Linear Models for Classification

Dama Putra Sarpanda | Ferza Reyaldi | Irvan Kurniawan
09021181924016 | 09021281924060 | 09021181924159

Discriminant Functions

Fungsi Diskriminan dengan Dua Kelas

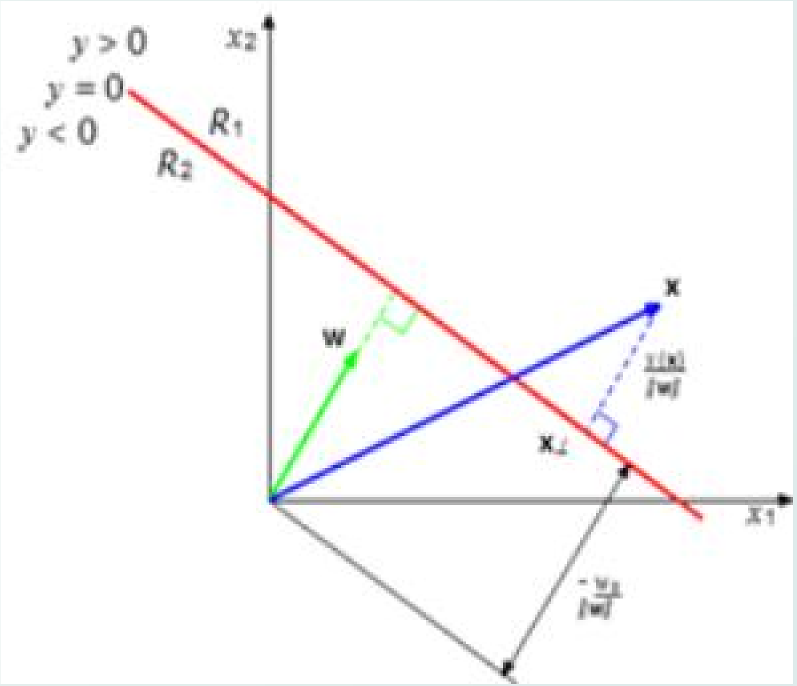
- Mulailah dengan 2 masalah kelas
  $t \in \{0, 1\}$
- Diskriminan linier sederhana

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

terapkan fungsi ambang batas untuk

mendapatkan klasifikasi

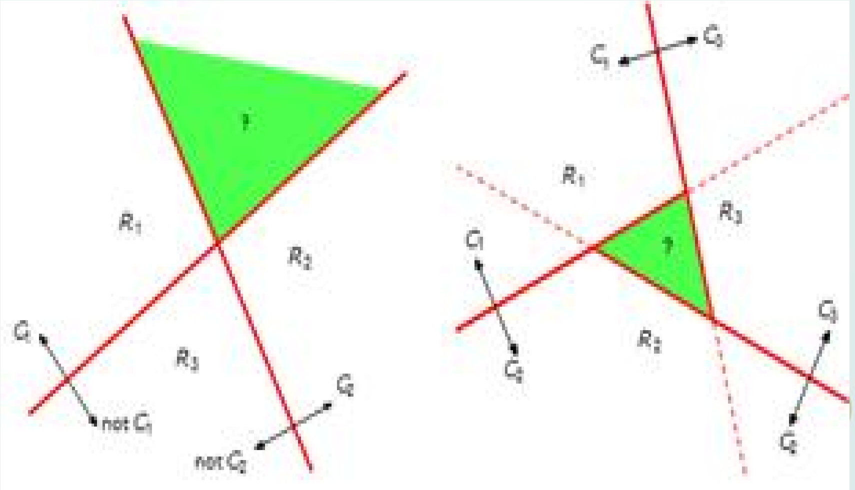
- Proyeksi $\boldsymbol{x}$ dalam $\boldsymbol{w}$ Adalah $\frac{wT\,x}{||w}$



Fungsi Diskriminan dengan Beberapa Kelas

- Diskriminan linier antara dua kelas terpisah dengan hyperplane
- Bagaimana cara menggunakan ini untuk beberapa kelas?

