SVM untuk Regresi

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

- *Preprocessing*: ekstraksi fitur dan representasi data, misal dalam bentuk vektor $\mathbf{x}_i = (x_1, x_2, .., x_D)^T$
- *Learning*: pemilihan model dan penentuan parameter model, 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 model.

Learning

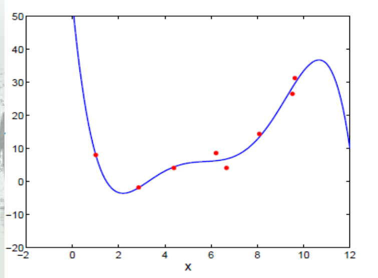
Diberikan data pelatihan $\mathbf{x}_i$, $i = 1$ sd N, dan/atau $\mathbf{t}_i$, $i = 1$ sd N

- 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

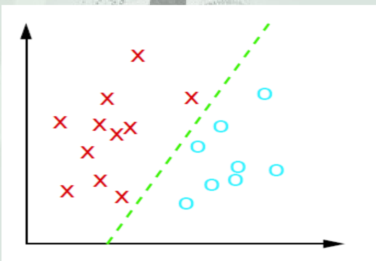
- Unsupervised Learning. Data pelatihan tidak disertai target, yaitu $\mathbf{x}_i$, i = 1 sd N. Tujuan pembelajaran adalah membagun model yang dapat menemukan komponen/variabel/fitur tersembunyi pada data pelatihan, yang dapat digunakan untuk: pengelompokan (*clustering*), reduksi dimensi (*dimension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

Supervised Learning

- Regresi
  - Nilai output t_i bernilai kontinu (riil)
  - Bertujuan memprediksi output dari data baru dengan akurat



- Klasifikasi
  - Nilai output 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 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).

Model Linear

Pendekatan Nonparametrik

- Pendekatan lain adalah pendekatan nonparametrik, yaitu menetapkan data pelatihan sebagai pusat-pusat fungsi basis. Selanjutnya, memilih sebagian dari fungsi-funsgi 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.

Metode Kernel

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 perkalian dalam (*inner product*) pada ruang fitur.

Metode Kernel

Fungsi Kernel : Contoh

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

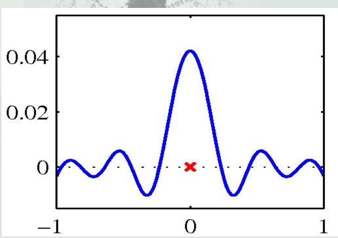
$$k(\mathbf{x},\mathbf{x}') = \phi(||\mathbf{x}-\mathbf{x}'||) = \exp(-||\mathbf{x}-\mathbf{x}'||^2 / 2s^2)$$

  dimana **x**' adalah „inti“ yang dipilih dari data pelatihan.

