MANCHESTER 1824
The University of Manchester

COMP24112: Machine Learning

Chapter 6: Training and Optimisation, I

Dr. Tingting Mu

Email: tingting.mu@manchester.ac.uk

Content

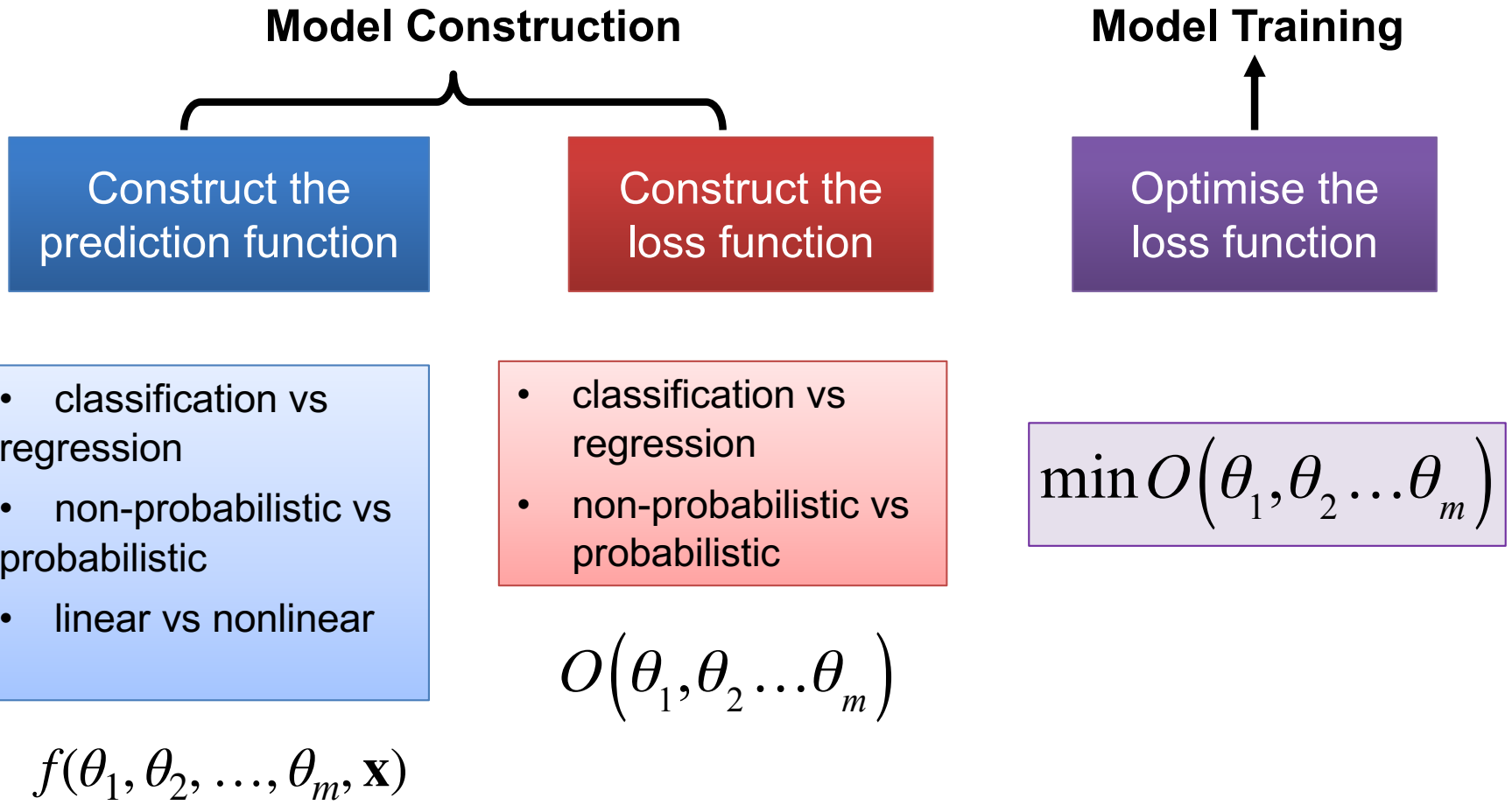
MANCHESTER 1824
The University of Manchester

- Training by optimality condition
- Linear least squares solution

Revisit Machine Learning Pipeline

The University of Manchester

- The pipeline for building a machine learning system.



