CHEME 5820: Machine Learning for Engineers **Spring 2025**

Lecture 4c: Support Vector Machines (SVMs)

Lecturer: Jeffrey Varner

1 Introduction

In this lecture, we will discuss Support Vector Machines (SVMs). SVMs are a powerful class of supervised learning algorithms that can be used for classification and regression tasks. SVMs are based on the concept of decision planes that define decision boundaries. A decision plane is a hyperplane that separates the data into classes. The goal of SVMs is to find the optimal decision plane that maximizes the margin between the classes. The margin is the distance between the decision plane and the closest data points from each class. SVMs are particularly useful for high-dimensional data and can handle non-linear decision boundaries using the kernel trick. In this lecture, we will discuss the basic concepts of SVMs, including the underlying optimization problem, the kernel trick, and the soft margin SVM.

2 Basic Concepts of SVMs

Support Vector Machines (SVMs) are a class of supervised learning algorithms that can be used for classification and regression tasks. The goal of SVMs is to find the optimal decision plane that maximizes the margin between the classes. The decision plane is defined by the equation:

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{1}$$

where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the data point, and b is the bias term. To estimate the decision plane, SVMs use the training data to find the optimal weight vector $\mathbf{w}$ and bias term b that separate the data into classes. These parameters are estimated by solving an optimization problem that minimizes the norm of the weight vector subject to the constraints that the data points are correctly classified. The optimization problem for SVMs can be formulated as:

$$\min_{\mathbf{w},b} \frac{1}{2}\left|\left|\mathbf{w}\right|\right|^2 \tag{2}$$

subject to the constraints:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \ldots, n \tag{3}$$

where $\mathbf{w}$ is the weight vector, b is the bias term, $\mathbf{x}_i$ is the i-th data point, y_i is the label of the i-th data point, and n is the number of data points.

2.1 Soft Margin SVM

The soft margin SVM is an extension of the basic SVM that allows for some misclassification errors. The soft margin SVM introduces a slack variable ξ_i for each data point, which measures the distance of the data point from the decision plane. The optimization problem for the soft margin SVM can be formulated as

follows:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i \tag{4}$$

subject to the constraints:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, n \tag{5}$$

3 Kernel Functions

Kernel functions in machine learning are mathematical tools that enable algorithms to operate in high-dimensional spaces without explicitly computing the coordinates in those spaces. Kernel functions are used in a variety of machine learning algorithms, including Support Vector Machines (SVMs), kernelized regression, and kernelized clustering. Kernel functions have a few different interpretations. For example, kernel functions can be thought of as similarity measures, i.e., they quantify the similarity between pairs of data points in a high-dimensional space. They are also implict mappings of data into a high-dimensional space, where the data becomes linearly separable (which is useful for classification algorithms). Thus, kernel functions are a powerful tool that we are going to use for many applications in machine learning.

Kernel functions are useful, but what are they? There are several types of kernel functions, but the most common ones are the linear kernel, polynomial kernel, and radial basis function (RBF) kernel. Suppose we have a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, 2, \ldots, n\}$, where the features $\mathbf{x}_i \in \mathbb{R}^m$ are m-dimensional vectors and the labels are binary $y_i \in \{-1, 1\}$. A kernel function $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a function that takes fearure vectors points $\mathbf{x}_i$ and $\mathbf{x}_j$ and computes a scalar value that represents the similarity between the two data points. For example, the linear kernel function computes the dot product between the two data points, which is a measure of their similarity:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j \tag{6}$$

On the other hand, a polynomial kernel is defined as:

$$k_d(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^d \tag{7}$$

where d is the degree of the polynomial. The radial basis function (RBF) kernel is defined as:

$$k_\gamma(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \, ||\mathbf{x}_i - \mathbf{x}_j||_2^2) \tag{8}$$

where γ is a scaling factor, and $||\cdot||_2^2$ is the squared Euclidean norm. If we define γ as $\frac{1}{2\sigma^2}$, the RBF kernel looks like a Gaussian function, without the normalization constant. Of course, not all functions can be used as kernel functions (Defn. 1).

Definition 1. *(Kernel Function) A function $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a valid kernel function if and only if the Gram matrix $\mathbf{K}$ is positive semidefinite for all possible choices of the data points $\mathbf{x}_i$, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. This is equivalent to saying that all eigenvalues of the Gram matrix K are non-negative. Further, for all real value vectors $\mathbf{x}$, the Gram matrix $\mathbf{K}$ must satisfy $\mathbf{x}^\top \mathbf{K} \mathbf{x} \geq 0$.*

Given a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, the decision function of the SVM can be written as:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{9}$$

where α_i are the Lagrange multipliers, y_i are the labels of the data points, and b is the bias term.

4 Summary and Conclusions

In this lecture, we discussed the basic concepts of Support Vector Machines (SVMs). SVMs are a class of supervised learning algorithms that can be used for classification and regression tasks.

References