Project 1

FYS-STK4155

Håvard Skåli, Erik Røset & Oskar Idland
University of Oslo, Department of Physics
(Dated: September 30, 2024)

https://github.com/Oskar-Idland/FYS-STK4155-Projects

I. INTRODUCTION

In this work, we explore regression analysis and resampling methods in the context of machine learning, focusing on both theoretical and practical aspects. The primary goal is to develop a solid understanding of various regression techniques, including Ordinary Least Squares (OLS), Ridge, and Lasso regression, and their application to both synthetic and real-world data. We will also investigate how resampling methods such as bootstrap and cross-validation can help assess model performance.

We begin by applying the abovementioned methods to the Franke function, a widely used test function in numerical analysis, and extend the analysis to real topographic data specify what type of data. Our approach involves fitting polynomial regression models of varying complexity to the Franke function and studying the bias-variance tradeoff, an essential concept in machine learning. This allows us to evaluate the impact of model complexity, noise, and the size of training data on the accuracy and generalizability of the models. The overarching aim is to gain insights into how different regression methods handle overfitting, model complexity, and data variability, and to develop a framework for critically evaluating model performance using statistical and resampling techniques. remove / rewrite?

In section II we present relevant background theory, including central concepts such as model bias, model variance and the bias-variance tradeoff, as well as the Franke function. The most important expressions introduced here are derived in appendix A. Our methodology is explained in section III, specifically regression analysis, where we focus on OLS, Ridge and Lasso regression. We also present the advantages and disadvantages of two crucial resampling methods; bootstrapping and cross-validation, both of which will be used in this work. In this section we also present our dataset and specify how we implement the methods, while our code is placed in appendix C maybe not?. The results of our analyses are presented in section IV, and in section V we discuss our findings in light of what we would expect from the regression variants and resampling methods implemented edit after discussion is written. Lastly, in section VI we summarize and conclude the main findings of our work mention reflection?.

II. THEORY

A. Bayesian Statistics

maybe remove section if not relevant. find sources
Bayesian statistics is a branch of statistics that differs from traditional frequentist approaches (like maximum likelihood estimation) by incorporating prior beliefs or knowledge into the analysis. In the Bayesian framework, we update our beliefs about model parameters based on observed data, leading to a more flexible approach to uncertainty quantification. The essence of Bayesian inference is captured by Bayes' theorem, which relates prior beliefs, likelihood, and observed data to produce an updated (posterior) belief about a parameter or model. Bayes' theorem is written as:

$$P(\boldsymbol{\beta}|\mathbf{y},\mathbf{X}) = \frac{P(\mathbf{y}|\mathbf{X},\boldsymbol{\beta})\cdot P(\boldsymbol{\beta})}{P(\mathbf{y})}. \tag{1}$$

Here $P(\boldsymbol{\beta}|\mathbf{y},\mathbf{X})$ is the posterior distribution, representing our updated belief about the model parameters $\boldsymbol{\beta}$, given the data $\mathbf{y}$ and design matrix $\mathbf{X}$. Furthermore, $P(\mathbf{y}|\mathbf{X},\boldsymbol{\beta})$ is the likelihood function, representing the probability of the data given the model and its parameters. This is the same likelihood function used in maximum likelihood estimation. The prior distribution $P(\boldsymbol{\beta})$ represents our prior beliefs about the parameters $\boldsymbol{\beta}$ before seeing the data, and lastly $P(\mathbf{y})$ is the marginal likelihood, a normalizing constant that ensures the posterior is a valid probability distribution.

In Bayesian regression, we treat the regression coefficients $\boldsymbol{\beta}$ as random variables with a prior distribution. After observing the data, we update our belief about $\boldsymbol{\beta}$ to obtain a posterior distribution. This provides not only a point estimate for $\boldsymbol{\beta}$, but also an uncertainty estimate (posterior variance) for each parameter. For example, in Ridge regression we effectively place a Gaussian prior on the parameters $\boldsymbol{\beta}$, centered around zero, with variance controlled by the hyperparameter λ. This discourages large parameter values (regularization), and leads to a tradeoff between fitting the data well and keeping the model coefficients small.

