

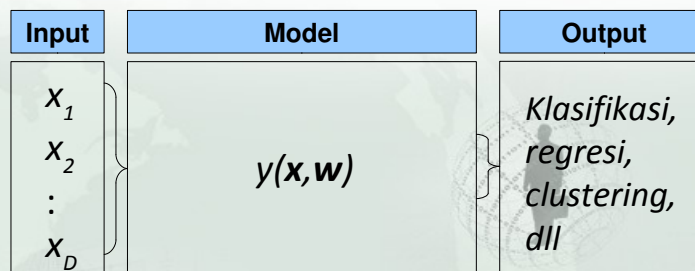
Radial Basis Function Networks

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



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

Learning

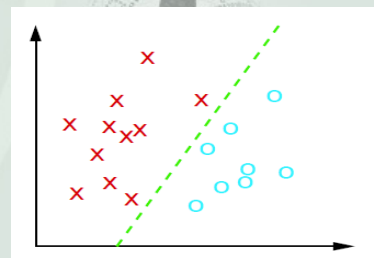
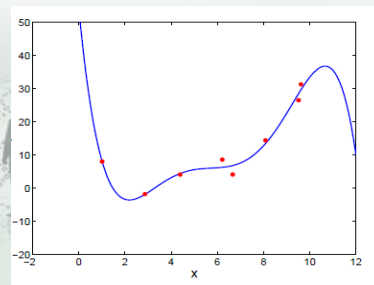
Diberikan data pelatihan $\mathbf{x}_i, i = 1 \text{ sd } N$, dan/atau $\mathbf{t}_i, i = 1 \text{ sd } N$

- **Supervised Learning.** Data pelatihan disertai target, yaitu $\{\mathbf{x}_i, \mathbf{t}_i\}, i = 1 \text{ sd } N$. Tujuan pembelajaran adalah membangun model yang dapat menghasilkan output yang benar untuk suatu data input, misal untuk regresi, klasifikasi, regresi ordinal, ranking, dll
- **Unsupervised Learning.** Data pelatihan tidak disertai target, yaitu $\mathbf{x}_i, i = 1 \text{ sd } N$. Tujuan pembelajaran adalah membangun model yang dapat menemukan komponen/varianbel/fitur tersembunyi pada data pelatihan, yang dapat digunakan untuk: pengelompokan (*clustering*), reduksi dimensi (*dimension reduction*), rekomendasi, dll

3

Supervised Learning

- **Regresi**
 - Nilai output \mathbf{t}_i bernilai kontinu (riil)
 - Bertujuan memprediksi output dari data baru dengan akurat
- **Klasifikasi**
 - Nilai output \mathbf{t}_i bernilai diskrit (kelas)
 - Bertujuan mengklasifikasi data baru dengan akurat



4

Model Linear

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

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

Dimana $\mathbf{x} = (x_1, x_2, \dots, x_D)^T$ adalah variabel input, dan $\mathbf{w} = (w_0, w_1, \dots, w_D)^T$ adalah parameter, $\boldsymbol{\phi}(\mathbf{x})$ 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 $\boldsymbol{\phi}(\mathbf{x})$, misal fungsi linear, fungsi polinomial, fungsi gaussian, fungsi sigmoidal, dll

5

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*)

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

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

1

D^3

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

6

Model Linear

Pendekatan Alternatif

- Pendekatan alternatif adalah membuat fungsi basis adaptif terhadap data pelatihan dengan jumlah fungsi basis ditentukan didepan.
- Dengan kata lain, menggunakan bentuk **parametrik tidak linear** dimana nilai-nilai parameter adaptif terhadap data pelatihan selama proses training.
- Contoh metode yang menggunakan pendekatan ini adalah *neural networks* (NN).

7

Model Linear

Pendekatan Nonparametrik

- Pendekatan lain adalah pendekatan nonparametrik, yaitu menetapkan data pelatihan sebagai pusat-pusat fungsi basis. Selanjutnya, memilih sebagian dari fungsi-fungsi basis tersebut selama proses pelatihan untuk menjadi fungsi-fungsi basis dari model final.
- Dasarnya adalah bahwa data real biasanya memiliki sifat mulus, artinya perubahan sedikit pada data input hanya akan memberikan sedikit perubahan pada output
- Fungsi basis yang banyak digunakan pada pendekatan nonparametrik ini adalah fungsi kernel.