- Metode satu lawan satu: buat pengklasifikasi K-1, antara Ck dan yang lainnya
- Metode satu lawan satu: buat pengklasifikasi $K(K$-1$)/2$, di antara semua pasangan



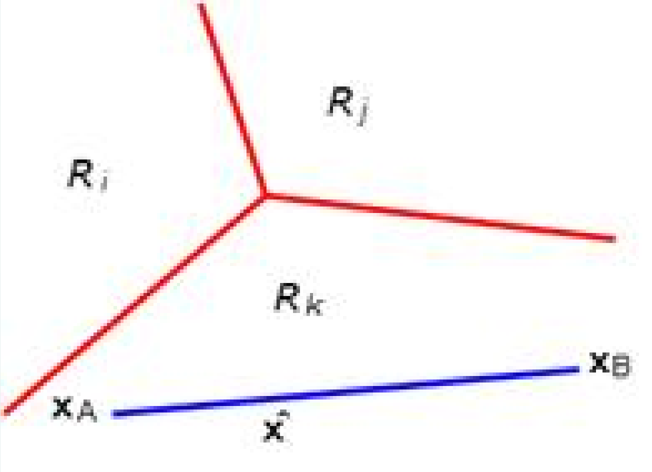
Fungsi Diskriminan dengan Beberapa Kelas

- Solusinya adalah membangun *K* fungsi linier
  - $$y_k(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_{k0}$$

  tetapkan *x* ke kelas arg maxk *yk*(*x*)

- Memberikan wilayah keputusan konveks yang terhubung

$$\begin{aligned} \hat{\mathbf{x}} &= \lambda \mathbf{x}_A + (1-\lambda)\mathbf{x}_B \\ y_k(\hat{\mathbf{x}}) &= \lambda y_k(\mathbf{x}_A) + (1-\lambda) y_k(\mathbf{x}_B) \\ \Rightarrow y_k(\hat{\mathbf{x}}) &> y_j(\hat{\mathbf{x}}),\ \forall j \neq k \end{aligned}$$



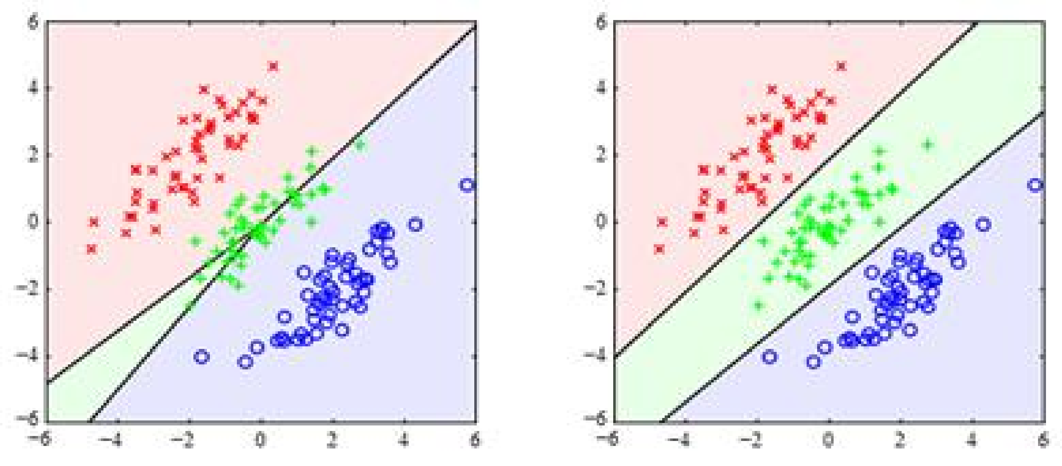
Kuadrat Terkecil untuk Klasifikasi

- Bagaimana kita mempelajari batas keputusan ($\boldsymbol{w}_k$, w_{k0})?
- Salah satu pendekatan adalah dengan menggunakan kuadrat terkecil, mirip dengan regresi
- Temukan $\boldsymbol{W}$ untuk meminimalkan kesalahan kuadrat pada semua contoh dan semua komponen vektor label:

$$E(\boldsymbol{W}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\left(y_k(\boldsymbol{x}_n) - t_{nk}\right)^2$$

- Beberapa aljabar, menggunakan invers semu seperti dalam regresi

Permasalahan Kuadrat Terkecil



- Mudah dipisahkan oleh hyperplanes, tetapi tidak ditemukan menggunakan kuadrat terkecil!
- Mengatasi masalah ini dengan model yang lebih baik
- Pertama, lihat kriteria yang berbeda untuk diskriminan linier

