# SVM untuk Regresi Ordinal

**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

## Model Linear

- Model dasar yang akan digunakan adalah model linear, yaitu model yang merupakan kombinasi linear dari fungsi basis, yaitu:

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

  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 fungsi basis

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

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

## Solusi Praktis

- Untuk menerapkan metode ini pada masalah skala besar, pendekatan umum yang dilakukan adalah membuat fungsi basis dapat beradaptasi dengan data pembelajaran, dan mengeset data pembelajaran tersebut sebagai pusat fungsi basis. Selanjutnya, memilih sebagian dari data pembelajaran tersebut selama proses training (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: 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

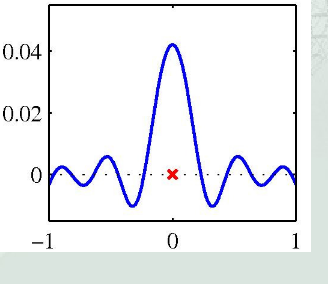
- Salah satu contoh fungsi kernel yang banyak digunakan adalah fungsi Gaussian, yaitu:

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

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

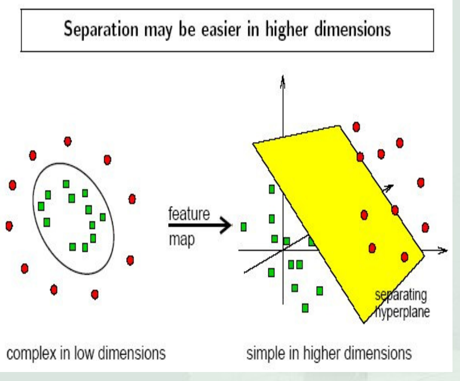
- Misal untuk data x=5 dengan target t=0.04, maka data tsb dapat digambarkan dengan fungsi basis sbb:

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



# 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

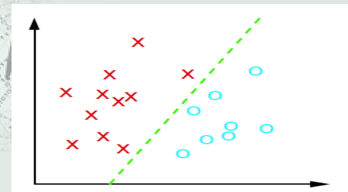
- Secara umum, ada dua cara penggunaan metode kernel pada *machine learning*, yaitu:
  - Penggunaan langsung, yaitu fungsi kernel digunakan sebagai fungsi basis dari model machine learning tersebut, contoh: *radial basis function*
  - Penggunaan tidak langsung melalui *kernel trick*, yaitu merepresentasikan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

# Support Vector Machine

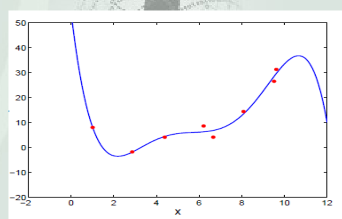
## Klasifikasi dan Regresi

Diberikan data pembelajaran $\{\mathbf{x}_n, t_n\}$, $n = 1$ sd $N$

- Klasifikasi
  - $t_n$ bernilai diskrit (kelas) berhingga tidak terurut (skala nominal)
  - Permasalahan: menentukan *decision boundary* yang dapat mengklasifikasi data dengan benar



- Regresi
  - $t_i$ bernilai kontinu (bilangan real)
  - Permasalahan: menentukan fungsi regresi yang dapat memprediksi nilai data dengan benar



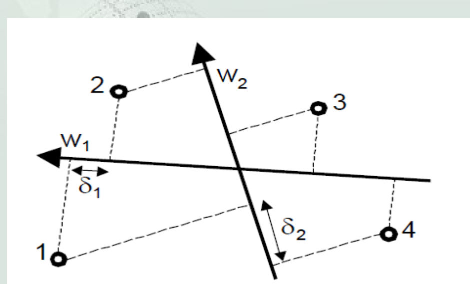



# Support Vector Machine

## Regresi Ordinal

Diberikan data pembelajaran $\{\mathbf{x}_n, t_n\}$, $n = 1$ sd $N$

- Regresi Ordinal
  - $t_n$ bernilai diskrit (kelas) berhingga seperti klasifikasi
  - Terdapat urutan diantara elemen $t_n$ (skala ordinal) seperti regresi
  - Permasalahan: menentukan fungsi regresi ordinal yang dapat mengklasifikasikan data dengan benar



