

Support Vector Regressor

1. Cara kerja Algoritma

Secara garis besar, *Support Vector Regression* (SVR) adalah metode regresi yang didasarkan pada *Support Vector Machine* (SVM) untuk menemukan hyperplane yang memprediksi nilai output dalam batas margin tertentu. Batas margin ini menjadi variabel untuk menentukan seberapa besar kesalahan yang dapat diterima dalam model dan akan menemukan garis yang sesuai (atau *hyperplane* dalam dimensi yang lebih tinggi) agar sesuai dengan data.

Garis prediksi memiliki rumus:

$$\sum_{k=1}^K w_k \phi(x_{i,k}) + b$$

SVR berusaha untuk meminimalkan nilai $\frac{1}{2} \|w\|^2$ (fungsi objektif) dengan harapan bisa meminimalkan error. Garis prediksi memiliki batasan:

$$\begin{aligned} y_i - wx_i - b &\leq \epsilon \\ wx_i + b - y_i &\leq \epsilon \end{aligned}$$

Simbol ϵ /epsilon adalah nilai margin dan merupakan hyperparameter. Semakin besar nilai epsilon maka semakin toleran prediksi terhadap error.

Terdapat dua jenis SVR berdasarkan toleransinya terhadap titik data di luar margin, yaitu hard-margin dan soft-margin. Hard-margin berarti sama sekali tidak boleh ada titik di luar margin sedangkan soft-margin memperbolehkan adanya titik data di luar margin. Seberapa toleran soft-margin diwakili oleh hyperparameter C yang berbanding terbalik dengan jumlah titik di luar margin. Rumus fungsi objektif soft-margin:

$$\text{MIN } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n |\xi_i|$$

ξ (ξ_i) adalah jarak titik terhadap batas margin.

Kernel pada SVR adalah suatu trik untuk seakan-akan membawa bidang data ke dimensi yang lebih tinggi untuk digambar garis lurusnya lalu dikembalikan lagi ke dimensi semula. Terdapat banyak sekali jenis kernel yang diwakili dengan suatu fungsi ($k(x)$). Untuk menerapkannya, fungsi kernel akan dikali dengan w pada rumus garis gradien.

2. Perbandingan Hasil Evaluasi Model

Model yang saya buat memiliki score yang lebih buruk daripada model dari scikit learn dengan hyperparameter yang sama. Hal ini terjadi karena scikit learn tidak menggunakan optimizer gradient descent melainkan menyelesaikanya dengan *quadratic programming solver*.

```
Custom R2 Score: 0.3487960099119829  
Sklearn R2 Score: 0.4610472806025431  
Custom Mean Squared Error: 24228.916156427513  
Sklearn Mean Squared Error: 20052.457370223772
```

3. Improvement

Untuk mempercepat waktu kalkulasi, model perlu menggunakan *sparse* matriks. Selain itu, penambahan kernel-kernel lain juga bisa menambah versatilitas/penggunaan model pada kasus-kasus tertentu.