Support Vector Machines for Breast Cancer Classification

Optimization Course Project

Diego Rossi 1073945

April 17, 2025

Introduction

- Classification of breast cancer tumors as benign or malignant.
- Feature selection using linear SVM (norm 1) to reduce dimensionality.
- Implementation of non-linear SVMs (norm 1) with Gaussian, Polynomial, and Sigmoid kernels to classify tumors.
- Optimization of hyperparameters through grid search.
- K-fold Cross Validation for performance evaluation.
- Comparison based on accuracy, recall, precision, and F1-score.

Wisconsin Diagnostic Dataset

- 30 numerical features computed from digitized images of fine needle aspirates (FNA) of breast tumors.
- Features describe characteristics of cell nuclei:
  - Radius, Texture, Perimeter, Area, Smoothness, etc.
  - Each measured by: **mean**, **standard error**, and **worst** value.
- Total: 10 base features $\times$ 3 descriptors $=$ 30 features.
- **Label:**
  - `B` $=$ Benign
  - `M` $=$ Malignant

Linear SVM Models

Model 1: Linear SVM (Arbitrary Norm)

$$\min_{w,\gamma,y} \quad \nu\, e^T y + \|w\|$$

$$\text{s.t.} \quad D\,(A\,w - e\,\gamma) + y \geq e, \quad y \geq 0.$$

Model 2: Linear SVM with L1 Norm (SVM1)

$$\min_{w,\gamma,s,y} \quad \nu\, e^T y + e^T s$$

$$\text{s.t.} \quad D\,(A\,w - e\,\gamma) + y \geq e, \quad -s \leq w \leq s, \quad y \geq 0.$$

MATLAB implementation of SVM1

```
cvx_begin quiet
    cvx_solver Mosek
    variables w(n) gam s(n) y(m)

    minimize (nu*sum(y) + sum(s))

    subject to
        D * (A*w - gam*ones(m,1)) + y >= ones(m,1);
        -s <= w <= s;
        y >= 0;
cvx_end
```

Non Linear SVM Models

Model 3: Non Linear SVM

$$\min_{u,\gamma,y} \quad \nu\, e^T y + \|u\|_p$$

$$\text{s.t.} \quad D\Big(K(A, A^T)\, D\, u - e\,\gamma\Big) + y \geq e, \quad y \geq 0.$$

Model 4: Non Linear SVM with L_1 Norm

$$\min_{u,\gamma,s,y} \quad \nu\, e^T y + e^T s$$

$$\text{s.t.} \quad D\Big(K(A, A^T)\, D\, u - e\,\gamma\Big) + y \geq e, \quad -s \leq u \leq s, \quad y \geq 0.$$

Separating Surface: $K(x, A)^T Du = \gamma$

MATLAB implementation of GSVM L1 Norm

```
cvx_begin quiet
cvx_solver mosek
variables u(l_train) gam y(l_train) s(l_train)

minimize(nu * sum(y) + sum(s))

subject to
    D_train * (K_train * D_train * u - gam*ones(l_train,1)) + y >= ones(l_train,1);
    -s <= u <= s;
    y >= 0;
cvx_end


y_pred = sign(K_test * D_train * u - gam);
```

Preprocessing and Feature Selection with Linear SVM

- **Initial dataset:** 30 features per instance.
- **Label mapping:** $\mathtt{B} \to -1$, $\mathtt{M} \to +1$.
- **Normalization:** Standardized all features to ensure consistent scale.
- **Feature selection:** Performed using a Linear SVM with L1 norm (**SVM1**).
- **Selected features (9):**
  - `texture1`, `concave_point1`, `fractal_dimension1`.
  - `radius2`.
  - `radius3`,`texture3`, `smoothness3`, `concavity3`, `symmetry3`.

Kernel Implementations for Classification

- **Gaussian Kernel:**

$$K(x_i, x_j) = \exp(-\sigma||x_i - x_j||^2)$$

```
K_train = exp(-sigma * pdist2(A_train, A_train, 'euclidean').^2);
```

- **Polynomial Kernel:**

$$K(x_i, x_j) = (x_i \cdot x_j + c)^p$$

```
K_train = (A_train * A_train' + c) .^ p;
```

- **Sigmoid Kernel:**

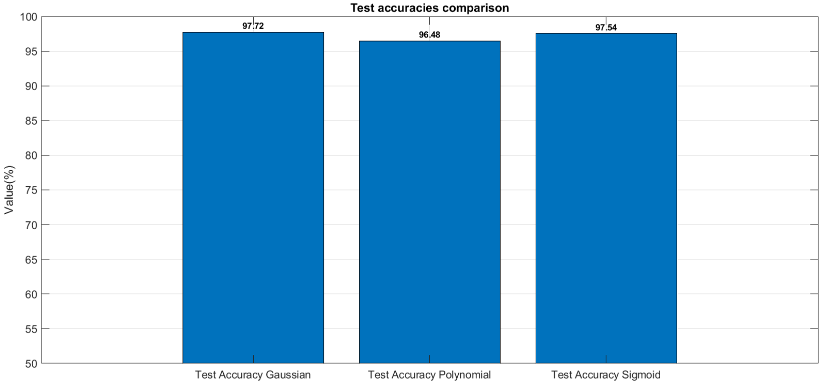
$$K(x_i, x_j) = \tanh(\gamma x_i \cdot x_j + r)$$

```
K_train = tanh(g * (A_train * A_train') + r);
```

Hyperparameter Optimization

- Grid search approach used to optimize:
  - Gaussian kernel: σ, ν
  - Polynomial kernel: c, p, ν
  - Sigmoid kernel: γ, r, ν
- Best hyperparameters selected based on highest accuracy.
- 10-fold cross-validation to evaluate model performance.

Test Accuracy Performance



Performance Metrics Comparison

Kernel	Train Acc.	Test Acc.	Precision	Recall	F1 Score
Gaussian	97.89%	97.72%	98.59%	95.39%	96.94%
Polynomial	98.59%	96.48%	97.02%	93.35%	95.06%
Sigmoid	98.14%	97.54%	96.84%	96.75%	96.71%

- **Gaussian**: Highest test accuracy and precision and F1 score.
- **Polynomial**: Lowest test accuracy and Recall.
- **Sigmoid**: Most balanced overall performance.
- All kernels achieve test accuracy above 95%.

Conclusion

- Feature selection reduced dimensionality effectively while maintaining high classification performance.
- All kernels achieved over 95% test accuracy.
- The Gaussian kernel achieved the best test accuracy.
- The Sigmoid kernel showed the most balanced trade-off between precision and recall.
- The polynomial kernel gives good test accuracy, but the recall is quite low.

GitHub Repository:

`github.com/drossi15/Optimization-Project`

Thank you for your attention!