

Support Vector Machine (SVM)

Formulasi Umum

Support Vector Machine (SVM) merupakan algoritma klasifikasi margin-maksimum yang mendukung pemisahan linier maupun non-linier melalui fungsi kernel. Diberikan dataset pelatihan $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, dengan $\mathbf{x}_i \in \mathbb{R}^d$ dan label biner $y_i \in \{-1, 1\}$, SVM berusaha menemukan hiperplane pemisah $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$ yang memaksimalkan margin terhadap kelas berbeda, dengan $\phi(\cdot)$ merupakan pemetaan implisit dari data ke ruang fitur berdimensi lebih tinggi.

Kernel dan Parameterisasi

Implementasi mendukung empat jenis kernel:

- **Linear:** $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$
- **Polynomial:** $K(\mathbf{x}, \mathbf{x}') = (\gamma \cdot \mathbf{x}^\top \mathbf{x}' + \text{coef0})^d$
- **Radial Basis Function (RBF):** $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$
- **Sigmoid:** $K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \cdot \mathbf{x}^\top \mathbf{x}' + \text{coef0})$

Nilai γ ditetapkan sebagai:

$$\gamma = \begin{cases} \frac{1}{d \cdot \text{Var}(\mathbf{X})}, & \text{jika gamma = scale} \\ \text{nilai eksplisit}, & \text{lainnya} \end{cases}$$

Optimisasi

Model dioptimalkan menggunakan *gradient descent* terhadap parameter dual α_i dan bias b . Fungsi objektif yang diminimalkan adalah:

$$L(\alpha, b) = \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i f(\mathbf{x}_i)) + \frac{1}{2C} \|\alpha\|^2$$

Gradien dihitung secara efisien menggunakan batch kernel, dan parameter dikoreksi dengan pembatasan $0 \leq \alpha_i \leq C$. Proses dihentikan berdasarkan konvergensi fungsi kerugian dengan toleransi tertentu.

Inferensi

Nilai keputusan dihitung sebagai:

$$f(\mathbf{x}) = \sum_{i \in \text{SV}} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b$$

di mana SV adalah himpunan *support vectors*.

Prediksi kelas dilakukan dengan:

$$\hat{y} = \begin{cases} \text{kelas}_1, & f(\mathbf{x}) \geq 0 \\ \text{kelas}_0, & \text{lainnya} \end{cases}$$

Estimasi probabilitas dikalkulasi melalui pendekatan **Platt scaling**:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-f(\mathbf{x}))}$$

Linear SVM

Optimalisasi Langsung

Pada kasus kernel linear, implementasi disederhanakan dengan mengoptimasi bobot \mathbf{w} dan bias b secara langsung:

$$L(\mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)) + \frac{C}{2} \|\mathbf{w}\|^2$$

Gradien dari fungsi kerugian dihitung sebagai:

- $\nabla_{\mathbf{w}} = C\mathbf{w} - \frac{1}{n} \sum_{i \in \text{violated}} y_i \mathbf{x}_i$
- $\nabla_b = -\frac{1}{n} \sum_{i \in \text{violated}} y_i$

Dengan kriteria konvergensi berbasis perubahan absolut fungsi kerugian.

Prediksi

Prediksi dilakukan melalui:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

dan probabilitas dengan:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-f(\mathbf{x}))}$$

Perbandingan

Performa model dari Scikit-learn sedikit lebih baik dibanding versi from scratch dalam hal akurasi dan F1-score. Ini dapat dikaitkan dengan:

Optimalisasi Scikit-learn dalam memilih parameter regularisasi (C) dan kernel.

Algoritma Quadratic Programming atau liblinear yang sangat efisien.