8

Metode Kernel

Fungsi Kernel : Definisi

- Fungsi kernel adalah suatu fungsi k yang mana untuk semua vektor input \mathbf{x}, \mathbf{z} akan memenuhi kondisi

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

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

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

9

Metode Kernel

Fungsi Kernel : Contoh

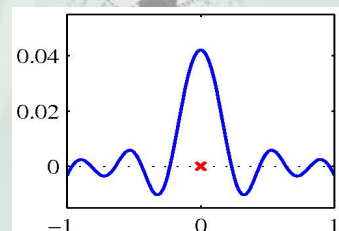
- Salah satu contoh fungsi kernel yang banyak digunakan adalah *Gaussian radial basis function* (RBF), yaitu:

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

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

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

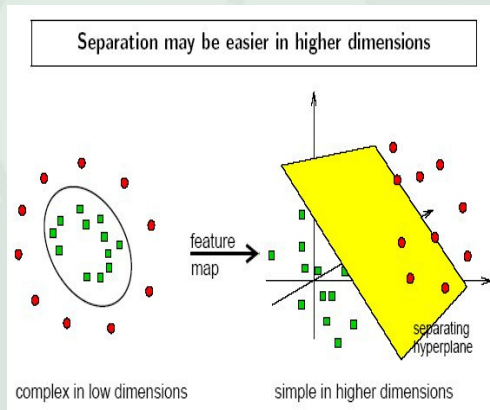
$$\begin{aligned} w * k(x, 5) &= 0.04 * \phi(\|x - 5\|) \\ &= 0.04 * \exp(-\|x - 5\|^2 / 2s^2) \end{aligned}$$



10

Metode Kernel

Fungsi 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
- 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

11

Metode Kernel

Fungsi 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: *kernel linear regression*, *kernel Perceptron*, **support vector machine**, dll

RBF Networks

Definisi

- Radial Basis Function (RBF) networks adalah model linear dimana fungsi basis berupa *radial basis function*, yaitu fungsi yang tergantung pada jarak antara argumennya, yaitu $\phi(\|\mathbf{x}-\mathbf{x}_n\|)$, sehingga modelnya berbentuk:

$$y(x) = \sum_{n=1}^M w_n \phi(\|\mathbf{x}-\mathbf{x}_n\|)$$

dimana w_n adalah parameter bobot, \mathbf{x}_n adalah pusat fungsi basis radial

- RBF networks merupakan suatu *homogeneous kernels*, yaitu:

$$k(\mathbf{x}, \mathbf{x}_n) = k(\|\mathbf{x}-\mathbf{x}_n\|) = \phi(\|\mathbf{x}-\mathbf{x}_n\|)$$

- Misal fungsi kernel Gaussian:

$$\phi(\|\mathbf{x}-\mathbf{x}_n\|) = \exp(-\|\mathbf{x}-\mathbf{x}_n\|^2/2\sigma^2)$$

13

RBF Networks

Formulasi Masalah Pembelajaran

- Asumsikan kita memiliki data pembelajaran $\{\mathbf{x}_n, t_n\}$, $n = 1$ sd N . Selanjutnya, misal semua \mathbf{x}_n dijadikan sebagai pusat dari fungsi basis, maka modelnya akan menjadi:

$$y(x) = \sum_{n=1}^N w_n \phi(\|\mathbf{x}-\mathbf{x}_n\|)$$

Dengan memisalkan $\phi_n(\mathbf{x}) = \phi(\|\mathbf{x}-\mathbf{x}_n\|)$, kita dapat mengekspresikan t_i sebagai:

$$t_i = \sum_{n=1}^N w_n \phi_n(x_i) + e_i, \text{ dimana } i = 1 \text{ sd } N$$

dimana e_i adalah error antara t_i dan $y(\mathbf{x}_i)$

14

RBF Networks

Formulasi Masalah

- Persamaan sebelumnya dapat ditulis dalam bentuk SPL:

$$\mathbf{t} = P\mathbf{w} + \mathbf{e}$$

dimana

$$\mathbf{t} = [t_1 \dots t_N]^T$$

$$\mathbf{e} = [e_1 \dots e_N]^T$$

$$\mathbf{w} = [w_1 \dots w_N]^T$$

$$P = [\phi_1 \dots \phi_N], \text{ dimana } \phi_i = [\phi_i(\mathbf{x}_1) \dots \phi_i(\mathbf{x}_N)]^T, \text{ dengan } i = 1 \text{ sd } N$$

