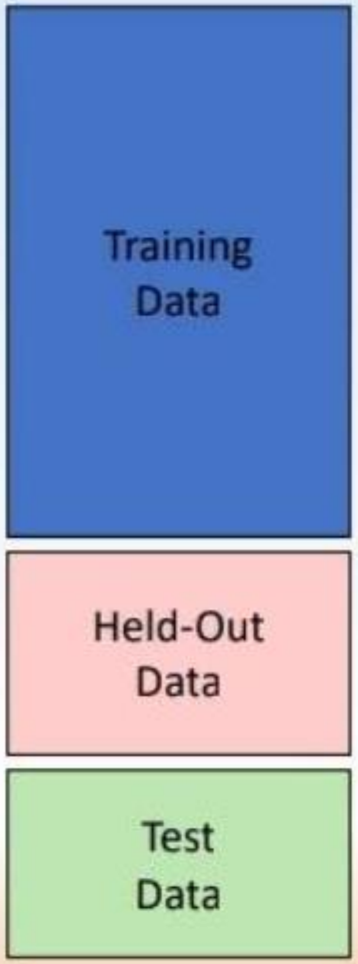
Regression 2

Welcome!

Overfitting

How do we check that we are not overfitting?

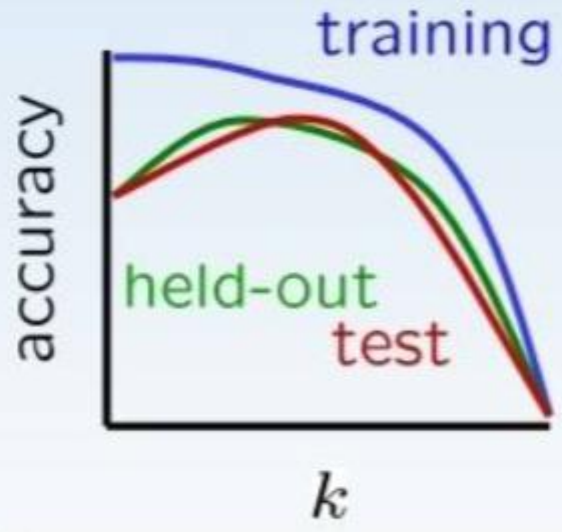
- **Data**: labeled instances, e.g. emails marked spam/ham
  - Training set
  - Held out set
  - Test set
- **Experimentation cycle**
  - Learn parameters (e.g. model probabilities) on training set
  - Compute accuracy of test set
  - Very important: **never** “peek” at the test set!
- **Evaluation**
  - Accuracy: fraction of instances predicted correctly
- **Overfitting and generalization**
  - Want a classifier which does well on *test* data
  - Overfitting: fitting the training data very closely, but not generalizing well



Recap | Tuning

Tuning on Held-Out Data

- Now we’ve got two kinds of unknowns
  - Parameters: the probabilities $P(X|Y)$, $P(Y)$
  - **Hyperparameters**: e.g. the amount / type of smoothing to do, k, α
- What should we learn where?
  - Learn **parameters** from **training data**
  - Tune **hyperparameters** on **different data**
    - Why?
  - For each value of the hyperparameters, train and test on the held-out data
  - Choose the best value and do a final test on the test data



Mean Squared Error (MSE)

- $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$

Root Mean Squared Error (RMSE)

- $\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}$

Mean Absolute Error (MAE)

- $\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$

Mean Absolute Percentage Error (MAPE)

- $\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i-\widehat{y_i}}{y_i}\right|$

