COMP 551 Assignment 2 Report

Author Jinghan Zheng, Yunjie Xiao, Tailai Li

Abstract

In this assignment, we work with synthetic data to explore key concepts in linear regression, model complexity, and regularization. The experiments aim to deepen understanding of the bias-variance trade-off, cross-validation, and the effects of L1 and L2 penalties on model behavior and optimization. In Task 1, we generate nonlinear synthetic data and fit linear regression models using Gaussian basis functions of varying complexity to study underfitting and overfitting. Task 2 extends this analysis by repeating experiments to quantify bias and variance through multiple model fits and error visualizations (test and training). In Task 3, we incorporate L1 (Lasso) and L2 (Ridge) regularization, using 10-fold cross-validation to select the optimal regularization strength and examine how λ influences bias, variance, and generalization. Finally, Task 4 visualizes the impact of regularization on the loss surface and optimization paths. Our main findings are that using Gaussian bases, too few bases underfit and too many overfit, with $D = 10$ optimal. L1 promotes sparsity, L2 shrinks smoothly. Contour plots reveal L1 solutions align with axes while L2 remains elliptical. Cross-validation balances bias-variance, and ISTA accelerates L1 optimization, producing exact sparse solutions efficiently.

Keywords: Synthetic Data, Linear Regression, Overfitting, Underfitting, Gaussian Basis Functions, Bias-Variance Tradeoff, Regularization with Cross-Validation, L1, L2, Generalization.

1 Task 1

1.1 Data Generation & Non-Linear Base Functions Model Fitting

We generated 100 data points sampled from the nonlinear function

$$y(x) = (\log x + 1)\cos(x) + \sin(2x) + \varepsilon,$$

where $x \sim \text{Uniform}(0, 10)$ and $\varepsilon \sim \mathcal{N}(0, 1)$.

To capture nonlinear patterns using linear regression, we transformed the inputs using Gaussian basis functions defined as

$$\phi(x; \mu, \sigma) = \exp\left[-\frac{(x-\mu)^2}{\sigma^2}\right],$$

where $\sigma = 1$ and the means are spaced uniformly:

$$\mu_i = x_{\min} + \frac{x_{\max} - x_{\min}}{D-1}\, i, \quad i = 0, 1, \ldots, D-1.$$

Models were fitted with increasing complexity by varying the number of basis functions $D = 0, 5, 10, \ldots, 45$. The data were split into training and validation sets, and the sum of squared errors (SSE) was computed for each model to select the number of Gaussian bases D that minimized the validation error, balancing underfitting and overfitting.

1.2 SSE for Training and Validation Analysis

The dataset was divided into training and validation sets, and the sum of squared errors (SSE) was computed for each model with different numbers of Gaussian basis functions. The optimal model complexity was chosen as the number of bases D that minimized the validation SSE, effectively balancing underfitting and overfitting.

Number of Bases (D)	Training SSE	Validation SSE
0	362.16	110.09
5	183.07	75.69
10	72.87	**26.01**
15	66.21	34.97
20	60.72	91.42
25	49.32	567.85
30	44.79	3530.10
35	38.09	5993.56
40	36.48	54307.71
45	36.72	75734.07

Table 1: Training and validation SSE for different numbers of Gaussian bases. The validation error is minimized when $D = 10$.

From Table 1, the training SSE consistently decreases as the number of Gaussian bases D increases, indicating that models with more basis functions fit the training data more closely, and the training error decreases as the basis function increases. However, the validation SSE decreases only up to $D = 10$, where it reaches its minimum value of 26.01, and then rises sharply for larger D. It demonstrates that underfitting (when too few bases) occurs when the model fails to capture the nonlinear structure, and overfitting (when too many bases) occurs when the model fits noise instead of the true data. The validation set thus helps identify the optimal model complexity-$D = 10$ gives the lowest validation error-which balances underfitting and overfitting.

