

Support Vector Machines

Ali Akbar Septiandri

December 16, 2017

untuk Astra Graphia IT

1. Pendahuluan
2. Batas Keputusan dan Margin Maksimum
3. SVM Non-linear

1. VanderPlas, J. (2016). Python Data Science Handbook. (In Depth: Support Vector Machines) <https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html>
2. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann. (Section 6.3)

Pendahuluan

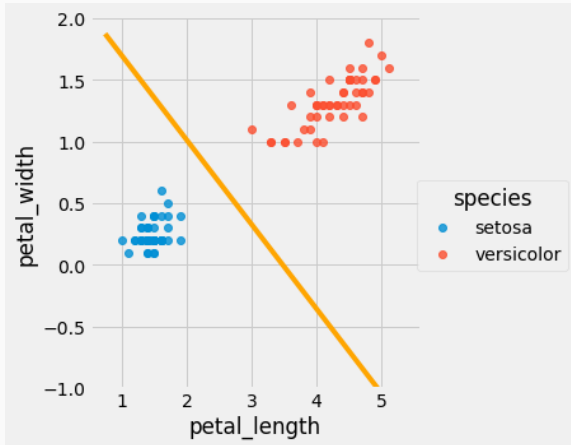
- Support vector machines (SVM) adalah salah satu yang paling sering digunakan dan cukup efektif
- Merupakan kombinasi dari dua ide
 - Klasifikasi margin maksimum
 - “Kernel trick”
- SVMs adalah *linear classifier*, seperti regresi logistik

$w^T x$ adalah **panjang** dari **proyeksi** x ke w (jika w adalah vektor unit), i.e. $b = w^T x$.

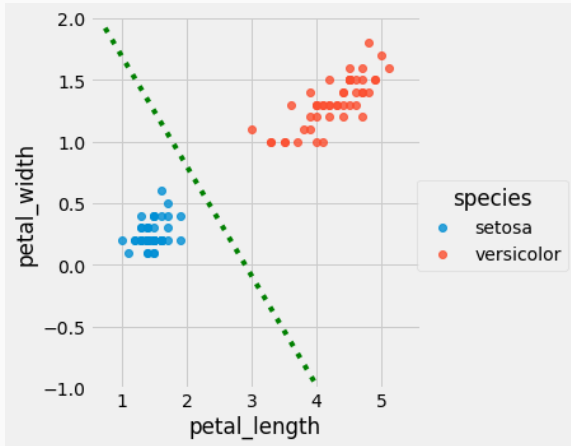
Untuk setiap *linear classifier*,

- Data latih $(\mathbf{x}_i, y_i), i = 1, \dots, n. y_i \in \{-1, +1\}$
- Hyperplane $\mathbf{w}^T \mathbf{x} + w_0 = 0$
- Dalam kasus SVM, kita akan menggunakan -1 alih-alih 0 karena lebih memudahkan matematikanya

