

Chapter 5: Support Vector Machines

Support Vector Machine (SVM) adalah salah satu model Machine Learning yang paling kuat dan serbaguna. Model ini mampu melakukan tugas klasifikasi linear maupun non-linear, regresi, dan bahkan deteksi anomali (*outlier detection*). SVM sangat efektif pada dataset yang kompleks namun berukuran kecil hingga menengah, dan menjadi salah satu model yang wajib dipahami dalam fundamental Machine Learning.

1. Klasifikasi Linear SVM

Ide dasar di balik SVM adalah menemukan batas keputusan (decision boundary) yang terbaik untuk memisahkan dua kelas.

- Klasifikasi Margin Lebar (Large Margin Classification)

SVM tidak hanya mencari garis yang memisahkan kelas, tetapi mencari "jalan" (street) terlebar yang memisahkan kedua kelas tersebut. Batas keputusan SVM adalah garis yang berada tepat di tengah jalan ini. Metode ini disebut klasifikasi margin lebar karena memaksimalkan jarak antara batas keputusan dan instance pelatihan terdekat dari kelas mana pun.

Instance pelatihan yang terletak di tepi jalan disebut **support vectors**. Vektor-vektor inilah yang "menopang" atau menentukan batas keputusan. Jika kita memindahkan *support vector*, batas keputusan akan berubah. Sebaliknya, instance lain yang tidak berada di tepi jalan tidak memiliki pengaruh apa pun.

- Soft Margin Classification

Pendekatan di atas, yang disebut hard margin classification, hanya berfungsi jika data dapat dipisahkan secara linear dengan sempurna dan sangat sensitif terhadap outlier. Untuk data di dunia nyata, pendekatan yang lebih fleksibel diperlukan.

Soft margin classification adalah pendekatan yang lebih realistis. Tujuannya adalah untuk menyeimbangkan dua hal:

1. Menjaga "jalan" selebar mungkin.
2. Membatasi pelanggaran margin (*margin violations*), yaitu instance yang berakhir di tengah jalan atau bahkan di sisi yang salah.

Tingkat keseimbangan ini dikendalikan oleh *hyperparameter C*.

- **C rendah:** Menghasilkan jalan yang lebih lebar tetapi dengan lebih banyak pelanggaran margin. Ini adalah bentuk **regularisasi yang lebih tinggi**, karena model lebih digeneralisasi dan tidak terlalu terpengaruh oleh setiap instance data.
- **C tinggi:** Menghasilkan jalan yang lebih sempit dengan pelanggaran margin yang lebih sedikit. Model akan berusaha sangat keras untuk mengklasifikasikan setiap instance dengan benar. Ini adalah bentuk **regularisasi yang lebih rendah** dan berisiko mengalami *overfitting* jika datanya *noisy*.
-

2. Klasifikasi Non-Linear SVM

Banyak dataset tidak dapat dipisahkan secara linear. SVM dapat menangani data non-linear menggunakan sebuah teknik cerdas yang disebut **kernel trick**.

- Kernel Trick

Ide dasarnya adalah dengan menambahkan fitur baru ke data untuk membuatnya dapat dipisahkan secara linear di ruang dimensi yang lebih tinggi. Kernel trick adalah sebuah "trik" matematis yang memungkinkan SVM mendapatkan hasil yang sama seolah-olah telah menambahkan banyak fitur dimensi tinggi, tanpa benar-benar harus membuatnya. Ini sangat efisien secara komputasi.

Beberapa kernel yang populer adalah:

1. **Polynomial Kernel:** Mensimulasikan penambahan fitur polinomial. Jika data Anda memiliki pola kurva polinomial, kernel ini bisa sangat efektif. Tingkat kompleksitasnya diatur oleh *hyperparameter* **degree**.
2. **Gaussian RBF Kernel:** Kernel ini sangat kuat dan sering menjadi pilihan utama. Cara kerjanya didasarkan pada **kemiripan** (*similarity*) setiap instance dengan *landmark* tertentu. Pengaruh setiap *landmark* menyebar keluar seperti lonceng Gaussian. *Hyperparameter* **gamma (γ)** mengontrol lebar "lonceng" ini:
 - **gamma rendah:** Membuat kurva lonceng lebar. Pengaruh setiap instance lebih luas, menghasilkan batas keputusan yang lebih mulus dan **regularisasi lebih tinggi**.
 - **gamma tinggi:** Membuat kurva lonceng sempit. Pengaruh setiap instance lebih terbatas, menghasilkan batas keputusan yang lebih tidak teratur dan berliku-liku. Ini adalah **regularisasi lebih rendah** dan berisiko *overfitting*.

3. Regresi SVM

SVM juga dapat digunakan untuk tugas regresi. Tujuannya sedikit berbeda dari klasifikasi.

- **Tujuan:** Jika klasifikasi SVM mencoba mencari jalan terlebar *di antara* kelas, maka Regresi SVM mencoba memasukkan sebanyak mungkin instance *ke dalam* jalan.
- **Margin (ϵ):** Lebar jalan dikendalikan oleh *hyperparameter* **epsilon (ϵ)**. Model tidak akan "dihukum" untuk instance yang prediksinya berada di dalam margin ϵ . Dengan kata lain, model ini **ϵ -insensitive**.
- **Pelanggaran Margin:** Seperti pada klasifikasi, Regresi SVM juga menggunakan *hyperparameter* **C** untuk mengontrol seberapa besar "hukuman" yang diberikan untuk instance yang berada di luar margin.

Kesimpulan

Support Vector Machines adalah model yang elegan dan kuat. Kemampuannya untuk melakukan klasifikasi margin lebar dan menangani data non-linear melalui *kernel trick* (terutama dengan kernel RBF) menjadikannya alat yang sangat efektif untuk berbagai masalah. Meskipun mungkin lebih lambat pada dataset yang sangat besar dibandingkan beberapa model lain, kinerjanya seringkali sangat baik, terutama pada dataset berukuran kecil hingga menengah.

yang kompleks. Pemahaman tentang *hyperparameter* kunci seperti C , γ , dan ϵ sangat penting untuk menggunakannya secara efektif.