2 Task 2

In task 2, we repeat the fitting process from Task 1 ten times using newly generated noisy data. For each number of Gaussian basis functions, we fit the model, plot all fitted curves together, and compute the average training and test errors. By observing this average training and validation SSE, we see that it follows almost the same pattern as

Table 2: Average Training and Validation SSE for Different Numbers of Gaussian Bases

D	Train SSE	Val SSE	D	Train SSE	Val SSE
0	354.65	1.28×10^{2}	25	47.22	4.90×10^{1}
5	209.75	6.97×10^{1}	30	41.96	1.88×10^{3}
10	61.77	3.30×10^{1}	35	37.41	2.00×10^{5}
15	56.47	3.66×10^{1}	40	33.92	1.03×10^{7}
20	50.72	4.09×10^{1}	45	33.12	1.02×10^{7}

the single fitting model in task 1. In terms of the bias-variance tradeoff, we observed that For small D, the model is too simple and cannot capture the data, indicating high bias and low variance (underfitting). For large D, validation SSE increases dramatically while training SSE remains low, indicating low bias but high variance (overfitting). We should balance the bias and variance. The validation SSE is minimized at $D = 10$ (33.0), making it the optimal choice for balancing bias and variance, in which the training error is also not very high.

3 Task 3

1. Plots showing the train and validation error vs. λ for L1 and L2 regularization.

2. Plots showing the bias squared, variance, (Bias^2 + Variance), and (Bias^2 + Variance + noise_variance) vs. λ for L1 and L2 regularization.

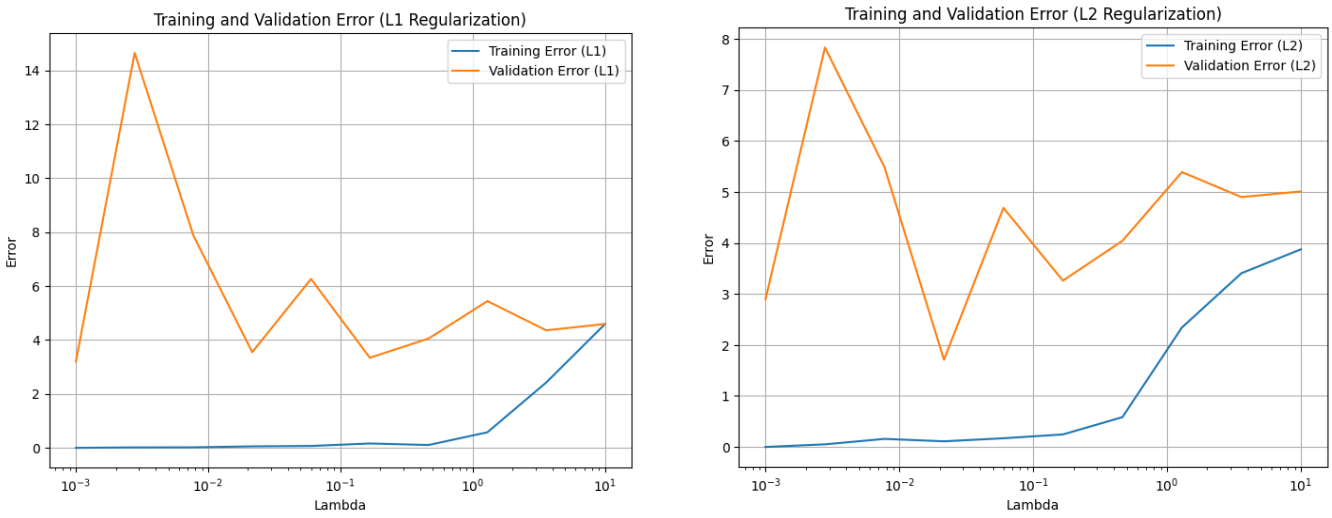


Figure 1: Train and validation error vs. λ for L1 (left) and L2 (right) regularization

