **momentum**: ‘0.34
- **scale_factor**: `4.6416e-05`
- **max_constraints**: `60`

Iterations	MSE	Time (s)
60	10.7976	19.13
90	10.7354	59.04

Unlike the Abalone dataset results, here the two SVR models show a highly synchronized learning behavior, supported by the minimal discrepancy in their MSE values

The time-dependent behavior is noteworthy: while the function evaluation shows a substantial initial discrepancy, the difference almost completely disappears after 10 seconds, resulting in virtually indistinguishable outputs.

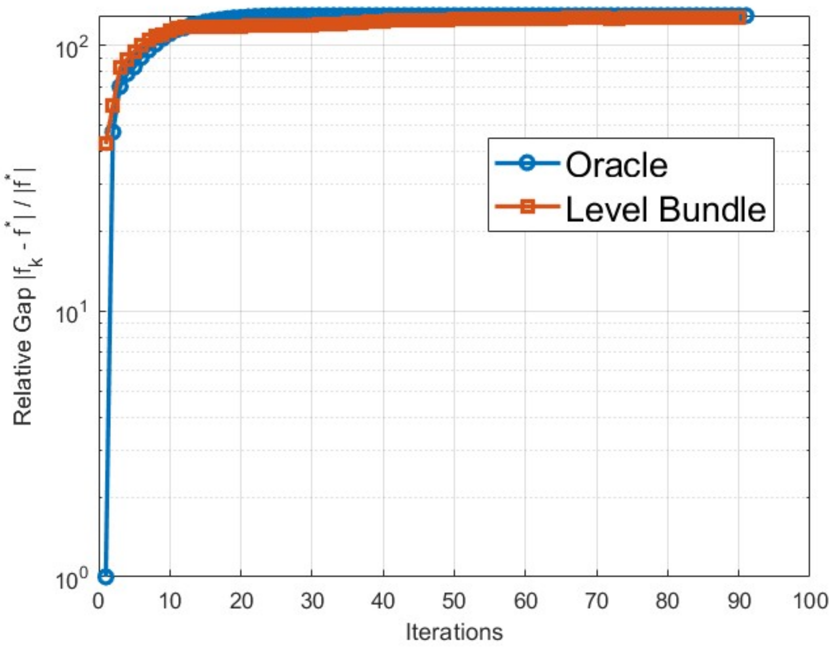

Figure 3: Relative gap between the Oracle and LBM function evaluations

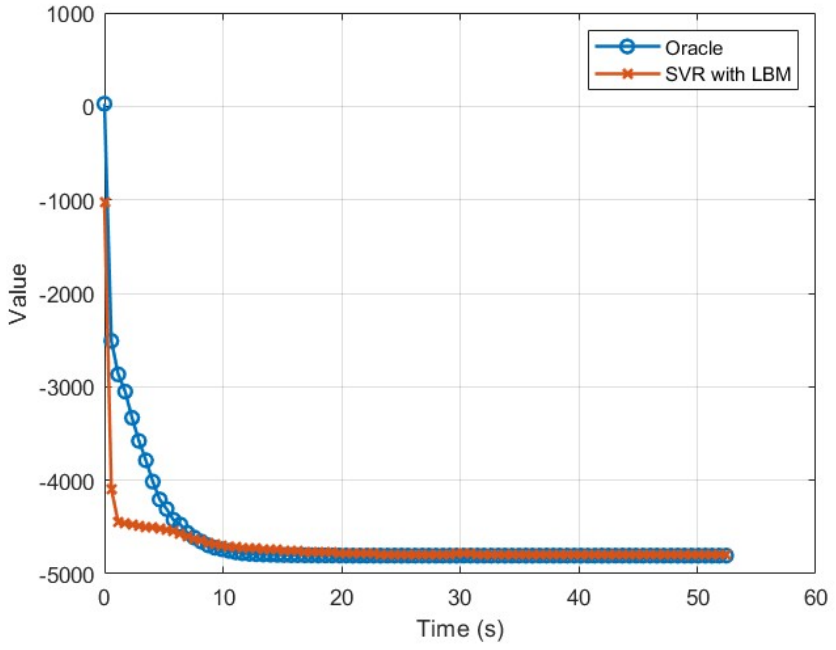

Figure 4: Plot of function values at same time t

Red Whine

Oracle

Iterations	MSE	Time (s)
60	0.055518	33.87
90	0.055519	59.08

SVR with Level Bundle Method (LBM)

- **tol**: `1e-2`
- **theta**: `0.4878`
- **lr** (learning rate): `4.1169e-06`
- **momentum**: `0.654`
- **scale_factor**: `3.7483e-05`
- **max_constraints**: `60`

Iterations	MSE	Time (s)
60	0.057641	26.29
90	0.055734	51.52

As with the Red Wine dataset, here too we observe no significant differences between the two SVR models: both exhibit excellent and comparable convergence, evidenced by the small MSE gap and similar computational time.

