Data Analysis and Machine Learning Project 2

Andrei Kukharenka, Anton Fofanov and Anna Gribkovskaya
FYS-STK 4155

December 29, 2018

Abstract

This project is devoted to study of Ising model using different statistical methods. We begin with linear regression then move to classification of spin configurations using either logistics regression and neural network. LASSO regression provide the best results among the regression methods with $R^2 = 1$ and $MSE = 10^{-6}$ for $\lambda = 10^{-3}$.
Neural network and logistic regression (ADD).

1 Introduction

Data science is one of the most rapidly developing parts of information technologies nowadays. The increase of computer power allow us to analyze huge amounts of data and this require some specific methods and techniques to be studied. Some of them have been already under consideration in the previous project, for example simple regression methods - linear, Ridge and Lasso. In this project we aim to tackle a classification problem using logistics regression. After that our goal is to move towards the neural network. The problem we are going to use as a test bed is Ising model.

Structure of the report. The first part is a theoretical description of the 1D and 2D Ising model. Second part is a brief description of the methods. After this we move toe the results and discussion part. The last part is conclusion where we present a brief summary of what have been dona and also discuss some possibilities for further research.

2 Problem description

This project is mostly based on the work of Metha et al. [1] and that's why we are using the problem formulation provided in this article. However, Ising model is a well known model in physics and one may find many studies devoted to the model. For example, it's one of the natural choices to study Monte Carlo

simulations, as it have been done here (REF to project 4).

Ising model is a model that allows us to to compute the energy of a system of spins. Each spin can take only two values ± 1. The interaction is allowed only for closest neighbors. Generally speaking the Ising model provides us a simple approach to model the phase transitions of a ferromagnet. Phase transitions are transitions between ordered or disordered states. As soon as ordered state is more preferable for lower temperatures and disordered state is more preferred for higher temperature there shroud be some critical temperature when the switch is happening. In the project we will study 1D and 2D Ising models and the periodical boundary conditions are used. For the 1D Ising model there is no phase transition, while for the 2D the phase transition is happening at the critical temperature $T_c/J = 2/\log(1 + \sqrt{2}) \approx 2.26$.

2.1 1D Ising model

The Hamiltonian for the classical 1D Ising model is given by

$$H = -J \sum_i^N S_i S_{i+1}, \qquad S_i \in \{\pm 1\}, \tag{1}$$

where N is number of particles in the system, and S_i is a spin pointing up or down.

2.2 2D Ising model

The Hamiltonian for the classical 2D Ising model is given by

$$H = -J \sum_{\langle ij \rangle}^N S_i S_j, \qquad S_j \in \{\pm 1\}, \tag{2}$$

where the lattice site indices i,j run over all nearest neighbors of a 2D square lattice, and J is some arbitrary interaction energy scale. Onsager proved that this model undergoes a thermal phase transition in the thermodynamic limit from an ordered ferromagnet with all spins aligned to a disordered phase.

2.3 Ising model for statistical methods

In order to apply the regression methods to the Ising model we need to rewrite the equations in terms of so-called coupling coefficient J_{ij} as:

$$H = -\sum_{\langle ij \rangle}^N J_{ij} S_i S_j. \tag{3}$$

Now we can apply regression to determine the J_{ij}. In order to use classification methods, for example logistics regression, we use already prepared 2D spin configurations that have been prepared by Metha et al. in [1].

3 Methods

3.1 Logistic Regression

In the

3.2 Neural networks

Neural networks are very popular in the field of machine learning. The name "neural" refer to the fact that such networks are supposed to mimic a biological system of communicating neurons. Neural network is a network of layers each of them containing an arbitrary number of neurons. The connection in this case is represented by a weight.