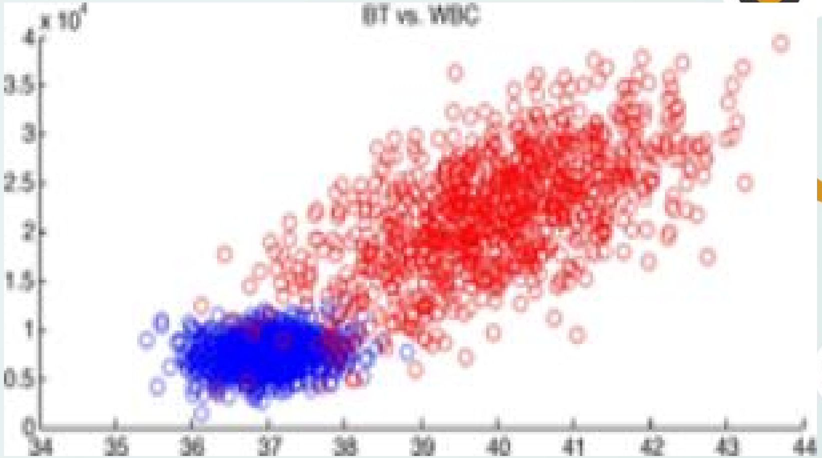
Diskriminan Linier Fisher

- Diskriminan linier dua kelas bertindak sebagai proyeksi:

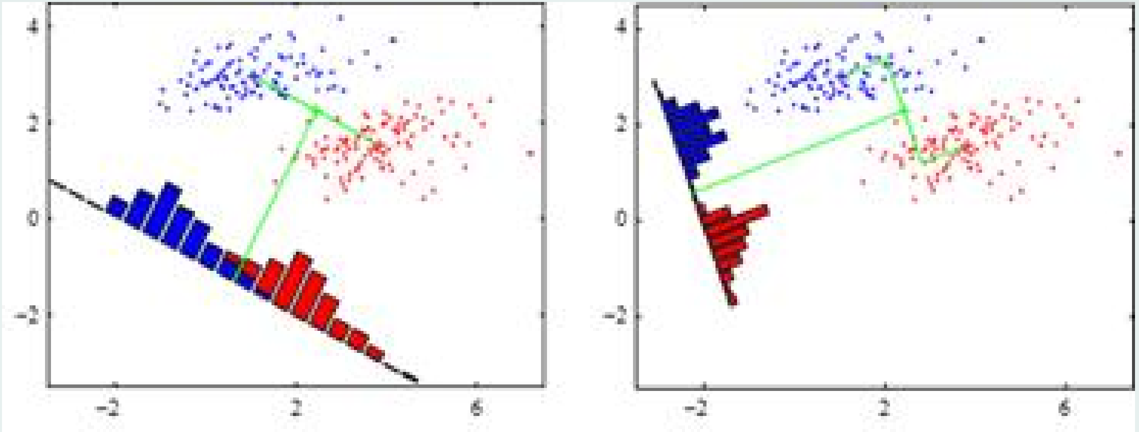
$$y = \boldsymbol{w}^T \boldsymbol{x} \geq -w_0$$

diikuti oleh ambang batas

- Ke arah mana kita harus memproyeksikan?
- Yang memisahkan kelas "baik"



Diskriminan Linier Fisher



- Natural ide adalah memproyeksikan ke arah garis yang menghubungkan sarana kelas
- Namun, bermasalah jika kelas memiliki varian dalam arah ini
- Kriteria Fisher: memaksimalkan rasio pemisahan antar kelas (antar) terhadap variansi intra kelas (di dalam)

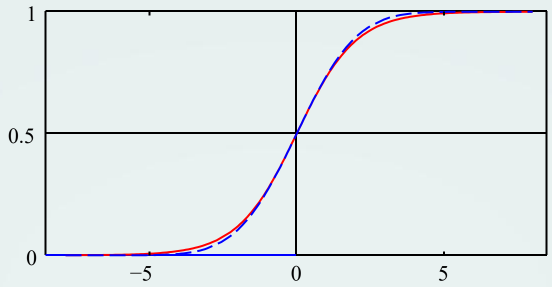
Ringkasan FLD

- FLD adalah teknik pengurangan dimensi
- Kriteria pemilihan proyeksi berdasarkan label kelas

Masih memiliki outlier (misalnya contoh kuadrat terkecil sebelumnya)

Probabilistic Generative Models

Logistic Sigmoid



- Fungsi $\sigma(a) = \frac{1}{1+\exp(-a)}$ diketahui sebagai logistic sigmoid
- Fungsi ini mengubah axis menjadi $[0, 1]$
- Berupa bilangan continuous dan bisa diturunkan
- Menghindari masalah yang diakibatkan oleh least-square error fitting yang terlalu tepat (nanti)

Multi-class Extension

- Ada generalisasi dimana logistic sigmoid $K > 2$ kelas:

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{\sum_j p(\boldsymbol{x}|C_j)p(C_j)}$$