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

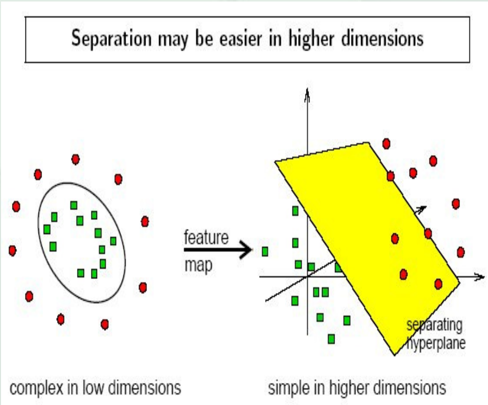
$$w * k(x,5) = 0.04 * \phi(||x-5||)$$

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



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

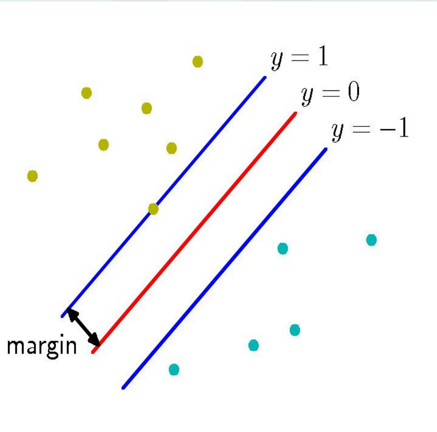
Metode Kernel

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

Support Vector Machine

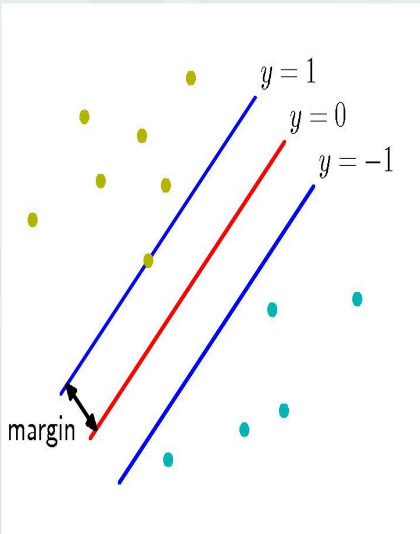
Maximum Margin



- Untuk menentukan *decision boundary* (DB), yaitu suatu model linear atau *hyperplane* y(**x**) dengan parameter **w** dan b, SVM menggunakan konsep *margin* yang didefiniskan sebagai jarak terdekat antara DB dengan sembarang data training
- Dengan memaksimumkan *margin*, maka akan didapat suatu DB tertentu

Support Vector Machine

Maximum Margin



- Kenapa maksimum ?
- Berdasarkan intuisi, margin maksimum adalah pilihan yang aman karena jika terjadi sedikit kesalahan pada data maka akan memberikan kemungkinan terkecil terjadi kesalahan klasifikasi
- Berdasarkan teori, yang merupakan basis dari metode SVM, maksimum margin akan memberikan kapabilitas generalisasi terbaik (VC theory, 1960-1990)

SVM untuk Regresi

Bentuk Umum

- Model linear yang akan digunakan sebagai fungsi regresi memiliki bentuk umum sbb:

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

dimana ***x*** adalah vektor input, ***w*** adalah parameter bobot, $\phi(\mathbf{x})$ adalah fungsi transformasi fitur, dan *b* adalah suatu bias

SVM untuk Regresi

Formulasi Masalah

- Diberikan data pembelajaran $\{\mathbf{x}_n, t_n\}$, n = 1 ... N. Pada model regresi linear sebelumnya, untuk mendapatkan parameter bobot ***w*** dan bias *b*, kita meminimumkan fungsi error yang diregularisasi berikut ini:

$$E(w,b) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n)-t_n)^2 + \frac{\lambda}{2}\|w\|^2$$

- Pada model SVR, kita akan meminimumkan fungsi error yang diregularisasi berikut ini:

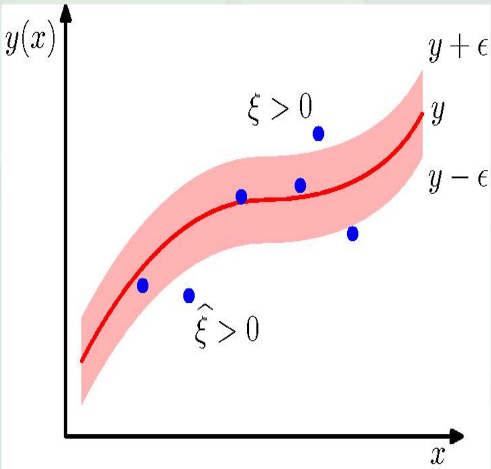
$$E(w,b) = C\sum_{n=1}^{N}E_\varepsilon(y(x_n)-t_n) + \frac{1}{2}\|w\|^2$$

dimana $E_\varepsilon(.)$ adalah fungsi error *ε-insensitive* yang didefinisikan sbb:

$$E_\varepsilon(y(x_n)-t_n) = \begin{cases} 0, & jika\ |y(x_n)-t_n| < \varepsilon \\ |y(x_n)-t_n|-\varepsilon, & jika\ lainnya \end{cases}$$