- Selanjutnya, parameter bobot \mathbf{w} dapat diperoleh dengan menggunakan metode *least quares*, yaitu dengan memecahkan SPL normal berikut ini:

$$P^T \mathbf{t} = P^T P \mathbf{w}$$

15

RBF Networks

Contoh Kasus

Seorang ahli biologi telah melakukan eksperimen sebanyak 7 kali untuk melihat pertumbuhan bakteri berdasarkan kadar Nitrogen, dan diperoleh kondisi sbb:

Kadar Nitrogen (gram)	3	4	6	7	8	9
Pertumbuhan Bakteri	1	3	4	6	8	8

Tentukan RBF Networks berdasarkan data tsb !

Solusi:

Dari persoalan diatas diketahui $\mathbf{x} = \{3, 4, 6, 7, 8, 9\}$, $t = \{1, 3, 4, 6, 8, 8\}$, $N=6$

Selanjutnya fungsi basis radial yang digunakan adalah Gaussian dengan $\sigma = 0.5$, yaitu:

$$\phi(||\mathbf{x}-\mathbf{x}_n||) = \exp(-||\mathbf{x}-\mathbf{x}_n||^2/0.5)$$

16

RBF Networks

Contoh Kasus

Dengan menggunakan semua data x sebagai pusat fungsi basis radial, maka:

$$P = \begin{bmatrix} 1.00 & 1.05 & 1.50 & 2.05 & 3.08 & 5.05 \\ 1.05 & 1.00 & 1.20 & 1.50 & 2.05 & 3.08 \\ 1.50 & 1.20 & 1.00 & 1.05 & 1.20 & 1.50 \\ 2.05 & 1.50 & 1.05 & 1.00 & 1.05 & 1.20 \\ 3.08 & 2.05 & 1.20 & 1.05 & 1.00 & 1.05 \\ 5.05 & 3.08 & 1.50 & 1.20 & 1.05 & 1.00 \end{bmatrix}$$

Dengan menyelesaikan SPL $t = Pw$ menggunakan metode *least squares*, yaitu menyelesaikan SPL $P^t t = P^T P w$, maka parameter bobot dapat ditentukan, yaitu:

$$w = [227, -1029, 5387, -8457, 4744, -884]^T$$

Sehingga RBF Networks yang terbentuk adalah

$$y(x) = 227 * \phi(\|x - 3\|) - 1029 * \phi(\|x - 4\|) + 5387 * \phi(\|x - 6\|) \\ - 8457 * \phi(\|x - 7\|) + 4744 * \phi(\|x - 8\|) - 884 * \phi(\|x - 9\|)$$

17

RBF Networks

Metode Umum

- Karena ada satu fungsi basis untuk setiap data pembelajaran, maka model yang terbentuk akan menjadi mahal secara komputasi ketika melakukan prediksi untuk data baru. Untuk itu dibutuhkan mekanisme untuk menentukan lokasi pusat fungsi basis yang optimal, antara lain:
 - Dipilih secara acak dari data pembelajaran
 - Menggunakan algoritma *unsupervised learning*, misal: metode *K-Means*
 - Pendekatan secara sistematis, misal: metode *orthogonal least square*
- Setelah jumlah lokasi pusat fungsi basis ditentukan, misal M , maka parameter bobot dapat ditentukan dengan menggunakan metode *least squares*

$$y(x) = \sum_{n=1}^M w_n \phi(\|x - x_n\|)$$

18

RBF Networks

Pemilihan Secara Acak

1) Dengan memilih 3, 6, dan 9 sebagai pusat fungsi basis radial, maka:

$$P = \begin{bmatrix} 1.00 & 1.50 & 5.05 \\ 1.05 & 1.20 & 3.08 \\ 1.50 & 1.00 & 1.50 \\ 2.05 & 1.05 & 1.20 \\ 3.08 & 1.20 & 1.05 \\ 5.05 & 1.50 & 1.00 \end{bmatrix}$$

