

Nama : Halida Fiadnin

NIM : G6401211142

## **LAPORAN KULIAH UMUM ANALISIS ALGORITMA**

Judul : Computational Complexity of Machine Learning Algorithm

Pembicara : Aries Fitriawan, S.Komp, M.Kom

Waktu : 30 September 2023

- kompleksitas pada machine learning
- kompleksitas komputasi pada sudut pandang praktis
- contoh kasus yang tdk disampaikan pada kuliah praktisi ini

### **A. Kompleksitas pada Machine Learning**

Konsep kompleksitas pada Machine Learning terbagi menjadi beberapa bagian, yaitu :

#### **1. Paradigma Komputasi**

Ada dua paradigma komputasi :

- a) Komputasi keras menangani permasalahan yang memiliki solusi eksak, dan solusi yang bersifat perkiraan/tidak pasti tidak dapat diterima. Ini adalah komputasi konvensional, dan sebagian besar mata kuliah algoritma berhubungan dengan komputasi keras.
- b) Komputasi lunak, di sisi lain, melihat pada teknik untuk memecahkan masalah yang tidak dapat dipecahkan dalam waktu terbatas. Sebagian besar algoritme pembelajaran mesin termasuk dalam kategori ini.

#### **2. Teori kompleksitas**

Ada dua teori kompleksitas :

- a) Kompleksitas waktu dapat dilihat sebagai ukuran seberapa cepat atau lambat kinerja suatu algoritma untuk ukuran masukan. Kompleksitas waktu selalu diberikan sehubungan dengan beberapa ukuran masukan ( $n$ ).
- b) Kompleksitas ruang dapat dilihat sebagai jumlah memori ekstra yang Anda perlukan untuk menjalankan algoritma Anda. Seperti waktu kompleksitas, itu juga diberikan sehubungan dengan beberapa ukuran masukan ( $n$ ).

#### **3. Kompleksitas berdasarkan jenis Algoritma**

Ada tiga kompleksitas berdasarkan jenis algoritma nya :

- a) Supervised Learning adalah jenis pembelajaran mesin di mana model dipelajari dari data berlabel. Kompleksitas waktu dalam Supervised Learning sangat tergantung pada jumlah besar data yang memiliki label. Selain itu, ruang kompleksitas juga dipengaruhi oleh struktur model yang digunakan. Semakin kompleks modelnya, semakin besar ruang kompleksitas yang diperlukan.

- b) Unsupervised Learning, di sisi lain, melibatkan pembelajaran dari data tanpa label. Kompleksitas Unsupervised Learning bergantung pada kemampuan algoritma untuk menemukan pola dalam data tanpa bimbingan. Misalnya, algoritma seperti K-Means Clustering memiliki kompleksitas yang berkaitan dengan jumlah cluster yang perlu ditemukan.
- c) Deep Learning, yang melibatkan penggunaan Neural Networks dengan banyak lapisan dan neuron, memiliki kompleksitas tersendiri. Kompleksitasnya terkait dengan jumlah lapisan dan neuron dalam jaringan. Selain itu, waktu dan ruang kompleksitas untuk pelatihan model Deep Learning sangat bergantung pada ukuran dataset yang digunakan dan arsitektur jaringan yang dipilih.

#### 4. Kompleksitas Algoritma Spesifik

Ada tujuh kompleksitas algoritma yang spesifik

- K-Nearest Neighbour (KNN):
  - Prediction phase:  $O(np)$  untuk KNN-KD Tree dan  $O(np^2)$  untuk KNN-Brute Force.
  - Training phase:  $O(np \log(n))$  untuk KD-Tree.
- Decision Tree:
  - Training phase:  $O(n^2p)$  untuk GINI and Entropy.
  - Prediction phase:  $O(p)$  untuk worst scenario.
- Random Forest:
  - Training phase:  $O(n^2p \cdot n\text{-trees})$  untuk GINI and Entropy.
  - Prediction phase:  $O(p \cdot n\text{trees})$  untuk worst scenario.
- Logistic Regression:
  - Training phase:  $O(p^2n + p^3)$  untuk menemukan vector weight pada linear regresi.
  - Prediction phase:  $O(p)$  untuk worst case.
- Support Vector Machine (Kernel):
  - Training phase:  $O(n^2p + n^3)$ .
  - Prediction phase:  $O(n_{sv}p)$ .
- Neural Networks:
  - Training phase:  $O(n_{epoch}n^5 (pn_{I1} + n_{I1}n_{I2} + \dots))$ .
  - Prediction phase:  $O(pn_{I1} + n_{I1}n_{I2} + \dots)$ .
- K-Means Clustering:
  - Training phase:  $O(nkl)$ .
  - Prediction phase:  $O(npk)$ .