Friedman #1

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon$$

Where ϵ is the gaussian noise and x is a vector of feature $x = (x_1, x_2, ..., x_{10})$.

This function have the following interesting properties:

- **Complex Nonlinearity**: Includes both linear and nonlinear components:
  - Nonlinear term: $10 \cdot \sin(\pi x_1 x_2)$ – a trigonometric relationship introducing nonlinearity
  - Quadratic term: $20 \cdot (x_3 - 0.5)^2$ – a quadratic relationship
  - Linear terms: $10x_4 + 5x_5$ – direct linear relationships

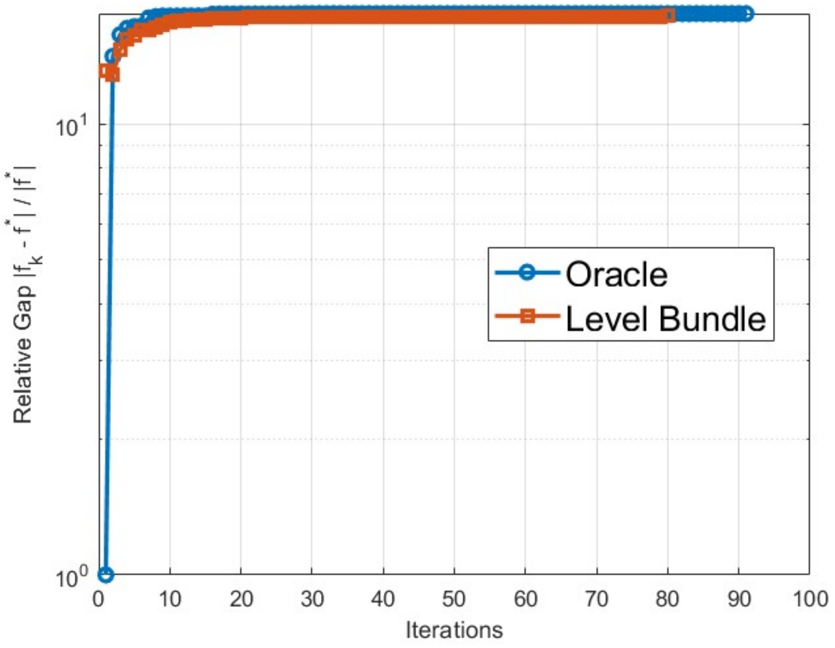

Figure 5: Relative gap between the Oracle and LBM function evaluations

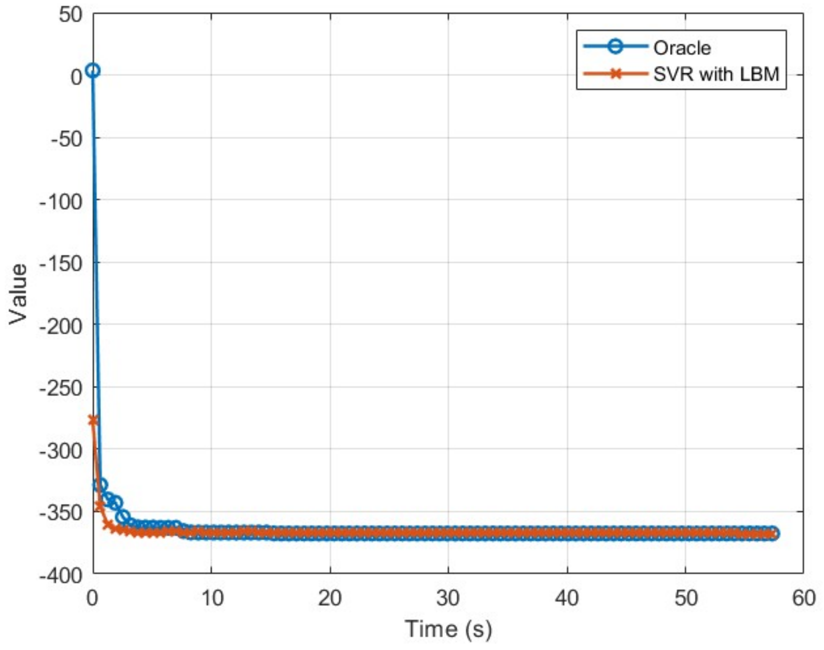

Figure 6: Plot of function values at same time t

- **Variable Interactions**: the $\sin(\pi x_1 x_2)$ component represents an interaction between x_1 and x_2, testing the algorithm's ability to capture dependencies between variables.
- **Known Theoretical Limit**: given the noise standard deviation, the minimum theoretical error can be calculated, providing a benchmark for evaluating algorithm optimality.

