

Machine Learning

Assignment 2

CLO1

Ida Bagus Dwi Satria Kusuma
1301140297

October 20, 2017

1. **(5 points)** Explain about univariate linear regression.

Univariate Linear Regression adalah salah satu jenis regresi linear, di mana variable bebas yang terlibat hanya satu.[1] Regresi ini dimodelkan menggunakan persamaan[2] :

$$f(x) = \omega_1 x + \omega_0 \quad (1)$$

dimana w_0 dan w_1 adalah parameter dari model yang dinamakan koefisien regresi (**regression coefficients**). Salah satu cara untuk mencari nilai ω_0 dan ω_1 adalah dengan menggunakan metode ***Least Square*** (selain *least square*, untuk mencari kedua nilai ini dapat menggunakan cara lain seperti *maximum likelihood* atau algoritma *gradient descent*[1]), yang mencoba untuk menemukan nilai parameter (ω_0 dan ω_1) yang meminimalkan *sum of the squared error* [2]

$$SSE = \sum_N^{i=1} [y_i - f(x_i)]^2 = \sum_N^{i=1} [y_i - \omega_1 x - \omega_0]^2 \quad (2)$$

yang juga dikenal sebagai **residual sum of squares**[2].

Masalah optimasi ini dapat diselesaikan dengan mengambil turunan parsial dari E yang berhubungan dengan ω_0 dan ω_1 , dan menyelesaikan sistem persamaan linear yang sesuai [2]

$$\frac{\partial E}{\partial \omega} = -2 \sum_N^{i=1} [y_i - \omega_1 x - \omega_0] = 0 \quad (3)$$

$$\frac{\partial E}{\partial \omega} = -2 \sum_N^{i=1} [y_i - \omega_1 x - \omega_0] x_i = 0 \quad (4)$$

Persamaan tersebut dapat diringkas menggunakan persamaan matriks berikut, yang juga disebut sebagai **normal equation** [2]:

$$\begin{pmatrix} N & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix} \begin{pmatrix} \omega_0 \\ \omega_1 \end{pmatrix} = \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix}$$

Normal equation ini kemudian dapat diekspresikan sebagai :

$$\hat{\omega}_0 = \bar{y} - \hat{\omega} \bar{x} \quad (5)$$

$$\hat{\omega}_1 = \frac{\sigma_{xy}}{\sigma_{xx}} \quad (6)$$

dimana $\hat{x} = \sum_i x_i / N$, $\hat{y} = \sum_i y_i / N$, dan

$$\sigma_{xy} = \sum_i (x_i - \bar{x})(y_i - \bar{y})$$

$$\sigma_{xx} = \sum_i (x_i - \bar{x})^2$$

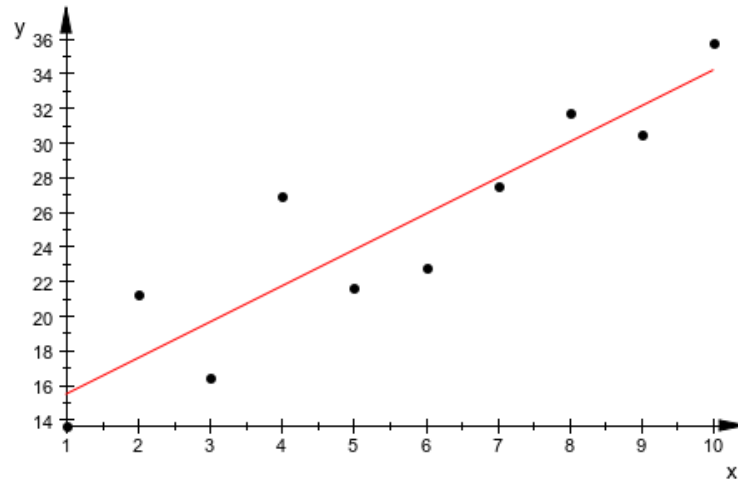
$$\sigma_{yy} = \sum_i (y_i - \bar{y})^2$$

