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

Lecture 4c: Kernel Functions and Kernelized Regression

Lecturer: Jeffrey Varner

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

In this lecture, we will discuss the following topics:

- **Positive definite kernel functions**: A positive definite kernel function $k : \mathbb{R}^\star \times \mathbb{R}^\star \to \mathbb{R}$ is a function that takes two vector arguments and returns a scalar that is in some sense a *similarity* measure of the two input vectors. A positive definite kernel function produces a kernel matrix $\mathbf{K}$ that is positive (semi)definite.
- **Kernel machines**: A kernel machine is a class of machine learning algorithms that uses kernel functions to implicitly transform input data into a high-dimensional feature space, enabling the solution of non-linear problems using linear classifiers without explicitly computing the coordinates in that space.
- **Kernel regression**: Kernel regression is a technique that uses kernel functions to estimate (potentially) non-linear relationships between variables by assigning weights to data points based on their proximity to a point of interest, allowing for flexible modeling without assuming a specific functional form.

1 Introduction

In this lecture, we will discuss kernel functions and kernel regression. Kernel functions 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 various machine learning algorithms, including Support Vector Machines (SVMs), kernelized regression, and kernelized clustering. Today, we will focus on kernel regression, a non-parametric regression technique that uses kernel functions to estimate the relationship between the input and output variables. Let's start by discussing the basic concepts of kernel functions and then consider kernel regression.

2 Kernel Functions

Kernel functions are mathematical tools in machine learning that enable algorithms to operate in high-dimensional spaces without explicitly computing the coordinates of those spaces. Kernel functions are used in various 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 considered similarity measures, quantifying the similarity between pairs of data points in a high-dimensional space. They also implicitly map data into a high-dimensional space, where the data becomes linearly separable (which is helpful for classification algorithms). Thus, kernel functions are powerful tools that we will use for many machine learning applications.

A kernel function $k : \mathbb{R}^\star \times \mathbb{R}^\star \to \mathbb{R}$ is a function that takes a pair of vectors $\mathbf{v}_i \in \mathbb{R}^\star$ and $\mathbf{v}_j \in \mathbb{R}^\star$ as arguments, e.g., a pair of feature vectors, a feature vector and a parameter vector, or any two vectors of compatible size and computes a scalar value that represents the similarity (in some sense) between the two

vector arguments. For example, the linear kernel function computes the dot product between two vectors:

$$k(\mathbf{v}_i, \mathbf{v}_j) = \mathbf{v}_i^\top \mathbf{v}_j \tag{1}$$

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

$$k_d(\mathbf{v}_i, \mathbf{v}_j) = (1 + \mathbf{v}_i^\top \mathbf{v}_j)^d \tag{2}$$

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

$$k_\gamma(\mathbf{v}_i, \mathbf{v}_j) = \exp(-\gamma \, ||\mathbf{v}_i - \mathbf{v}_j||_2^2) \tag{3}$$

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

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

Kernel functions can also be combined to create more complex kernel functions using the concept of kernel composition. For example, the sum of two valid kernel functions is also a valid kernel function. The product of two valid kernel functions is also a valid kernel function. Multiplying a kernel function by a scalar is also a valid kernel function, etc. See the CS 4780 Lecture Notes (Fall 2018) for more details on kernel composition.

3 Kernel Regression

Kernel regression is a non-parametric regression technique that uses kernel functions to estimate the relationship between the input and output variables. Kernel regression is a powerful tool for modeling complex relationships in data and is widely used in machine learning applications. Suppose we have a 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 target variables are continuous values $y_i \in \mathbb{R}$, e.g., the price of a house, the temperature, etc. The basic idea behind kernel regression is to estimate the output variable y as a weighted average of the output variables of the training data points, where the kernel functions determine the weights. The kernel regression function is defined as:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{4}$$

where α_i are the weights, and $k(\mathbf{x}_i, \mathbf{x})$ is the kernel function that measures the similarity between the input features $\mathbf{x}_i$ and $\mathbf{x}$.

3.1 Kernel Ridge Regression

Suppose we consider a linear regression problem of the form:

$$\hat{\mathbf{y}} = \hat{\mathbf{X}} \theta \tag{5}$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{n \times p}$ is a data matrix with the transpose of the augmented feature vectors $\hat{\mathbf{x}}^\top \in \mathbb{R}^p$ on the rows, and θ is an unknown parameter vector $\theta \in \mathbb{R}^p$ where $p = m + 1$. We can estimate the (expected value) of the parameter vector θ by minimizing the least squares loss function:

$$\hat{\theta} = \arg\min_{\theta} \left|\left|\mathbf{y} - \hat{\mathbf{X}}\theta\right|\right|_2^2 + \lambda \, ||\theta||_2^2 \tag{6}$$

where $\mathbf{y}$ is the target variable vector, and $\lambda \geq 0$ is a regularization parameter. When we include the L2 penalty term, this is referred to as ridge regression, but we'll refer to it as regularized least squares regression. If we include a different penalty, we would have a different type of regression, e.g., L1 penalty would be Lasso regression. The (regularized) least squares solution for the (expected value) of the parameters θ is given by:

$$\hat{\theta}_\lambda = \left(\hat{\mathbf{X}}^\top \hat{\mathbf{X}} + \lambda \, \mathbf{I}\right)^{-1} \hat{\mathbf{X}}^\top \mathbf{y} \tag{7}$$

where λ is the regularization parameter, and $\mathbf{I}$ is the identity matrix.

