Metode Kernel

Dr. rer. nat. Hendri Murfi

Intelligent Data Analysis (IDA) Group

Departemen Matematika, Universitas Indonesia – Depok 16424

Telp. +62-21-7862719/7863439, Fax. +62-21-7863439, Email. hendri@ui.ac.id

Machine Learning

Input	Metode	Output
x_1 x_2 : x_D	$y(\mathbf{x},\mathbf{w})$	*Klasifikasi, regresi, clustering, dll*

- *Preprocessing*: representasi data (dalam bentuk vektor), pemilihan/ekstraksi fitur dari data, misal $\mathbf{x}_i = (x_1, x_2, .., x_D)^T$
- *Learning*: pemilihan model dan penentuan parameter metode, misal **w**, berdasarkan data pelatihan (*training data*)
- *Testing*: pengujian metode dengan data penguji (*testing data*) yang tidak sama dengan data pelatihan, sehingga didapat nilai estimasi untuk kapabilitas generalisasi dari metode.

Learning

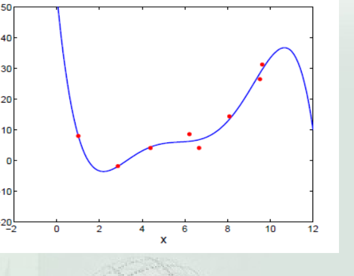
Diberikan data pelatihan $\boldsymbol{x}_i$, *i = 1 sd N,* dan/atau $\boldsymbol{t}_i$, *i = 1 sd N*

- Supervised Learning. Data pelatihan disertai target, yaitu {$\mathbf{x}_i$, $\mathbf{t}_i$}, i = 1 sd N. Tujuan pembelajaran adalah membangun model yang dapat menghasilkan output yang benar untuk suatu data input, misal untuk regresi, klasifikasian, regresi ordinal, ranking, dll

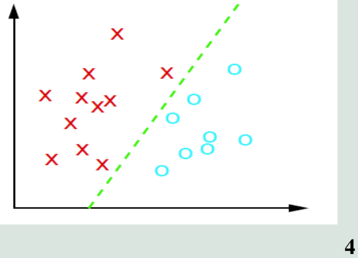
- Unsupervised Learning. Data pelatihan tidak disertai target, yaitu $\mathbf{x}_i$, i = 1 sd N. Tujuan pembelajaran adalah membagun model yang dapat menemukan komponen/variabel/fitur tersembunyi pada data pelatihan, yang dapat digunakan untuk: pengelompokan (*clustering*), reduksi dimensi (*dimension reduction*), rekomendasi, dll

Supervised Learning

- Regresi
  - Nilai output $\boldsymbol{t}_i$ bernilai kontinu (riil)
  - Bertujuan memprediksi output dari data baru dengan akurat



- Klasifikasi
  - Nilai output $\boldsymbol{t}_i$ bernilai diskrit (kelas)
  - Bertujuan mengklasifikasi data baru dengan akurat



Model Linear

- Model yang umum digunakan untuk menyelesaikan masalah klasifikasi dan regresi adalah model linear, yaitu model yang merupakan kombinasi linear dari fungsi basis:

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

  Dimana $\mathbf{x} = (x_1, x_2, ..., x_D)^T$ adalah variabel input, dan $\mathbf{w} = (w_0, w_1, ..., w_D)^T$ adalah parameter, $\phi(\mathbf{x})$ adalah *fungsi basis*, *M* adalah jumlah total parameter dari model

- Biasanya, $\phi_0(\mathbf{x}) = 1$, sehingga w_0 berfungsi sebagai bias

- Ada banyak pilihan yang mungkin untuk fungsi basis $\phi(\mathbf{x})$, misal fungsi linear, fungsi polinomial, fungsi gaussian, fungsi sigmoidal, dll