The artificially build neural network should be able to behave similarly to a real neural network in human brains. By the analogy how the neurons connected in the human brain the mathematical model was established which can be used for both classification and regression problem. Starting from the simplest possible example the neural network can be represented as the following(see figure 1). The neural network has input and output layers, alongside with some amount
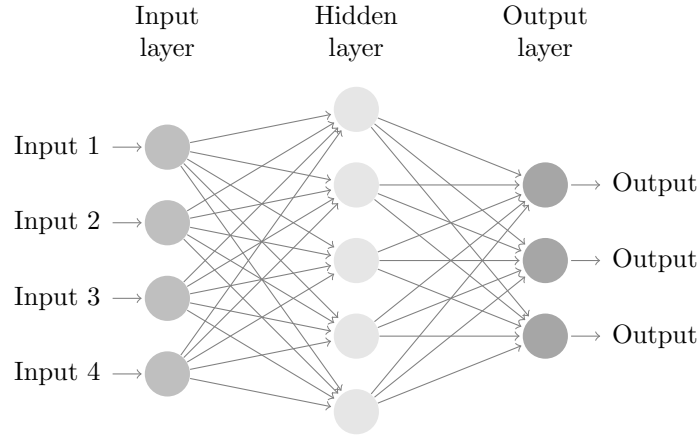

Figure 1: Neural network scheme.

of the hidden layers located between input and output. The way neural network operates is activations in one layer determine the activation of the next layer. It can be assigned a weight to each one of the connections between our neuron and the neuron from the following layer. Then the weighed sum of all activations from the input layer is calculated. To be able to handle numerically the values of the activation sum(these values in general can have any values depending on the size of neural network), one needs to have these values to be in interval $[0, 1]$. To do this an activation function is used. A common function to use is the sigmoid function. The different activation functions we tested in this project will be described below in a greater details.

So the activation function is a basically a way to determine how positive this particular weighed sum is. To find a threshold for activation of one neuron one needs to add some number to the weighed sum. To be more precise, to subtract it from the weighed sum. This number is called the bias and it determines how large the weighted sum needs to be to activate a neuron. Described procedure was for just a one neuron from the hidden layer, where all the neurons from the input layer were connected with. This trail of thoughts can be repeated for the each neuron in the hidden layer, so each neuron has an individual weight and one dedicated bias. We need to find all the right weights and biases. Mathematically it can be expressed via linear algebra language. Organizing all the inputs into a vector A, such so for input layer($l = 0$):

$$A^{(0)} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \\ \vdots \\ x_m^{(0)} \end{bmatrix} \tag{4}$$

We can construct a weight matrix $(W^{(l)})$associated with each layer, where each row represents a connection between one layer and a particular neuron in the next layer. By denoting via vector B a row-vector, containing all the biases, the activation of the next layer can be expressed via the following expression:

$$A^{(l+1)} = \sigma(W^{(l)} A^{(l)} + B) \tag{5}$$

For the input layer one applies the latter equation to the input vector and then its activation function. The second layer receives the output of the first layer and the whole scheme is repeated up to output layer.

The learning process of the neuron network is just an adjustment of the weights and biases. So when the neural network gets an unknown data as an input, the output will be correct. We define a cost function which might be seen as a representation of the correlation between input and output. Squared difference between desired result and the output of the neural network can be considered as a measure here. So the cost function is a multidimensional function of all the weights and biases which should provide us a score how good our neural network performs. It is naturally to employ some kind of multi-dimensional minimization algorithms to find a minimum of the cost function. The most simple approach is to use the steepest descent algorithm [1].