Dengan menyelesaikan SPL $t = Pw$ menggunakan metode *least squares*, yaitu menyelesaikan SPL $P^t t = P^T P w$, maka parameter bobot dapat ditentukan, yaitu:

$$w = [-0.80, 10.15, -2.68]^T$$

Sehingga RBF Networks yang terbentuk adalah

$$y(x) = -0.80 * \phi(| |x - 3| |) + 10.15 * \phi(| |x - 6| |) - 2.68 * \phi(| |x - 9| |)$$

19

RBF Networks

Pemilihan Dengan Unsupervised Learning

2) Dengan memilih rata-rata nilai x , yaitu $x = 6.1667$, sebagai pusat fungsi basis radial, maka:

$$P = \begin{bmatrix} 3.50 \\ 1.80 \\ 1.00 \\ 1.09 \\ 1.52 \\ 2.73 \end{bmatrix}$$

Dengan menyelesaikan SPL $t = Pw$ menggunakan metode *least squares*, yaitu menyelesaikan SPL $P^t t = P^T P w$, maka parameter bobot dapat ditentukan, yaitu:

$$w = [1.95]^T$$

Sehingga RBF Networks yang terbentuk adalah

$$y(x) = 1.95 * \phi(| |x - 6.1667| |)$$

20

RBF Networks

Pemilihan Secara Sistematis

- Selain dengan menggunakan metode *least squares* biasa seperti yang sudah dijelaskan sebelumnya, parameter bobot \mathbf{w} dapat juga diperoleh dengan menggunakan metode *orthogonal least squares (OLS)*
- Selain memiliki propertis numerik yang lebih superior dibandingkan dengan metode *least squares* biasa, metode OLS juga memungkinkan untuk memilih lokasi pusat fungsi basis secara lebih sistematis

21

Orthogonal Least Squares

Metode Faktorisasi Matrik

- Metode OLS menggunakan metode faktorisasi matrik untuk menyelesaikan SPL ($\mathbf{t} = P\mathbf{w} + \mathbf{e}$), yaitu metode yang akan memfaktorkan matrik koefisien (P) menjadi matrik ortogonal
- Ada beberapa metode faktorisasi matrik ortogonal yang dapat digunakan, antara lain:
 - **Faktorisasi QR**, dengan menggunakan teknik Householder Reflection, Given Rotation, Classical Gram-Schmidt, dll
 - **Single Value Decomposition (SVD)**

22

Orthogonal Least Squares

Penentuan Nilai Bobot

$$t = P w + e$$

Faktorisasi QR --> $P = SA$

S adalah matrik orthogonal ($S^T S = \text{diagonal}$)

$$A = \begin{bmatrix} 1 & a_{1,2} & \dots & a_{1,N} \\ 0 & 1 & \vdots & \vdots \\ \vdots & \vdots & \ddots & a_{N-1,N} \\ 0 & \dots & 0 & 0 \end{bmatrix}, \text{ matrik segitiga atas}$$

$$(1) t = S g + e$$

$$g = (S^T S)^{-1} S^T t = D^{-1} S^T t \text{ (solusi dengan metode least squares)}$$

$$(2) A w = g$$

A dan g diketahui maka w dapat dicari dengan substitusi balik

23

Orthogonal Least Squares

Faktorisasi QR: Metode Classical Gram-Schmidt

Matrik P difaktorkan menjadi matrik orthogonal S dan matrik segitiga atas A dengan cara sbb:

$$s_1 = p_1$$

untuk $k=2, \dots, N$:

$$\alpha_{ik} = \frac{w_i^T p_k}{w_i^T w_i}, 1 \leq i < k$$

$$s_k = p_k - \sum_{i=1}^{k-1} \alpha_{ik} s_i$$

Contoh:

Faktorisasi QR dari:

$$P = \begin{bmatrix} 1 & 5 \\ 5 & 1 \end{bmatrix}$$

adalah

$$S = \begin{bmatrix} 1 & 4.6154 \\ 5 & -0.9231 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0.3846 \\ 0 & 1 \end{bmatrix}$$