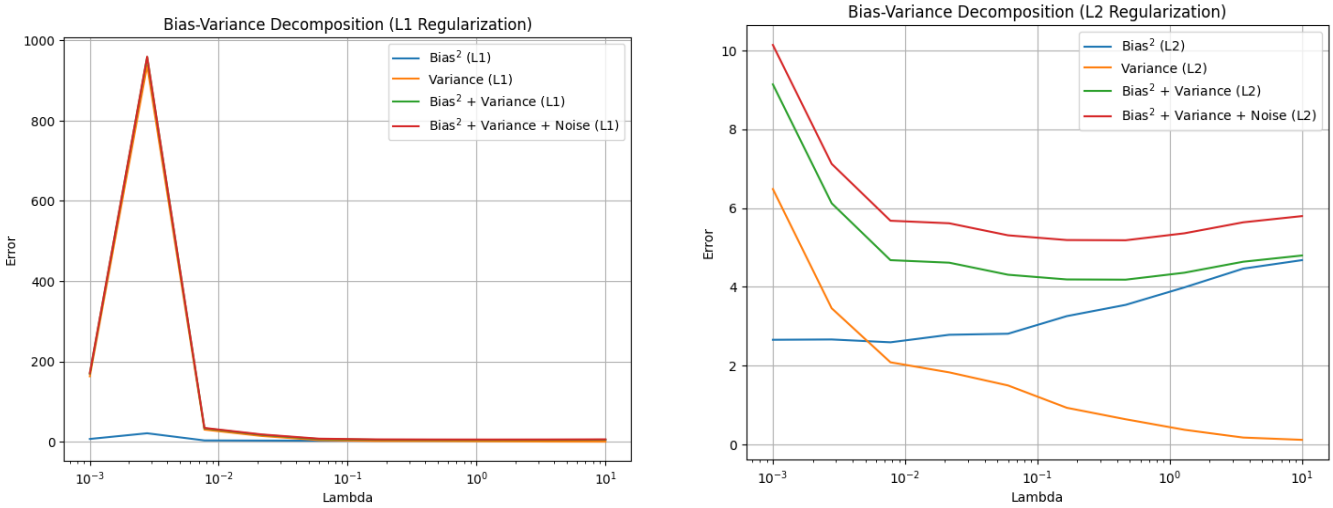


Figure 2: Bias squared, variance, (Bias^2 + Variance) and (Bias^2 + Variance + noise_variance) vs. λ for L1 (left) and L2 (right) regularization

3. Analysis of the results

- L1 Regularization:
  - Bias-squared: As λ increases, bias-squared increases slightly, then decreases slightly and increases again.
  - Variance: As λ increases, variance increases sharply at first and then decreases sharply.
  - Bias-squared + Variance: As λ increases, this value first increases sharply and then decreases sharply.
  - Bias-squared + Variance + noise_variance: Same as without noise variance.
  - Reasoning: When λ is small, the regularization is weak, leading to low bias and high variance. Although bias increases slightly at first, it's too small to be important. As λ increases, the model becomes more constrained, leading to a significant change in variance and a slight increase in bias. Overall, variance dominates, so the sum follows the variance trend.
- L2 Regularization:
  - Bias-squared: As λ increases, bias squared increases steadily.
  - Variance: As λ increases, variance decreases steadily.
  - Bias-squared + Variance: As λ increases, this value decreases first and then increases.
  - Bias-squared + Variance + noise_variance: Same as without noise variance but with a small offset.
  - Reasoning: When λ is small, the regularization is weak, the model tends to overfit leading to low bias and high variance. As λ increases, the model becomes more constrained, leading to a steady increase in bias as it underfits the data, while variance decreases steadily. Overall, the sum of bias-squared and variance initially decreases and then increases, indicating a trade-off between bias and variance. And there's a lowest point where the model achieves an optimal balance between bias and variance.

4. Selection of the optimal λ

- Optimal Lambda for L2 Regularization: 0.0215 (Validation Error: 1.7094)
- Reasoning: For L2 regularization, as λ increases, the validation error appears to increase sharply and then decrease to reach a minimum point before increasing again overall. This is likely due to the bias-variance trade-off, where a small λ leads to overfitting (high variance) and a large λ leads to underfitting (high bias). The optimal λ balances these two effects, which is the value in the middle range.
- Optimal Lambda for L1 Regularization: 0.0010 (Validation Error: 3.1976)
- Reasoning: For L1 regularization, as λ increases, the validation error appears to increase sharply and then decreases and then fluctuates at a low level overall. The discussion of the bias-variance trade-off is similar to L2 regularization. However, the least validation error is achieved at small λ values but it's almost the same as other theoretical minimum points. Which could be due to the simplicity of the underlying relationship in the data.

