Linear Regression and Regularization

by Dane Brown

“Life is ten percent what you experience and ninety percent how you respond to it.”

Scoring a Regression Model

- *mean_squared_error*: squared error between **predicted** and **true value** for every data point in the training set, averaged across all data points.

- *explained_variance_score*: the degree a model can explain the variation or dispersion of the test data. Measured using the correlation coefficient.

- *r2_score*: The R^2 score is closely related to the explained variance score, but uses an unbiased variance estimation. It is also known as the coefficient of determination.

cvML Methodology

- **Initialization:** Call the `cv` or `scikit` model by name to create an empty instance of the model
- **Set parameters:** default
- **Train the model:** `train` or `fit` is used to fit the model to some data
- **Predict new labels:** use `predict`, to guess the labels of new (**unseen**) data
- **Score the model:** refer to slide 10 & 13: works for both `cv` or `scikit`

Linear Regression

- Continuously predict new outcomes
- Describe a target variable with a linear combination of features

- Dataset: Boston house pricing
  - simply predict housing prices
  - no classifying of labels (into classes)
  - $\hat{y} = w_1 f_1 + w_2 f_2$
  - if f_1 and f_2 are today's price for two houses
  - train w_1 and w_2 to learn tomorrow's price
- OpenCV does not offer any good implementation of linear regression...

Linear Regression

`Check it out -> CV_ML`

- The goal is to predict the value of homes in several Boston neighborhoods in the 1970s, using information such as
  - crime rate
  - property tax rate
  - distance to employment centers
  - highway accessibility
  - etc.

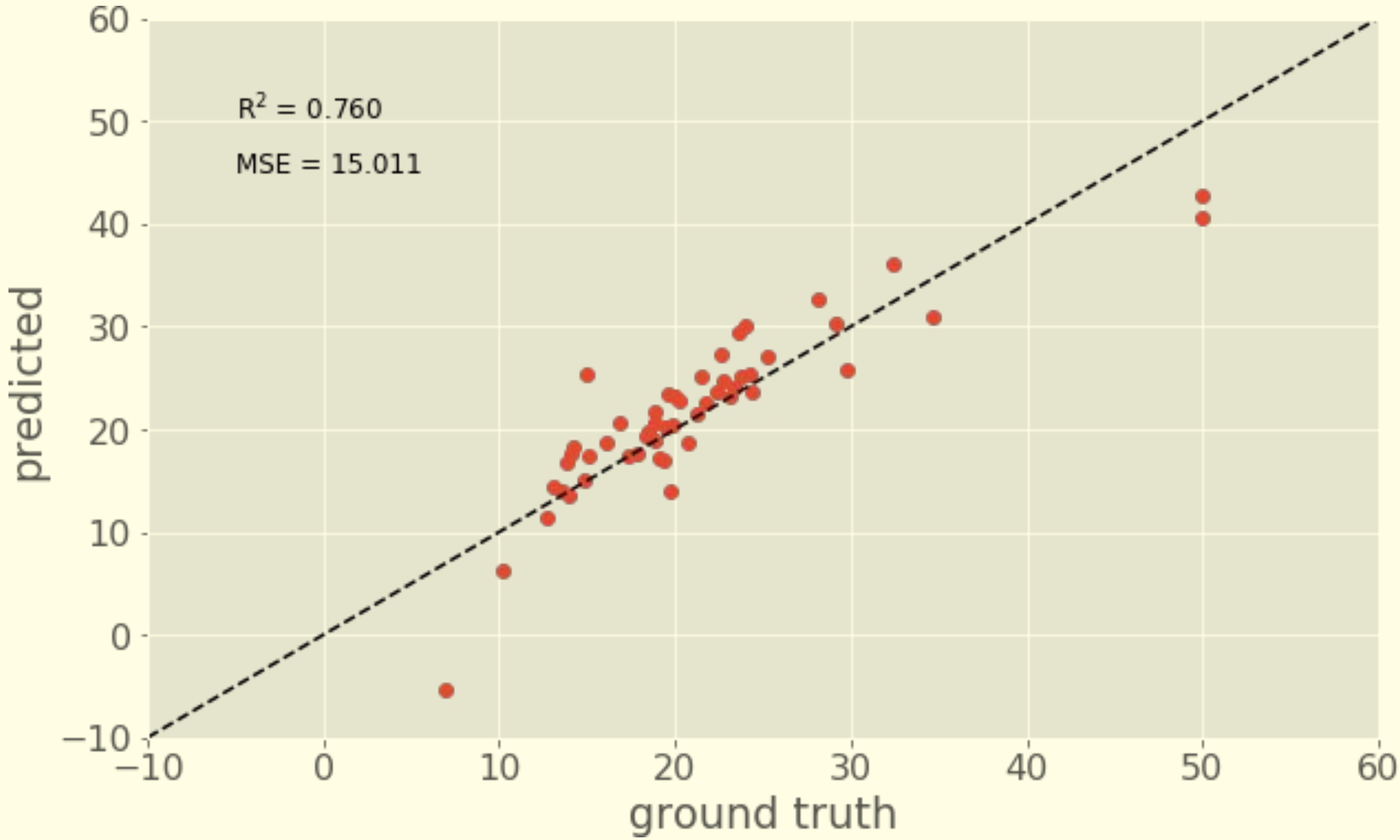
Linear Regression Results

The model is noticeably off at certain points...

Linear Regression Results

- The model tends to be off the most at peaks
  - i.e. really high or really low housing prices
  - peak values of data points 12, 18, and 42.

Linear Regression Results



Linear Regression Results

- A perfect model has all data points on the dashed diagonal, since **y_pred** would equal **y_true**
- Deviations from the diagonal indicate model errors: the model was not able to explain some variance in the data .
- R-squared shows that the model explained 76% of the scatter in the data, with an MSE of 15.011.
- ML project: These are the kind of numbers to use in comparisons and results chapter of your thesis!

Overfitting

- An algorithm might work really well on the training set, but poorly on unseen data
- This means the model does not generalize well
- This is especially common in tree-based classifiers
- This can also happen for other classifiers, including **regression** problems

Overfitting: How to Reduce it

- Use Regularization
- L1 norm of the Manhattan distance:
  - This adds a term to the scoring function that is proportional to the sum of all absolute weight values
  - Also known as **Lasso** regression
- L2 norm of the Euclidean distance:
  - This adds a term to the scoring function that is proportional to the sum of all squared weight values.
  - Removes strong outliers in the weight vector much more than the L1 norm
  - Also known as **Ridge** regression

Overfitting: How to Reduce it

- Original Linear regression:
  - linreg = linear_model.LinearRegression()

- Lasso regression
  - linreg = linear_model.Lasso()

- Ridge regression
  - linreg = linear_model.RidgeRegression()

Lasso vs. Ridge

Neither are better in general

Each tend to do well under conditions:

- **Lasso**
  - a small number of significant parameters and the others are close to zero significance
  - i.e. when few predictors influence the response
- **Ridge**
  - many significant parameters of similar value
  - i.e. when most predictors impact the response

- The boston dataset has many predictor variables