24

Orthogonal Least Squares

Pemilihan Lokasi Pusat Fungsi Basis

Metode OLS dapat digunakan untuk mendefinisikan lokasi pusat fungsi basis yang signifikan berdasarkan suatu rasio pereduksi error (*error reduction ratio (err)*) sbb:

$$t = Sg + e \longrightarrow t^T t = \sum_{i=1}^N g_i^2 s_i^T s_i + e^T e$$

$$e^T e = t^T t - \sum_{i=1}^N g_i^2 s_i^T s_i \longrightarrow \frac{e^T e}{t^T t} = 1 - \frac{\sum_{i=1}^N g_i^2 s_i^T s_i}{t^T t}$$

$$[err]_i = \frac{g_i^2 s_i^T s_i}{t^T t}, 1 \leq i \leq N$$

Metode OLS akan memilih data pembelajaran yang memiliki nilai *err* terbesar pada setiap iterasi sebagai pusat fungsi basis

25

Orthogonal Least Squares

Algoritma

- Diberikan data pembelajaran $\{x_r, t_r\}, n=1 \text{ sd } N$ dan suatu fungsi basis radial $\phi(\cdot)$
- Bentuk matrik koefisien P (semua data dijadikan pusat)

Langka Pertama, untuk $1 \leq i \leq M$

$$s_1^{(i)} = p_i$$

$$g_1^{(i)} = \frac{(s_1^{(i)})^T t}{(s_1^{(i)})^T s_1^{(i)}}$$

$$[err]_1^{(i)} = \frac{(g_1^{(i)})^2 (s_1^{(i)})^T s_1^{(i)}}{t^T t}$$

cari

$$[err]_1^{(i_1)} = \max [err]_1^{(i)}, 1 \leq i \leq M$$

dan pilih

$$s_1 = s_1^{(i_1)} = p_{(i_1)}$$

$$g_1 = g_1^{(i_1)}$$

- Iterasi pertama adalah untuk menentukan kolom pertama matrik $S (s_1)$ yang dipilih dari vektor kolom matrik P yang akan memberikan nilai *err* terbesar atau yang akan mereduksi error terbesar
- Selanjutnya solusi *least squares* yang bersesuaian juga ditentukan, yaitu nilai g_1

26

Orthogonal Least Squares

Algoritma

Langka ke- k ($k \geq 2$), untuk $1 \leq i \leq M$

$$\alpha_{jk}^{(i)} = \frac{w_j^T p_i}{w_j^T w_j}, 1 \leq j < k$$

$$s_k^{(i)} = p_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} s_j$$

$$g_k^{(i)} = \frac{(s_k^{(i)})^T t}{(s_k^{(i)})^T s_k^{(i)}}$$

$$[err]_k^{(i)} = \frac{(g_k^{(i)})^2 (s_k^{(i)})^T s_k^{(i)}}{t^T t}$$

cari

$$[err]_k^{(i_k)} = \max [err]_k^{(i)}, 1 \leq i \leq M, i \neq i_1, \dots, i_{k-1}$$

dan pilih

$$s_k = s_k^{(i_k)}$$

$$g_k = g_k^{(i_k)}$$

$$\alpha_{jk} = \alpha_{jk}^{(i_k)}$$

- Algoritma disamping adalah penentuan vektor kolom matrik S selanjutnya (s_k), vektor kolom matrik A (α_{jk}), dan solusi *least square* g_k

- Pemilihan pusat akan berhenti jika

$$1 - \sum_{j=1}^M [err]_j < \rho$$

dimana $0 < \rho < 1$ adalah suatu toleransi.

27

Regularized OLS

Regularization

- Pada aplikasi praktis, kita sering menemukan kondisi dimana ketersediaan data pembelajaran terbatas, sehingga fenomena *overfitting*, yaitu peningkatan error diluar data pembelajaran, sering terjadi
- Salah satu teknik yang umum digunakan untuk mengontrol fenomena *overfitting* adalah *regularization*, yaitu menambahkan suatu pinalti (*regularizer*) yang berupa parameter bobot pada fungsi error, yaitu: $e^T e + \lambda w^T w$
- Penambahan pinalti tsb bertujuan supaya bobot yang dihasilkan berupa bilangan yang tidak terlalu besar, sehingga fluktuasi prediksi tidak terlalu besar (kurva menjadi lebih mulus)

28

Regularized OLS

Penentuan Nilai Bobot

$$t = P w + e$$

Faktorisasi QR --> $P = SA$

S adalah matrik orthogonal ($S^T S = \text{diagonal}$)

$$A = \begin{bmatrix} 1 & a_{1,2} & \dots & a_{1,N} \\ 0 & 1 & \dots & \vdots \\ \vdots & \vdots & \ddots & a_{N-1,N} \\ 0 & \dots & 0 & 0 \end{bmatrix}, \text{ matrik segitiga atas}$$

