4. Linear Regression

Mittwoch, 13. November 2019 11:31

In-class Exercises

Problem 7: Assume that we are given a dataset, where each sample x_i and regression target y_i is generated according to the following process

$$x_i \sim \text{Uniform}(-10, 10)$$
$$y_i = ax_i^3 + bx_i^2 + cx_i + d + \epsilon_i, \quad \text{where} \quad \epsilon_i \sim \mathcal{N}(0,1) \quad \text{and} \quad a, b, c, d \in \mathbb{R}.$$

The 3 regression algorithms below are applied to the given data. Your task is to say what the bias and variance of these models are (low or high). Provide a 1-2 sentence explanation to each of your answers.

a) Linear regression

b) Polynomial regression with degree 3

c) Polynomial regression with degree 10

Bias: How well can model fit data?

Variance: How much does prediction change when data changes (a little)?

Bias-variance trade-off

a) Bias high, low variance
Straight line is underfitting the data

b) Bias low, variance low
Same model as generating process

c) Bias low, variance high
Overfitting the data, model too powerful

Problem 8: Given is a training set consisting of samples $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ with respective regression targets $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$.

Alice fits a linear regression model $f(\boldsymbol{x}_i) = \boldsymbol{w}^T \boldsymbol{x}_i$ to the dataset using the closed form solution for linear regression (normal equations).

Bob has heard that by transforming the inputs $\boldsymbol{x}_i$ with a vector-valued function $\boldsymbol{\phi}$, he can fit an alternative function, $g(\boldsymbol{x}_i) = \boldsymbol{v}^T \boldsymbol{\phi}(\boldsymbol{x}_i)$, using the same procedure (solving the normal equations). He decides to use a linear transformation $\boldsymbol{\phi}(\boldsymbol{x}_i) = \boldsymbol{A}^T \boldsymbol{x}_i$, where $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ has full rank.

a) Show that Bob's procedure will fit the same function as Alice's original procedure, that is $f(\boldsymbol{x}) = g(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^D$ (given that $\boldsymbol{w}$ and $\boldsymbol{v}$ minimize the training set error).

$$g(\vec{x}) = \vec{v}^T \phi(\vec{x}) = \vec{v}^T (A^T \vec{x}) = \underbrace{(A\vec{v})^T}_{\vec{w}} \vec{x} = \vec{w}^T \vec{x}$$

$$\vec{v} = A^{-1} \vec{w}$$

b) Can Bob's procedure lead to a lower training set error than Alice's if the matrix $\boldsymbol{A}$ is not invertible? Explain your answer.

A not invertible, so any 2 vector could be transformed to the same point, so Bob cannot assign different values to these $\Rightarrow$ Bob's model atmost as powerful as Alice's. $\Rightarrow$ NO

Problem 9: See Jupyter notebook `inclass_04_notebook.ipynb`.

Homework

1 Least squares regression

Problem 1: Let's assume we have a dataset where each datapoint, $(\boldsymbol{x}_i, y_i)$ is weighted by a scalar factor which we will call t_i. We will assume that $t_i > 0$ for all i. This makes the sum of squares error function look like the following:

$$E_{\text{weighted}}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N} t_i \left[\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) - y_i\right]^2$$

Find the equation for the value of $\boldsymbol{w}$ that minimizes this error function.

Furthermore, explain how this weighting factor, t_i, can be interpreted in terms of
1) the variance of the noise on the data and
2) data points for which there are exact copies in the dataset.

$$E = \frac{1}{2}(\Phi\vec{w} - \vec{y})^T T (\Phi\vec{w} - \vec{y})$$

$T = \text{diag}(t_0, \ldots, t_n)$

$$w^* = (\Phi^T T \Phi)^{-1}\Phi^T T\vec{y}$$

$$p(\vec{y}|\Phi, \vec{w}, \beta) = \prod_i \mathcal{N}(y_i|\vec{w}^T\phi(\vec{x}_i), \beta^{-1})$$

β ← precision

$$\ln p(\vec{y}|\Phi, \vec{w}, \beta) = \sum_i \ln \mathcal{N}(y_i|\vec{w}^T\phi(\vec{x}_i), \beta^{-1}) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_{LS}(\vec{w})$$

$$\beta E_{LS}(\vec{w}) = \frac{1}{2}\sum_i \beta(\vec{w}^T\phi(\vec{x}_i) - y_i)^2$$

$\beta \to t_i$

$t_i \to$ precision per sample!

t_i is the effective number of replicated observations

2 Ridge regression

Problem 2: Show that the following holds: The ridge regression estimates can be obtained by ordinary least squares regression on an augmented dataset: Augment the design matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N\times M}$ with M additional rows $\sqrt{\lambda}\boldsymbol{I}_{M\times M}$ and augment $\boldsymbol{y}$ with M zeros.

Problem 3: Derive the closed form solution for ridge regression error function

$$E_{\text{ridge}}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) - y_i)^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

Additionally, discuss the scenario when the number of training samples N is smaller than the number of basis functions M. What computational issues arise in this case? How does regularization address them?

3 Multi-output linear regression

Problem 4: In class, we only considered functions of the form $f : \mathbb{R}^n \to \mathbb{R}$. What about the general case of $f : \mathbb{R}^n \to \mathbb{R}^m$? For linear regression with multiple outputs, write down the loglikelihood formulation and derive the MLE of the parameters.

