CS 584 – Machine Learning

Topic: Regression

Mustafa Bilgic

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

Motivation

- So far, we talked about classification, where the target variable Y is discrete
  - Find-S, candidate elimination, decision trees, naïve Bayes, logistic regression, neural networks
- If the target variable Y is continuous, the task is called regression
- Examples
  - Recommendations (ratings)
  - Economics and finance (credit score, house prices, stock prices, consumption, etc.)
  - Weather forecasting (temperature, humidity, wind speed, etc.)
  - and more …

Representation

- x: the input vector, the object
- r: the true value of the regression
- $f(x)$: the true underlying function
- $g(x)$: the estimated underlying function
- ϵ: the noise
- $p(r|x)$: the conditional distribution of r and x
- D: the dataset that consists of $\langle x, r \rangle$ pairs

Regression Function

- $r = f(x) + \epsilon$
- ϵ is the noise (i.e., what the model cannot capture)
- Noise can exist due to several reasons
  - The input variables are insufficient to capture everything there is to capture
  - Noisy sensors
- ϵ is typically assumed to be a zero mean Gaussian with constant variance σ^2
  - $\epsilon \sim \mathcal{N}(0, \sigma^2)$

How To Learn $f(x)$

- One typical approach that we have already seen:
  - Assume a parametric form
  - Formulate an objective function
  - Optimize (maximize or minimize) it
- Disclaimer: not all learning approaches are parametric

Maximize Conditional Log Likelihood

- Assuming $\epsilon \sim \mathcal{N}(0, \sigma^2)$, then
  - $p(r \mid x) \sim \mathcal{N}(g(x \mid w), \sigma^2)$
- Given the dataset D that has N instances and the parameter vector w, the conditional log-likelihood
  - $\text{CLL} = \sum \ln\big(p(r[d] \mid x[d])\big)$
  - $\text{CLL} = \sum \ln\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r[d]-g(x[d] \mid w))^2}{2\sigma^2}}\right)$
  - $\text{CLL} = -N \ln\big(\sqrt{2\pi}\sigma\big) - \frac{1}{2\sigma^2} \sum (r[d] - g(x[d] \mid w))^2$

Maximize CLL = Minimize Squared Loss

- $\underset{w}{\operatorname{argmax}} CLL =$
- $\underset{w}{\operatorname{argmax}} -N \ln\left(\sqrt{2\pi}\sigma\right) - \frac{1}{2\sigma^2} \sum (r[d] - g(x[d] \mid w))^2$
- $\underset{w}{\operatorname{argmin}} \sum (r[d] - g(x[d] \mid w))^2$

$g(x|w)$

- So far, we did not specify what $g(x|w)$ is
- One popular approach is to assume a linear function
- Assume in each instance x has k features
  - $x = \langle x_1, x_2, \cdots, x_k \rangle$
- Then, a polynomial of degree one linear regression is
  - $g(x|w) = w_0 + \sum_{i=1}^{k} w_i x_i$ $= \sum_{i=0}^{1} w_i x_i$

$x_0 = 1$

Gradient of the Squared Loss

- See OneNote

Regression Examples

- See OneNote

REGULARIZATION

- L_2 regularization
  - Minimize squared-loss + L_2 penalty
  - Also called Ridge regression
- L_1 regularization
  - Minimize squared-loss + L_1 penalty
  - Also called Lasso regression
- See OneNote for derivation

POLYNOMIAL REGRESSION – ARBITRARY DEGREE

- Simply change the input representation by adding new features that correspond to powers and products
- See https://scikit-learn.org/stable/modules/preprocessing.html#polynomial-features

r 2 2
(3 5
(10 9

RSE

$$\frac{(2-2)^2+(3-5)^2+(10-9)^2}{(2-5)^2+(3-5)^2+(10-5)^2}$$

- $RSE = \frac{\sum(r-g)^2}{\sum(r-\bar{r})^2}$
- RSE = closer to 1, if our prediction is as good/bad as predicting the mean all the time
- RSE = closer to 0 means we have a better fit
- Coefficient of determination
  - $R^2 = 1 - RSE$

Other Regression Approaches

- There are other approaches besides linear regression, such as
  - Decision tree regression
  - Support vector regression
  - Neural networks
  - …

liner activation

Scikit-Learn

- Least squares
  - http://scikit-learn.org/stable/modules/linear_model.html#ordinary-least-squares
  - https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- Ridge
  - http://scikit-learn.org/stable/modules/linear_model.html#ridge-regression
  - https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
- Lasso
  - http://scikit-learn.org/stable/modules/linear_model.html#lasso
  - https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

TensorFlow

- https://www.tensorflow.org/tutorials/keras/regression