(1) $t = S g + e$

$g = (S^T S + \lambda I_N)^{-1} S^T t = D^{-1} S^T t$ (solusi dengan metode *least squares*)

(2) $A w = g$

A dan g diketahui maka w dapat dicari dengan substitusi balik

29

Regularized OLS

Pemilihan Lokasi Pusat Fungsi Basis

Metode OLS dapat digunakan untuk mendefinisikan lokasi pusat fungsi basis yang signifikan berdasarkan suatu rasio pereduksi error (*regularized error reduction ratio - rerr*) sbb:

$$t = S g + e \longrightarrow t^T t = \sum_{i=1}^N g_i^2 s_i^T s_i + e^T e$$

$$e^T e + \lambda g^T g = t^T t - \sum_{i=1}^N g_i^2 (s_i^T s_i + \lambda)$$

$$\frac{(e^T e + \lambda g^T g)}{t^T t} = 1 - \frac{\sum_{i=1}^N g_i^2 (s_i^T s_i + \lambda)}{t^T t}$$

$$[rerr]_i = \frac{g_i^2 (s_i^T s_i + \lambda)}{t^T t}, \quad 1 \leq i \leq N$$

Metode ROLS akan memilih data pembelajaran yang memiliki nilai *rerr* terbesar pada setiap iterasi sebagai pusat fungsi basis

30

Regularized OLS

Algoritma

- Diberikan data pembelajaran $\{\mathbf{x}_r, \mathbf{t}_r\}$, $n=1$ sd N dan suatu fungsi basis radial $\phi(\cdot)$
- Bentuk matrik koefisien P (semua data dijadikan pusat)

Langka Pertama, untuk $1 \leq i \leq M$

$$s_1^{(i)} = p_i$$

$$g_1^{(i)} = \frac{(s_1^{(i)})^T \mathbf{t}}{((s_1^{(i)})^T s_1^{(i)} + \lambda)}$$

$$[err]_1^{(i)} = \frac{(g_1^{(i)})^2 ((s_1^{(i)})^T s_1^{(i)} + \lambda)}{\mathbf{t}^T \mathbf{t}}$$

cari

$$[err]_1^{(i)} = \max [err]_1^{(i)}, 1 \leq i \leq M$$

dan pilih

$$s_1 = s_1^{(i)} = p_{(i)}$$

$$g_1 = g_1^{(i)}$$

- Iterasi pertama adalah untuk menentukan kolom pertama matrik S (s_1) yang dipilih dari vektor kolom matrik P yang akan memberikan nilai *err* terbesar atau yang akan mereduksi error terbesar
- Selanjutnya solusi *least quares* yang bersesuaian juga ditentukan, yaitu nilai g_1

31

Regularized OLS

Algoritma

Langka ke-k ($k \geq 2$), untuk $1 \leq i \leq M$

$$\alpha_{jk}^{(i)} = \frac{w_j^T p_i}{w_j^T w_j}, 1 \leq j < k$$

$$s_k^{(i)} = p_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} s_j$$

$$g_k^{(i)} = \frac{(s_k^{(i)})^T \mathbf{t}}{((s_k^{(i)})^T s_k^{(i)} + \lambda)}$$

$$[err]_k^{(i)} = \frac{(g_k^{(i)})^2 ((s_k^{(i)})^T s_k^{(i)} + \lambda)}{\mathbf{t}^T \mathbf{t}}$$

cari

$$[err]_k^{(i)} = \max [err]_k^{(i)}, 1 \leq i \leq M, i \neq i_1, \dots, i_{k-1}$$

dan pilih

$$s_k = s_k^{(i)}$$

$$g_k = g_k^{(i)}$$

$$\alpha_{jk} = \alpha_{jk}^{(i)}$$

- Algoritma disamping adalah penentuan vektor kolom matrik S selanjutnya (s_k), vektor kolom matrik A (α_{jk}), dan solusi *least square* g_k
- Pemilihan pusat akan berhenti jika

$$1 - \sum_{j=1}^M [err]_j < \rho$$

dimana $0 < \rho < 1$ adalah suatu toleransi.