The R^2 Formula

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Adjusted R-squared

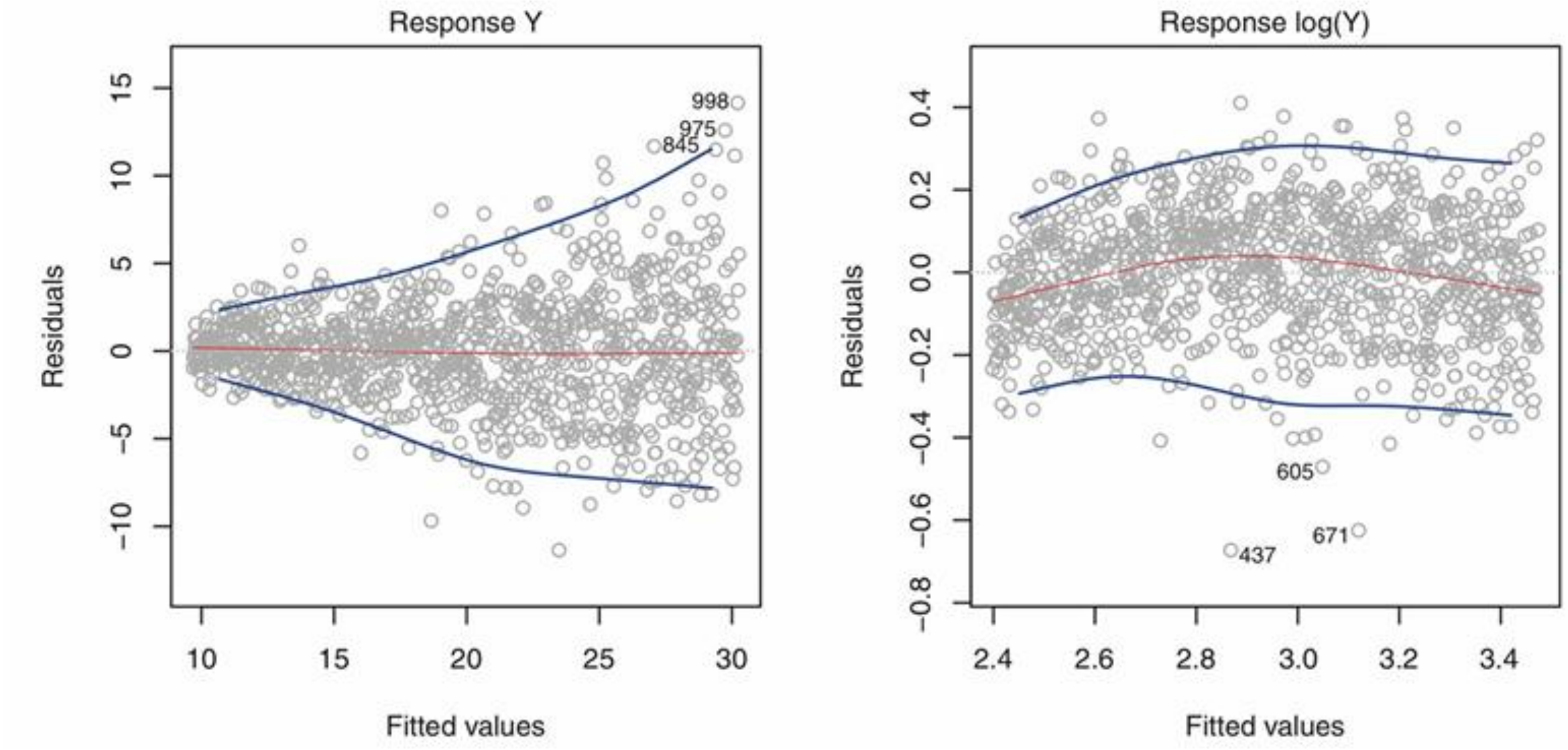
$$\overline{R^2} = 1 - (1 - R^2) \cdot \frac{n - 1}{n - \mathrm{k} - 1}$$

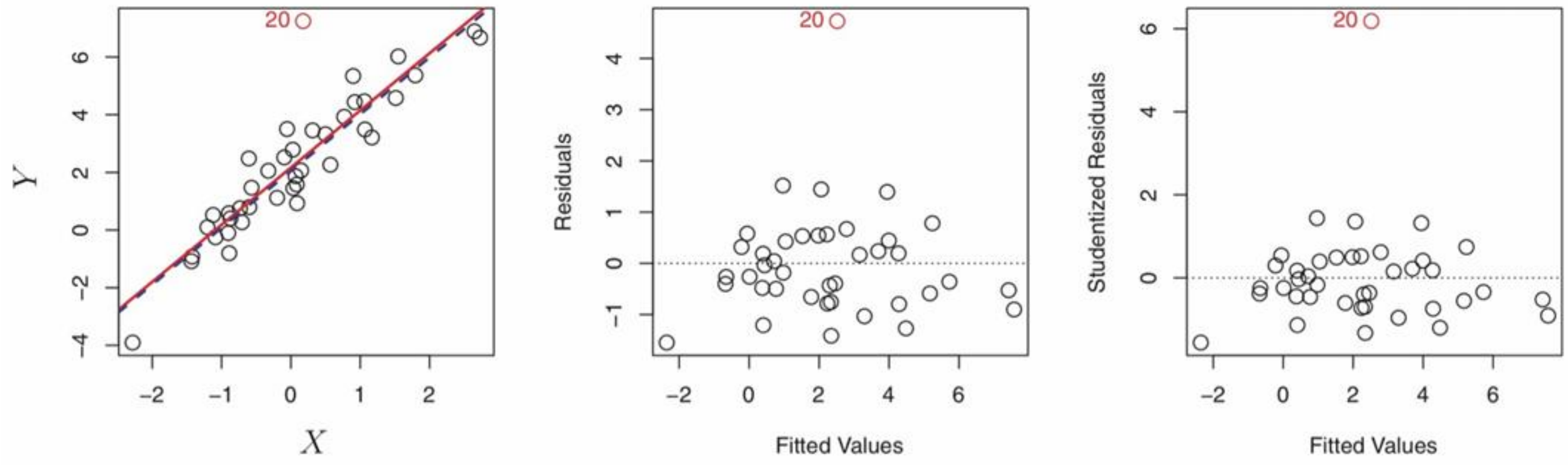
Bayesian Information Criterion (BIC)