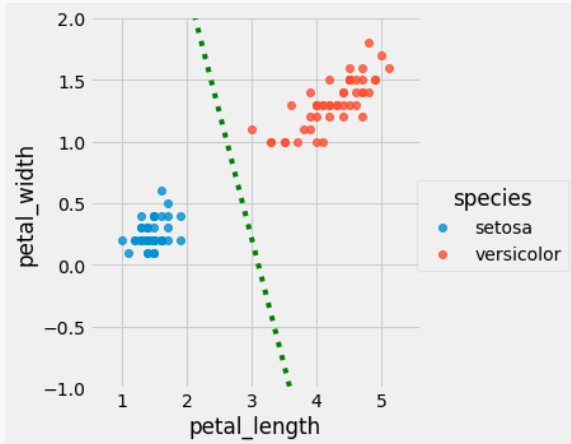
Batas Keputusan



Batas Keputusan



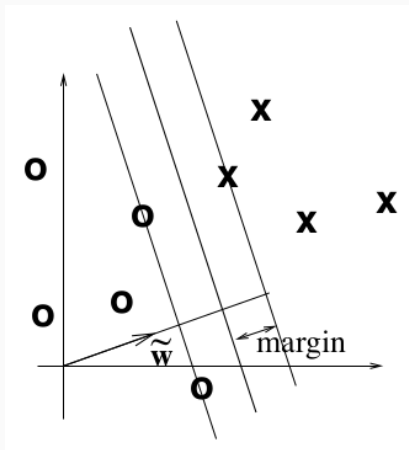
Batas Keputusan



Batas Keputusan dan Margin Maksimum

Ide: Maksimalkan Marginnya

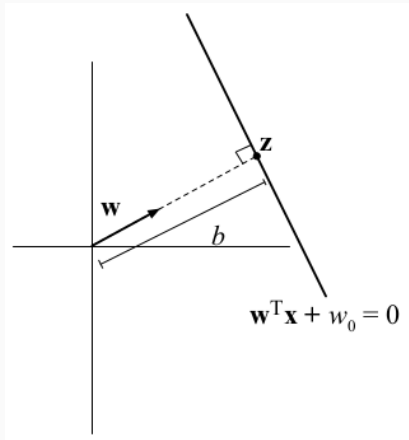
Margin adalah jarak antara batas keputusan dengan data latih terdekat



Menghitung Margin

- Dilakukan dengan menghitung jarak terdekat dari titik pusat $O(0,0)$ ke batas keputusan terlebih dahulu
- Memanfaatkan fakta bahwa w tegak lurus dengan batas keputusan

Menghitung Jarak ke Hyperplane



Diberikan...

- $\mathbf{w}^T \mathbf{x} + w_0 = 0$

Diberikan...

- $\mathbf{w}^T \mathbf{x} + w_0 = 0$
- panjang $\|z\| = b$ sehingga $b \frac{\mathbf{w}}{\|\mathbf{w}\|} = \mathbf{z}$

Diberikan...

- $\mathbf{w}^T \mathbf{x} + w_0 = 0$
- panjang $\|\mathbf{z}\| = b$ sehingga $b \frac{\mathbf{w}}{\|\mathbf{w}\|} = \mathbf{z}$
- berarti $\mathbf{w}^T \mathbf{z} + w_0 = 0$

Diberikan...

- $\mathbf{w}^T \mathbf{x} + w_0 = 0$
- panjang $\|z\| = b$ sehingga $b \frac{\mathbf{w}}{\|\mathbf{w}\|} = \mathbf{z}$
- berarti $\mathbf{w}^T \mathbf{z} + w_0 = 0$
- substitusi nilai \mathbf{z} , kita akan mendapatkan

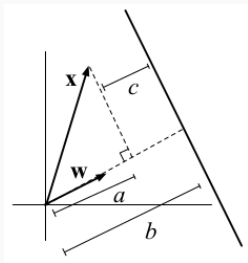
$$\mathbf{w}^T \frac{b\mathbf{w}}{\|\mathbf{w}\|} + w_0 = 0$$

$$\frac{b\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} + w_0 = 0$$

$$b = -\frac{w_0}{\|\mathbf{w}\|}$$

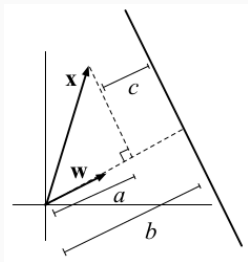
- ingat bahwa $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T \mathbf{w}}$

Menghitung Jarak ke Hyperplane



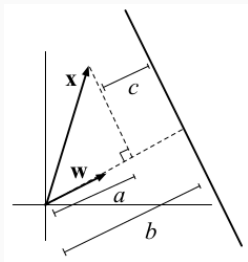
- Kita ingin c , jarak dari x ke hyperplane

Menghitung Jarak ke Hyperplane



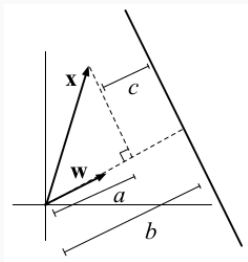
- Kita ingin c , jarak dari \mathbf{x} ke hyperplane
- Kita tahu $c = |b - a|$, dengan a adalah panjang proyeksi \mathbf{x} ke \mathbf{w}

Menghitung Jarak ke Hyperplane



- Kita ingin c , jarak dari \mathbf{x} ke hyperplane
- Kita tahu $c = |b - a|$, dengan a adalah panjang proyeksi \mathbf{x} ke \mathbf{w}
- Berapa nilai a ?

