Regression

Ahmed Bou-Rabee and Stephen Boyd

EE103
Stanford University

November 2, 2016

Outline

Regression model

Example

Feature engineering

Regression model

- we assume there is an approximate relation between n-vector x and scalar y: $y \approx f(x)$
- x is called *feature vector* or *regressor*
- y is called *outcome* or *dependent variable*
- *regression model* is affine function of x given by

$$\hat{y} = \hat{f}(x) = x^T \beta + v$$

  where $\beta \in \mathbf{R}^n, v \in \mathbf{R}$ are *model parameters*
- n-vector β is *weight vector*, scalar v is *offset*
- the regressors x_i are typically shifted and scaled to be on approximately the same scale
  (say, with a mean of 0 and standard deviation of 1)

Measurements/data

- we have N *samples* or *examples*

$$(x_1, y_1), \ldots, (x_N, y_N)$$

- define $n \times N$ matrix $X = [x_1 \cdots x_N]$ and N-vector $y = (y_1, \ldots, y_N)$
- define N-vector $\hat{y} = (\hat{f}(x_1), \ldots, \hat{f}(x_N))$ (predicted outcomes)
- can express predictions as

$$\hat{y} = X^T \beta + v\mathbf{1}$$

- prediction error N-vector (on data) is

$$\hat{y} - y = X^T \beta + v\mathbf{1} - y$$

Regression

- choose β, v to minimize sum square prediction error

$$\left\|X^T\beta + v\mathbf{1} - y\right\|^2 = \left\|\left[\mathbf{1}\ X^T\right]\left[\begin{array}{c} v \\ \beta \end{array}\right] - y\right\|^2$$

- a least squares problem with variables β, v
- solution

$$\left[\begin{array}{c} \hat{v} \\ \hat{\beta} \end{array}\right] = \left(\left[\mathbf{1}\ X^T\right]^T\left[\mathbf{1}\ X^T\right]\right)^{-1}\left[\mathbf{1}\ X^T\right]^T y$$

Validation

- we want $y \approx \hat{f}(x)$ on *new, unseen data*
- when this happens, we say model *generalizes*
- to check this, we reserve some of the data as a *test set*, leaving the rest of the data as a *training set*
- we *fit* the model by regression on the training set
- we *test* the model on the test data set
- if the RMS prediction error on the test set is similar to the RMS prediction on the training set, we have (some) confidence in the regression model
- if the RMS test prediction error is much larger than the RMS training error, the model is *over-fit*, and we don't trust it

Outline

Wine quality/rating

- 1599 red wines
- 11-feature-vector x
- outcome y is median of expert ratings (integer between 1 and 10)
- $\mathbf{avg}(y) = 5.6$, $\mathbf{std}(y) = 0.8$
- split data into training set (1279 samples) and test set (320 samples)

Regressors

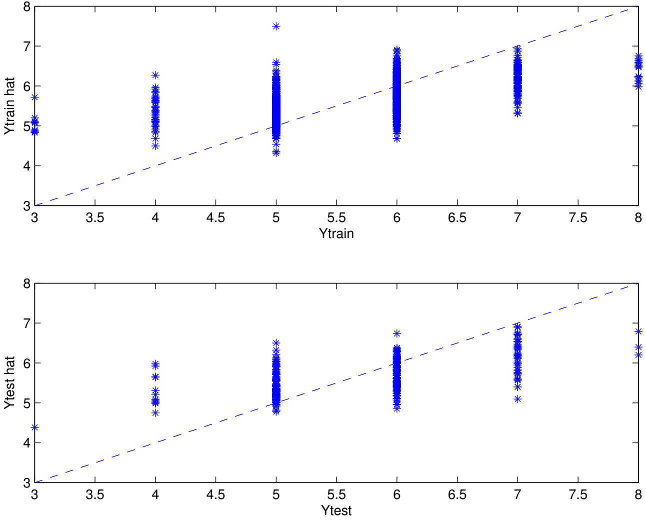
x_1	fixed acidity
x_2	volatile acidity
x_3	citric acid
x_4	residual sugar
x_5	chlorides
x_6	free sulfur dioxide
x_7	total sulfur dioxide
x_8	density
x_9	pH
x_{10}	sulphates
x_{11}	alcohol