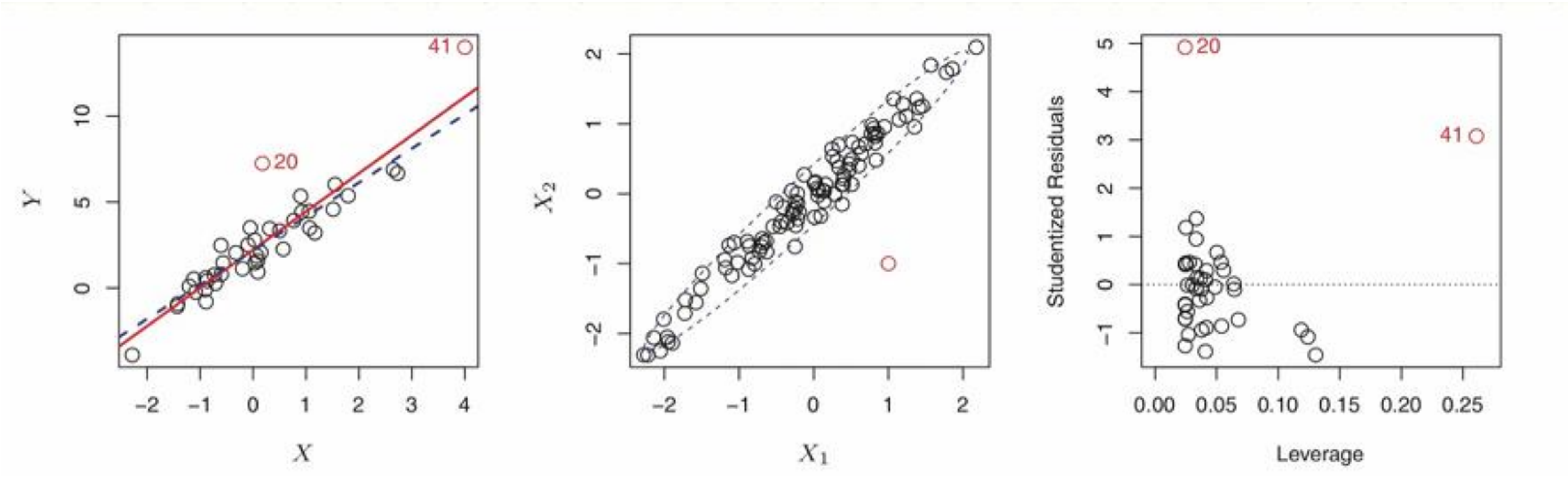
- $\mathrm{BIC} = \ln(n) \cdot k - 2\ln(\hat{L})$

- $\hat{L} = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \widehat{y_i})^2}{2\sigma^2}\right)$

Regression Problems!


Response Y
Residuals
Fitted values
998
975
845
Response log(Y)
Residuals
Fitted values
605
671
437


20
Y
X
−2
−1
0
1
2
−4
−2
0
2
4
6
Residuals
Fitted Values
−1
0
1
2
3
4
−2
0
2
4
6
Studentized Residuals
Fitted Values



41
20
Y
X
X_2
X_1
Studentized Residuals
Leverage

Gradient descent

- $w^{(t+1)} = w^{(t)} + \alpha \sum_{i=1}^{n} \left(y_i - w^{(t)^\top} x_i\right) x_i$

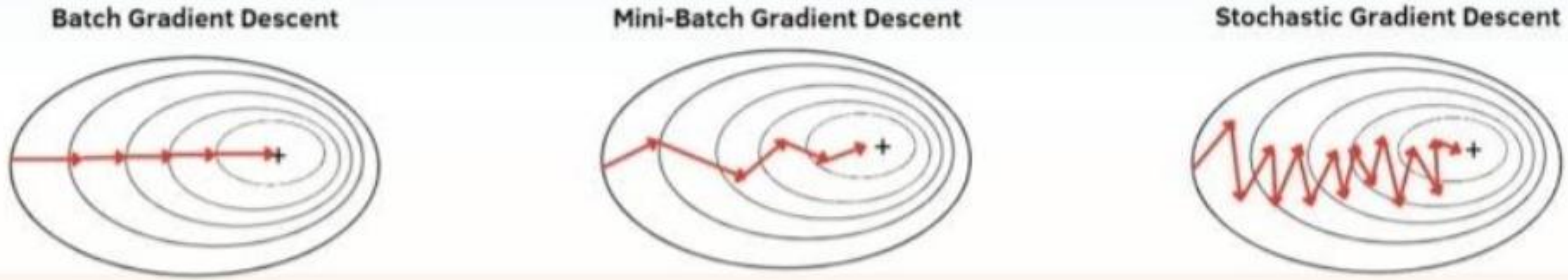
- Code example!

Gradient Descents

Example

If you have 1000 training examples and are using:

- Batch gradient descent: 1 epoch = 1 iteration (all 1000 examples used for one update)
- Mini-batch gradient descent with batch size 100: 1 epoch = 10 iterations
- Stochastic gradient descent: 1 epoch = 1000 iterations



Machine Learning | Neural Network | Training

Gradient Descents

Method	Data Used	Speed	Stability	Memory Usage	Convergence
Batch	All data	Slowest	Most stable	Highest	Smooth, deterministic
Stochastic	1 example	Fastest per step	Least stable	Lowest	Noisy, may oscillate
Mini-Batch	Small subset	Fast	Moderate	Moderate	Moderate noise