3.3 Activation functions

The sigmoid or so-called logistic function has been widely used in the hidden layers of the neural network:
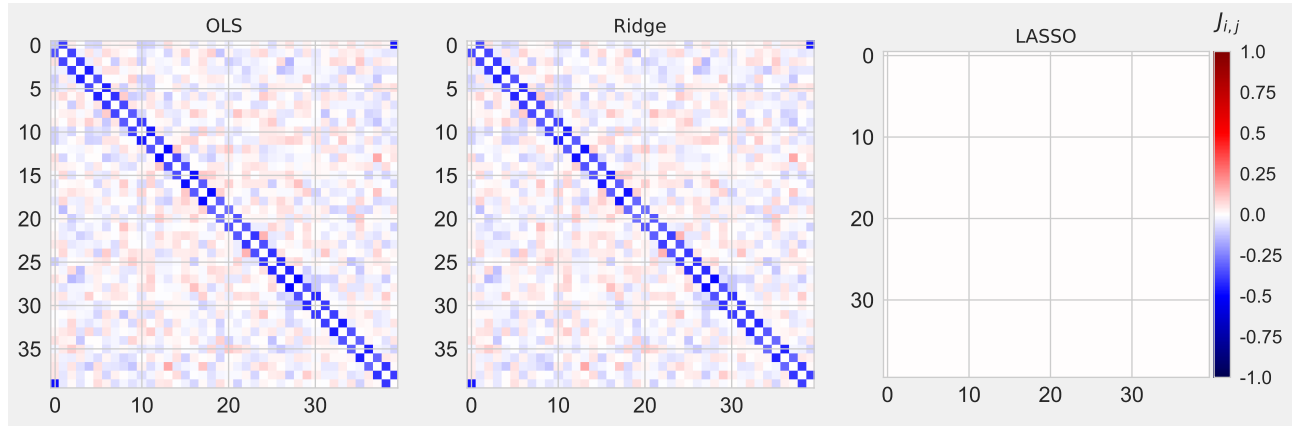
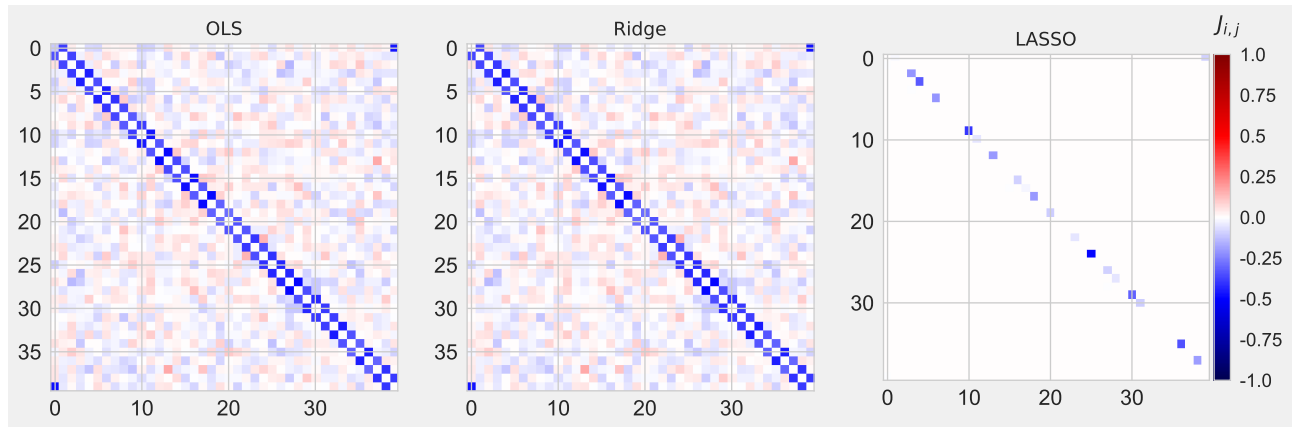$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Figure 2: λ=10.0

Figure 3: λ=1.0

As an alternative one might use the hyperbolic tangent as the activation function. Another example is the rectified linear unit (ReLU), which is motivated by a biological analogy(neurons would be reither activated or not):