The basic idea of kernel regression is to rewrite the parameter vector $\hat{\theta}_\lambda$ as a weighted sum of the augmented feature variables:

$$\hat{\theta}_\lambda \equiv \sum_{i=1}^{n} \alpha_i \hat{\mathbf{x}}_i \tag{8}$$

where α_i are the weights (that we need to estimate), and $\hat{\mathbf{x}}_i$ are the augmented feature vectors. Then for some (new) feature vector $\hat{\mathbf{z}}$, i.e., a vector not in the training set, the predicted output $\hat{y}$ is given by:

$$\begin{aligned} \hat{y} = \hat{\mathbf{z}}^\top \theta &= \sum_{i=1}^{n} \alpha_i \left\langle \hat{\mathbf{z}}, \mathbf{x}_i \right\rangle \quad \mid \text{Replace inner product with kernel} \\ &= \hat{\mathbf{z}}^\top \theta \simeq \sum_{i=1}^{n} \alpha_i \, k(\hat{\mathbf{z}}, \mathbf{x}_i) \end{aligned}$$

where $k(\hat{\mathbf{z}}, \mathbf{x}_i)$ denotes a kernel function (similarity score) between a new (augmented) feature vector and $\hat{\mathbf{z}}$ and the (known) training feature vector $\hat{\mathbf{x}}_i$. The question is how to estimate the weights α_i.

As it turns out the weights α_i have an analytical solution. However, before we go through this derivation, we need to introduce a slightly different form for the original θ_λ solution (Lemma 1).

Lemma 1. *First, for any data matrix* $\mathbf{X}$*, output vector* $\mathbf{y}$ *and regularization parameter* $\lambda \geq 0$*, we can show:*

$$\hat{\mathbf{X}}^\top \left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top + \lambda \, \mathbf{I}\right)^{-1} \mathbf{y} = \left(\hat{\mathbf{X}}^\top \hat{\mathbf{X}} + \lambda \, \mathbf{I}\right)^{-1} \hat{\mathbf{X}}^\top \mathbf{y}$$

Thus, we can rewrite the regularized least squares solution for the (expected value) of the parameters θ *as:*

$$\hat{\theta}_\lambda = \hat{\mathbf{X}}^\top \left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top + \lambda \, \mathbf{I}\right)^{-1} \mathbf{y}$$

The proof of this lemma can be found in the ECE595/STAT598 Course Notes, Prof. S Chan, Purdue University.

Starting from Eqn 7 and using Lemma 1, we equate the two expressions for $\hat{\theta}_\lambda$:

$$\begin{aligned}
\left(\hat{\mathbf{X}}^\top\hat{\mathbf{X}} + \lambda\,\mathbf{I}\right)^{-1}\hat{\mathbf{X}}^\top\mathbf{y} &= \sum_{i=1}^{n}\alpha_i\hat{\mathbf{x}}_i \quad \text{| rewrite right hand side in vector form} \\
\hat{\mathbf{X}}^\top\left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top + \lambda\,\mathbf{I}\right)^{-1}\mathbf{y} &= \hat{\mathbf{X}}^\top\alpha \quad \text{| multiply by } \hat{\mathbf{X}} \\
\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top + \lambda\,\mathbf{I}\right)^{-1}\mathbf{y} &= \hat{\mathbf{X}}\hat{\mathbf{X}}^\top\alpha \quad \text{| substitute } \mathbf{K}' = \hat{\mathbf{X}}\hat{\mathbf{X}}^\top \\
\mathbf{K}'\left(\mathbf{K}' + \lambda\,\mathbf{I}\right)^{-1}\mathbf{y} &= \mathbf{K}'\alpha \quad \text{| multiply by the inverse of } \mathbf{K}' \\
\left(\mathbf{K}' + \lambda\,\mathbf{I}\right)^{-1}\mathbf{y} &= \alpha
\end{aligned}$$

For an inner product kernel, the matrix $\mathbf{K}'$ is the Gram matrix $\mathbf{K}$ with elements $K_{ij} = \hat{\mathbf{x}}_i^\top\hat{\mathbf{x}}_j$.

4 Summary and Conclusions

In this lecture, we discussed kernel functions and kernel regression. Kernel functions 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 various machine learning algorithms, including Support Vector Machines (SVMs), kernelized regression, and kernelized clustering. They have several interpretations, including similarity measures and implicit data mappings into high-dimensional spaces. We introduced the linear kernel, polynomial kernel, and radial basis function (RBF) kernel as examples of kernel functions, and discussed the properties of valid kernel functions. Finally, we discussed kernel regression, a non-parametric regression technique that uses kernel functions to estimate the relationship between the input and output variables. Kernel regression is a powerful tool for modeling complex relationships in data and is widely used in machine learning applications.