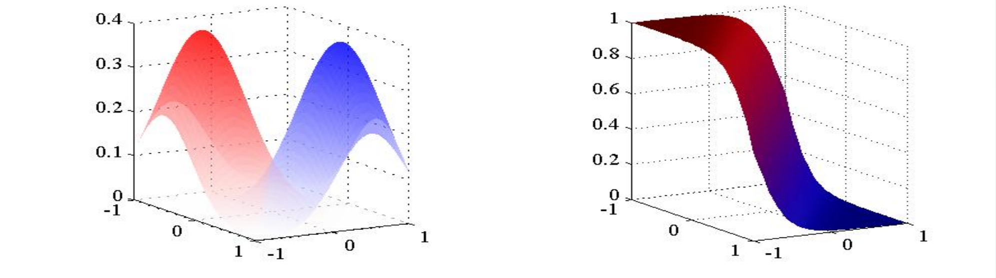
$$= \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

dimana $a_k = \ln p(\boldsymbol{x}|C_k)p(C_k)$

Atau, fungsi softmax

- Jika beberapa $a_k >> a_j$, $p(C_k|\boldsymbol{x})$ menuju 1

Gaussian Class-Conditional Densities



- Kembali ke *a* di logistic sigmoid untuk 2 kelas
- Mari berasumsi densitas class-conditional $p(\boldsymbol{x}|C_k)$ adalah Gaussian, dan memiliki kovariansi yang sama dengan matrix Σ:

$$p(\boldsymbol{x}|C_k) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$

- *a* mengambil bentuk sederhana:

$$a = \ln \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_2)p(C_2)} = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

  - Catatan bahwa ketentuan kuadrat $\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x}$ batal

Maximum Likelihood Learning

- Kita bisa memasukkan parameter ke dalam model ini menggunakan kemiripan maksimum
  - Parameternya adalah $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma^{-1}, p(C_1) \equiv \pi, p(C_2) \equiv 1 - \pi$
  - Dirujuk sebagai θ
- Untuk datapoint $\boldsymbol{x}_n$ dari kelas C_1 ($t_n = 1$):

$$p(\boldsymbol{x}_n, C_1) = p(C_1)p(\boldsymbol{x}_n|C_1) = \pi N(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \Sigma)$$

- Untuk datapoint $\boldsymbol{x}_n$ dari kelas C_2 ($t_n = 0$):

$$p(\boldsymbol{x}_n, C_2) = p(C_2)p(\boldsymbol{x}_n|C_2) = (1 - \pi)N(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \Sigma)$$

Maximum Likelihood Learning

- Kemiripan data latihan adalah:

$$p(\boldsymbol{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = \prod_{n=1}^{N} [\pi \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \Sigma)]^{t_n} [(1-\pi)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \Sigma)]^{1-t_n}$$

- Seperti biasa, ln adalah teman kita:

$$A(\boldsymbol{t}; \theta) = \sum_{n=1}^{N} \underbrace{t_n \ln \pi + (1 - t_n) \ln(1 - \pi)}_{\pi} + \underbrace{t_n \ln \mathcal{N}_1 + (1 - t_n) \ln \mathcal{N}_2}_{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}}$$

- Maksimal untuk setiapnya secara terpisah

Maximum Likelihood Learning - Class Priors

- Memaksimalkan sehubungan dengan kelas prior parameter π sangat mudah:

$$\frac{\partial}{\partial \pi} A(\boldsymbol{t}; \theta) = \sum_{n=1}^{N} \frac{t_n}{\pi} - \frac{1 - t_n}{1 - \pi}$$

$$\Rightarrow \pi = \frac{N_1}{N_1 + N_2}$$

- N_1 dan N_2 adalah angka untuk poin - poin latihan di setiap kelas
- Prior hanyalah sebagian dari poin - poin di setiap kelas

Maximum Likelihood Learning - Gaussian Parameters

- Parameter yang lain juga bisa ditemukan dalam bentuk yang sama
- Rata - rata kelas :

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \boldsymbol{x}_n$$

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \boldsymbol{x}_n$$

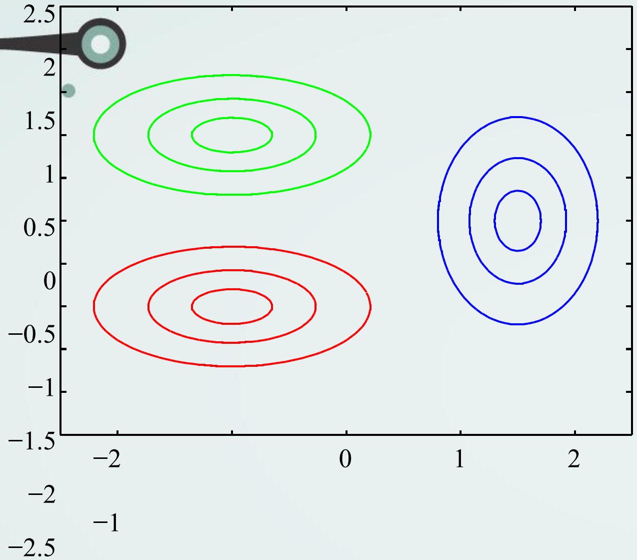
  - Rata - rata contoh latihan dari setiap kelas