B. Properties of Predictive Models

maybe change title of section

explain what y, ytilde etc. represent. see week 38

explain cost function, MSE, score function

include analytic part of week 37

1. Predicted Values

In the context of predictive modeling, $\tilde{\mathbf{y}}$ represents the model's predicted values of the target variable $\mathbf{y}$. These predictions are made based on the features and parameters learned from the training data. For new or unseen data, the true target values $\mathbf{y}$ are often unknown, but the model generates an estimate $\tilde{\mathbf{y}}$ that approximates these values.

The performance of the model's predictions can be analyzed by decomposing the error into several components. The error of $\tilde{\mathbf{y}}$ stems from three main sources: noise variance σ^2, model bias and model variance. The latter two are expressed as

$$\text{Bias}\left[\tilde{y}\right] = \mathbb{E}\left[\left(\mathbf{y} - \mathbb{E}\left[\tilde{\mathbf{y}}\right]\right)^2\right], \tag{2}$$

$$\text{Var}\left[\tilde{y}\right] = \mathbb{E}\left[\left(\tilde{\mathbf{y}} - \mathbb{E}\left[\tilde{\mathbf{y}}\right]\right)^2\right]. \tag{3}$$

Each of these represents a different aspect of the total error that affects the model's performance.

2. Model Bias

Model bias refers to the systematic error introduced by the model's inability to capture the true underlying relationship between the independent and dependent variables. This error arises when the model makes incorrect assumptions about the data or oversimplifies the relationship. For example, a linear model trying to fit highly non-linear data will result in high bias because the model cannot flexibly represent the complexity of the data.

From the expression (2) we see that the bias measures how far the average model prediction $\mathbb{E}[\tilde{\mathbf{y}}]$ is from the true target value $\mathbf{y}$. A high bias occurs when the model is too simple, leading to underfitting, where it fails to capture the underlying structure of the data.

3. Model Variance

Model variance refers to the sensitivity of the model to fluctuations in the training data. A model with high variance fits the training data very closely but may perform poorly on new, unseen data. This happens because the model has "memorized" the noise in the training data rather than capturing the true underlying pattern. High variance typically arises in overly complex models that are capable of capturing minute details, which may not generalize well to new data.

We see from the expression (3) for the model variance that it represents the variability in the model's predictions across different training sets. A high model variance indicates that the model is overly sensitive to the specific training data, leading to overfitting, where the model performs well on the training set but poorly on the test set.

4. Bias-Variance Tradeoff

An ideal model strikes a balance between bias and variance, where the model is flexible enough to capture the underlying patterns in the data but not so flexible that it fits the noise. At this point, the model minimizes the total error, including both bias and variance, as well as the irreducible noise variance. This balance is referred to as the bias-variance tradeoff, and is a fundamental concept in machine learning, specifically when building predictive models. We will study this in great detail throughout this work, as the model error

$$\mathbb{E}\left[(\mathbf{y} - \tilde{\mathbf{y}})^2\right] = \text{Bias}[\tilde{y}] + \text{Var}[\tilde{y}] + \sigma^2 \tag{4}$$

is minimized through finding the correct balance between the two first terms. This expression is derived in appendix A.

C. The Franke Function

The Franke function is a widely used synthetic function in numerical analysis and computational mathematics, particularly in the fields of interpolation, regression analysis, and surface fitting. It is a two-dimensional function defined over the unit square $[0,1]\times[0,1]$, combining several Gaussian functions of varying width and amplitude to simulate complex surface behavior. The function is defined as:

$$\begin{aligned} f(x,y) =& \frac{3}{4}\exp\left\{\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right)\right\} \\ &+ \frac{3}{4}\exp\left\{\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right)\right\} \\ &+ \frac{1}{2}\exp\left\{\left(-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right)\right\} \\ &- \frac{1}{5}\exp\left\{\left(-(9x-4)^2 - (9y-7)^2\right)\right\}. \end{aligned} \tag{5}$$

The function produces a surface with both smooth and non-smooth regions, making it an excellent benchmark for testing regression and interpolation algorithms. It simulates real-world data that may contain both complex variations and noise, mimicking scenarios that are encountered in practical data analysis tasks.