4 Task 4

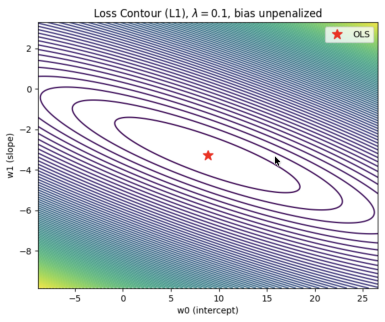
We generated synthetic linear data following the relationship

$$y = -3x + 8 + 2\varepsilon,$$

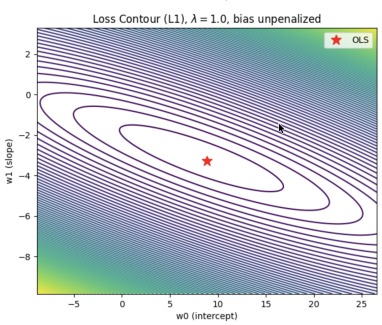
where ε is Gaussian noise with mean 0 and variance 1. We then applied both L1 and L2 regularization using gradient descent to fit the data. Then, we generate 30 data points with x uniformly distributed between 0 and 10.

4.1 Contour plots of the loss function for L1 and L2 regularization with varying strengths.

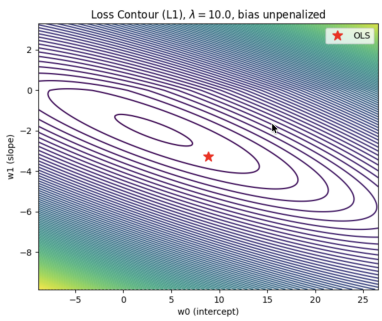
- The graph for lambda = 0.01 and 100 are not shown due to page limit.



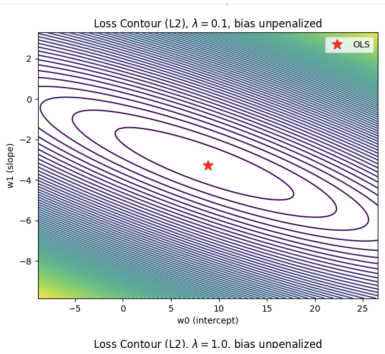
L1 ($\lambda = 0.1$)



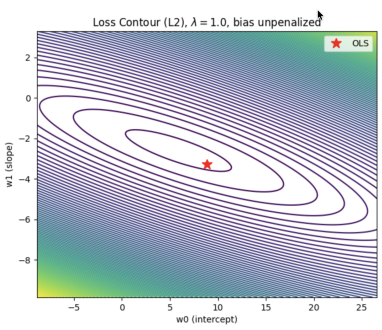
L1 ($\lambda = 1$)



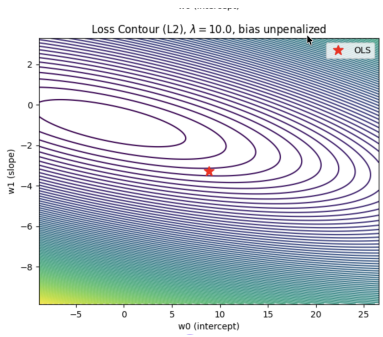
L1 ($\lambda = 10$)



L2 ($\lambda = 0.1$)