$\vec{y}_i \sim \mathcal{N}(W\vec{x}_i, \Sigma)$ ($W\phi(\vec{x}_i)$)

$p(Y|W,\Sigma) = \prod_i \mathcal{N}(W\vec{x}_i, \Sigma)$

$\ln p(Y|W,\Sigma) = \text{const} + \sum_i \frac{1}{2}(\vec{y}_i - W\vec{x}_i)^T\Sigma^{-1}(\vec{y}_i - W\vec{x}_i) =$

Cholesky decomposition: $\Sigma = LL^T$, $\Sigma^{-1} = L^{-T}L^{-1}$ (since Σ is symmetric and PSD)

$= C + \sum_i \frac{1}{2}(L^{-1}(\vec{y}_i - W\vec{x}_i))^T(L^{-1}(\vec{y}_i - W\vec{x}_i)) =$

$L^{-1}(\vec{y}_i - W\vec{x}_i) = L^{-1}\vec{y}_i - L^{-1}W\vec{x}_i = \hat{y}_i - \hat{W}\vec{x}_i$

$= C + \sum_i \frac{1}{2}(\hat{y}_i - \hat{W}\vec{x}_i)^T(\hat{y}_i - \hat{W}\vec{x}_i) =$

$= C + \frac{1}{2}\text{Tr}(\hat{Y} - X\hat{W})^T(\hat{Y} - X\hat{W})$

$\hat{W}_{MLE} = (X^TX)^{-1}X^T\hat{Y}$, $W_{MLE} = L\hat{W}_{MLE}$

$\text{Tr}[(Y-XW)^T(Y-XW)] =$

$= \sum_j [(Y-XW)^T(Y-XW)]_{jj} =$

$= \sum_j \sum_i (Y-XW)_{ij}(Y-XW)_{ij} =$

$= \sum_j \sum_i (Y_{ij} - [XW]_{ij})^2 =$

samples, dimensionality of $\vec{y}$

$= \sum_i (\vec{y}_i - \vec{x}_i^T W)^T(\vec{y}_i - \vec{x}_i^T W)$

4 Comparison of Linear Regression Models

Problem 5: We want to perform regression on a dataset consisting of N samples $\boldsymbol{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ and $\boldsymbol{y} \in \mathbb{R}^N$).

Assume that we have fitted an L_2-regularized linear regression model and obtained the optimal weight vector $\boldsymbol{w}^* \in \mathbb{R}^D$ as

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}^T\boldsymbol{x}_i - y_i)^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

Note that there is no bias term.

Now, assume that we obtained a new data matrix $\boldsymbol{X}_{new}$ by scaling all samples by the same positive factor $a \in (0, \infty)$. That is, $\boldsymbol{X}_{new} = a\boldsymbol{X}$ (and respectively $\boldsymbol{x}_i^{new} = a\boldsymbol{x}_i$).

a) Find the weight vector $\boldsymbol{w}_{new}$ that will produce the same predictions on $\boldsymbol{X}_{new}$ as $\boldsymbol{w}^*$ produces on $\boldsymbol{X}$.

$\hat{y} = \vec{w}^T\vec{x}_i = \vec{w}_{new}^T\vec{x}_i^{new} = \vec{w}_{new}^T a\vec{x}_i$

$\Leftrightarrow \vec{w}_{new} = \frac{\vec{w}}{a}$

b) Find the regularization factor $\lambda_{new} \in \mathbb{R}$, such that the solution $\boldsymbol{w}^*_{new}$ of the new L_2-regularized linear regression problem

$$\boldsymbol{w}^*_{new} = \arg\min_{\boldsymbol{w}} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{w}^T \boldsymbol{X}^{new}_i - y_i)^2 + \frac{\lambda_{new}}{2} \boldsymbol{w}^T \boldsymbol{w}$$

will produce the same predictions on $\boldsymbol{X}_{new}$ as $\boldsymbol{w}^*$ produces on $\boldsymbol{X}$.

Provide a mathematical justification for your answer.

I) $w^* = (X^T X + \lambda I)^{-1} X^T \vec{y}$

$w^*_{new} = (X^T_{new} X_{new} + \lambda_{new} I)^{-1} X^T_{new} \vec{y} =$

$= a (a^2 X^T X + \lambda_{new} I)^{-1} X^T \vec{y} =$

$= \frac{1}{a} (X^T X + \frac{\lambda_{new}}{a^2} I)^{-1} X^T y$

w^*, because $w^*_{new} = \frac{w^*}{a}$

$\lambda = \frac{\lambda_{new}}{a^2} \Leftrightarrow \lambda_{new} = a^2 \lambda$

II) $w^*_{new} \overset{!}{=} \frac{w^*}{a} = \frac{1}{a} \arg\min_{\vec{w}} \frac{1}{2} \sum_i (\vec{w}^T \vec{x}_i - y_i)^2 + \frac{\lambda}{2} \vec{w}^T \vec{w} =$

$= \frac{1}{a} \arg\min_{\vec{w}} \frac{1}{2} \sum_i \left(\frac{\vec{w}^T}{a} \cdot a\vec{x}_i - y_i \right)^2 + \frac{a^2 \lambda}{2} \frac{\vec{w}^T}{a} \frac{\vec{w}}{a} =$

result: $\vec{w}$

$= \frac{a}{a} \arg\min_{\vec{w}_{new} = \frac{\vec{w}}{a}} \frac{1}{2} \sum_i (\vec{w}^T_{new} \vec{x}^{new}_i - y_i)^2 + \frac{a^2 \lambda}{2} \vec{w}^T_{new} \vec{w}_{new}$

result: $\frac{\vec{w}}{a}$

$\lambda_{new} = a^2 \lambda$

5 Programming Task: Least squares regression

Problem 6: Load the notebook `04_homework_linear_regression.ipynb` from Piazza. Fill in the missing code and run the notebook. Convert the evaluated notebook to pdf and add it to the printout of your homework.

Note: We suggest that you use Anaconda for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks and how to convert them to other formats, consult the Jupyter documentation and nbconvert documentation.