matLearn: User-Friendly Large-Scale Machine Learning

Student: Jennifer She | Supervisor: Mark Schmidt | School: University of British Columbia
Contact Information: x.she@alumni.ubc.ca

Background

Machine learning involves studying and working with algorithms that can learn from and make predictions on data. These algorithms are becoming essential to help us make sense of and make use of the ever-growing quantity of data collected across many fields of science, engineering and business. Some simple examples where these algorithms can be used include:

- Using symptoms exhibited by a patient and patient records to predict whether the patient is likely to have an illness
- Using past prices for a stock to determine whether the stock should be bought, held or sold

Introduction & Purpose

This project focused on putting together a software package of fundamental machine learning algorithms in MATLAB. The product is around 60 machine learning models, and over 40 demonstrations of using these models with simulated datasets.

Significance of this package:

- the package is built using modern numerical optimization techniques which scales up to large datasets (advantageous over existing tools)
- the models have a unified format for inputs and outputs making the package very user-friendly
- the demonstrations and structured nature of the package allows it to be used for educational purposes

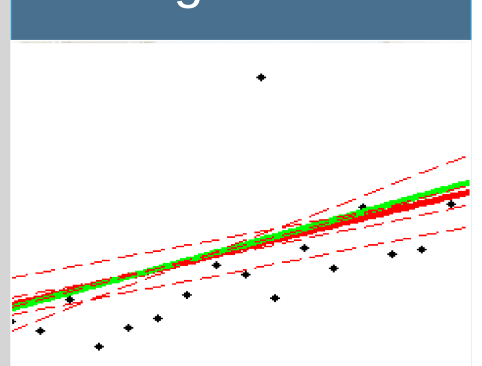
Problem Type

This project, for the most part, focused on supervised regression, binary classification and classification problems.

$$\vec{x} \rightarrow \boxed{} \rightarrow y$$

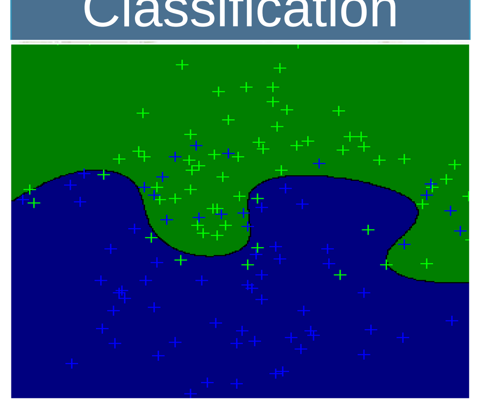
explanatory variables — response variable

Regression



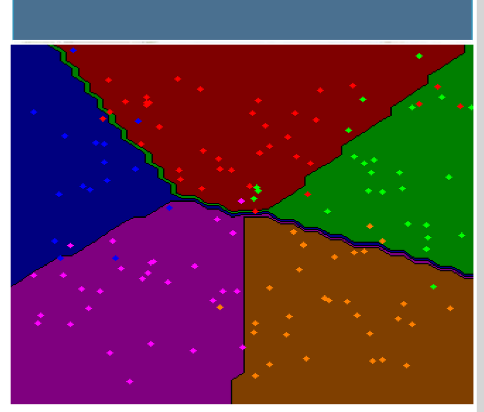
Continuous response variable

Binary Classification



Binary response variable

Classification

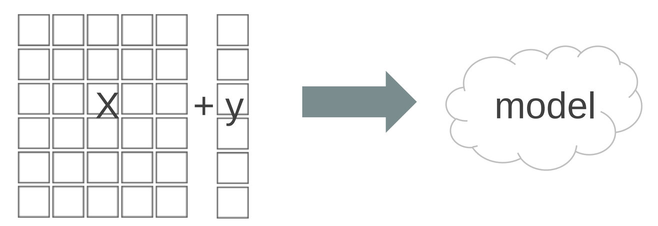


Discrete response variable

Model Structure

Step 1: Training

Create model using training data X and y



Input Specifications:

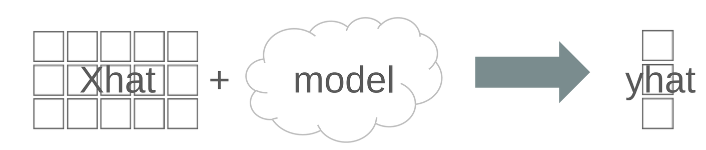
- X: n-by-d design matrix where each column is a different explanatory variable, and each row is a set of data for the explanatory variables
- y: n-by-1 target vector where each element is a value for the response variable corresponding to a row in X
- `options`: additional fields specified by documentation

Output:

- `model`: stores parameters needed to make predictions

Step 2: Prediction

Make predictions `yhat` given new data `Xhat` using model from training



Input Specifications:

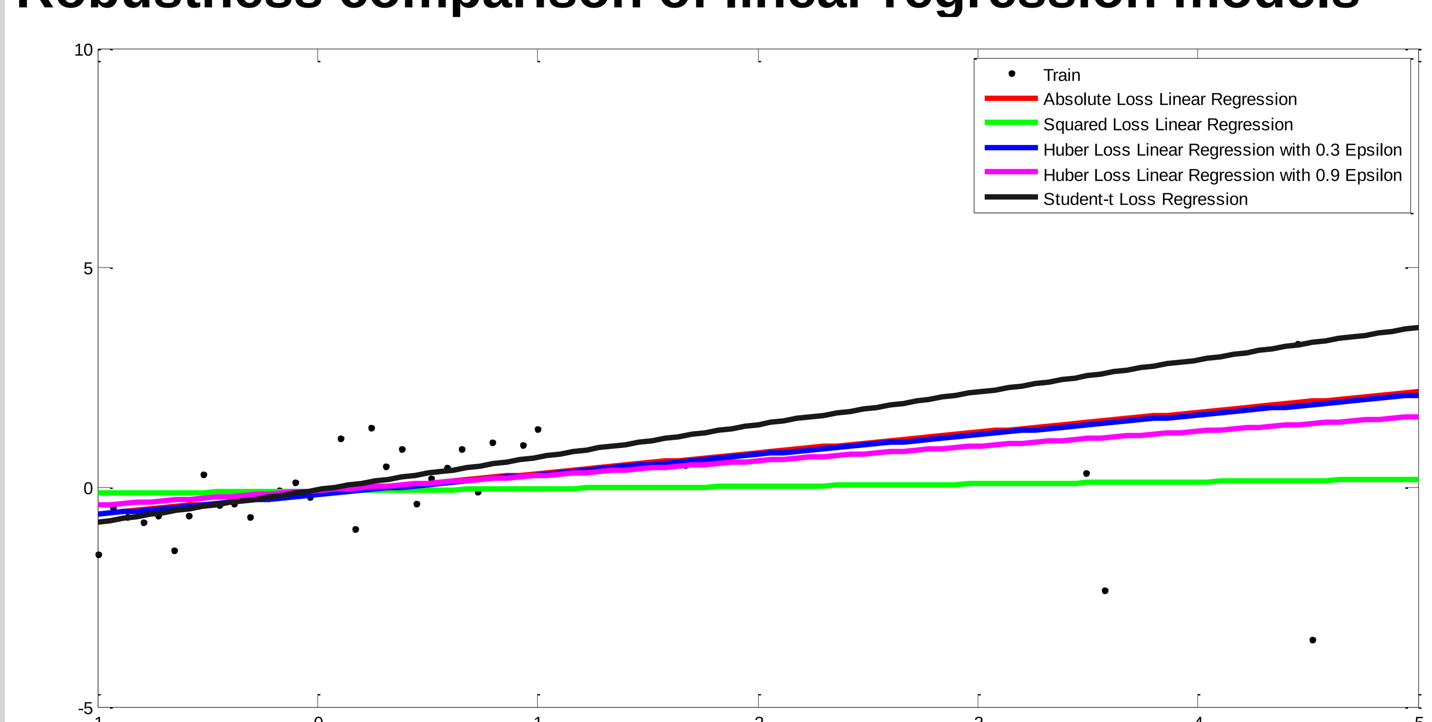
- `model`: stores parameters needed to make predictions
- `Xhat`: m-by-d matrix where each row is a set of data for the explanatory variables, but do not have corresponding values for the response variable

Output:

- `yhat`: m-by-1 vector of the predicted response values corresponding to Xhat using the model

