Support Vector Machines (SVM) Notes

Based on Professor Galletti's Medium Article

Core Concepts of Support Vector Machines (SVM)

- **Objective**: Identify the widest possible margin (or "street") that separates two classes in a dataset.
- **Decision Boundary**: A linear separator defined by:

$$w^T x + b = 0$$

- **Classification Rule**:
  - If $w^T x + b \geq 0$, classify as $+1$
  - If $w^T x + b < 0$, classify as -1
- **Margin Width**: Inversely proportional to $\|w\|$; maximizing the margin equates to minimizing $\|w\|$

Adjusting the Margin

- Multiplying both w and b by a positive constant c affects the margin width:
  - $0 < c < 1$: Margin expands
  - $c > 1$: Margin contracts
- **Support Vectors**: Points that lie exactly on the margin boundaries and are critical in defining the optimal hyperplane.

SVM via Perceptron Algorithm

Initialization

- Start with random w and b

Hyperparameters

- Learning rate lr
- Expanding rate < 1
- Retracting rate > 1

Training Loop

1. For each epoch:
   (a) Pick random sample (x, y)
   (b) Compute prediction: $y_{\text{pred}} = w^T x + b$
   (c) Update based on:
      - Correctly classified and within margin: retract
      - Correctly classified and outside margin: expand
      - Misclassified: update w and b to correct

Dual Formulation (Nonlinear Case)

- Introduce Lagrange multipliers α_i
- Prediction function becomes:
$$y_{\text{pred}} = \sum_j \alpha_j (x_j^T x) + b$$
- Training Loop (similar to above but update α_i instead of w)

Kernel Trick

- **Purpose**: To handle non-linearly separable data
- Replace dot product $x_j^T x$ with kernel $K(x_j, x)$
- **Common Kernels**: s
- Updated prediction function:
$$y_{\text{pred}} = \sum_j \alpha_j K(x_j, x) + b$$

Summary

- SVMs aim to find a maximum margin hyperplane separating two classes
- Perceptron algorithm can be adapted to enforce margin constraints
- Dual formulation allows for kernel-based non-linear classification

Reference: Professor Galletti's Medium article