Support Vector Machines (SVM)

Sunday Adedayo Falade (21062644)

In this session, let us talk about...

1. **Introduction**
   Who am I? What am I talking about today? anything else...

2. **Understanding SVM**
   What is SVM? What does it do? How does it work? Where can we apply SVM?

3. **Components of SVM**
   What is the idea behind it? What does the distance tell us? How can we work around problems encountered if any?

4. **SVM Implementation (Sklearn)**
   How does the code work? How can we evaluate the outcome? Can we improve it to be efficient if required?

5. **Conclusion**
   Putting it all together informatively

6. **References**
   Resources that aided my understanding of the SVM algorithm.

What is Support Vector Machines (SVM)

Support vector machines (SVM) is a supervised machine learning algorithm used to analyze data for classification and regression task and it is effective for high-dimensional datasets and complex decision boundaries.

Applications of SVM

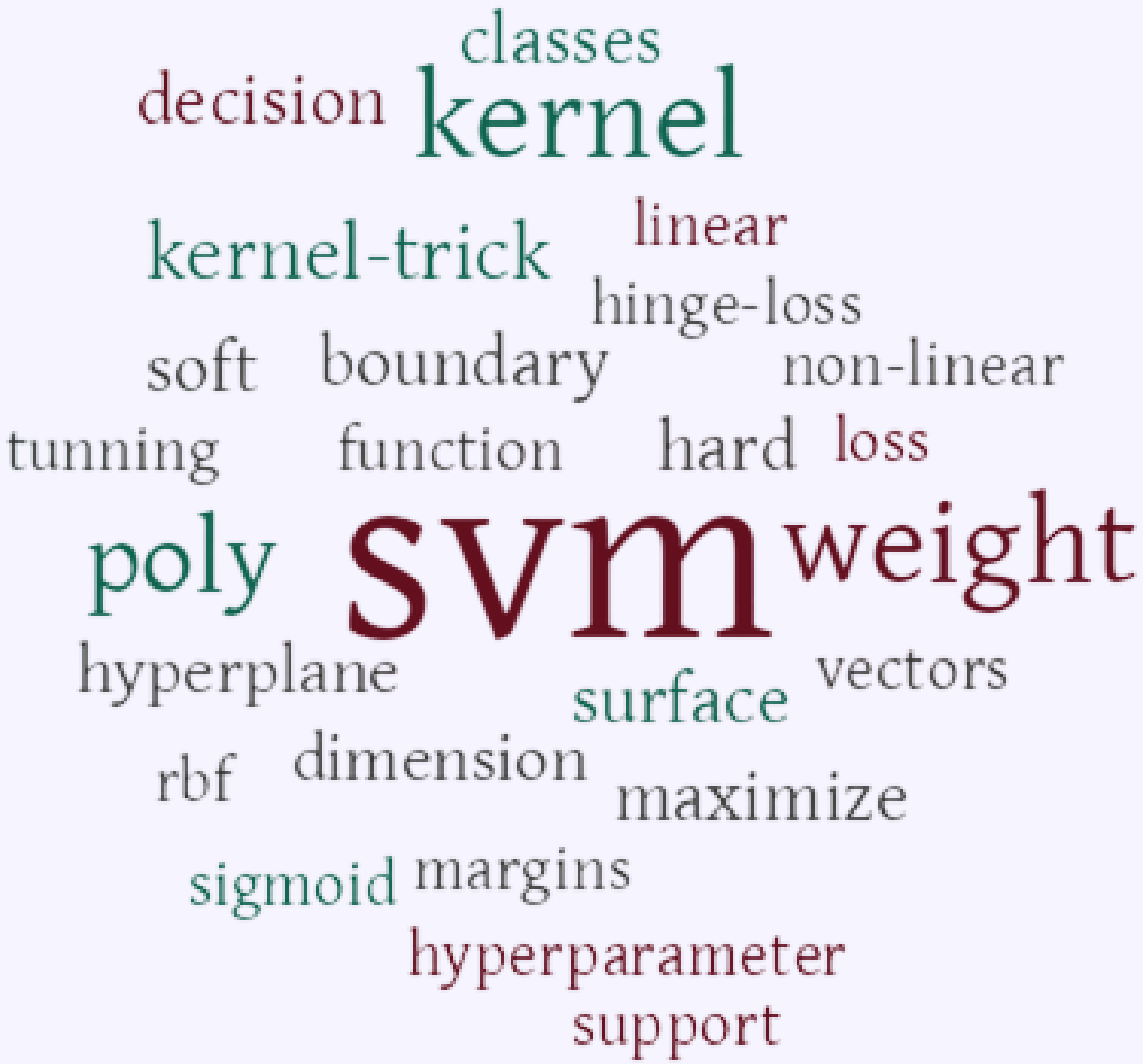
- **Face detection** – SVMc classify parts of the image as a face and non-face and create a square boundary around the face.
- **Text and hypertext categorization** – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compares with the threshold value.
- **Classification of images** – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.
- **Bioinformatics** – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems.
- **Handwriting recognition** – We use SVMs to recognize handwritten characters used widely.
- **Generalized predictive control(GPC)** – Use SVM based GPC to control chaotic dynamics with useful parameters.