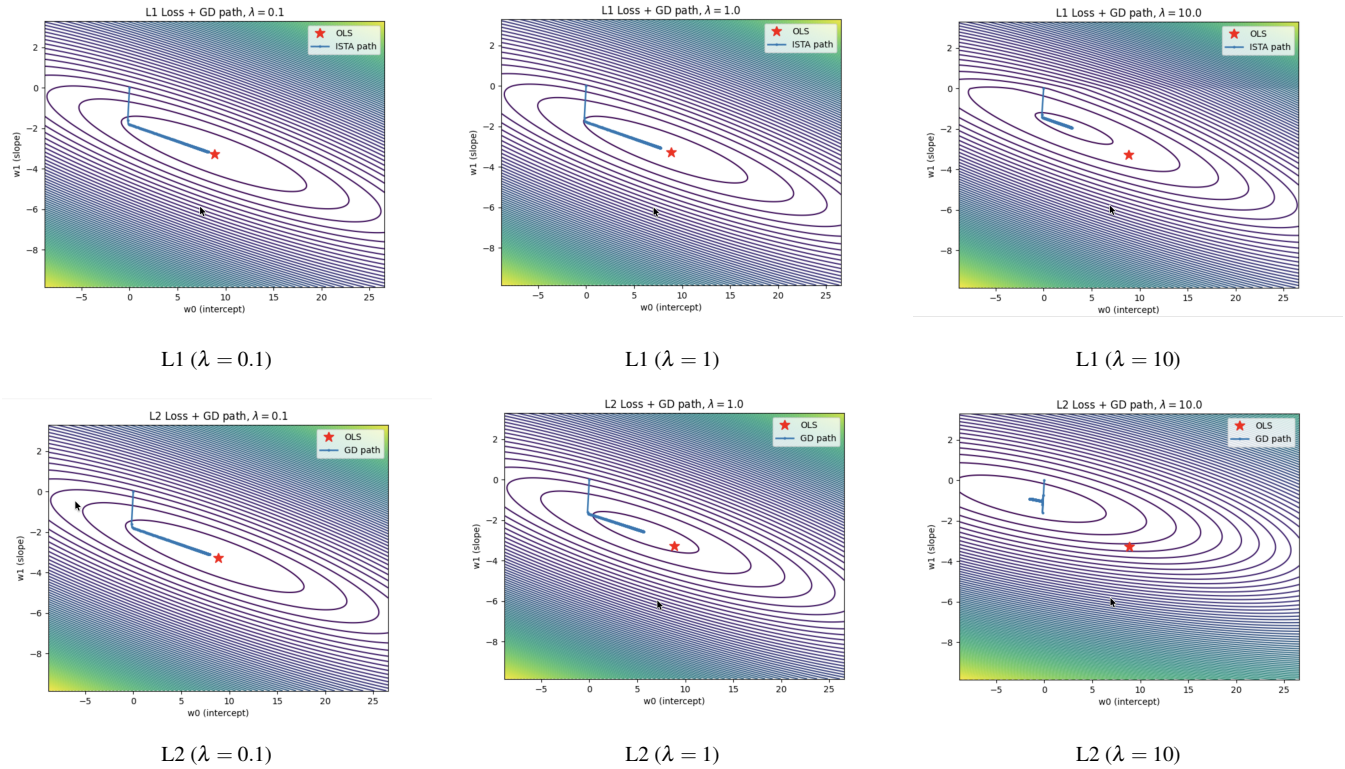
L2 ($\lambda = 1$)



L2 ($\lambda = 10$)

Figure 3: Loss contours for L1 and L2 regularization with different λ values.

4.2 Optimization Paths: Plots showing the trajectory of gradient descent for different regularization strengths.



L1 ($\lambda = 0.1$) L1 ($\lambda = 1$) L1 ($\lambda = 10$)

L2 ($\lambda = 0.1$) L2 ($\lambda = 1$) L2 ($\lambda = 10$)

Figure 4: Loss contours and optimization trajectories for L1 and L2 regularization with different λ values.

4.3 Analysis

4.3.1 L1 vs. L2: What the Plots Reveal

- **L_1 promotes sparsity.**
  In the L_1 contour plots, the level sets are *diamond* with sharp corners on the coordinate axes. When these diamonds intersect the quadratic MSE bowl, the minimizer often lands exactly on an axis, i.e., one coefficient becomes zero.

- **L_2 shrinks but rarely zeros.**
  The contours ofL_2 are *elliptical*; the quadratic penalty $\lambda \|w\|_2^2$ pulls solutions towards the origin smoothly. As λ grows, both b (if penalized) and a shrink uniformly, but the coefficients do not reach exactly zero unless the data term forces it. The GD paths under L_2 are smooth, radially contracting curves that settle at a small-norm solution rather than sparse points aligned with the axis.

4.3.2 Different values of lambda affect the optimization paths and loss landscape

- **Small λ ($\approx 10^{-2}$–10^{-1}):**
  The landscapes appear close to the unregularized mean squared error (MSE) bowl. For L_2 regularization, gradient descent paths meander along the curved valley before convergence. For L_1 regularization, trajectories behave similarly to plain gradient descent and seldom reach exactly zero coefficients.

- **Moderate λ (≈ 1):**
  The elliptical contours of L_2 tighten as λ increases—optimization paths become shorter and the conditioning improves, resulting in faster and more stable convergence. The weights shrink noticeably but typically remain nonzero. In contrast, L_1 contours become more diamond-shaped; optimization trajectories hit the coordinate axes earlier, and we observe exact zeros in the penalized coefficients due to the soft-thresholding effect.

- **Large λ ($\gg 1$):**
  The penalty term dominates the objective function. Under L_2 regularization, coefficients are driven toward

small-norm solutions (strong shrinkage) but still remain nonzero. Under L_1 regularization, large λ values create broad regions where the optimum lies on a coordinate axis (and, if the bias were penalized, at the corners), leading to sparse solutions where one or more coefficients are exactly zero.