# SVM untuk Regresi Ordinal

## Bentuk Umum

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

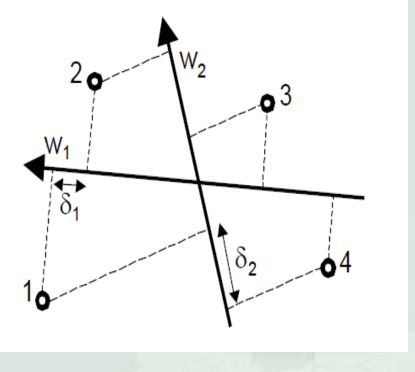
$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x})$$

dimana ***x*** adalah vektor input, ***w*** adalah parameter bobot, dan $\phi(\boldsymbol{x})$ adalah fungsi pemetaan



# SVM untuk Regresi Ordinal

## Formulasi Masalah



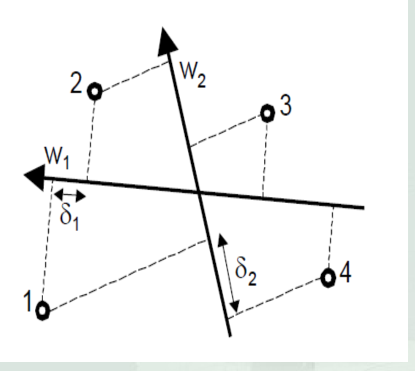

- Diberikan data pembelajaran $\{\boldsymbol{x}_n, t_n\}$, $n = 1$ sd $N$, dimana $\boldsymbol{x}_n \in R^D$, dan $t_n \in \{r_1, r_2, ..., r_P\}$ adalah suatu skala ordinal.
- Permasalahan adalah menentukan fungsi $y(\boldsymbol{x})$ sedemikian sehingga untuk sembarang pasangan data training $(\boldsymbol{x}_i, t_i)$ dan $(\boldsymbol{x}_j, t_j)$ maka

$$y(\boldsymbol{x}_i) > y(\boldsymbol{x}_j) \iff t_i > t_j$$

<u>Contoh:</u> Diberikan data training $(\boldsymbol{x}_1, 4), (\boldsymbol{x}_2, 3), (\boldsymbol{x}_3, 2), (\boldsymbol{x}_4, 1)$, maka $w_1$ adalah fungsi yang benar

# SVM untuk Regresi Ordinal

## Formulasi Masalah




- Misal $P$ adalah himpunan dari pasangan *(i,j)* dimana $\mathbf{x}_i$ memiliki ranking yang lebih tinggi dari $\mathbf{x}_j$, yaitu $P = \{(i,j) : t_i > t_j\}$

  <u>Contoh:</u> dari contoh sebelumnya, maka *P = {(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)}*

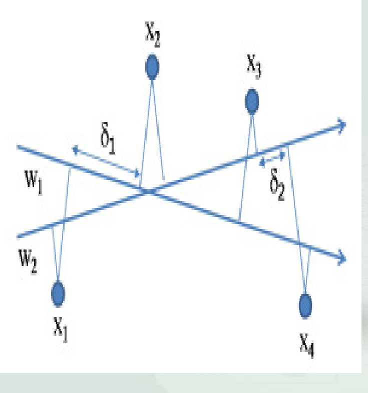
- Maka penentuan fungsi $y(\mathbf{x})$ harus memenuhi kondisi berikut ini:

$$\forall (i,j) \in P : w^T \phi(x_i) \geqslant w^T \phi(x_j)$$



# SVM untuk Regresi Ordinal

## Formulasi Masalah




- Asumsikan semua data dapat di ranking dengan benar (*linearly rankable*), maka jarak antara proyeksi dua data $\mathbf{x}_i$ dan $\mathbf{x}_j$ pada $\mathbf{w}$ adalah:

$$\frac{w^T(\phi(x_i) - \phi(x_j))}{\|w\|}$$

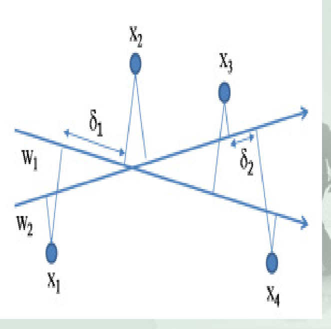
- Margin ($\delta_i$) adalah jarak terdekat dari proyeksi dua data training pada vektor bobot **w.** Sehingga, memaksimumkan margin dapat dideskripsikan sbb:

$$\underset{w}{arg\,max} \left\{ \frac{1}{\|w\|} \min_{i,j} [w^T(\phi(x_i) - \phi(x_j))] \right\}$$

# SVM untuk Regresi Ordinal

## Formulasi Masalah

- Solusi langsung dari masalah optimasi sebelumnya akan sangat kompleks, sehingga perlu dikonversi ke masalah yang ekivalen yang lebih mudah diselesaikan




- Salah satu metode adalah menggunakan bentuk kanonik, yaitu:

$$w^T(\phi(x_i)-\phi(x_j)) = 1$$

untuk data yang terdekat. Selanjutnya, semua data pembelajaran harus memenuhi kondisi berikut ini:

$$w^T(\phi(x_i)-\phi(x_j)) \geqslant 1$$



# SVM untuk Regresi Ordinal

## Formulasi Masalah

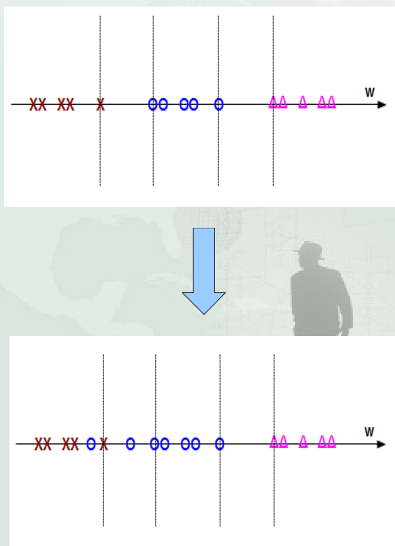
- Selanjutnya, masalah optimasi sebelumnya dapat disederhanakan menjadi memaksimum 1/||w|| yang ekivalen dengan meminimumkan $||w||^2$
- Sehingga masalah optimasi-nya menjadi (bentuk *primal*) [1]:

$$\arg\min_{w} \ \frac{1}{2}\|w\|^2$$
$$s.t \quad \forall(i,j) \in P : w^T(\phi(x_i)-\phi(x_j)) \geqslant 1$$

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

# SVM untuk Regresi Ordinal

## Soft Margin: Landasan

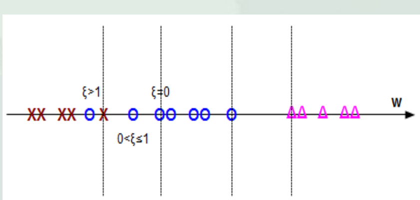



- Diberikan data pembelajaran $\{\mathbf{x}_n, t_n\}$, $n = 1$ sd $N$, dimana $\mathbf{x}_n \in R^D$, dan $t_n \in \{r_1, r_2, \ldots, r_P\}$ adalah suatu skala ordinal.
- Pada formulasi sebelumnya, semua data diasumsikan *linearly rankable* pada ruang fitur $\phi(\mathbf{x})$.
- Dalam aplikasi praktis, kondisi tersebut sering tidak dapat terpenuhi bahkan setelah data di transformasi ke ruang fitur $\phi(\mathbf{x})$.
- Masalah ini diatasi dengan membuat margin lunak (*soft margin*) yang memungkinkan beberapa data pada „urutan yang salah"