Demonstration Example #1

Robustness comparison of linear regression models

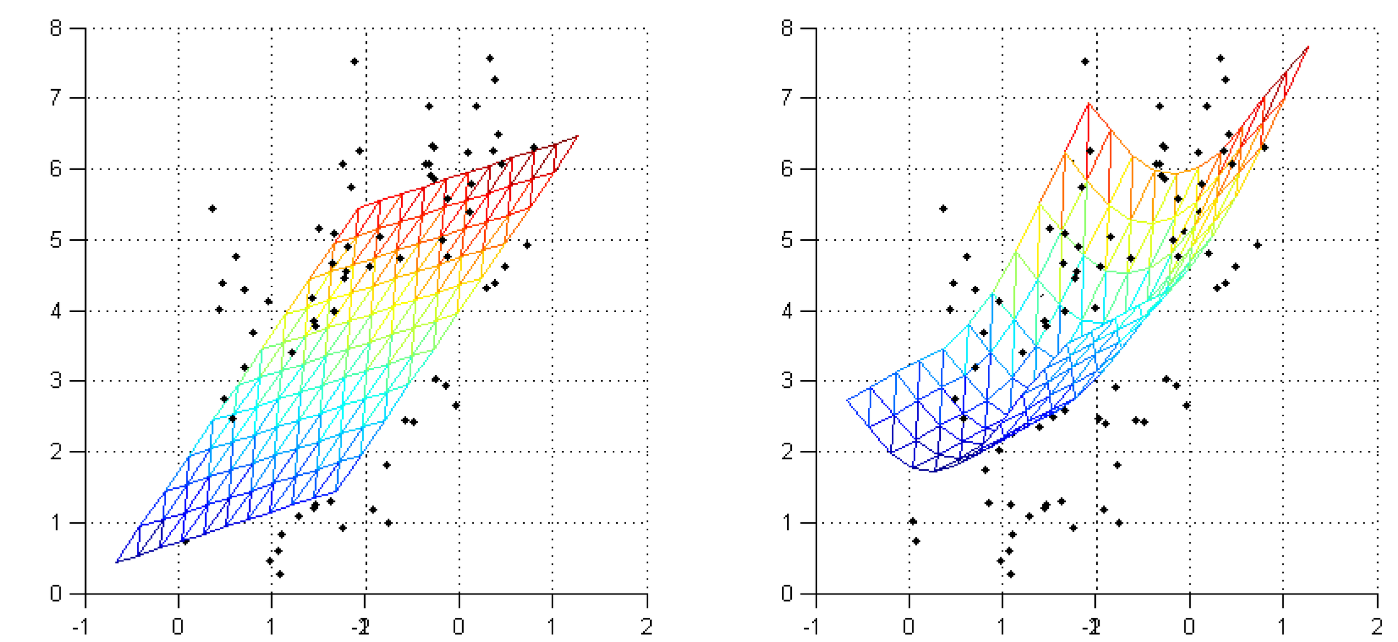


Conclusion:

- L2 (least squares) regression is generally the least robust against outliers because outliers are weighted more heavily
- Huber loss with a lower epsilon threshold appears to be more robust
- Student-t regression model seems to be the most robust

Demonstration Example #2

Linear regression model vs. additive regression model



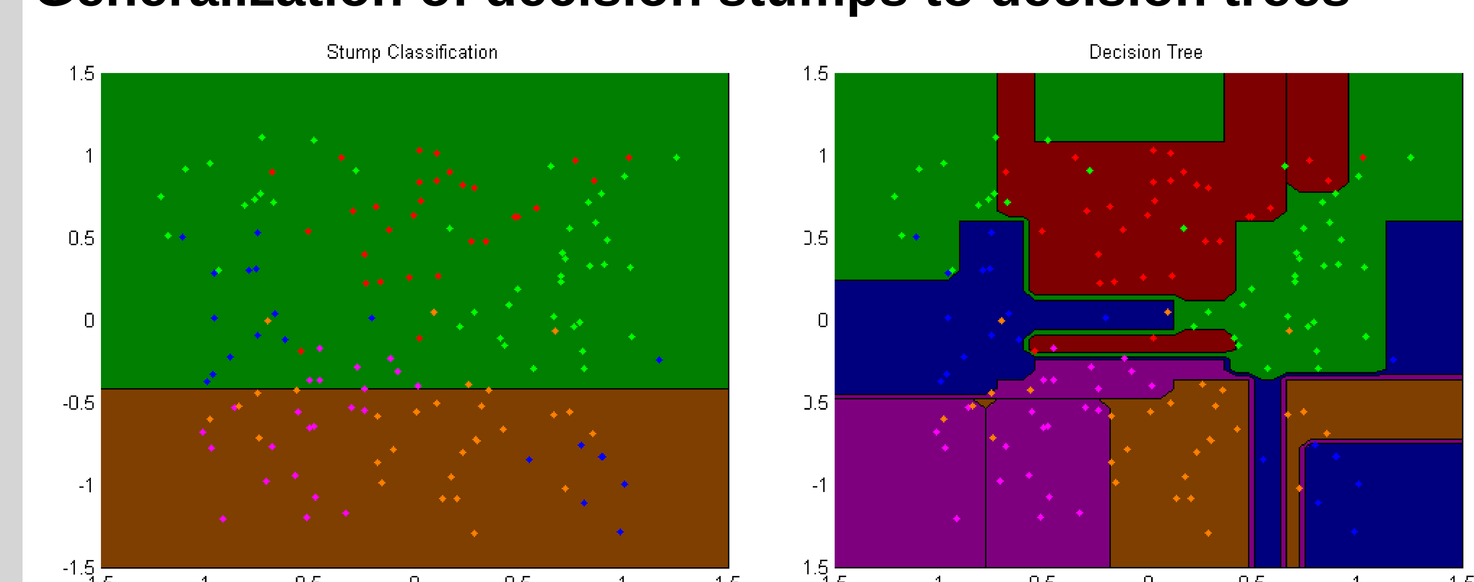
Least squares regression — Cubic splines additive regression

Conclusion:

- Generalized additive models (right) use non-linear functions on each explanatory variable, resulting in a non-linear model, thus are more flexible compared to linear models
- Generalized additive models can overfit to the small curvatures of the training data whereas linear models (left) can be more robust for linear data

Demonstration Example #3

Generalization of decision stumps to decision trees



Conclusion:

- Decision stump binary classification (left) is limited in its classification capability with slightly more complex data
- Decision tree classification model (right) recursively uses many binary decision stumps to allow for more accurate classification

Acknowledgement

I thank Mark Schmidt for much of the foundational code (for optimized minimization, plotting, etc.) required for the project and lots of help as a supervisor. Also, I thank the CPSC 540 2015W2 class for providing much of the training & prediction code to work with.