Menghitung Jarak ke Hyperplane



- Kita ingin c , jarak dari x ke hyperplane
- Kita tahu $c = |b - a|$, dengan a adalah panjang proyeksi x ke w
- Berapa nilai a ?
- $a = \frac{w^T x}{\|w\|}$

Persamaan untuk Margin

- Jarak tegak lurus dari titik \mathbf{x} ke hyperplane $\mathbf{w}^T \mathbf{x} + w_0 = 0$ adalah

$$\frac{1}{\|\mathbf{w}\|} |\mathbf{w}^T \mathbf{x} + w_0|$$

- Margin adalah jarak dari titik terdekat ke hyperplane

$$\min_i \frac{1}{\|\mathbf{w}\|} |\mathbf{w}^T \mathbf{x}_i + w_0|$$

Max Margin Optimization Problem

Tujuan kita adalah memaksimalkan margin dengan

$$\begin{aligned} & \max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|} \\ \text{subject to } & \mathbf{w}^T \mathbf{x}_i + w_0 \geq 0, & \text{for all } i \text{ with } y_i = 1 \\ & \mathbf{w}^T \mathbf{x}_i + w_0 \leq 0, & \text{for all } i \text{ with } y_i = -1 \\ & \min_i |\mathbf{w}^T \mathbf{x}_i + w_0| = 1 \end{aligned}$$

Max Margin Optimization Problem

Batasan pertama dan kedua dapat diubah menjadi

$$\begin{aligned} \max_{\mathbf{w}} \quad & \frac{1}{\|\mathbf{w}\|} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{x}_i + w_0 \geq +1, & \text{for all } i \text{ with } y_i = 1 \\ & \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1, & \text{for all } i \text{ with } y_i = -1 \\ & \min_i |\mathbf{w}^T \mathbf{x}_i + w_0| = 1 \end{aligned}$$

Max Margin Optimization Problem

Batasan ketiga dapat dibuang karena redundan

$$\begin{aligned} & \max_{\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|} \\ & \text{subject to } \mathbf{w}^T \mathbf{x}_i + w_0 \geq +1, & \text{for all } i \text{ with } y_i = 1 \\ & \quad \quad \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1, & \text{for all } i \text{ with } y_i = -1 \end{aligned}$$

Max Margin Optimization Problem

Dapat disederhanakan lagi menjadi

$$\begin{aligned} & \max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|} \\ & \text{subject to } y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq +1, \quad \text{for all } i \end{aligned}$$

Perhatikan bahwa memaksimalkan nilai $\frac{1}{\|\mathbf{w}\|}$ sama dengan meminimalkan $\|\mathbf{w}\|^2$

Jadi, optimasi vektor bobot pada SVM dapat didefinisikan sebagai

$$\begin{aligned} \min_{\mathbf{w}} \quad & \|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq +1 \quad , \text{ for all } i \end{aligned}$$

Ini dikenal sebagai *constrained optimization problem*.

Mencari Nilai Optimal

- Persoalan tadi dapat diselesaikan dengan Lagrange multipliers sehingga parameter optimalnya adalah

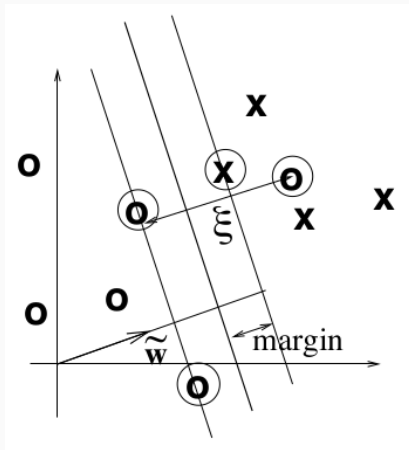
$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