SVM untuk Regresi

Formulasi Masalah

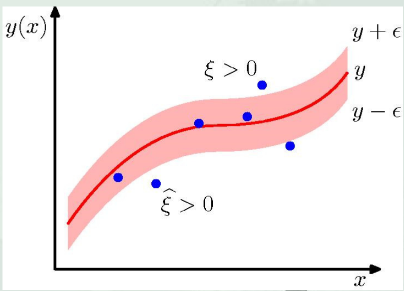


- Untuk data target t_n yang terletak pada *ε-tube* akan memenuhi kondisi:
  - $t_n \leq y(x_n) + \varepsilon$
  - $t_n \geq y(x_n) - \varepsilon$
- Untuk data target t_n yang terletak diluar *ε-tube,* maka seperti pendekatan *soft-margin*, kita membutuhkan varibel *slack* ξ_n, $\bar{\xi}_n$ dimana $\xi_n > 0$ untuk $t_n > y(\mathbf{x}_n) + \varepsilon$, dan $\bar{\xi}_n > 0$ untuk $t_n < y(\mathbf{x}_n) - \varepsilon$, sehingga:
  - $t_n \leq y(\mathbf{x}_n) + \varepsilon + \xi_n$
  - $t_n \geq y(\mathbf{x}_n) - \varepsilon - \bar{\xi}_n$

SVM untuk Regresi

Formulasi Masalah

- Selanjutnya, penentuan parameter model SVR dapat diformulasikan sebagai masalah optimasi sbb:



$$\underset{w,b}{\arg\min} \quad C\sum_{n=1}^{N}(\xi_n + \bar{\xi}_n) + \frac{1}{2}\|w\|^2$$

$$\begin{aligned} s.t. \quad & \xi_n \geq 0, \quad \bar{\xi}_n \geq 0 \\ & t_n \leq y(x_n) + \varepsilon + \xi_n \\ & t_n \geq y(x_n) - \varepsilon - \bar{\xi}_n \end{aligned}$$

- Dengan kata lain, penentuan nilai parameter **w** dan b menjadi masalah pemrograman kuadrat (*quadratic programming*), yaitu meminimumkan suatu fungsi kuadrat dengan syarat suatu pertidaksamaan linear

SVM untuk Regresi

Bentuk Dual: *Lagrange Multipliers*

- Untuk menyelesaikan pemrograman kuadrat tersebut, cara yang umum digunakan adalah mencari bentuk dual dengan menggunakan perkalian Lagrange (*Lagrange multipliers*) $a_n \geq 0$, $\bar{a}_n \geq 0$, $\mu_n \geq 0$, $\bar{\mu}_n \geq 0$ dengan satu pengali Lagrange untuk setiap kendala, untuk membentuk fungsi Lagrangian (*Lagrangian function*) sbb:

$$L(w, b, a, \bar{a}, \mu, \bar{\mu}) = C\sum_{n=1}^{N}(\xi_n + \bar{\xi}_n) + \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N}(\mu_n \xi_n + \bar{\mu}_n \bar{\xi}_n)$$
$$- \sum_{n=1}^{N} a_n(\varepsilon + \xi_n + y(x_n) - t_n) - \sum_{n=1}^{N} \bar{a}_n(\varepsilon + \bar{\xi}_n - y(x_n) + t_n)$$

SVM untuk Regresi

Bentuk Dual: *Lagrange Multipliers*

- Dengan menurunkan *L* terhadap **w,** *b*, ξ_n, dan $\bar{\xi}_n$ sama dengan nol, maka:

$$\frac{\partial L}{\partial w} = w - \sum_{n=1}^{N}(a_n - \bar{a}_n)\phi(x_n) = 0 \rightarrow w = \sum_{n=1}^{N}(a_n - \bar{a}_n)\phi(x_n)$$

$$\frac{\partial L}{\partial b} = \sum_{n=1}^{N} a_n - \sum_{n=1}^{N} \bar{a}_n = 0 \rightarrow \sum_{n=1}^{N}(a_n - \bar{a}_n) = 0$$

$$\frac{\partial L}{\partial \xi_n} = C - a_n - \mu_n = 0 \rightarrow a_n + \mu_n = C$$

$$\frac{\partial L}{\partial \bar{\xi}_n} = C - \bar{a}_n - \bar{\mu}_n = 0 \rightarrow \bar{a}_n + \bar{\mu}_n = C$$