Jadi, model linear yang dihasilkan dari *minimum squared error* adalah [2]

$$f(x) = \bar{y} + \frac{\sigma_{xy}}{\sigma_{xx}} [x - \bar{x}]$$

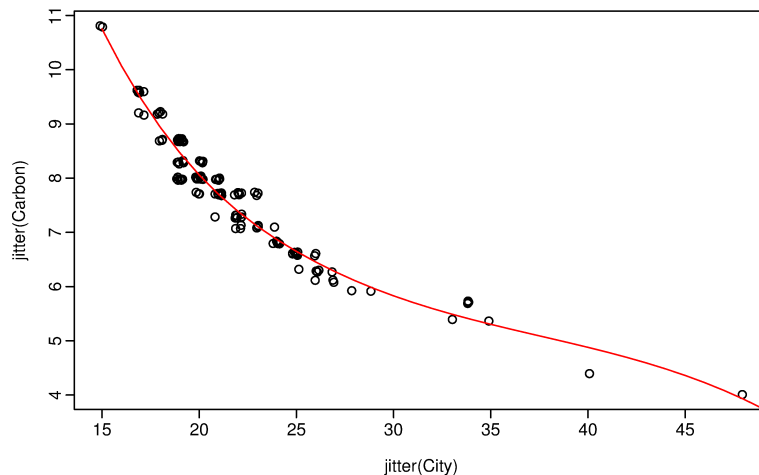
Jadi kesimpulannya, metode *least square* adalah pendekatan sistematis untuk mencocokkan model linear sebagai respon terhadap variabel y

dengan cara meminimalkan *squared error* antara nilai y yang sebenarnya dan yang telah diestimasi.



2. Explain about univariate non-linear regression.

Univariate non-linear regression adalah salah satu jenis regresi di mana hanya terdapat satu variabel bebas, namun berbeda dengan regresi linear, pada regresi jenis ini, hubungan antara variabel dependent dan variabel independennya (variable bebas) tidak linear. Selain itu, menurut [4], non-linear regression berarti '*the regression function is not simultaneously linear in the unknown parameter β_i* '



Salah satu contoh model regresi ini adalah model Michaelis-Menten

untuk kinetik enzim[3]

$$v = \frac{V_{max} [S]}{K_m + [S]}$$

yang dapat dituliskan menjadi

$$f(x, \beta) = \frac{\beta_1 x}{\beta_2 + x}$$

di mana β_1 adalah parameter V_{max} , β_2 adalah parameter K_m dan $[S]$ adalah variabel independent, x . Fungsi ini termasuk model non-linear karena fungsi ini tidak dapat diekspresikan sebagai kombinasi linear dari dua β . Contoh lain dari fungsi nonlinear adalah fungsi eksponensial, fungsi logaritmik, fungsi trigonometri, fungsi kuadrat, dan fungsi Gaussian[3].

3. **(5 points)** Explain about multivariate linear regression

Multivariate linear regression adalah regresi linear yang variabel bebas yang terlibat tidak hanya satu, melainkan beberapa variabel bebas. [3] Karena input yang digunakan lebih dari satu dimensi, maka model regresi linear ini berbeda dengan model regresi linear univariat, yaitu

$$h_w(x) = w_0 + w_1x_1 + w_2x_2 + \dots + w_mx_m$$

$$h_w(x) = w_0 + \sum_{i=0}^m w_ix_i$$

di mana w juga merupakan nilai yang akan dicari sedemikian sehingga nilai w menjadi optimal dan x merupakan variabel bebas atau input. Untuk mencari nilai w , kita dapat menggunakan cara yang sama dengan regresi linear univariat, yaitu dengan menggunakan metode *least square*, *maximum likelihood*, atau algoritma *gradient descent*.

Selain itu, untuk mendefinisikan regresi linear multivariat, juga dapat menggunakan bentuk persamaan lain, menurut [5], yaitu

$$y = X\omega + \epsilon$$

di mana *residual* (sis) $\epsilon_i = y_i - \omega \cdot x_i$ mengindikasikan error yang dihasilkan oleh vektor berat w pada titik (x_i, y_i) . Cara untuk menemukan besar ω adalah dengan meminimalkan *sum of squared residual*,

$$\sum_{i=1}^n \epsilon_i^2 = \|\epsilon\|_2^2$$

4. **(5 points)** Explain about multivariate non-linear regression

Multivariate non-linear regression adalah regresi linear yang melibatkan variabel bebas lebih dari satu (sama seperti regresi linear multivariate), namun berbeda dengan regresi linear multivariat, bidang hasilnya tidak linear (lurus).

Menurut [5], kita dapat memodelkan fungsi nonlinear dengan membuat input matriksnya. Misal untuk

$$y_i = f(x_i) = \omega_0 + \omega_1 x_{i1} + \omega_2 x_{i2} + \omega_3 x_{i3} + \omega_4 x_{i1} x_{i2}$$

kita membuat matrix inputnya menjadi

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{11}x_{12} \\ 1 & x_{21} & x_{22} & x_{21}x_{22} \\ 1 & x_{31} & x_{32} & x_{31}x_{13} \\ 1 & x_{41} & x_{42} & x_{41}x_{42} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \text{ dan } \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \end{pmatrix}$$

5. **(5 points)** Explain the advantages of Multi-Layer Perceptron (MLP).