- Perhatikan bahwa hyperplane yang optimal hanya dibentuk dari beberapa titik saja, yang disebut sebagai **support vectors**
- $\alpha_i = 0$ untuk *non-support*
- Tidak ada minimum lokal

Prediksi pada data baru \mathbf{x} adalah

$$\begin{aligned}f(\mathbf{x}) &= \text{sign}(\mathbf{w}^T \mathbf{x} + w_0) \\&= \text{sign}\left(\sum_{i=1}^n \alpha_i y_i (\mathbf{x}^T \mathbf{x}) + w_0\right)\end{aligned}$$

Apa yang terjadi kalau kasusnya seperti ini?



- Solusi: Gunakan “slack” variabel $\xi_i \geq 0$ untuk tiap data latih
- Idenya seperti memberikan penalti pada *ridge regression*

SVM Non-linear

- Dapat dibuat non-linear dengan fungsi basis seperti pada regresi

- Dapat dibuat non-linear dengan fungsi basis seperti pada regresi
- Ada cara khusus menggunakan **kernel**

- Dapat dibuat non-linear dengan fungsi basis seperti pada regresi
- Ada cara khusus menggunakan **kernel**
- Komputasinya lebih cepat

- Dapat dibuat non-linear dengan fungsi basis seperti pada regresi
- Ada cara khusus menggunakan **kernel**
- Komputasinya lebih cepat
- Topik matematika lanjut, jadi dibahas ide umumnya saja

- Transformasi \mathbf{x} menjadi $\phi(\mathbf{x})$
- Algoritma yang linear hanya bergantung pada $\mathbf{x}^T \mathbf{x}$ sehingga transformasinya hanya bergantung pada $\phi(\mathbf{x})^T \phi(\mathbf{x})$
- Gunakan fungsi kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ sedemikian sehingga

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x})^T \phi(\mathbf{x})$$

Support Vector Machine

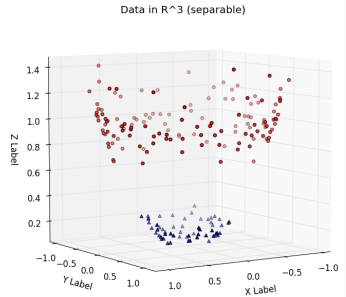
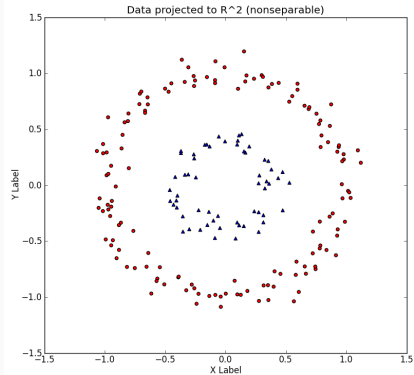
- Support vector machine pada dasarnya adalah *kernelized maximum margin classifier*
- Ingat kembali bahwa prediksi pada SVM adalah

$$\hat{y} = \text{sign}(\mathbf{w}^T \mathbf{x} + w_0) = \text{sign}\left(\sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + w_0\right)$$

- Dengan kernel, didapat

$$\hat{y} = \text{sign}\left(\sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + w_0\right)$$

Kernel Trick



Kernel popular lain: RBF

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\alpha^2}$$

Pros:

- Modelnya padat, hanya perlu memori yang sedikit
- Fase prediksi bisa sangat cepat
- Bisa bekerja dengan baik dalam dimensi tinggi
- “Kernel tricks”

Cons:

- $O(N^3)$ dalam kasus terburuk, hanya bisa diperbaiki hingga $O(N^2)$
- Sangat bergantung pada parameter ξ
- Tidak probabilistik

**Salindia ini dipersiapkan dengan banyak
mengadaptasi dari Nigel Goddard (2014)**

Terima kasih