D. Terrain Data

III. METHODS

A. Regression Analysis

Regression analysis is a fundamental statistical technique used to model the relationship between one or more independent variables (also known as predictors or features) and a dependent variable (or target). The goal of regression analysis is to find the mathematical relationship that best explains the variation in the dependent variable based on the values of the independent variables.

At its core, regression analysis seeks to find a mathematical function that relates the independent variables to the dependent variable. In its most basic form, the relationship between a dependent variable $\mathbf{y}$ and an independent variable $\mathbf{x}$ is modeled as

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\epsilon}, \tag{6}$$

where $f(\mathbf{x})$ is the function we are trying to estimate, which represents the relationship between $\mathbf{x}$ and $\mathbf{y}$, and $\boldsymbol{\epsilon}$ is the error term, representing the part of the variation in $\mathbf{y}$ that is not explained by the model (due to noise or other unobserved factors).

1. Ordinary Least Squares

The Ordinary Least Squares (OLS) method is one of the most fundamental and widely used techniques in regression analysis. Its objective is to find the best-fitting line or curve for a given set of data by minimizing the sum of the squared differences between the observed values and the predicted values. These squared differences are called residuals, and minimizing them ensures the best possible fit of the model to the data. continue, introduce expressions (week 38)

2. Ridge

While OLS provides a foundational method for regression, it can lead to problems when the data has high multicollinearity or when there are more features than data points, leading to overfitting. OLS attempts to minimize the sum of squared errors, but it does not impose any restrictions on the model complexity. This often results in high variance when the model learns to fit noise in the training data. To address this, Ridge regression introduces a regularization term to the OLS cost function, penalizing large coefficients and preventing the model from becoming overly complex. By introducing the hyperparameter λ, Ridge regression balances the trade-off between bias and variance, allowing us to control model complexity and fit the data more effectively. double check, introduce expressions (week 37)

3. Lasso

Like Ridge regression, Lasso regression (Least Absolute Shrinkage and Selection Operator) is a regularization technique designed to improve the generalizability of the model by introducing a penalty term to the cost function. While Ridge regression uses the L2 norm (squared magnitude of coefficients) for regularization, Lasso regression employs the L1 norm, which leads to a different form of regularization. This difference has important implications, particularly for feature selection and sparse models. double check, introduce expressions

B. Resampling Methods

Resampling methods are statistical techniques used to generate additional data samples from the available data. These methods are particularly useful in machine learning and data analysis when the dataset is limited, and we want to better assess the performance of a model. The primary goal of resampling is to estimate the accuracy of a model by splitting the data into different subsets or generating new samples of the data, repeatedly fitting the model, and evaluating its performance on different sets of data. In this work we implement two commonly used resampling techniques: bootstrap and cross-validation.

1. Bootstrap

The Bootstrap method is a powerful resampling technique used to estimate the uncertainty and variability of a model by repeatedly drawing random samples, with replacement, from the original dataset. This resampling creates multiple "bootstrapped" datasets, each the same size as the original, but with some samples appearing multiple times and others potentially omitted.

The process of bootstrap resampling approximates the underlying distribution of a statistic—whether it's model performance, a parameter estimate, or prediction error—without requiring strong parametric assumptions about the data. This makes it particularly useful when the theoretical distribution of a statistic is unknown or difficult to calculate. Bootstrap methods are frequently used to estimate confidence intervals, standard errors, and model variance in both regression and classification tasks.

2. Cross-Validation

Cross-validation is another resampling technique that involves partitioning the dataset into several distinct subsets (or "folds"), and then systematically training the model on one subset while testing it on another. A common form is k-fold cross-validation, where the dataset is

divided into k equally-sized folds. The model is trained on $k-1$ folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the test set exactly once.

Steps in k-fold Cross-Validation: 1. Split the dataset into k equal-sized folds (typically $k = 5$ or $k = 10$). 2. Train the model on $k-1$ folds and evaluate it on the remaining fold. 3. Repeat the process k times, so that each fold is used as a test set exactly once. 4. Calculate the average performance across all folds.