$$\text{ReLU}(x) = \max(0, x)$$

The rectified linear unit usually performs the best [1]. In our project we use all mentioned activation functions.
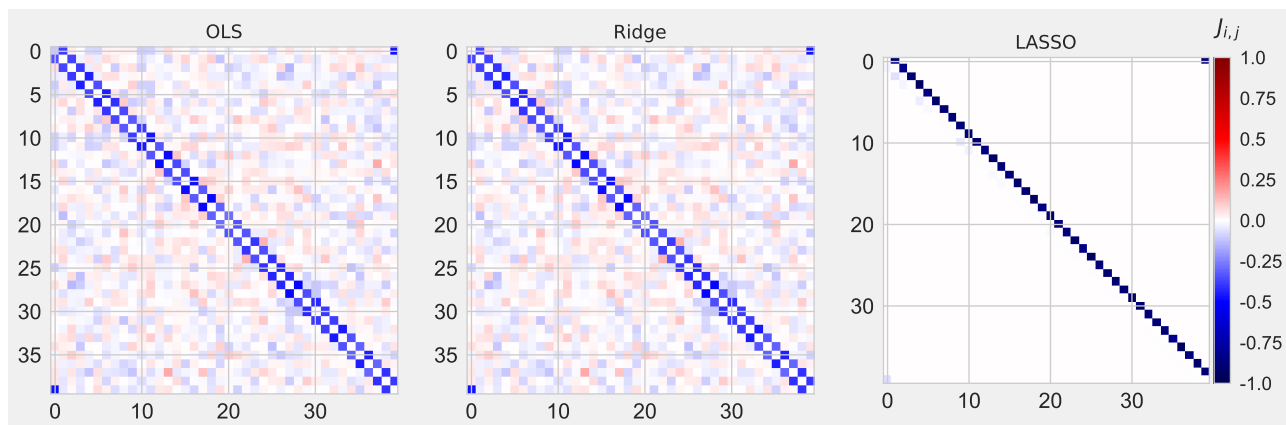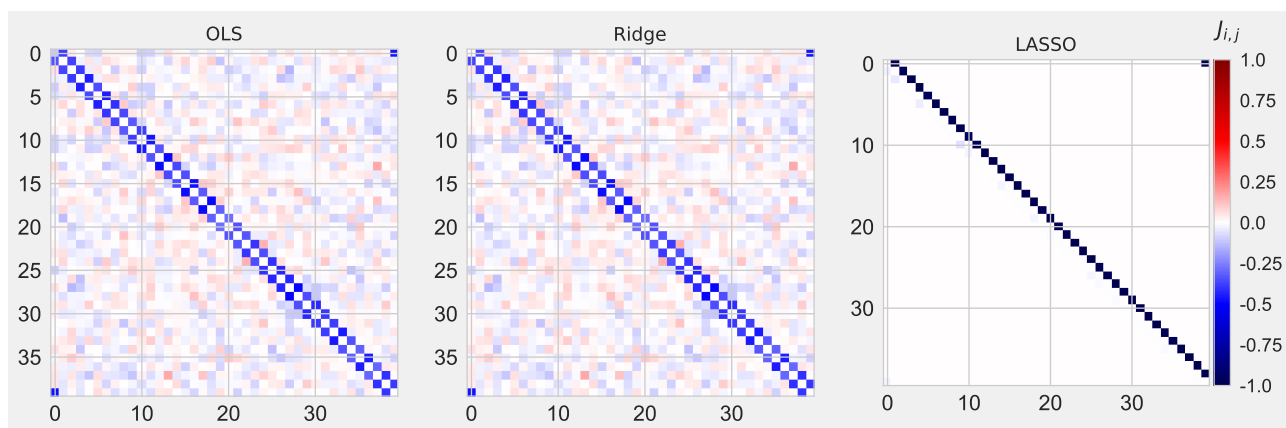
5