SVM untuk Regresi

Bentuk Dual: Fungsi Kernel

- Substitusikan hasil turunan tsb ke persamaan Lagrangian, yaitu:

$$L = C\sum_{n=1}^{N}(\xi_n+\bar{\xi}_n)+\frac{1}{2}\|w\|^2 - \sum_{n=1}^{N}(\mu_n\xi_n+\bar{\mu}_n\bar{\xi}_n)$$
$$-\sum_{n=1}^{N}a_n(\varepsilon+\xi_n+y(x_n)-t_n) - \sum_{n=1}^{N}\bar{a}_n(\varepsilon+\bar{\xi}_n-y(x_n)+t_n)$$

sehingga menjadi:

$$\tilde{L}(a,\bar{a}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n-\bar{a}_n)(a_m-\bar{a}_m)k(x_n,x_m)$$
$$-\varepsilon\sum_{n=1}^{N}(a_n+\bar{a}_n)+\sum_{n=1}^{N}(a_n-\bar{a}_n)t_n$$

SVM untuk Regresi

Bentuk Dual

- Sehingga, bentuk *dual* dari masalah SVR adalah:

$$\underset{a,\bar{a}}{arg\ max} \quad -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n-\bar{a}_n)(a_m-\bar{a}_m)k(x_n,x_m)$$
$$-\varepsilon\sum_{n=1}^{N}(a_n+\bar{a}_n)+\sum_{n=1}^{N}(a_n-\bar{a}_n)t_n$$

$$s.t. \quad 0\leqslant a_n\leqslant C, \quad 0\leqslant\bar{a}_n\leqslant C$$
$$\sum_{n=1}^{N}(a_n-\bar{a}_n)=0$$

- Representasi dual yang dihasilkan adalah berupa pemrograman kuadrat juga, akan tetapi memiliki kendala yang lebih sederhana (*bound-constrained optimization*).

SVM untuk Regresi

Solusi Bentuk Dual: Algoritma

- Ada beberapa algoritma dan perangkat lunak yang telah dikembangkan untuk memecahkan masalah optimisasi dari SVM, antara lain:
  - SMO [2]
  - LibSVM [3] (http://www.csie.ntu.edu.tw/~cjlin/libsvm)
  - SVM^{Light} [4] (http://svmlight.joachims.org)

SVM untuk Regresi

Solusi Bentuk Dual: Nilai Bobot

- Misal, solusi dari pemrograman kuadrat bentuk dual tersebut adalah $\boldsymbol{a}^*$ dan $\bar{\boldsymbol{a}}^*$, maka:

$$w = \sum_{n=1}^{N} (a_n^* - \bar{a}_n^*) \phi(x_n)$$

$$y(x) = w^T \phi(x) + b = \sum_{n=1}^{N} (a_n^* - \bar{a}_n^*) k(x, x_n) + b$$

- Hanya data dengan $a_n^* > 0$, $\bar{a}_n^* > 0$ (*support vectors*) yang berperan pada model SVR, sehingga dapat ditulis menjadi:

$$y(x) = \sum_{m \in S} (a_m^* - \bar{a}_m^*) k(x, x_m) + b$$

dimana *S* adalah himpunan indeks dari *support vectors*

SVM untuk Regresi

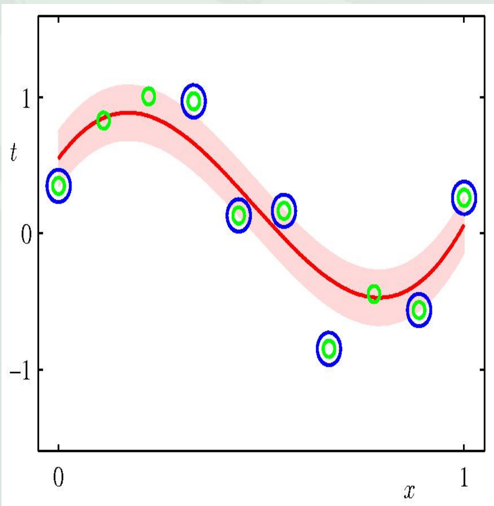
Solusi Bentuk Dual: Karush-Kuhn-Tucker

Karush-Kuhn-Tucker (KKT) conditions yang menyatakan bahwa solusi yang diperoleh adalah optimal, atau pada solusi tersebut variabel dan kendala dari bentuk dual bertemu, adalah sbb:

$$\begin{aligned} a_n(\epsilon + \xi_n + y_n - t_n) &= 0 \qquad [KKT-1] \\ \bar{a}_n(\epsilon + \bar{\xi}_n - y_n + t_n) &= 0 \qquad [KKT-2] \\ (C - a_n)\xi_n &= 0 \qquad [KKT-3] \\ (C - \bar{a}_n)\bar{\xi}_n &= 0 \qquad [KKT-4] \end{aligned}$$

SVM untuk Regresi

Solusi Bentuk Dual: Support Vectors



- *Support vectors* adalah data training dengan nilai $a_n^* > 0$, $\bar{a}_n^* > 0$
- Berdasarkan [KKT-1] maka $\epsilon + \xi_n + y_n - t_n = 0$. Dengan kata lain, data training terletak pada *boundary* atas ($\xi_n = 0$) atau diatas *boundary* atas ($\xi_n > 0$)
- Berdasarkan [KKT-2] maka $\epsilon + \bar{\xi}_n - y_n + t_n = 0$. Dengan kata lain, data training terletak pada *boundary* bawah ($\xi_n = 0$) atau dibawah *boundary* bawah ($\xi_n > 0$)

SVM untuk Regresi

Solusi Bentuk Dual: Nilai Bias

- Selanjutnya, *b* dapat dicari dengan memperhatikan kondisi sebelumnya, sbb:

$$\begin{array}{ll} (1) & 0 < a_n < C \quad (kendala,\ slide\ 16) \\ (2) & (C - a_n)\xi_n = 0 \quad (KKT-3, slide\ 18) \\ (3) & a_n(\varepsilon + \xi_n + y_n - t_n) = 0 \quad (KKT-1, slide\ 18) \\ (4) & y(x) = w^T \phi(x) + b \end{array}$$

Dari (2) maka $\xi_n = 0$, dari (3) maka $\varepsilon + y_n - t_n = 0$, selanjutnya b dapat dicari dari (4) sbb:

$$\begin{aligned} b &= y(x_n) - w^T \phi(x_n) \\ &= t_n - \varepsilon - w^T \phi(x_n) \\ &= t_n - \varepsilon - \sum_{m \in S} (a_m^* - \bar{a}_m^*) k(x_n, x_m) \end{aligned}$$

SVM untuk Regresi

Prediksi Data Baru

- Misal diberikan data ***z***, maka prediksi dari data tersebut ditentukan berdasarkan fungsi regresi *y(**x**)*, yaitu: *y(**z**)*

SVM untuk Regresi

Black Box

- Pada pembahasan SVM pada presentasi ini, beberapa bagian masih bersifat *black box*, yaitu:
  - ***VC Theory***, yaitu teori yang mendasari metode margin maksimum yang menunjukkan bahwa margin maksimum akan memberikan generalisasi error terkecil [1]
  - Algoritma penyelesaian masalah pemrograman kuadrat (bentuk dual dari *soft margin*), misal algoritma SMO [2], LibSVM [3], SVM^{Light} [4]

Referensi

(1) C. H. Bishop. *Pattern Recognition and Machine Learning*, Springer, 2006 (Bab 7.1, Bab 7.1.1, Bab 7.1.3, Appendix E)

(2) J. C. Platt. *Fast training of support vector machines using sequential minimal optimization*. In B. Schoelkopf, C. J. C. Burges, and A. J. Smola (Eds), Advances in Kernel Methods – Support Vector Learning, pp. 185-208, MIT Press, 1999

(3) R. E. Fan, P. H. Chen, C. J. Lin. *Working set selection using second order information for training SVM*. Journal of Machine Learning Research 6, 1889-1918, 2005

(4) T. Joachim. *Making Large-Scale SVM Learning Practical*. In B. Schoelkopf, C. J. C. Burges, and A. J. Smola (Eds), Advances in Kernel Methods – Support Vector Learning, pp. 169-184, MIT Press, 1999