# SVM untuk Regresi Ordinal

## Soft Margin: Formulasi Masalah




- Untuk merealisasikan *soft margin* ini, diperkenalkan variabel *slack*, $\xi_{ij} \geq 0$, (i,j) ∈ P, dengan satu variabel *slack* untuk masing-masing pasangan data pembelajaran
- Variabel *slack* tersebut bernilai

$$w^T(\phi(x_i) - \phi(x_j)) \geqslant 1 - \xi_{ij}$$

- Sehingga, $\xi_{ij} = 0$ adalah pasangan data yang terletak pada *margin* atau diranking dengan benar, $0 < \xi_{ij} \leq 1$ adalah pasangan data yang terletak didalam margin, dan $\xi_{ij} > 1$ adalah pasangan data yang salah ranking

# SVM untuk Regresi Ordinal

## Soft Margin: Formulasi Masalah

- Bentuk primal dari masalah optimasi untuk *hard margin* adalah:

$$\underset{w}{\arg\min}\ \frac{1}{2}\|w\|^2$$
$$s.t \qquad \forall(i,j)\in P : w^T(\phi(x_i)-\phi(x_j)) \geqslant 1$$

sementara bentuk primal dari masalah optimasi untuk *soft margin* adalah:

$$\underset{w}{\arg\min}\ \frac{1}{2}\|w\|^2 + C\sum_{(i,j)\in P}\xi_{ij}$$
$$s.t. \qquad \forall(i,j)\in P : w^T(\phi(x_i)-\phi(x_j)) \geqslant 1-\xi_{ij}$$
$$\xi_{ij}\geqslant 0$$

dimana parameter C > 0 akan mengkontrol *trade-off* antara lebar margin (kapabilitas generalisasi) dan jumlah pasangan data training yang tertukar urutannya (kesalahan urutan)



# SVM untuk Regresi Ordinal

## Soft Margin: Formulasi Masalah

- Mentuk primal dari masalah optimasi untuk *soft margin* [1] adalah:

$$\underset{w}{\arg\min}\ \frac{1}{2}\|w\|^2 + C\sum_{(i,j)\in P}\xi_{ij}$$
$$s.t. \qquad \forall(i,j)\in P : w^T(\phi(x_i)-\phi(x_j)) \geqslant 1-\xi_{ij}$$
$$\xi_{ij}\geqslant 0$$

dimana parameter C > 0 akan mengkontrol *trade-off* antara lebar margin (kapabilitas generalisasi) dan jumlah pasangan data training yang tertukar urutannya (kesalahan urutan)

- Masalah optimisasi diatas tampak ekivalen dengan masalah optimisasi dari SVM untuk klasifikasi pada pasangan vektor jarak $\phi(\mathbf{x}_i)$ - $\phi(\mathbf{x}_j)$. Sehingga dapat diselesaikan dengan pendekatan yang digunakan oleh SVM untuk klasifikasi

# SVM untuk Regresi Ordinal

## Solusi Masalah: Lagrange Multipliers

Untuk menyelesaikan pemrograman kuadrat tersebut, yaitu mencari titik stationer dari fungsi beberapa variabel dengan satu atau lebih kendala, digunakan perkalian Lagrange (*Lagrange multipliers*) $a_{ij}$, $\mu_{ij} \geq 0$ untuk membentuk fungsi Lagrangian (*Lagrangian function*) sbb:

$$L(w, a, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i,j\in P}\xi_{ij} - \sum_{i,j\in P} a_{ij}\{w^T(\phi(x_i)-\phi(x_j))-1+\xi_{ij}\} - \sum_{i,j\in P}\mu_{ij}\xi_{ij}$$