32

Simulasi

Mackey-Glass Time Series Prediction

- Data simulasi dibagun dengan menggunakan persamaan berikut ini:

$$s(t+1) = \begin{cases} (1-b) * s(t-1) + \frac{a * s(t-\tau)}{1 + s(t-\tau)^{10}} & \text{untuk } t \geq \tau + 1 \\ 0.3 & \text{untuk } 1 \leq t \leq \tau \end{cases}$$

dimana $a = 0.2$, $b = 0.1$, $\tau = 18$

- Data simulasi tersebut terdiri dari 1000 titik dengan diberikan noise berdistribusi $N(0, 0.1)$, dimana 500 data pertama sebagai data pembelajaran, dan 500 lainnya sebagai data pengujian
- Masalah: memprediksi nilai $s(t+1)$ berdasarkan nilai-nilai $s(t)$, $s(t-1)$, ..., $s(t-5)$
- Model: masalah regresi dengan fitur input berdimensi 6, yaitu $[s(t), s(t-1), s(t-2), s(t-3), s(t-4), s(t-5)]$, dan target adalah bilangan real, yaitu $[s(t+1)]$

33

Simulasi

Mackey-Glass Time Series Prediction

- Fungsi basis yang digunakan adalah fungsi Gaussian, yaitu: $\phi(||\mathbf{x}-\mathbf{x}_n||) = \exp(-||\mathbf{x}-\mathbf{x}_n||^2/2\sigma^2)$
- Parameter model, yaitu σ dan λ , dapat ditentukan dengan menggunakan teknik *cross-validation*, dimana untuk algoritma OLS hanya parameter σ yang perlu dioptimasi. Pada simulasi ini digunakan algoritma genetik untuk optimasi parameter tsb.
- Akurasi prediksi diukur dengan menggunakan *root mean squares error* (RMSE), yaitu:

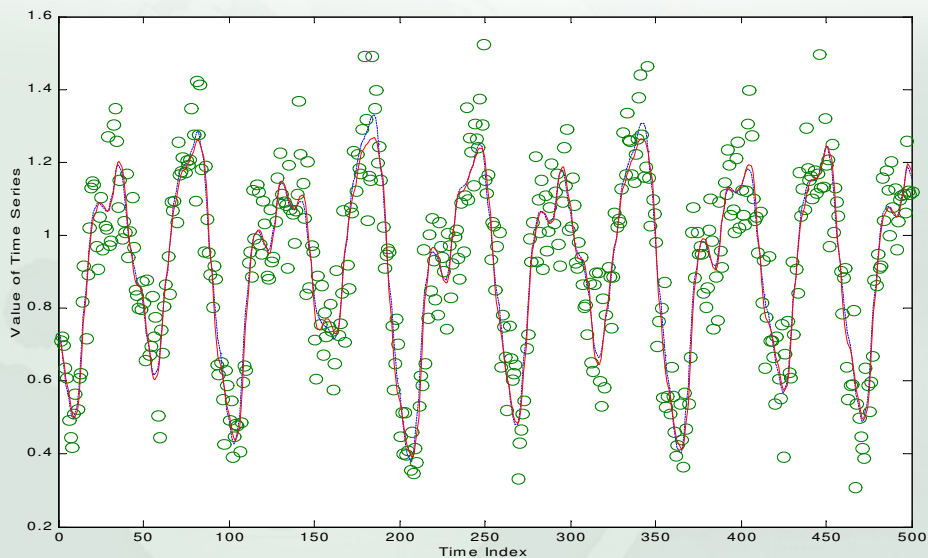
$$RMSE = \sqrt{MSE} = \sqrt{\sum_{i=1}^N (t_i - y(x_i))^2}$$

- Hasil:
 - Algoritma OLS memberikan akurasi RMSE sebesar 0.000417
 - Algoritma ROLS memberikan akurasi RMSE sebesar 0.000294

34

Simulasi

Mackey-Glass Time Series Prediction



Ket: garis putus-putus biru adalah fungsi Mackey-Glass, o adalah data pembelajaran, garis merah adalah prediksi dari ROLS

