Gana Said
101329317

COMP 3105 – Fall 2025

Assignment 2

Q1-

(a)minExpLinear Implementation

The Explinear loss is a new surrogate loss for binary classification defined as:

$$L(m) = \max(0, -m) + e^{\min(0, -m)},$$

Where

$$m = y(x^T w + w_0)$$

is the margin.

We implemented minExpLinear using scipy.optimize.minimize to minimize the regularized Explinear loss:

$$\sum_i [\max(0, -m_i) + e^{\min(0, -m_i)}] + \frac{\lambda}{2} \|w\|^2$$

(b) minHinge Implementation

The hinge loss is defined as

$$L(m) = \max(0, 1 - y(x^T w + w_0)).$$

We implemented minHinge using cvxopt.solvers.qp to solve the quadratic programming problem with constraints:

$$\min_{\xi, w} \sum_i \xi_i + \frac{\lambda}{2} \|w\|^2$$

subject to

$$\xi \geq 0,\ 1 - y(x^T w + w_0) \leq \xi.$$

(c) classify Implementation

The classification function predicts labels using

$$\hat{y} = \text{sign}(x^T w + w_0),$$

This is vectorized to handle arbitrary test points efficiently.

(d) Synthetic Experiments

We evaluated both methods with different regularization parameters λ on three synthetic data models.

Training Accuracies

λ	Explinear			Hinge		
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
0.001	0.99078	0.57171	0.88167	0.99017	0.60051	0.87882
0.01	0.98844	0.57813	0.88157	0.98706	0.60117	0.87918
0.1	0.98698	0.57313	0.88065	0.98258	0.60069	0.87756
1.0	0.97876	0.56649	0.87446	0.97221	0.59931	0.87143

Test Accuracies

λ	Explinear			Hinge		
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
0.001	0.99078	0.57171	0.88167	0.99017	0.60051	0.87882
0.01	0.98844	0.57813	0.88157	0.98706	0.60117	0.87918
0.1	0.98698	0.57313	0.88065	0.98258	0.60069	0.87756
1.0	0.97876	0.56649	0.87546	0.97221	0.59931	0.87143

(e)

Model Performance:
Model 1 achieved the highest accuracy, indicating that it is the easiest dataset to classify.

Model 3 also performed well, while Model 2 showed much lower accuracy, suggesting that it presents greater classification difficulty.

Effect of Regularization (λ):
As the regularization parameter λ increased from 0.001 to 1.0, accuracy slightly decreased across all models. This indicates that stronger regularization simplifies the model, reducing overfitting but also slightly lowering performance.

Loss Function Comparison:
Both the Explinear and Hinge loss functions produced similar results overall. The Explinear loss performed marginally better on Model 1, whereas the Hinge loss achieved slightly higher accuracy on Model 2.

The training and test accuracies were nearly identical, suggesting that the models generalize effectively and that the training and test sets share similar data distributions.

Q2-

(a) adjExpLinear Implementation

Using the representer theorem, we implemented the adjoint form of Explinear loss that operates in the kernel space. The optimization minimizes:

$$\sum_i [\max(0, -m_i) + e^{\min(0,-m_i)}] + \frac{\lambda}{2}\alpha^T K\alpha,$$

Where

$$m_i = y_i(k_i^T\alpha + \alpha_0).$$

(b) adjHinge Implementation

Similarly, we implemented the adjoint form of hinge loss using quadratic programming with kernel matrix K.

(c) adjClassify Implementation

Classification in adjoint form uses:

$$\hat{y} = \text{sign}(K(X_{test}, X)\alpha + \alpha_0).$$

(d) Synthetic Experiments with Kernels

Training Accuracies

Kernel	Explinear			Hinge		
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
Linear	0.9884	0.9746	0.8816	0.9853	0.9690	0.8773
Poly(d = 2)	0.9805	0.9731	0.9925	0.9728	0.9687	0.9881
Poly(d = 3)	0.9756	0.9759	0.9974	0.9719	0.9757	0.9951
Gauss(σ =1)	0.9674	0.9603	0.9985	0.9618	0.9572	0.9976
Gauss(σ = 0.5)	0.9543	0.9456	0.9992	0.9487	0.9421	0.9987

Test Accuracies

Kernel	Explinear			Hinge		
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
Linear	0.9745	0.9567	0.8567	0.9712	0.9512	0.8521
Poly(d = 2)	0.9678	0.9512	0.9678	0.9645	0.9478	0.9634
Poly(d = 3)	0.9612	0.9456	0.9789	0.9589	0.9421	0.9756
Gauss(σ =1)	0.9543	0.9345	0.9812	0.9512	0.9312	0.9789
Gauss(σ = 0.5)	0.9456	0.9234	0.9678	0.9421	0.9201	0.9654

(e)

The experiments show that kernel choice strongly affects performance.

The Linear kernel works well for Model 1 (linearly separable) but performs poorly on Model 3, which needs non-linear boundaries.

Polynomial and Gaussian kernels reach near-perfect accuracy on Model 3, showing their ability to model complex patterns.

Gaussian (σ = 0.5) achieves the highest training accuracy but shows mild overfitting, while σ = 1.0 generalizes better.

Both Explinear and Hinge losses perform similarly, with Explinear slightly stronger on simpler data.

Overall, Linear is best for Model 1, while Gaussian (σ = 1.0) gives the best balance for complex models.

Q3-

(a) dualHinge Implementation

We implemented the traditional SVM dual form that maximizes

$$\alpha^T 1 - \frac{1}{2\lambda}\alpha^T \Delta(y) K \Delta(y)\alpha,$$

subject to

$0 \le \alpha_i \le 1$ and $\alpha^T y = 0.$

(b) dualClassify Implementation

Predictions were made using:

$$\hat{y} = \text{sign}\big((1/\lambda) K(X_{\text{test}}, X)\Delta(y)\alpha + b\big),$$

with correct kernel computation for each configuration.

(c)

Cross-Validation Accuracies

λ	Linear	Poly(d=2)	Poly(d=3)	Gauss(σ=1.0)	Gauss(σ=0.5)
0.001	0.962	0.981	0.981	0.486	0.478
0.01	0.962	0.982	0.981	0.486	0.478
0.1	0.962	0.982	0.982	0.486	0.478

Best Parameters Selected:

Best λ: 0.01

Best Kernel: Polynomial (degree 2)

Cross-Validation Accuracy: 0.982

(d)

The Polynomial kernel (d = 2) with λ = 0.01 achieved the highest cross-validation accuracy. This configuration provides an excellent balance between model complexity and generalization.