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

# Lecture 5c: Support Vector Machines (SVMs)

*Lecturer: Jeffrey Varner*

**Disclaimer**: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

The key concepts covered in this lecture include:

- Support Vector Machines (SVMs)

# 1 Introduction

In the previous lectures, we've discussed several binary classification algorithms, including the perceptron, logistic regression, K-nearest neighbors, and the concept of Kernel functions and the kerner trick. In this lecture, we will conclude our dicussion (for now) of classification algorithms by introducing support vector machines (SVMs), a powerful and versatile machine learning algorithm that can be used for both classification and regression tasks. SVMs are particularly well-suited for binary classification problems, where the goal is to separate data points into two classes using a hyperplane, e.g., like the perceptron. However, SVMs are based on the concept of finding an *optimal seperating hyperplane* (not just some seperating hyperplane) that maximizes the margin between the two classes.

# 2 Support Vector Machines (SVMs)

Support vector machines (SVMs) are a class of supervised learning algorithms that can be used for both classification and regression tasks. In many ways, SVMs are similar to the perceptron and logistic regression, as they are also based on the concept (in the simplest case) of finding a hyperplane that separates data points into different classes. Suppose, we have dataset $\mathcal{D} = \{(\hat{\mathbf{x}}_i, y_i) \mid i = 1, 2, \dots, n\}$, where $\hat{\mathbf{x}}_i \in \mathbb{R}^p$ is an *augmented* feature vector ($m$ features with additional 1 last entry to model the bias) and $y_i \in \{-1, 1\}$ is the corresponding class label. Then, the goal of SVMs is to find the hyperplane $\mathcal{H} = \{\hat{\mathbf{x}} \mid \langle\hat{\mathbf{x}}, \theta\rangle = 0\}$ that separates the data points into two classes (those points above the hyperplane, and those points below the hyperplane), where $\theta \in \mathbb{R}^p$ ($p = m + 1$) is the normal vector to the hyperplane, or alternatively, the parameters of the model that we need to estimate. So far, this is similar to the perceptron and logistic regression, but the key difference is that SVMs aim to find the *optimal* hyperplane in some sense. Let's explore the notion of an *optimal hyperplane* in more detail.

## 2.1 Maximum Margin Classifier

Suppose we have found a hyperplane $\mathcal{H}$ that separates the data points into two classes. Then, the margin of the hyperplane is defined as the distance $\gamma$ between the hyperplane and the closest data point from either class. Thus, the margin is a measure of how well the hyperplane separates the two classes, and the goal of a maximizing margin SVM classifier is to find the hyperplane that maximizes the margin. Let's develop a model for the margin of the seperating hyperplane.

Consider some feature vector $\hat{\mathbf{x}} \in \mathbb{R}^m$. Let $\mathbf{d}$ denote the vector from the hyperplane $\mathcal{H}$ to the feature vector $\hat{\mathbf{x}}$. Finally, let the point $\mathbf{p}$ be the projection of $\hat{\mathbf{x}}$ onto the hyperplane $\mathcal{H}$. Because the vector $\mathbf{d}$ is orthogonal to the hyperplane $\mathcal{H}$, we can write $\mathbf{d} = \hat{\mathbf{x}} - \mathbf{p}$. Futher, the vector $\mathbf{d}$ can be written as some scalar multiple of the normal vector $\theta$, i.e., $\mathbf{d} = \lambda\theta$. Then we can find the value of $\lambda$ by taking the dot product of $\mathbf{d}$ with the normal vector $\theta$:

$$\begin{aligned}
\hat{\mathbf{p}} &= \hat{\mathbf{x}} - \mathbf{d} \quad | \text{ take dot product with } \theta \\
\langle \hat{\mathbf{p}}, \theta \rangle &= \langle \hat{\mathbf{x}}, \theta \rangle - \langle \mathbf{d}, \theta \rangle = 0 \quad | \text{ substitute } \mathbf{d} = \lambda\theta \\
\langle \hat{\mathbf{p}}, \theta \rangle &= \langle \hat{\mathbf{x}}, \theta \rangle - \lambda \langle \theta, \theta \rangle = 0 \quad | \text{ solve for } \lambda \\
\lambda &= \frac{\langle \hat{\mathbf{x}}, \theta \rangle}{\langle \theta, \theta \rangle}
\end{aligned}$$