4.4 Creativity: Using Proximal Gradient (ISTA) [1] for L1 Instead of Plain GD

Motivation. For L1-regularized problems (e.g., Lasso), the objective $F(w) = \frac{1}{2N}\|Xw - y\|_2^2 + \lambda\|w\|_1$ is non-smooth at zero. Plain (sub)gradient descent (GD) suffers from slow and unstable convergence around $w_j \approx 0$ and rarely produces exact zeros. To better exploit the problem structure and promote true sparsity, we replace subgradient GD with the *Iterative Shrinkage-Thresholding Algorithm* (ISTA), a proximal-gradient method.

Method: With our scaling of the data term $f(w) = \frac{1}{2N}\|Xw - y\|_2^2$, the gradient is $\nabla f(w) = \frac{1}{N}X^\top(Xw - y)$ and a valid stepsize is $\alpha \le 1/L$, where $L = \|X\|_2^2/N$. ISTA updates are:

$$w_{k+1} = \mathrm{prox}_{\alpha\lambda\|\cdot\|_1}(w_k - \alpha\nabla f(w_k)) = \mathrm{soft}\Big(w_k - \alpha\tfrac{1}{N}X^\top(Xw_k - y),\, \alpha\lambda\Big),$$

where $\mathrm{soft}(u,\tau) = \mathrm{sign}(u)\cdot\max(|u| - \tau, 0)$ is applied elementwise. Following standard practice, we do not penalize the intercept; thus the proximal step is applied only to the non-bias coefficients.

Algorithm:

Input: X, y, λ, steps T, stepsize $\alpha = \frac{0.9}{L}$

Initialize: $w_0 = \mathbf{0}$

For $k = 0, \ldots, T-1$: $\begin{cases} u_k = w_k - \alpha\,\frac{1}{N}X^\top(Xw_k - y), \\ \text{(no bias shrink)} \quad w_{k+1} = \big[u_k^{(\text{bias})},\ \mathrm{soft}(u_k^{(\text{non-bias})},\ \alpha\lambda)\big]. \end{cases}$

Convergence and Complexity: ISTA enjoys an $\mathcal{O}(1/k)$ convergence rate in objective value and yields *exact sparsity* via soft-thresholding. Each iteration costs $\mathcal{O}(Nd)$ (a matrix–vector multiply and a proximal step), same order as GD, but with better practical behavior near zero. (Optionally, FISTA adds Nesterov momentum to achieve $\mathcal{O}(1/k^2)$.)

Visualization: On the L1 loss contours (diamond-shaped), ISTA trajectories bend toward an axis and then slide along it, reflecting soft-thresholding; this produces exact zeros in the penalized coefficients. In contrast, subgradient GD exhibits zig-zag behavior around the kinks and converges more slowly.

5 Discussion and Conclusions

Using Gaussian basis functions, we observed that too few bases led to underfitting while too many caused overfitting, with the optimal model at $D = 10$. Repeated experiments confirmed this bias-variance tradeoff, where we have to find the optimal balance between bias and variance to do better generalization. Through L1 (Lasso) and L2 (Ridge) regularization, we saw that L1 encourages sparsity by setting some weights to zero, while L2 shrinks weights smoothly. Overall, cross-validation proved essential for balancing bias and variance and selecting optimal model complexity. For originality, compared to plain gradient descent, ISTA yields faster, more stable optimization and naturally sets some coefficients to zero, highlighting the practical advantage of using structure-aware algorithms for sparse models. Future investigations could explore nonlinear models with adaptive basis functions, alternative optimization techniques like FISTA or coordinate descent, and regularization methods such as elastic net for combining the strengths of L1 and L2.

6 Statement of Contributions

Tailai Li led the data generation and model fitting experiments (Tasks 1–2). Yunjie Xiao focused on implementing and analyzing regularization methods (Task 3). Jinghan Zheng developed the visualization of loss surfaces, optimization paths, and the ISTA implementation (Task 4). All members jointly wrote the paper and reviewed the final submission.

References

[1] Neal Parikh and Stephen Boyd. *Proximal Algorithms*, volume 1 of *Foundations and Trends in Optimization*. Now Publishers Inc., 2014.