Model Linear

Kutukan Dimensi

- Model linear memiliki sifat-sifat yang penting baik dari aspek komputasi maupun analitik. Penggunaan model linear dengan pendekatan parametrik pada metode klasik memiliki keterbatasan pada aplikasi praktis disebabkan oleh kutukan dimensi *(curse of dimensionality)*

$$y(x, w) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

1

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

D^3

untuk model beorde M, maka pertumbuhan jumlah parameter w proposional dengan D^M

Model Linear

Pendekatan Alternatif

- Pendekatan alternatif adalah menentukan jumlah fungsi basis didepan, akan tetapi masing-masing fungsi basis tersebut adaptif terhadap semua data pembelajaran.
- Dengan kata lain, menggunakan bentuk **parametrik tidak linear** dimana nilai-nilai parameter adaptif selama proses training.
- Contoh metode yang menggunakan pendekatan ini adalah *neural networks* (NN).

Model Linear

Pendekatan Nonparametrik

- Untuk menerapkan metode ini pada masalah skala besar, pendekatan umum yang dilakukan adalah membuat fungsi basis dapat beradaptasi dengan data pembelajaran, dan menetapkan data pembelajaran tersebut sebagai pusat fungsi basis. Selanjutnya, memilih sebagian dari data pembelajaran tersebut selama proses pelatihan (nonparametrik).
- Dasarnya adalah bahwa data real biasanya memiliki sifat mulus, artinya perubahan sedikit pada data input hanya akan memberikan sedikit perubahan pada output
- Penggunaan fungsi kernel sebagai fungsi basis adalah salah satu contoh pendekatan seperti ini yang banyak digunakan saat ini.

Metode Kernel

Fungsi Kernel : Contoh

- Salah satu contoh fungsi kernel yang banyak digunakan adalah *Gaussian radial basis function* (RBF), yaitu:

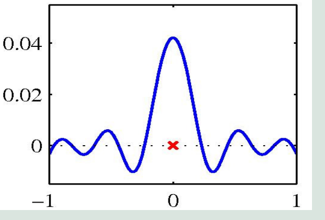
$$k(\mathbf{x},\mathbf{x}') = \phi(||\mathbf{x}-\mathbf{x}'||) = exp(-||\mathbf{x}-\mathbf{x}'||^2 / 2s^2)$$

  dimana $\mathbf{x}'$ adalah „inti“ yang dipilih dari data pelatihan.

- Contoh: fungsi basis dengan pusat/inti x' = 5 dan bobot w = 0.04 dapat digambarkan sbb:

$$w * k(x,5) = 0.04 * \phi(||x-5||)$$

$$= 0.04 * exp(-||x-5||^2 / 2s^2)$$



Fungsi Kernel

Definisi

- Fungsi kernel adalah suatu fungsi *k* yang mana untuk semua vektor input ***x,z*** akan memenuhi kondisi

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

  dimana $\phi(.)$ adalah fungsi pemetaan dari ruang input ke ruang fitur

- Dengan kata lain, fungsi kernel adalah fungsi hasil kali dalam (*inner product*) pada ruang fitur.

Fungsi Kernel

Contoh

- Contoh 1:

  $k(\mathbf{x},\mathbf{z}) = (\mathbf{x}^T\mathbf{z})^2$ adalah fungsi kernel untuk $\mathbf{x}, \mathbf{z} \in R^2$

  Bukti:

  misal $\mathbf{x}=(x_1,x_2)$, $\mathbf{z}=(z_1,z_2)$ maka

$$
\begin{aligned}
(\mathbf{x}^T\mathbf{z})^2 &= (x_1z_1 + x_2z_2)^2 \\
&= x_1^2z_1^2 + 2x_1z_1x_2z_2 + x_2^2z_2^2 \\
&= (x_1^2, \sqrt{2}\, x_1x_2, x_2^2)^T (z_1^2, \sqrt{2}\, z_1z_2, z_2^2) \\
&= \phi(\mathbf{x})^T\phi(\mathbf{z})
\end{aligned}
$$

  sehingga $k(\mathbf{x},\mathbf{z}) = (\mathbf{x}^T\mathbf{z})^2$ adalah suatu fungsi kernel dengan fungsi pemetaan $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}\, x_1x_2, x_2^2)$, yaitu suatu fungsi pemetaan dari R^2 ke R^3