MLP adalah algoritma yang sangat baik untuk digunakan untuk regresi dan *mapping*. Ia dapat digunakan untuk memetakan sinyal input dengan dimensi-N ke sinyal output dimensi-M, dan pemetaan ini juga bisa non-linear[6]

Selain itu, MLP juga memiliki kemampuan untuk memodelkan relasi yang lebih kompleks antara input dan output[2] dengan akurasi yang tinggi[2] (akurasi juga bergantung pada fungsi aktivasi yang digunakan).

6. **(5 points)** Explain the disadvantages of MLP.

Karena memiliki beberapa perceptron, MLP memerlukan waktu *training* yang lebih lama dan dilakukan berulang kali sampai jumlah tertentu untuk mendapatkan akurasi yang lebih baik, apalagi jika *hidden-node* atau *hidden-layer*nya besar/banyak.

7. **(5 points)** Explain the similarities between MLP and Support Vector Machine (SVM)

Dalam mengklasifikasikan data, MLP dan SVM sama-sama mencari garis yang memisahkan data yang disebut *decision-boundary*[2].

8. **(5 points)** Explain the differences between MLP and SVM.

Dalam menentukan *decision-boundary*, MLP akan membuat *decision-boundary* secara acak, kemudian seiring berjalannya learning, MLP akan mengupdate *decision-boundary*-nya dengan mengupdate bias atau *synaptic weight*-nya[2].

Sedangkan SVM menggunakan perhitungan tertentu, bahkan akan memproses data terlebih dahulu menggunakan fungsi kernel (jika diperlukan atau jika data tidak dapat dipisahkan secara linear) untuk kemudian dipisahkan menggunakan *decision-boundary* dengan *learning rate* tertentu[2].

9. **(5 points)** Describe the influences of number of neurons to the MLP model.

Jumlah neuron dalam tiap *layer* berbeda-beda. Pada *input layer*, jumlah neuron harus sesuai dengan jumlah atribut yang akan diproses. Sedangkan pada *hidden layer*, jumlah neuron ≥ 1 . Pada layer ini, jumlah neuron yang terbaik dicari dengan mencoba mengganti jumlah neuron (trial and error). Yang terakhir, pada *output layer*, jumlah neuron disesuaikan dengan banyaknya kelas.[2][7]

10. **(5 points)** Describe the influences of type of activation function to the MLP model.

Tanpa menggunakan fungsi aktivasi, *ouput signalnya* adalah fungsi linear sederhana.[8] Selain itu, fungsi aktivasi penting untuk proses *learning* dan digunakan untuk mengubah nilai *input* menjadi nilai tertentu, yang kemudian digunakan untuk mengaktifkan atau menonaktifkan neuron.[2]

Pemilihan fungsi aktivasi bisa dengan cara mencoba sesuai kebutuhan. namun dari beberapa literatur, untuk *hidden layer*, sebaiknya menggunakan ReLu. Dan jika ternyata model kita memiliki banyak *dead neuron* selama proses *training*, maka sebaiknya kita menggunakan fungsi ReLu atau Maxout.

11. **(5 points)** Describe the influences of learning rate to the MLP model.

Pada MLP, *learning rate* memiliki pengaruh pada perubahan *decision-boundary*. Semakin besar nilai *learning rate* maka semakin besar perubahan *decision-boundary* dan sebaliknya. *Learning rate* yang kecil, mengakibatkan sebuah model membutuhkan proses *learning* yang lebih banyak atau epoch yang lebih banyak, untuk mendapatkan *decision-boundary* yang optimal. Sedangkan *learning rate* yang besar, mengakibatkan *decision-boundary* sulit atau bahkan tidak dapat mencapai titik optimal.

12. **(5 points)** Explain how MLP obtains optimal decision boundary.

Untuk mendapatkan *decision-boundary* yang optimal, akan memproses inputan hingga mendapatkan *error*, dengan menentukan nilai bobot random dna memprosesnya hingga *layer* output. Kemudian, nilai *error* tersebut diproses lagi ke *layer input* dengan arah terbalik. Hasil dari proses ini kemudian digunakan untuk mengupdate nilai bobot yang akan diproses lagi. Langkah bolak-balik atau maju-mundur ini akan dilakukan berulang-ulang hingga nilai *error* tertentu atau sebanyak epoch yang telah ditentukan. Nilai akhir atau nilai bobot yang telah di-*update* akan membantu mempengaruhi MLP dalam membentuk *decision-boundary* yang optimal.