(regressors are shifted and scaled so mean ≈ 0, std. dev. ≈ 1)

Results

model	RMS train error	RMS test error
constant	0.80	0.83
regression	0.65	0.64

Results



8
7
6
5
4
3
Ytrain hat
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8
Ytrain
8
7
6
5
4
3
Ytest hat
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8
Ytest

Regression model parameters

x_1	fixed acidity	0.06
x_2	volatile acidity	-0.18
x_3	citric acid	-0.03
x_4	residual sugar	0.02
x_5	chlorides	-0.07
x_6	free sulfur dioxide	0.05
x_7	total sulfur dioxide	-0.09
x_8	density	-0.05
x_9	pH	-0.06
x_{10}	sulphates	0.15
x_{11}	alcohol	0.30
1	(constant)	5.62

5-fold validation

- divide data (1599 samples) into 5 *folds* (each with ≈ 320 samples)
- for $i = 1, \ldots, 5$ train on all folds except i
- then test regression model on fold i

- results:

test fold	train RMS	test RMS
1	0.65	0.64
2	0.64	0.68
3	0.65	0.62
4	0.64	0.66
5	0.64	0.66

- suggests regression model can predict quality on new wines with an RMS error around 0.66 or so

Outline

Modifying features

- idea: replace feature x_i with some function of x_i
- *standarizing*: replace x_i with $(x_i - b_i)/a_i$
  - b_i is (approximately) mean of x_i across data set
  - a_i is (approximately) standard deviation of x_i across data set

  (modified features have mean near zero, standard deviation near one)
  this is almost always done
- *winsorizing*: 'trim' values of x_i outside some range: replace x_i with

$$\begin{cases} 3 & x_i > 3 \\ x_i & |x_i| \leq 3 \\ -3 & x_i < -3 \end{cases}$$

  helps when there are some values that are 'outliers'

Modifying features

- *log transform*: replace x_i with $\log x_i$ (for $x_i > 0$)
  - good for features that vary over large range
  - variation for $x_i \geq 0$: replace x_i with $\log(x_i + 1)$

- Q: is transforming features a good idea?
- A: if RMS error on *validation set* is smaller

Augmenting features

- idea: augment original features with new functions of them
- *high/low values*: augment feature x_i with two new features
  - $x_i^{\mathrm{hi}} = \max\{x_i - 1, 0\}$
  - $x_i^{\mathrm{lo}} = \min\{x_i + 1, 0\}$
- *interactions*: add features of form $x_i x_j$
- custom augmented features are common in applications
  - last high/low price
  - price/earnings ratio

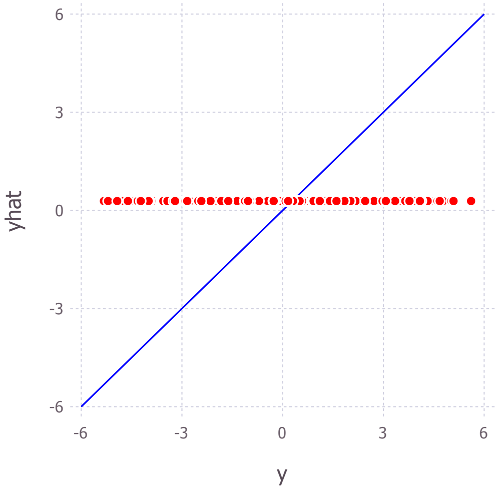
Example

- synthetic data set, with 1000 samples, 4 features
- divide into training set (800) and test set (200)
- first fit simple models, using zero or one regressor:

model	train RMS	test RMS
1	1.85	1.84
$1, x_1$	1.76	1.74
$1, x_2$	1.82	1.79
$1, x_3$	1.46	1.47
$1, x_4$	1.54	1.60