SVM works by finding the optimal hyperplane that maximizes the margin between different classes.

Components of SVM

In this section, I discuss about various components associated with support vector machines used to achieve the desired result for classifying data points. They are as follows:

1. Decision boundary
2. Support vectors
3. Margins (soft and hard margins)
4. Kernels
    - Linear
    - Non-linear (e.g., Radial Basis Function (RBF), Polynomial and Sigmoid)



Word cloud generated from worditout.com

Components of SVM

Decision Boundary

is the hyperplane that separates different classes in the feature space. It is positioned in such a way that it maximizes the margin (the distance between the hyperplane and the nearest support vectors from each class). The mathematical function for the hyperplane is given as: **$f(x) = W^T X + b$**

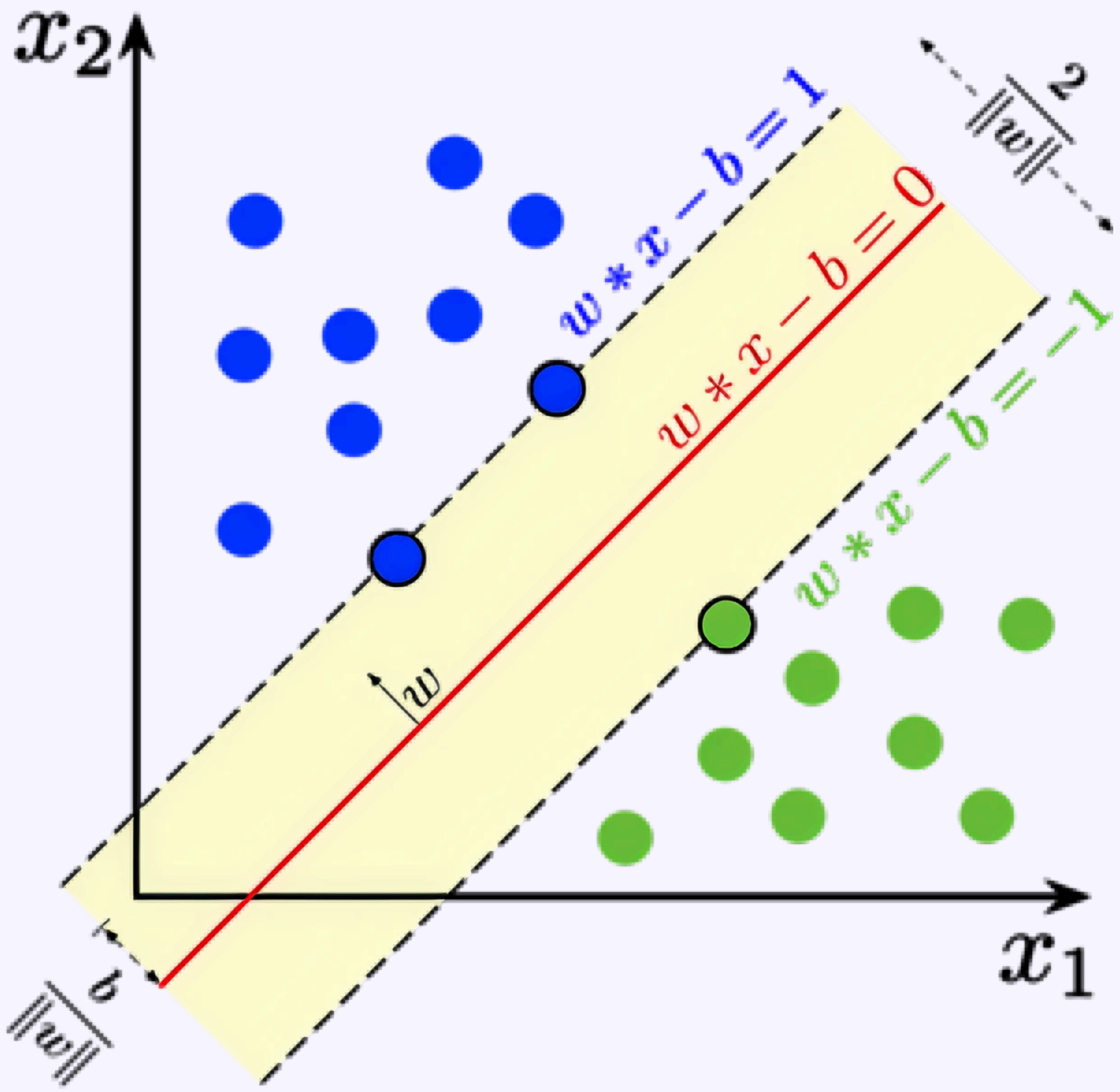
Where:

- w: Weight vector (defining the orientation of the hyperplane)
- b: Bias term (defining the position of the hyperplane)
- X: Input feature vector

The classification decision rule is based on the sign of f(x) such that if f(x) > 0, then Class 1 but if f(x) < 0, then Class -1.

Support Vectors

Data points that are nearest to the hyper-plane and affect the position and orientation of the hyper-plane. We have to select a hyperplane, for which the margin, i.e the distance between support vectors and hyper-plane is maximum.



SVM diagram from wikipedia

Components of SVM

Margins

The margin signifies the distance between the decision boundary and the closest data points from each class. Maximizing margin is essential as it provides a measure of robustness against noise and aids in better generalization to unseen data. Two approaches to margins in SVMs are Hard Margin and Soft Margin.

Hard Margin

In a hard margin SVM, the objective is to identify a hyperplane that completely separates data points belonging to different classes, ensuring a clear demarcation with the utmost margin width possible.

Soft Margin

Soft margin SVM introduces flexibility by allowing some margin violations (misclassifications) to handle cases where the data is not perfectly separable. Suitable for scenarios where the data may contain noise or outliers.

Components of SVM // Margins

Criteria	Hard Margin	Soft Margin
Objective Function	Maximize margin	Maximize margin, minimize margin violations.
Handling Noise	Sensitive, requires perfectly linearly seperable	Robust, handles noisy data with margin violations.
Regularization	Not applicable, no regularization parameter	Controlled by regularization parameter C.
Complexity	Simple, computationally efficient	May require more computational resources

Hard Margin vs Soft Margin in SVM by geeksforgeeks

Components of SVM // Kernels

Linear

The simplest form of kernel used in SVM. it is suitable when the data us linearly separable meaning that a straight line (or hyperplane in higher dimensions) can effectively separate the classes. it is the dot product of the input samples

- It is represented as **K(x,y) = x.y**
- The higher the value, the more similar the points are.
- It is used for text classification problems such as spam detection

Polynomial

Allows SVM to model complex relationships by introducing polynomial terms. it is useful when the data is not linearly separable but still follows a pattern.

- It is represented as **K(x,y) = $(x.y+c)^d$ .where d is the degree of the polynomial and C is the polynomial coefficients**
- It is used in complex problems like image recognition where relationships between features can be non-linear

Radial Basis Function

The most widely used kernel in SVM. it maps the data into infinite dimensional space making it highly effective for complex classification problems. It measures similarity between two data points in infinite dimensions and then approaches classification by majority vote.

- It is used when data has a smooth, continous distribution and requires a flexible boundary

$$K(\mathbf{x}_1, \mathbf{x}_2) \quad \exp\left(\; \gamma \cdot \|\mathbf{x}_1 - \mathbf{x}_2\|^2 \right)$$

Sigmoid

Inspired by neural network and behaves similarly to the activation function of a neuron. it is based on the hyperbolic tangent function and is suitable for neural networks and other non-linear classifiers.