# SVM untuk Regresi Ordinal

## Solusi Masalah: Lagrange Multipliers

$$L(w, a, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i,j\in P}\xi_{ij} - \sum_{i,j\in P} a_{ij}\{w^T(\phi(x_i)-\phi(x_j))-1+\xi_{ij}\} - \sum_{i,j\in P}\mu_{ij}\xi_{ij}$$

- Dengan menurunkan *L(**w**, a, μ)* terhadap ***w*** sama dengan nol, maka:

$$\frac{\partial L}{\partial w} = w - \sum_{i,j\in P} a_{ij}(\phi(x_i)-\phi(x_j)) = 0 \rightarrow w = \sum_{i,j\in P} a_{ij}(\phi(x_i)-\phi(x_j))$$

$$\frac{\partial L}{\partial \xi_{ij}} = C - a_{ij} - \mu_{ij} = 0 \rightarrow a_{ij} = C - \mu_{ij}$$

# SVM untuk Regresi Ordinal

## Solusi Masalah: Fungsi Kernel

- Substitusikan hasil turunan tsb ke persamaan Lagrangian, yaitu:

$$L(w,a,\mu) = \frac{1}{2}\|w\|^2 + C\sum_{i,j\in P}\xi_{ij} - \sum_{i,j\in P} a_{ij}\{w^T(\phi(x_i)-\phi(x_j))-1+\xi_{ij}\} - \sum_{i,j\in P}\mu_{ij}\xi_{ij}$$

sehingga menjadi:

$$\tilde{L}(a) = \frac{1}{2}\sum_{ij\in P}\sum_{uv\in P} a_{ij}a_{uv}(\phi(x_i)-\phi(x_j))^T(\phi(x_u)-\phi(x_v)) - \sum_{ij\in P}\sum_{uv\in P} a_{ij}a_{uv}(\phi(x_i)-\phi(x_j))^T(\phi(x_u)-\phi(x_v)) + \sum_{ij\in P} a_{ij}$$

$$= \sum_{ij\in P} a_{ij} - \frac{1}{2}\sum_{ij\in P}\sum_{uv\in P} a_{ij}a_{uv}(\phi(x_i)-\phi(x_j))^T(\phi(x_u)-\phi(x_v))$$

$$= \sum_{ij\in P} a_{ij} - \frac{1}{2}\sum_{ij\in P}\sum_{uv\in P} a_{ij}a_{uv}k(x_i-x_j, x_u-x_v)$$



# SVM untuk Regresi Ordinal

## Solusi Masalah: Bentuk Dual

- Sehingga, bentuk *dual* dari masalah *soft margin* adalah:

$$\underset{a}{arg\,max}\ \tilde{L}(a) = \sum_{ij\in P} a_{ij} - \frac{1}{2}\sum_{ij\in P}\sum_{uv\in P} a_{ij}a_{uv}k(x_i-x_j, x_u-x_v)$$

$$s.t. \qquad 0 \leqslant a_{ij} \leqslant C,\ \ ij\in P$$

# SVM untuk Regresi Ordinal

## Solusi Masalah: Nilai Bobot

- Misal solusi dari pemrograman kuadrat bentuk dual tersebut adalah ***a****, maka:

$$w = \sum_{ij \in P} a_{ij}^{*} (\phi(x_i) - \phi(x_j))$$

$$y(x) = w^T \phi(x) = \sum_{ij \in P} a_{ij}^{*} (\phi(x_i) - \phi(x_j)) \phi(x) = \sum_{ij \in P} a_{ij}^{*} k(x_i - x_j, x)$$