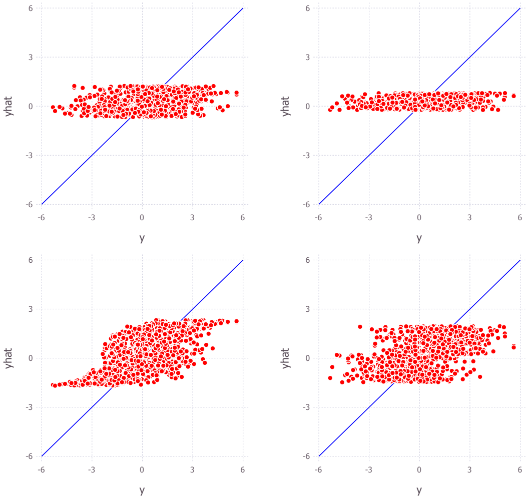
$\hat{y}$ versus y, constant model

(test set)



$\hat{y}$ versus y, single regressor models

(test set)



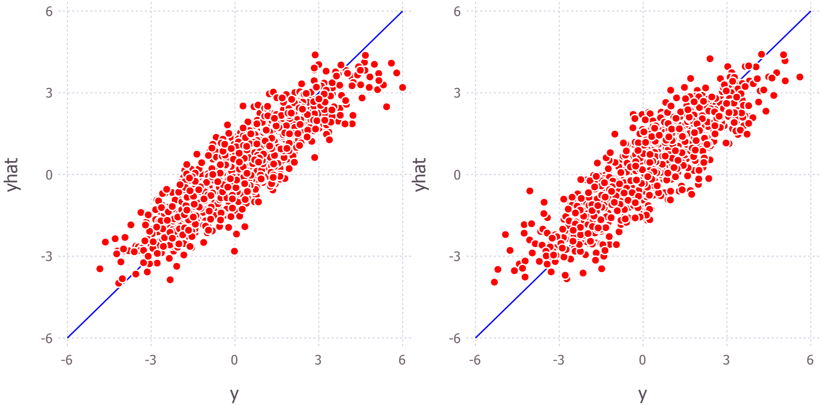
Basic regression

(regression with all features)

model	train RMS	test RMS
1	1.85	1.84
x_1	1.76	1.74
x_2	1.82	1.79
x_3	1.46	1.47
x_4	1.54	1.60
$1, x_1, x_2, x_3, x_4$	0.88	0.92

$\hat{y}$ versus y, basic regression

train and test sets



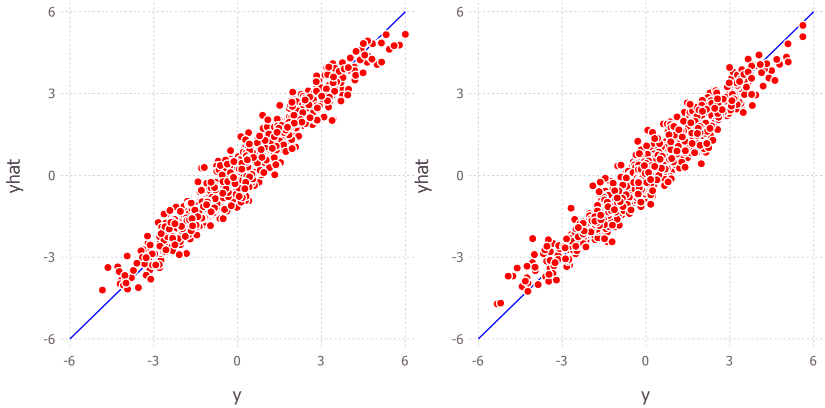
Augmenting features

- add new features $\max\{x_i - 1, 0\}$, $\min\{x_i + 1, 0\}$, $i = 1, \ldots, 4$
- augmented model has 13 features total

model	train RMS	test RMS
1	1.85	1.84
$1, x_1$	1.76	1.74
$1, x_2$	1.82	1.79
$1, x_3$	1.46	1.47
$1, x_4$	1.54	1.60
$1, x_1, x_2, x_3, x_4$	0.88	0.92
augmented	0.46	0.48

$\hat{y}$ versus y, augmented regression

with augmented features on train and test sets



Regression model with augmented features

- $\hat{y} = \beta_1 + (\beta_2 x_1 + \beta_6 \max\{x_1 - 1, 0\} + \beta_{10} \min\{x_1 + 1, 0\}) + \cdots$
- $\hat{y}$ is a sum of piecewise linear functions of x_i
- called a *generalized additive model*