- It is often used in neural networks and non-linear classifiers.
- It is represented by the formula:

$$K(\mathbf{x}_1, \mathbf{x}_2) \quad \tanh(\gamma \cdot \mathbf{x}_1^\top \mathbf{x}_2 + r)$$

where the kernel coefficient γ (gamma) controls the influence of each individual training sample on the decision boundary and r is the bias term (coef0) that shifts the data up or down.

Hinge Loss Function

Hinge Loss is a loss function utilized within machine learning to train classifiers that optimize to increase the margin between data points and the decision boundary. Hence, it is mainly used for maximum margin classifications. To ensure the maximum margin between the data points and boundaries, hinge loss penalizes predictions from the machine learning model that are wrongly classified, which are predictions that fall on the wrong side of the margin boundary and also predictions that are correctly classified but are within close proximity to the decision boundary.

This characteristic of the Hinge Loss function ensures machine learning models are able to predict the accurate classification of data points to their target value with confidence that exceeds the threshold of the decision boundary. This approach to machine algorithm learning enhances the model's generalization capabilities, making it effective for accurately classifying data points with a high degree of certainty.

The mathematical equation for Hinge Loss is:

$L(y, f(x)) = \max(0, 1 - y * f(x))$

where:

- L represents the Hinge Loss
- y is the true label or target value (-1 or 1)
- f(x) is the predicted value or decision function output

SVM Implementation (Sklearn)

In this section, I discuss the implementation of the SVM classifier using python scikit-learn library and how necessary steps taken to optimize parameter for better result.

Github link to view project: https://github.com/faadeola/support-vector-machines

```python
# Preprocess data for machine learning
preprocessor = ColumnTransformer([
    ('one_hot',OneHotEncoder(handle_unknown='ignore'),cat_columns),
    ('scaler',StandardScaler(),num_columns)],
    remainder = 'passthrough'
)
```

```python
# Create pipleine for data preprocessing and model selection
pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('pca', PCA()),
    ('model',SVC())
])


# Create parameter grid for the gridsearchcv
param_grid = {
    'model__kernel':['linear','rbf', 'poly'],
    'model__class_weight':[None,'balanced'],
    'model__C': [5, 10],
    'pca__n_components': [5,10,15]
}
```

```python
# Set up model to use GrdisearchCV
svc_model = GridSearchCV(pipeline, param_grid, cv=5)
```

Summary

Support Vector Machines (SVM) is widely used due to its ability to handle high-dimensional data, robustness to outliers, and effectiveness in small datasets. However, it can be computationally expensive for large datasets and requires proper tuning of hyperparameters. The introduction of regularization parameter in the case of the soft margin allows for flexibility in the way SVM performs classification task.

References

- GeeksforGeeks. (n.d.). Major kernel functions in support vector machine (SVM). Retrieved March 09, 2025, from https://www.geeksforgeeks.org/major-kernel-functions-in-support-vector-machine-svm/
- DataCamp. (n.d.). Loss function in machine learning: An overview. Retrieved March 12, 2025, from https://www.datacamp.com/tutorial/loss-function-in-machine-learning
- Wikipedia contributors. (n.d.). Support vector machine – Applications. Wikipedia, The Free Encyclopedia. Retrieved March 12, 2025, from https://en.wikipedia.org/wiki/Support_vector_machine#Applications
- FreeCodeCamp. (n.d.). SVM kernels: How to tackle nonlinear data in machine learning. Retrieved March 13, 2025, from https://www.freecodecamp.org/news/svm-kernels-how-to-tackle-nonlinear-data-in-machine-learning/
- Scikit-learn developers. (n.d.). SVM kernels example. Scikit-learn. Retrieved March 13, 2025, from https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html#sphx-glr-auto-examples-svm-plot-svm-kernels-py
- Analytics Vidhya. (2021, March). Beginner's guide to support vector machine (SVM). Retrieved March 16, 2025, from https://www.analyticsvidhya.com/blog/2021/03/beginners-guide-to-support-vector-machine-svm/

Thank You