



UNIVERSITY OF OSLO

FYS-STK3155

---

## Project 1

---

Eirik  
Felix  
Brage Wiseth

EIRIK@IFI.UIO.NO  
FELIX@IFI.UIO.NO  
BRAGEWI@IFI.UIO.NO

September 16, 2023

[HTTPS://GITHUB.COM/BRAGEWISETH/MACHINELEARNINGPROJECTS](https://github.com/bragewiseth/machinelearningprojects)

# Contents

1	Introduction . . . . .	3
2	Ordinary Least Squares . . . . .	4
3	Ridge . . . . .	4
4	Lasso . . . . .	4
	4.1 Discussion on Scaling . . . . .	4
	Bibliography . . . . .	7

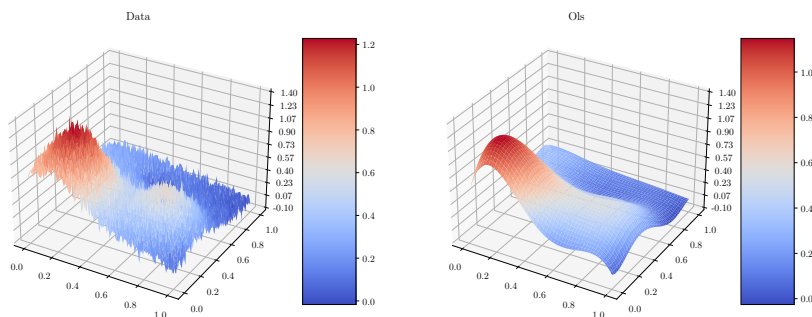
## Abstract

This paper describes the mixtures-of-trees model, a probabilistic model for discrete multidimensional domains. Mixtures-of-trees generalize the probabilistic trees of in a different and complementary direction to that of Bayesian networks. We present efficient algorithms for learning mixtures-of-trees models in maximum likelihood and Bayesian frameworks. We also discuss additional efficiencies that can be obtained when data are “sparse,” and we present data structures and algorithms that exploit such sparseness. Experimental results demonstrate the performance of the model for both density estimation and classification. We also discuss the sense in which tree-based classifiers perform an implicit form of feature selection, and demonstrate a resulting insensitivity to irrelevant attributes.

**Keywords:** Linear Regression, Fitting Polynomials, Scaling, Bias-Variance

## 1. Introduction

Franke function bla bla bla blablbisdjaidh sdhkadhjsiajdsdsah



## 2. Ordinary Least Squares

## 3. Ridge

## 4. Lasso

### 4.1 Discussion on Scaling

**Unscaled** sample design matrix fitting one-dimensional polynomial of degree 5

```
[[1.      0.      0.      0.      0.      0.      ]
 [1.      0.25    0.0625  0.01562 0.00391 0.00098]
 [1.      0.5     0.25    0.125   0.0625  0.03125]
 [1.      0.75    0.5625  0.42188 0.31641 0.2373 ]
 [1.      1.      1.      1.      1.      1.      ]]
```

**Scaled** sample design matrix fitting one-dimensional polynomial of degree 5

```
[[ 0.      -1.41421 -1.0171  -0.83189 -0.728   -0.66226]
 [ 0.      -0.70711 -0.84758 -0.7903   -0.71772 -0.65971]
 [ 0.       0.      -0.33903 -0.49913 -0.56348 -0.58075]
 [ 0.       0.70711  0.50855  0.29116  0.10488 -0.0433 ]
 [ 0.       1.41421  1.69516  1.83016  1.90431  1.94603]]
```

**Unscaled**  $\beta$  after fitting two-dimensional polynomial of degree 5

```
[ 0.4097   7.55124   3.79304 -32.85696 -14.83669 -8.81645  45.45889
 43.33221  20.70625  -7.63623 -21.24552 -51.81866 -7.53731 -29.60175
 28.57282   0.73824  18.29253  10.60883  -5.52465  16.60259 -16.13743]
```

**Scaled**  $\beta$  after fitting two-dimensional polynomial of degree 5

```
[ 2.21481  1.1093  -9.93941 -3.32124 -2.6612  13.11848  8.70306  4.14949
 -2.19979 -5.79259 -9.48697 -1.28166 -5.40093  7.77825  0.19055  3.09519
 1.60557 -0.83485  2.797   -4.1589 ]
```

```
0.010349396022903145
0.010349396024145656
0.02106077418650843
0.01782525371566323
```

the code for genertating this output <sup>1</sup>

First  $x_{\text{unscaled}}$  and  $x_{\text{scaled}}$  is not that different, the original data was close to zero-centered and not that spread out, which means that initaially by just looking at the data scaling is not that neceary. When we add the polynomial terms we can now see that some of the entries of  $X_{\text{unscaled}}$  get really small as an exsample  $0.1^5 = 0.00001$  this makes the collums of  $X_{\text{unscaled}}$  live in their own order of magnitude and scaling should be considered to bring

1. <https://github.com/bragewiseth/MachineLearningProjects/blob/main/project1/src/ols.py>[5]

them back to the same ish order of magnitude. The act of not scaling results in  $\beta$  spanning from  $-50$  to  $48$  while scaling gives a smaller span from  $-10$  to  $13$  (We would imagine that keeping everything in the same order of magnitude is something that the computer likes, perhaps reducing floatingpoint error). Now this alone may not justify why we should scale this dataset, as scaled and unscaled OLS yields the same MSE. However when doing ridge regression the cost function is directly dependent on the magnitude of  $\beta_i$ . This explains why the MSE for unscaled data is much higher than for scaled data in the ridge case. This leads us to conclude that we should scale the data, making it easier to tweek  $\lambda$  and giving us nicer numbers to work with. <sup>1</sup>

## Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642) and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637).

## Appendix A.

In this appendix we prove the following theorem from Section 6.2:

**Theorem** *Let  $u, v, w$  be discrete variables such that  $v, w$  do not co-occur with  $u$  (i.e.,  $u \neq 0 \Rightarrow v = w = 0$  in a given dataset  $\mathcal{D}$ ). Let  $N_{v0}, N_{w0}$  be the number of data points for which  $v = 0, w = 0$  respectively, and let  $I_{uv}, I_{uw}$  be the respective empirical mutual information values based on the sample  $\mathcal{D}$ . Then*

$$N_{v0} > N_{w0} \Rightarrow I_{uv} \leq I_{uw}$$

*with equality only if  $u$  is identically 0.* ■

**Proof.** We use the notation:

$$P_v(i) = \frac{N_v^i}{N}, \quad i \neq 0; \quad P_{v0} \equiv P_v(0) = 1 - \sum_{i \neq 0} P_v(i).$$

These values represent the (empirical) probabilities of  $v$  taking value  $i \neq 0$  and 0 respectively. Entropies will be denoted by  $H$ . We aim to show that  $\frac{\partial I_{uv}}{\partial P_{v0}} < 0 \dots$

*Remainder omitted in this sample. See <http://www.jmlr.org/papers/> for full paper.*

# Bibliography

- [1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Book in preparation for MIT Press.
- [3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- [4] Wessel N. van Wieringen. Lecture notes on ridge regression, 2023.
- [5] Brage Wiseth, Eirik ?, and Felix ? MachineLearningProjects. <https://github.com/bragewiseth/MachineLearningProjects>, September 2023.