- Hanya data dengan $a_{ij}^{*} > 0$ (*support vectors*) yang berperan pada fungsi regresi ordinal diatas, sehingga dapat ditulis menjadi:

$$y(x) = \sum_{ij \in S} a_{ij}^{*} k(x_i - x_j, x)$$

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



# SVM untuk Regresi Ordinal

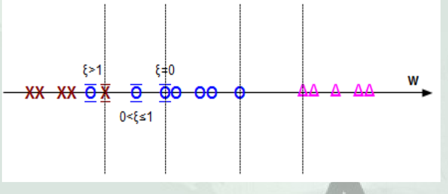
## Solusi Masalah: Karush-Kuhn-Tucker

*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:

$$\begin{aligned}
a_{ij} &\geqslant 0 && [KKT - S1] \\
y(x_{ij}) - 1 + \xi_{ij} &\geqslant 0 && [KKT - S2] \\
a_{ij}\{y(x_{ij}) - 1 + \xi_{ij}\} &= 0 && [KKT - S3] \\
\mu_{ij} &\geqslant 0 && [KKT - S4] \\
\xi_{ij} &\geqslant 0 && [KKT - S5] \\
\mu_{ij}\xi_{ij} &= 0 && [KKT - S6]
\end{aligned}$$

# SVM untuk Regresi Ordinal

Solusi Masalah: Support Vectors




- *Support vectors* adalah data training dengan nilai $a_{ij}^* > 0$, maka dari [KKT-S3]

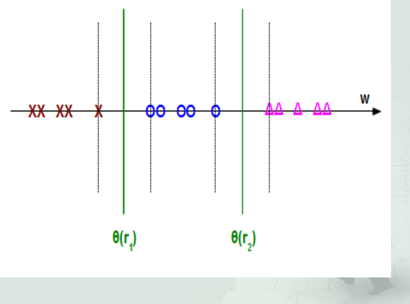
$$w^T(\phi(x_i) - \phi(x_j)) = 1 - \xi_{ij}$$

- Jika $a_{ij}^* < C$, maka berdasarkan hasil turunan bahwa $a_{ij} = C - \mu_{ij}$ maka $\mu_{ij} > 0$. Selanjutnya, berdasarkan [KKT-S6] $\mu_{ij}\xi_{ij} = 0$ maka $\xi_{ij} = 0$. Dengan kata lain adalah pasangan data training yang terletak pada margin
- Jika $a_{ij}^* = C$, maka $\mu_{ij} = 0$. Dari [KKT-S6] maka $\xi_{ij} \neq 0$. Dengan kata lain adalah pasangan data training yang terletak didalam margin ($0 < \xi_{ij} \leq 1$) atau salah ranking ($\xi_{ij} > 1$)



# SVM untuk Regresi Ordinal

Prediksi Data Baru




- Misal $g(\mathbf{x}) \in \{r_1, r_2, \ldots, r_p\}$ adalah skala ordinal, maka prediksi dari suatu data $\mathbf{z}$ adalah:

$$g(z) = r_i \Leftrightarrow y(z) \in [\theta(r_{i-1}), \theta(r_i)]$$

- Estimasi dari $\theta(r_k)$ diberikan oleh:

$$\theta(r_k) = \frac{y(x_1) + y(x_2)}{2}$$

dimana

$$(x_1, x_2) = \underset{(x_i, x_j) \in \Theta(k)}{arg\,min}\,[y(x_i) - y(x_j)]$$

$$\Theta(k) = \{(x_i, x_j) \mid t_i = r_k \wedge t_j = r_{k+1} \wedge 0 < a_{ij} < C\}$$

# SVM untuk Regresi Ordinal

## Algoritma & Perangkat Lunak

- Ada beberapa algoritma dan perangkat lunak yang telah dikembangkan untuk memecahkan masalah optimasi pada SVM untuk Regresi Ordinal, antara lain:
  - SVM$^{perf}$ [2] (http://svmlight.joachims.org)



# Referensi

(1) R. Herbrich, T. Graepel, K. Obermayer. *Large Margin Rank Boundaries for Ordinal Regression*. Advances in Large Margin Classifiers, pp. 115-132, MIT Press, 2000

(2) T. Joachims. *Training Linear SVMs in Linear Time*. Proceeding of the ACM Conference on Knowledge Discovery and Data Mining, 2006