This is precisely why the Friedman #1 function is particularly valuable: its theoretical minimum MSE (Mean Squared Error) is known in advance. Since we typically inject Gaussian noise with a variance of 1 ($\sigma = 1$) in benchmark tests, the optimal MSE should hover around this value. A significantly lower MSE (< 0.9) suggests overfitting to the noise, while a higher MSE (> 1.2) indicates underfitting.

Only for this function we have tweaked some common parameters:

Parameter	Value
Kernel	RBF(sigma=0.6)
C	7
Epsilon	0.05

Oracle

Iterations	MSE	Time (s)
60	1.045	2.0544
90	1.059	2.3194

SVR with Level Bundle Method (LBM)

- **tol**: `1e-2`
- **theta**: `0.9`
- **lr** (learning rate): `1e-06`
- **momentum**: `0.4`
- **scale_factor**: `3e-05`
- **max_constraints**: `60`

Iterations	MSE	Time (s)
60	1.077	4.6933
90	1.059	13.487

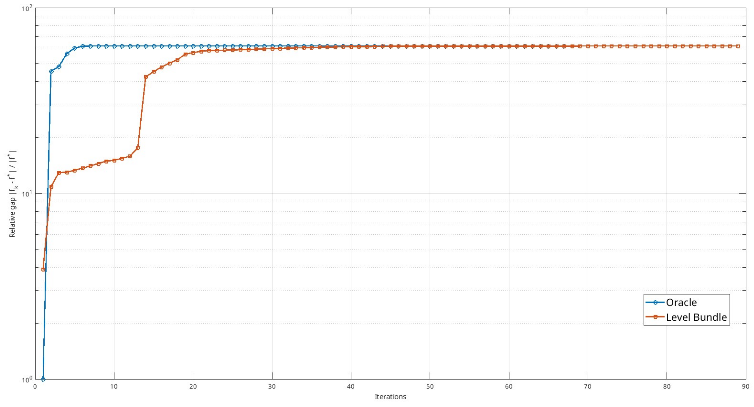

Figure 7: Relative gap between the Oracle and LBM function evaluations

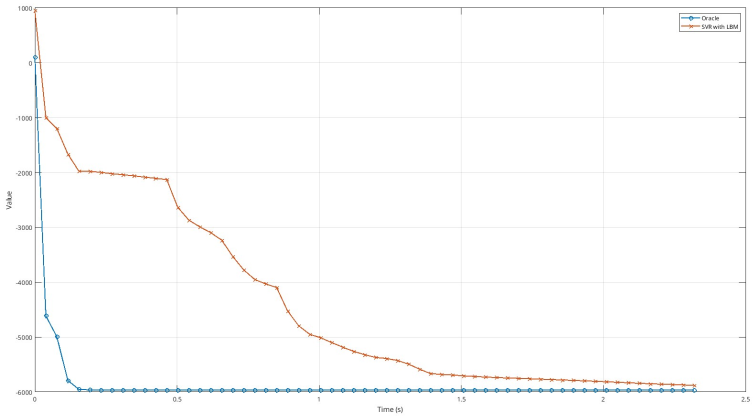

Figure 8: Plot of function values at same time t

This function is particularly interesting because, as the graphs show, both SVR implementations eventually converge to a solution, but the Oracle achieves it in significantly fewer iterations and less time. We observe that both SVRs stabilize around **25 iterations** and ~**2.5 seconds**. Before reaching these thresholds, their results varies dramatically.

Notably, unlike in other datasets, the Oracle outperforms the SVRs *in every aspect* here—likely due to the smaller (though complex) dataset size. This demonstrates that **for small datasets**, our SVR does not substantially outperform the Oracle. Conversely, for **large-scale datasets**, the Level Bundle Method handles constraints more efficiently and can even surpass the Oracle's solutions within reasonable timeframes.

Summary

Dataset	Metodo	Iterazioni	MSE	Tempo (s)
Abalone	Oracle	60	4.2186	292.2166
Abalone	LBM (SVR)	60	4.2819	28.2933
Abalone	Oracle	90	4.2186	796.9473
Abalone	LBM (SVR)	90	4.1963	88.0550
Airfoil	Oracle	60	10.7398	54.05
Airfoil	LBM (SVR)	60	10.7976	19.13
Airfoil	Oracle	90	10.7397	94.32
Airfoil	LBM (SVR)	90	10.7354	59.04
Red Whine	Oracle	60	0.055518	33.87
Red Whine	LBM (SVR)	60	0.057641	26.29
Red Whine	Oracle	90	0.055519	59.08
Red Whine	LBM (SVR)	90	0.055734	51.52
Friedman #1	Oracle	60	1.045	2.0544
Friedman #1	LBM (SVR)	60	1.077	4.6933
Friedman #1	Oracle	90	1.059	2.3194
Friedman #1	LBM (SVR)	90	1.059	13.487