13. **(5 points)** Explain how SVM obtains optimal hyperplane.

Untuk mendapatkan hyperplane yang optimal, SVM harus memaksimalkan *decision-boundary* atau dengan hyperplane dengan margin paling besar (yang membuat SVM sering disebut *maximal margin classifier*) [2].

Misal dalam *binary classification* yang terdiri dari contoh *training* sebanyak N . Setiap contoh dinotasikan oleh *tuple* $(\mathbf{x}_i, y_i)(i = 1, 2, \dots, N)$, dimana $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ berhubungan dengan set atribut contoh ke- i , $y_i \in \{-1, 1\}$ sebagai label kelasnya. *Decision-boundary* dari sebuah *linear classifier* dapat ditulis dalam bentuk[2]

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

di mana \mathbf{w} dan b adalah parameter dari model. Untuk memaksimalkan *margin*-nya, sama dengan meminimalkan *objective function* berikut[2]

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}$$

Namun, fungsi objektif ini bersifat kuadratik dan *constraints*-nya linear dalam parameter \mathbf{w} dan b , yang dikenal sebagai *convex optimization problem*, yang dapat diselesaikan menggunakan metode *Lagrange Multiplier*.

Fungsi objektif yang kemudian digunakan sesuai masalah, *Langrangian* untuk optimisasi:

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \lambda_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1)$$

Kemudian untuk meminimalkan *Langrarian*, kita harus menggunakan turunan L_p terhadap \mathbf{w} dan b dan menjadikannya 0 :

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i$$

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N \lambda_i y_i = 0$$

Namun, karena pengali dari *Langrage* tidak diketahui, maka kita tidak dapat mencari \mathbf{w} dan b . Untuk itu, perlu dilakukan transformasi pada pengali *Langrage*, yang diketahui sebagai kondisi Karush-Kuhn-Tucker :

$$\begin{aligned} \lambda_i &\geq 0 \\ \lambda_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1] &= 0 \end{aligned}$$

yang kemudian mengarah ke

$$L_D = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

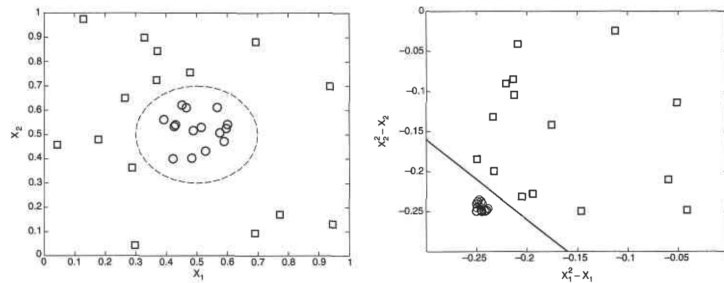
Kemudian, *decision-boundary*-nya dapat diekspresikan menggunakan [2]

$$\left(\sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \cdot \mathbf{x} \right) + b = 0$$

14. **(5 points)** Explain how SVM classifies data sets that are non-linearly separable.

Untuk mengklasifikasikan data yang *non-linearly separable*, SVM menggunakan fungsi kernel untuk memetakan data ke bidang baru yang memungkinkan SVM untuk membuat garis pemisah.

Contoh dampak transformasi pemetaan data pada bidang koordinat \mathbf{x} ke bidang koordinat $\Phi(\mathbf{x})$, yang membuat data dapat dipisahkan secara linear, dapat dilihat pada gambar berikut:



(a) Decision boundary in the original two-dimensional space.

(b) Decision boundary in the transformed space.

Referensi

- [1] https://id.wikipedia.org/wiki/Regresi_Linier
- [2] Introduction to Data Mining - Panning Tan, M. Steinbach
- [3] https://en.wikipedia.org/wiki/Nonlinear_regression
- [4] Regression book
- [5] Regression slide

- [6] <http://www.nickgillian.com/wiki/pmwiki.php/GRT/MLP>
- [7] Machine Learning - Tom Mitchell
- [8] <https://medium.com/towards-data-science/activation-functions-and-its-types-which-is-better-a9a5310cc8f>
- [9] Slide ANN-MLP Machine Learning