Figure 4: λ=0.1
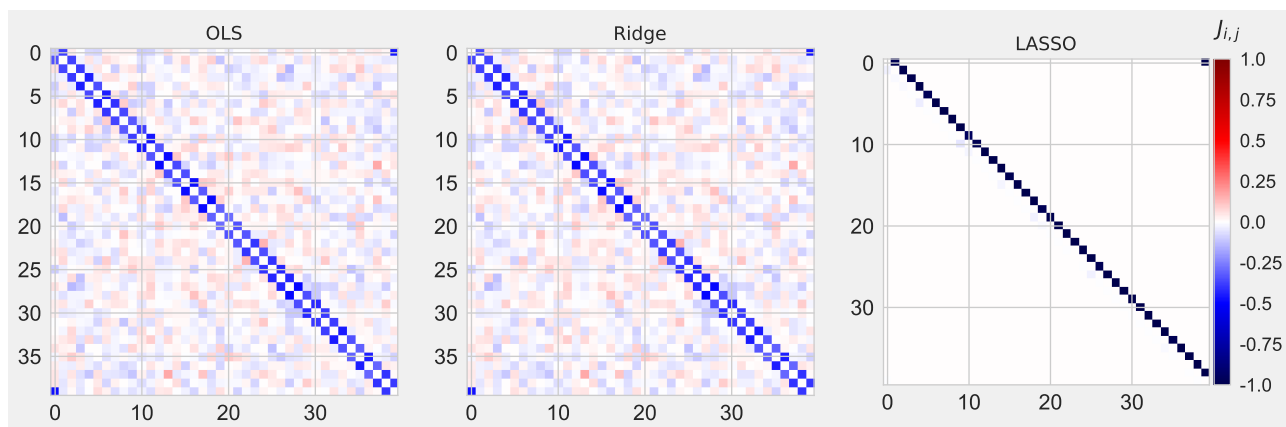

Figure 5: λ=0.01
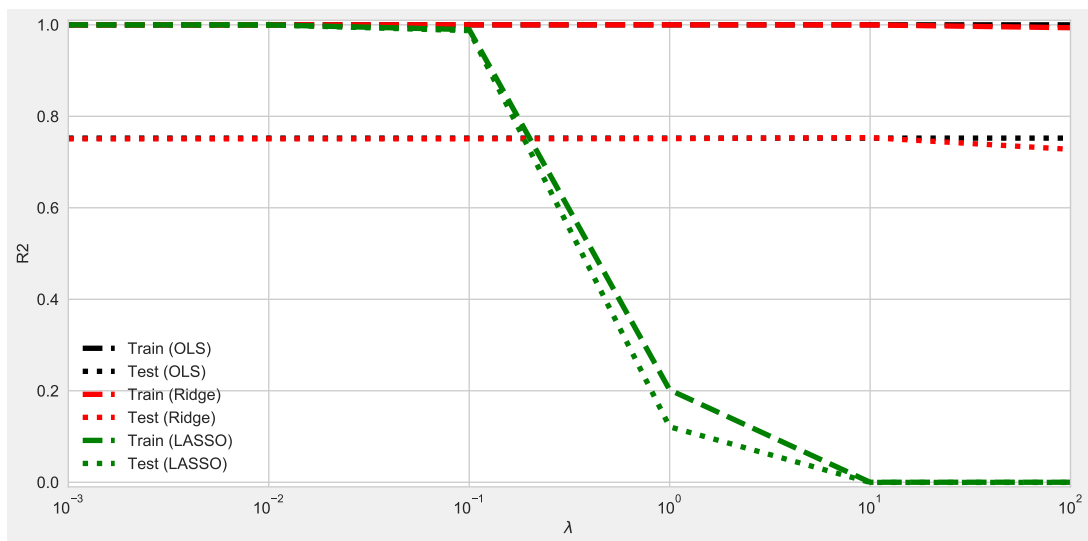
6

Figure 6: λ=0.001
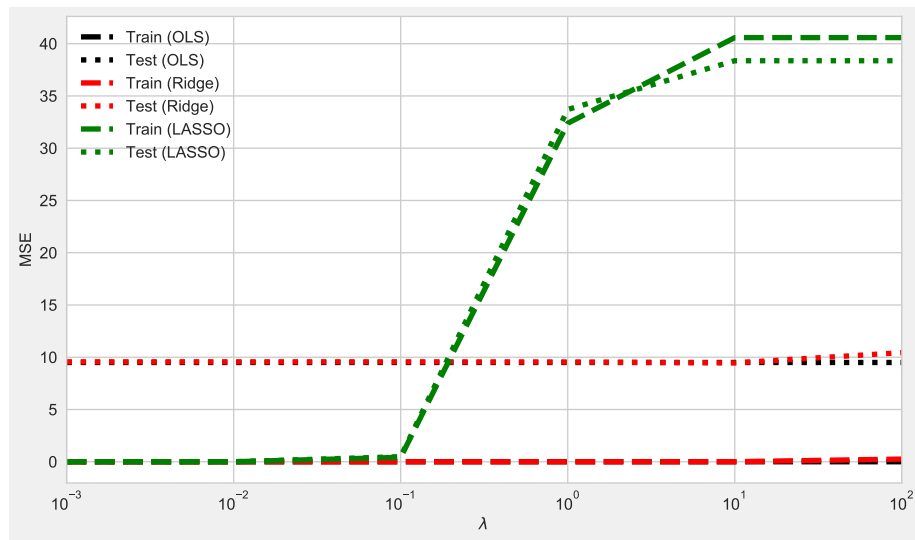

Figure 7: R2

Figure 8: MSE

4 Results and discussion

5 Conclusion

6 Further work

References

[1] Pankaj Metha et al. *A high-bias, low-variance introduction to Machine Learning for physicists* ArXiv:1803.08823.