## B. Kompleksitas komputasi pada sudut pandang praktis

Konsep kompleksitas pada Machine Learning terbagi menjadi beberapa bagian, yaitu :

### 1. Overfitting (Kelebihan Penyesuaian):

Kompleksitas Komputasi: Overfitting terjadi ketika sebuah model menjadi terlalu kompleks dan menangkap noise dalam data pelatihan daripada pola yang mendasarinya. Ini sering melibatkan pelatihan model dengan jumlah fitur yang besar, derajat polinomial yang tinggi, atau arsitektur jaringan saraf yang dalam. Kompleksitas komputasi dari overfitting dapat tinggi karena model harus menyesuaikan sejumlah besar parameter. Sudut Pandang Praktis: Overfitting dapat menghasilkan model yang mahal secara komputasi untuk dilatih dan dievaluasi. Mereka mungkin memerlukan banyak data dan penyetelan hiperparameter yang ekstensif untuk mencegah overfitting. Tingkat

kompleksitas yang tinggi dapat membuat sulit untuk mendeploy model-model tersebut dalam aplikasi waktu nyata.

**2. Underfitting (Kurang Penyesuaian):**

Kompleksitas Komputasi: Underfitting terjadi ketika model terlalu sederhana untuk memahami pola yang ada dalam data. Model yang terlalu sederhana mungkin memiliki kompleksitas komputasi yang rendah karena memiliki sedikit parameter.

Sudut Pandang Praktis: Meskipun underfitting memiliki kompleksitas komputasi yang rendah, model-model yang underfit akan kurang akurat dan tidak dapat memodelkan data dengan baik. Dalam konteks praktis, solusi yang terlalu sederhana mungkin tidak memberikan hasil yang memuaskan.

**3. Not Converge (Tidak Konvergen):**

Kompleksitas Komputasi: Ketika sebuah model tidak konvergen, artinya proses pelatihan tidak mencapai konvergensi ke solusi yang optimal. Hal ini dapat disebabkan oleh berbagai faktor, termasuk pengaturan hiperparameter yang tidak tepat atau masalah numerik.

Sudut Pandang Praktis: Ketidakkonvergenan dapat menghasilkan biaya komputasi yang tinggi karena pelatihan model mungkin memerlukan banyak iterasi yang tidak perlu. Dalam kasus yang lebih parah, model mungkin tidak dapat digunakan sama sekali.

**4. Good Fit (Fit yang Baik):**

Kompleksitas Komputasi: Pencapaian fit yang baik dalam konteks praktis adalah ketika model mampu menggambarkan pola dalam data tanpa terlalu sederhana atau terlalu kompleks. Kompleksitas komputasi dari fit yang baik bergantung pada ukuran dataset dan jenis algoritma yang digunakan.

Sudut Pandang Praktis: Model dengan fit yang baik seringkali merupakan pilihan terbaik dalam aplikasi praktis karena mampu memberikan hasil yang memadai tanpa memerlukan sumber daya komputasi yang berlebihan. Fit yang baik juga menghasilkan hasil yang lebih dapat diandalkan dan mudah untuk dikelola.

## **C. Contoh kasus yang tdk disampaikan pada kuliah praktisi ini**

Contoh perhitungan kompleksitas pada Machine Learning yaitu, Regresin Linier.

Regresi linier adalah metode machine learning sederhana yang digunakan untuk memodelkan hubungan linier antara variabel input dan variabel target.

Kompleksitas: Dalam pelatihan regresi linier, kompleksitas komputasi biasanya tergantung pada jumlah fitur ( $p$ ) dan jumlah sampel data pelatihan ( $n$ ). Algoritma pelatihan regresi linier memiliki kompleksitas  $O(np^2)$  dalam menyelesaikan persamaan normalnya.