We can now find the length of the vector $\mathbf{d}$ by computing $||\mathbf{d}||_2$:

$$\begin{aligned}
||\mathbf{d}||_2 &= \sqrt{\mathbf{d}^\top \mathbf{d}} \quad | \text{ substitute } \mathbf{d} = \lambda\theta \\
&= \sqrt{\lambda^2 \theta^\top \theta} \quad | \text{ substitute } \lambda = \frac{\langle \hat{\mathbf{x}}, \theta \rangle}{\langle \theta, \theta \rangle} \\
&= \sqrt{\frac{\langle \hat{\mathbf{x}}, \theta \rangle^2}{\langle \theta, \theta \rangle^2} \theta^\top \theta} \quad | \text{ substitute } \langle \theta, \theta \rangle = \theta^\top \theta \text{ and simplify} \\
&= \frac{\langle \hat{\mathbf{x}}, \theta \rangle}{||\theta||_2} \quad | \text{ where } ||\theta||_2 = \sqrt{\langle \theta, \theta \rangle}
\end{aligned}$$

The length of the distance vector $\mathbf{d}$ gives us the distance from a feature vector $\hat{\mathbf{x}}$ to the hyperplane $\mathcal{H}$. Thus, we can define the margin $\gamma_\theta$ of the hyperplane $\mathcal{H}$ as the distance between the hyperplane and the closest data point (Defn. 1).

**Definition 1.** *The margin $\gamma_\theta$ of a hyperplane $\mathcal{H}$ is given by the distance between the hyperplane and the closest data point:*

$$\gamma_\theta = \min_i \left\{ \frac{|\langle \hat{\mathbf{x}}_i, \theta \rangle|}{||\theta||_2} \right\}$$

*where we use the absolute value to account for the fact that the distance can be positive or negative, i.e., the data point can be on either side of the hyperplane. The margin and the hyperplane are scale invaraient, i.e., $\gamma_\theta = \gamma_{c\theta}$ for any $c \neq 0$.*

Now that have a model for the margin of the hyperplane, we can define the problem of finding the optimal hyperplane as an optimization problem. Ideally, we would like to estimate the parameters $\theta$ of the hyperplane that maximize the margin $\gamma_\theta$, i.e., the distance between the hyperplane and the closest data point is maxzimized:

$$\max_\theta \gamma_\theta \quad \text{subject to} \quad y_i \langle \hat{\mathbf{x}}_i, \theta \rangle \geq 0 \quad \forall i$$

We can substitute the definition of the margin $\gamma_\theta$ into the optimization problem, which gives as a new objective function:

$$\max_\theta \left[ \min_i \left\{ \frac{|\langle \hat{\mathbf{x}}_i, \theta \rangle|}{||\theta||_2} \right\} \right] \quad \text{subject to} \quad y_i \langle \hat{\mathbf{x}}_i, \theta \rangle \geq 0 \quad \forall i$$

However, we can factor out the norm of the normal vector $\theta$ from the objective function, which simplifies the optimization problem:

$$\max_\theta \frac{1}{||\theta||_2} \left[ \min_i \{ |\langle \hat{\mathbf{x}}_i, \theta \rangle| \} \right] \quad \text{subject to} \quad y_i \langle \hat{\mathbf{x}}_i, \theta \rangle \geq 0 \quad \forall i$$

Finally, we can use the scale invariance of the margin and the hyperplane to simplify the optimization problem further:

$$\max_{\theta} \frac{1}{||\theta||_2} \quad \text{subject to} \quad y_i \left\langle \hat{\mathbf{x}}_i, \theta \right\rangle \geq 0 \quad \forall i$$

where we have used the fact that the margin is scale invariant, i.e., we can always find a parameter vector $\theta$ such that $\min_i \{ | \left\langle \hat{\mathbf{x}}_i, \theta \right\rangle | \} = 1$. Putting all these ideas together (and using the fact that maximing the inverse of a function is equivalent to minimizing the function), we can define the optimization problem for the maximum margin classifier problem:

**Definition 2.** *The maximum margin classifier problem is defined as the optimization problem:*

$$\begin{aligned} \textit{find} \qquad & \min_{\theta} ||\theta||_2^2 \\ \textit{subject to} \qquad & y_i \left\langle \hat{\mathbf{x}}_i, \theta \right\rangle \geq 0 \quad \forall i \\ \textit{subject to} \qquad & \min_{i} \{ | \left\langle \hat{\mathbf{x}}_i, \theta \right\rangle | \} = 1 \end{aligned}$$

*where the objective function is the squared norm of the normal vector $\theta$, and the constraints ensure that the data points are correctly classified by the hyperplane, and the margin of the hyperplane is equal to 1.*

## 3 Summary and Conclusions

In this lecture, we introduced support vector machines (SVMs), a powerful and versatile machine learning algorithm that can be used for both classification and regression tasks. SVMs are particularly well-suited for binary classification problems, where the goal is to separate data points into two classes using an optimal hyperplane.

## References