Fungsi Kernel

Aproksimasi Nilai *Inner Product*

- Dari definisi dan contoh sebelumnya maka terlihat bahwa jika kita memiliki suatu fungsi kernel maka kita dapat menghitung *inner product* pada ruang fitur secara langsung dari ruang input tanpa secara eksplisit menghitung koordinat proyeksi masing-masing vektor input pada ruang fitur

- *Inner product* adalah operasi yang sangat penting karena sangat erat kaitannya dengan persoalan geometri dari data pada ruang fitur, misal untuk menghitung jarak:

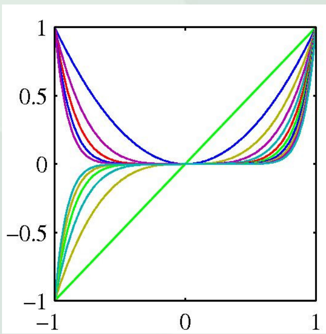
$$
\begin{aligned}
\|\phi(\mathbf{x})-\phi(\mathbf{z})\|^2 &= \phi(\mathbf{x})^T\phi(\mathbf{x}) + \phi(\mathbf{z})^T\phi(\mathbf{z}) - 2\phi(\mathbf{x})^T\phi(\mathbf{z}) \\
&= k(\mathbf{x},\mathbf{x}) + k(\mathbf{z},\mathbf{z}) - 2k(\mathbf{x},\mathbf{z})
\end{aligned}
$$

Fungsi Kernel

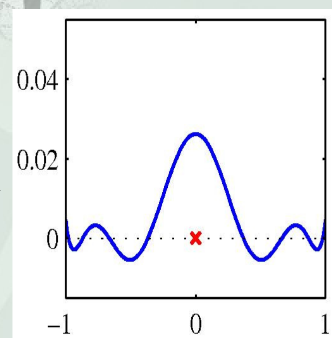
Konstruksi 1

- Pendekatan pertama adalah memilih suatu fungsi pemetaan $\phi(\mathbf{x})$ dan kemudian menggunakannya untuk mengkonstruksi kernel dengan cara sbb:

$$k(x, x') = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x') = \sum_{i=1}^{M} \phi_i(x)\phi_i(x')$$



$\phi_i(x) = x^i$



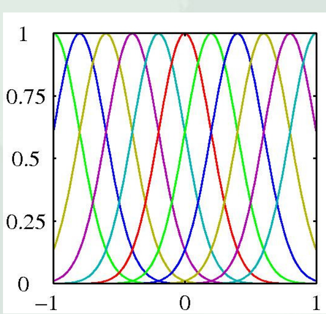
$k(x, x')$ *untuk* $x' = 0$

Fungsi Kernel

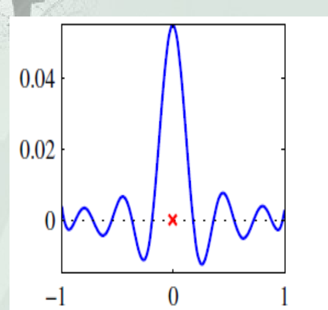
Konstruksi 1

- Contoh lain untuk fungsi basis Gaussian:

$$k(x, x') = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x') = \sum_{i=1}^{M} \phi_i(x)\phi_i(x')$$



$$\phi_i(x) = \exp\left\{-\frac{(x-\mu_i)^2}{2s^2}\right\}$$



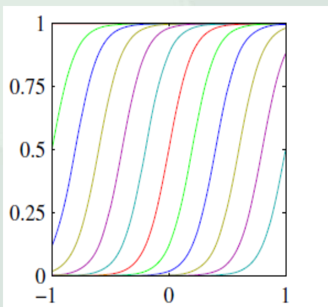
$k(x, x')$ *untuk* $x' = 0$

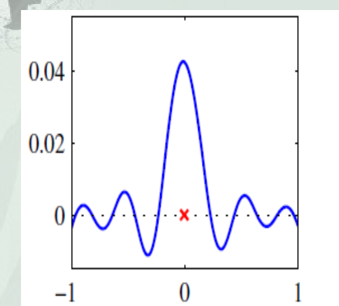
Fungsi Kernel

Konstruksi 1

- Contoh lain untuk fungsi basis sigmoid:

$$k(x, x') = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x') = \sum_{i=1}^{M} \phi_i(x)\phi_i(x')$$





$$\phi_i(x) = \sigma\left(\frac{x-\mu_i}{s}\right) \; dimana \; \sigma(a) = \frac{1}{1+\exp(-a)}$$

$$k(x, x') \; untuk \; x' = 0$$

Fungsi Kernel

Konstruksi 2

- Pendekatan alternatif adalah dengan cara mengkonstruksi fungsi kernel secara langsung. Selanjutnya, kita harus memastikan bahwa fungsi tersebut adalah suatu fungsi kernel yang benar, yaitu suatu *inner product* pada ruang fitur.

  Contoh:

  Pada Contoh 1, k(**x**,**x**') = ($\mathbf{x}^T\mathbf{x}'$)^2 adalah suatu fungsi kernel, karena ia merupakan juga suatu *inner product*, yaitu $k(\mathbf{x},\mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$, pada ruang fitur yang didefiniskan oleh fungsi pemetaan $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$

- Secara umum, untuk menunjukan bahwa $k(\mathbf{x},\mathbf{x}')$ adalah fungsi kernel tanpa secara eksplisit mendefiniskan $\phi(\mathbf{x})$ adalah berdasarkan kondisi berikut ini:

  „Kondisi cukup dan perlu untuk suatu fungsi $k(\mathbf{x},\mathbf{x}')$ menjadi suatu fungsi kernel yang valid adalah bahwa matrix **Gram K**, dimana elemen-elemenya adalah $k(\mathbf{x}_n,\mathbf{x}_m)$, harus semidefinit positip untuk semua pemilihan yang mungkin dari himpunan $\{\mathbf{x}_n\}$

Fungsi Kernel

Konstruksi 3

- Pendekatan lain untuk mengkonstruksi fungsi kernel baru, misal k(**x**,**x**'), adalah menggunakan fungsi kernel yang lebih sederhana yang sudah ada, misal k_1(**x**,**x**') dan k_2(**x**,**x**'), berdasarkan sifat-sifat berikut ini:

1. $k(x,x') = c\, k_1(x,x')$
2. $k(x,x') = f(x)\, k_1(x,x')\, f(x')$
3. $k(x,x') = q\,(k_1(x,x'))$
4. $k(x,x') = \exp(k_1(x,x'))$
5. $k(x,x') = k_1(x,x') + k_2(x,x')$
6. $k(x,x') = k_1(x,x')\, k_2(x,x')$
7. $k(x,x') = k_3(\phi(\mathbf{x}),\phi(\mathbf{x'}))$
8. $k(x,x') = x^T A\, x'$
9. $k(x,x') = k_a(x_a,x_a') + k_b(x_b,x_b')$
10. $k(x,x') = k_a(x_a,x_a')\, k_b(x_b,x_b')$

dimana c > 0 adalah konstanta, f(.) adalah suatu fungsi, q(.) adalah suatu polinomial dengan koefisien nonnegatif, $\phi(\mathbf{x})$ adalah suatu fungsi dari x ke R^M, $k_3(.,.)$ adalah suatu kernel pada R^M, A adalah suatu matrik simetri semidefinit positip, x_a dan x_b adalah variabel $x = (x_a,x_b)$, dan k_a dan k_b adalah fungsi kernel pada ruang yang bersesuaian

Fungsi Kernel

Konstruksi 3

Contoh:

$k(\mathbf{x},\mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||^2/2\sigma^2)$ adalah suatu fungsi kernel dikenal juga dengan nama fungsi kernel 'Gaussian' atau kernel radial basis function (RBF).