35

Simulasi

Composit Stock Index Prediction

- Data index saham gabungan (*composite stock index*) untuk simulasi adalah data dari Bursa Efek Jakarta
- Data pembelajaran berjumlah 214 (tahun 2001), sementara data pengujian berjumlah 242 (tahun 2002)
- Fitur yang umum digunakan untuk masalah ini antara lain: moving average (MA), momentum (M), relative strength index (RSI), stochastic (%K), moving average of stochastic (%D)
- Masalah: memprediksi harga index saham gabungan hari berikutnya $s(t+1)$ berdasarkan beberapa fitur, yaitu: $s(t)$, $s(t-1)$, MA_5 , MA_{10} , RSI_5 dan M
- Model: masalah regresi dengan fitur input berdimensi 6, yaitu $[s(t), s(t-1), MA_5, MA_{10}, RSI_5, M]$, dan target adalah bilangan real, yaitu $[s(t+1)]$

36

Simulasi

Composit Stock Index Prediction

- Fungsi basis yang digunakan adalah fungsi Gaussian, yaitu: $\phi(||\mathbf{x}-\mathbf{x}_n||) = \exp(-||\mathbf{x}-\mathbf{x}_n||^2/2\sigma^2)$
- Parameter model, yaitu σ dan λ , dapat ditentukan dengan menggunakan teknik *cross-validation*, dimana untuk algoritma OLS hanya parameter σ yang perlu dioptimasi. Pada simulasi ini digunakan algoritma genetik untuk optimasi parameter tsb.
- Akurasi prediksi diukur dengan menggunakan *root mean squares error* (RMSE), yaitu:

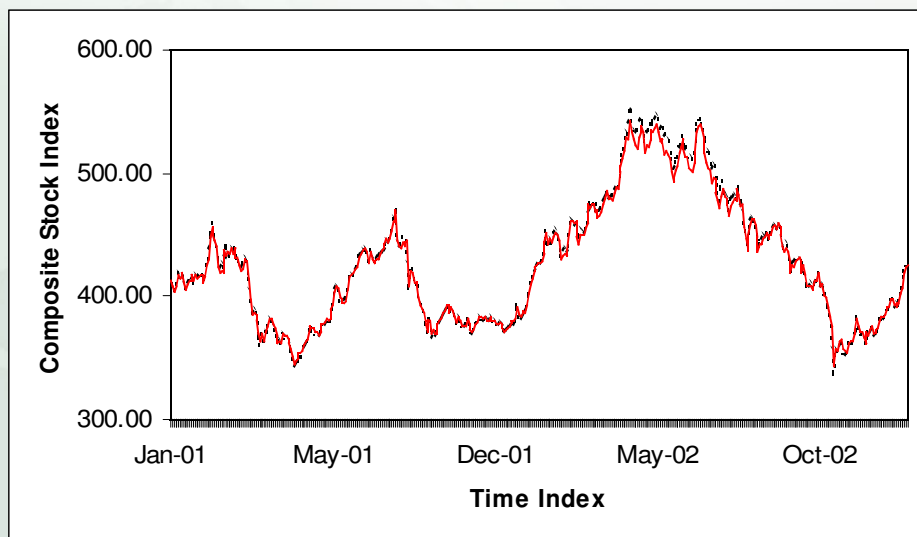
$$RMSE = \sqrt{MSE} = \sqrt{\sum_{i=1}^N (t_i - y(x_i))^2}$$

- Hasil:
 - Algoritma ROLS memberikan akurasi RMSE sebesar 0.141899

37

Simulasi

Composit Stock Index Prediction



Ket: garis putus-putus biru adalah data target, garis merah adalah prediksi dari ROLS

38



Referensi

- Bishop, C. H., *Pattern Recognition and Machine Learning*, Springer, 2006 (Bab 6.3)
- Chen, S., Cowan, C. F. N., Grant, P. M., *Orthogonal Least Squares Learning Algorithm for Radial Basis Functions Networks*, IEEE Transactions on Neural Networks, Vol. 2, No. 2, 1991
- Chen, S., Chng, E. S., Alkadhimi, K., *Regularized Orthogonal Least Squares Algorithm for Constructing Radial Basis Function Networks*, Int. J. Control, Vol. 64, No. 5, 1996
- Murfi, H., Kerami, D., *Regularized Orthogonal Least Squares-based Radial Basis Function Networks and Its Application on Time Series Prediction*, Proceeding of SEAMS ICOMA'03, 2003