Optimality Condition

MANCHESTER 1824
The University of Manchester

- Goal: To optimise the loss function, e.g., $\min O\left(\theta_1, \theta_2 \ldots \theta_m\right)$

- Optimisation theory tells us that the optimal $\theta = \left[\theta_1, \theta_2 \ldots \theta_m\right]$ should satisfy the following condition:

Each partial derivative of the objective function with respect to each input variable should be set as zero!

$$\frac{\partial O}{\partial \theta_1} = 0, \quad \frac{\partial O}{\partial \theta_2} = 0, \ldots \frac{\partial O}{\partial \theta_m} = 0$$

See "Maths Knowledge Overview" on derivative, partial derivative and gradient.

MANCHESTER 1824
The University of Manchester

Case study: (regularised) linear least square solution

Linear Least Squares: Single-output

- Apply the linear model to training samples: $\hat{y}_i = \mathbf{w}^T \tilde{x}_i$, *i=1,2…N*
- Sum-of-squares error loss:

$$O(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\left(\mathbf{w}^T\tilde{\mathbf{x}}_i - y_i\right)^2 = \frac{1}{2}\left\|\tilde{\mathbf{X}}\mathbf{w} - \mathbf{y}\right\|_2^2$$

An expanded feature vector for the i-th sample:

$$\tilde{\mathbf{x}}_i = \left[1, x_{i1}, x_{i2}, x_{i3} \ldots x_{id}\right]$$

An expanded feature matrix for N samples:

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \ldots & x_{Nd} \end{bmatrix}$$

A label vector for N samples:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

The coefficient vector:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Linear Least Squares: Multi-output

- Apply the linear model to training samples: $\hat{\mathbf{y}}_i = \mathbf{W}^T \tilde{\mathbf{x}}_i$, *i=1,2…N*
- Sum-of-squares error loss:

$$O(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{c} \left(\hat{y}_{ij} - y_{ij} \right)^2 = \frac{1}{2} \left\| \tilde{\mathbf{X}}\mathbf{W} - \mathbf{Y} \right\|_F^2$$

An expanded feature vector for the i-th sample:

$$\tilde{\mathbf{x}}_i = \left[1, x_{i1}, x_{i2}, x_{i3} \ldots x_{id}\right]$$

An expanded feature matrix for N samples:

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \ldots & x_{Nd} \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & y_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nc} \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} w_{01} & w_{02} & \cdots & w_{0c} \\ w_{11} & w_{12} & \cdots & w_{1c} \\ w_{21} & w_{22} & \cdots & w_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1} & w_{d2} & \cdots & w_{dc} \end{bmatrix}$$

Linear Least Squares: N>d

- If the minimal error of zero can be achieved, it is equivalent to solving the following linear systems:
  - Single-output case: $\tilde{\mathbf{X}}\mathbf{w} = \mathbf{y}$
  - Multi-output case: $\tilde{\mathbf{X}}\mathbf{W} = \mathbf{Y}$
- When there are **more training samples than input features** (N>d), the system is overdetermined.
- If $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ is invertible, there is a unique solution.
- Usually, rather than solving manually the linear equations. we use **normal equations** to find this solution.

Normal Equations: Single-output

The University of Manchester

- Gradient of the error function:

$$\nabla O(\mathbf{w}) = \begin{bmatrix} \frac{\partial O}{\partial w_1} \\ \frac{\partial O}{\partial w_2} \\ \vdots \\ \frac{\partial O}{\partial w_d} \end{bmatrix} = \tilde{\mathbf{X}}^T \tilde{\mathbf{X}} \mathbf{w} - \tilde{\mathbf{X}}^T \mathbf{Y}$$

See "Notes on linear and quadratic equations" on how to derive this.

- Re-express the gradient in an equivalent form:

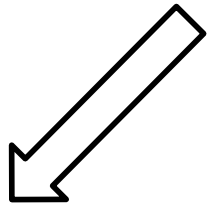
$$\nabla O(\mathbf{w}) = \sum_{i=1}^{N} \left(\tilde{\mathbf{x}}_i^T \mathbf{w} - y_i \right) \tilde{\mathbf{x}}_i$$

It is interesting to see this term carries the error information for each training sample.

Normal Equations: Single-output

- Optimality condition: set gradient to zero!

$$\nabla O(\mathbf{w}) = 0$$



$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{w} - \tilde{\mathbf{X}}^T\mathbf{Y} = 0 \Rightarrow \mathbf{w} = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{Y}$$

This equation is known as the **normal equation** for the least squares problem.