Bukti:

Karena $||\mathbf{x} – \mathbf{x}'||^2 = \mathbf{x}^T\mathbf{x} + (\mathbf{x}')^T\mathbf{x}' – 2\,\mathbf{x}^T\mathbf{x}$, maka $k(\mathbf{x},\mathbf{x}') = \exp(-\mathbf{x}^T\mathbf{x}/2\sigma^2)\ \exp(\mathbf{x}^T\mathbf{x}'/\sigma^2)\ \exp(-(\mathbf{x}')^T\mathbf{x}'/2\sigma^2)$.

Dengan menggunakan kondisi 1, 2 dan kondisi 4, dan bahwa $k_1(\mathbf{x},\mathbf{x}') = \mathbf{x}^T\mathbf{x}$ adalah fungsi kernel, maka k(**x**,**x**') adalah suatu fungsi kernel

Fungsi Kernel

Contoh Populer

- Linear :

  $k(x_i, x_j) = x_i^T x_j$

- Polynomial :

  $k(x_i, x_j) = (\gamma x_i^T x_j + r)^d$, dimana $\gamma > 0$

- Radial basis function (RBF) :

$$k(x_i, x_j) = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$$

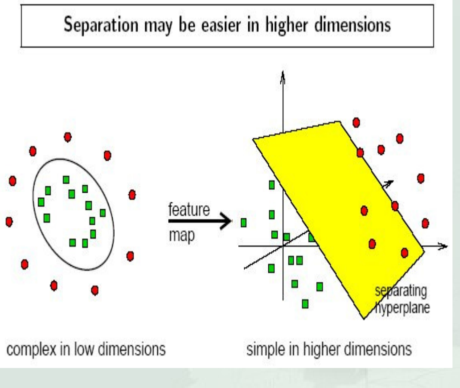
- Sigmoid :

$$k(x_i, x_j) = \tanh(\rho x_i^T x_j + r),$$

$$dimana\ \tanh(a) = 2\sigma(a) - 1,\ dan\ \sigma(a) = \frac{1}{1+\exp(a)}$$

Metode Kernel

Keuntungan



- Fungsi kernel memungkinkan kita untuk mengimplementasikan suatu model pada ruang dimensi lebih tinggi (ruang fitur) tanpa harus mendefinisikan fungsi pemetaan dari ruang input ke ruang fitur (Contoh pada halaman 11)
- Sehingga, untuk kasus yang *non-linearly separable* pada ruang input, diharapkan akan menjadi *linearly separable* pada ruang fitur
- Selanjutnya, kita dapat menggunakan *hyperplane* sebagai *decision boundary* secara efisien

Metode Kernel

Penggunaan

- Secara umum, ada dua cara penggunaan metode kernel pada *machine learning*, yaitu:
  - Penggunaan langsung, yaitu fungsi kernel digunakan sebagai fungsi basis dari model machine learning tersebut, contoh: *radial basis function networks*
  - Penggunaan tidak langsung melalui *kernel trick*, yaitu merepresentasikan suatu model kedalam representasi dual yang mengandung *inner product* dari fungsi pemetaan, contoh: *support vector machine*, *kernel linear regression*, *kernel Perceptron*, dll

Metode Kernel

Kernel Trick

$$\underset{w,b}{arg\,min} \ \frac{1}{2}\|w\|^2$$

$$s.t. \quad t_n(w^T \phi(x_n)+b) \geqslant 1, \ \ n=1,..,N$$

$$\begin{aligned}
\tilde{L}(a) &= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi(x_n)^T \phi(x_n) - \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi(x_n)^T \phi(x_n) - b\sum_{n=1}^{N}\sum_{m=1}^{N} a_n t_n + \sum_{n=1}^{N} a_n \\
&= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi(x_n)^T \phi(x_m) \\
&= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)
\end{aligned}$$

$$\underset{a}{arg\,max} \ \ \tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

$$s.t. \quad a_n \geqslant 0, \ \ n=1,..,N$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

Referensi

- Bishop, C. H., *Pattern Recognition and Machine Learning*, Springer, 2006 (Bab 6.1, Bab 6.2)
- Shawe-Taylor, J., Cristianini, N., *Kernel Method for Pattern Analysis*, Cambridge University Press, 2004