Normal Equations: Multi-output

- Gradient of error function.

$$\nabla O\left(\mathbf{W}\right) = \mathbf{X}^T\tilde{\mathbf{X}}\mathbf{W} - \tilde{\mathbf{X}}^T\mathbf{Y}$$

- Set gradient to zero to get optimal solution:

$$\mathbf{W} = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{Y}$$

Linear Least Squares: N<d

MANCHESTER 1824
The University of Manchester

- If the minimal error of zero can be achieved, it is equivalent to solving the following linear systems:
  - Single-output case: $\tilde{\mathbf{X}}\mathbf{w} = \mathbf{y}$
  - Multi-output case: $\tilde{\mathbf{X}}\mathbf{W} = \mathbf{Y}$
- When there are **less training samples than input features** (N<d), the system is underdetermined.
- When $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ is invertible, there are infinitely many solutions, and we want to use the one with minimum norm, which is
  - Single-output case: $\mathbf{w} = \tilde{\mathbf{X}}^T\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}\mathbf{y}$
  - Multi-output case: $\mathbf{W} = \tilde{\mathbf{X}}^T\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}\mathbf{Y}$

Normal Equations Implementation

MANCHESTER 1824
The University of Manchester

- The quantity $\left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T$ and $\tilde{\mathbf{X}}^T\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}$ are called Moore-Penrose inverse (or called pseudoinverse) of $\tilde{\mathbf{X}}$.
- Regardless the sample size and feature size, we can unify the solution.
  - Single-output case: $\mathbf{w} = \tilde{\mathbf{X}}^{\dagger}\mathbf{y}$
  - Multi-output case: $\mathbf{W} = \tilde{\mathbf{X}}^{\dagger}\mathbf{Y}$

More details on normal equations can be found in “Notes on Linear Least Squares Model”.

- You can calculate the pseudo inverse of a matrix using readily implemented library.

  numpy.linalg.pinv

Summary: Linear Least Squares Solution

The University of Manchester

Goal: To predict *c* target variables from *d* input variables using a linear least squares model based on *N* training samples.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix} \quad \tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}$$

Each row of the maxtrix $\mathbf{X}$ stores the d input variables of a training sample: $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots x_{id}]^T$.

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & y_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nc} \end{bmatrix}$$

Each row of the maxtrix $\mathbf{Y}$ stores the c output variables of a training sample $\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots y_{ic}]^T$.

Training a linear model by computing

$\mathbf{W} = \tilde{\mathbf{X}}^{+}\mathbf{Y}$

$\tilde{\mathbf{X}}$: add ones before the first column of $\mathbf{X}$

$$\mathbf{W} = \begin{bmatrix} w_{01} & w_{02} & \cdots & w_{0c} \\ w_{11} & w_{12} & \cdots & w_{1c} \\ w_{21} & w_{22} & \cdots & w_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1} & w_{d2} & \cdots & w_{dc} \end{bmatrix}$$

prediction function

$\hat{\mathbf{y}} = \mathbf{W}^T\tilde{\mathbf{x}}$

History: Least Squares Method

- This is known as **least squares fitting**, or **ordinal least squares solution**.
- Adrien-Marie Legendre (18/09/1752 - 10/01/1833, a French mathematician) proposes the "méthode des moindres carrés", known in English as the least squares method *(Adrien-Marie Legendre, Nouvelles méthodes pour la détermination des orbites des comètes, Paris: Firmin Didot. p. viii, 1805).*

Watercolor caricature of Adrien-Marie Legendre, by French artist Julien-Leopold Boilly, 1820.

l_2-Regularised Least Squares

- Let's look at the l_2-regularised least square model in single-output case:

$$\min O_\lambda(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \mathbf{w}^T\tilde{\mathbf{x}}_i\right)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- Set gradient to zero. The optimal weight vector is computed by

$$\mathbf{w} = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{Y}$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \text{ (identity matrix)}$$

single-output

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

multi-output

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & y_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nc} \end{bmatrix}$$

Hyperparameter λ

The University of Manchester

l_2-regularised least sqaures solution:

$$\mathbf{W} = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{Y}$$

- Hyperparameter: λ >0
  - It controls the balance between the data dependent error function and the regularisation term.
  - To distinguish it from those model parameters to be optimised during the learning process (e.g., ***W***), λ is referred to as a hyperparameter.
- Training, validation and testing:
  - Training set: A set of samples used for learning, that is to optimise the model parameters.
  - Validation set: A set of samples used to select (tune) the hyperparameters.
  - Test set: A set of samples used only to assess the performance of a fully-specified classifier (trained with selected hyperparameters).