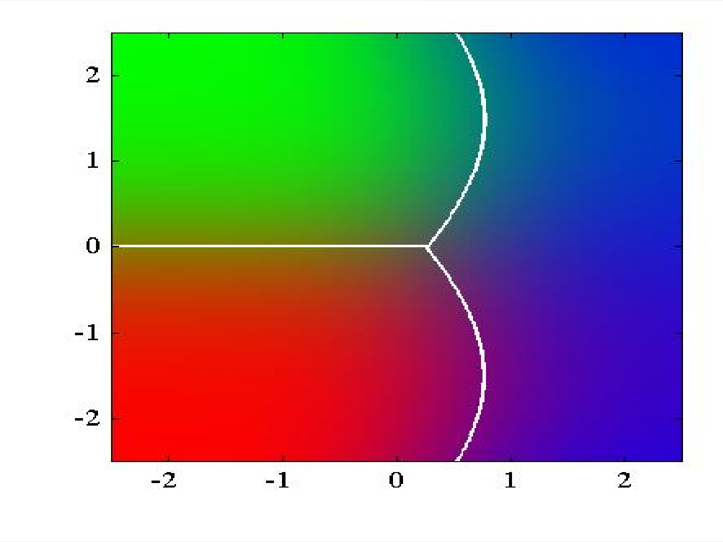
- Matrix kovariansi terbagi:

$$\Sigma = \frac{N_1}{N} \frac{1}{N_1} \sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{\mu}_1)(\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T + \frac{N_2}{N} \frac{1}{N_2} \sum_{n \in C_2} (\boldsymbol{x}_n - \boldsymbol{\mu}_2)(\boldsymbol{x}_n - \boldsymbol{\mu}_2)^T$$

  - Bobot rata - rata dari kovariansi kelas

Gaussian dengan Kovariansi berbeda





$$a = \ln \frac{p(x|C_b)p(C_b)}{p(x|C_r)p(C_r)}$$

$$a = \ln(p(x|C_b)) - \ln(p(x|C_r)) + \ln(p(C_b)) - \ln(p(C_r))$$

$-(x-\mu_b)^T \Sigma_b (x-\mu_b) + (x-\mu_r)^T \Sigma_r (x-\mu_r)$

Konstanta.

Rangkuman Probabilistic Generative Model

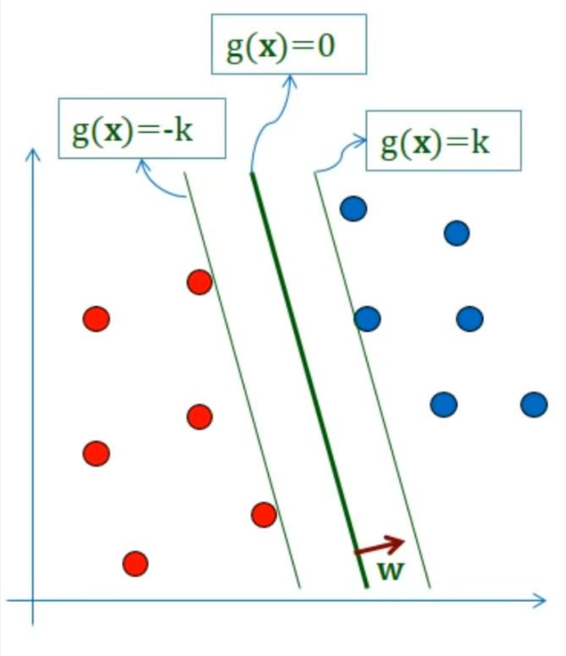
- Mencocokkan Gaussian menggunakan criterion machine learning sensitif terhadap outlier
- Bentuk linear sederhana untuk a di dalam logistic sigmoid terjadi lebih dari sekedar distribusi Gaussian
  - Muncul pada distribusi mana saja di dalam keluarga exponensial, sebuah kelas besar dari distribusi

Probabilistic Discriminative Models

Support Vector Machine

- Pengembangan dari Perceptron yang dikembangkan oleh Rosenblatt pada tahun 1958.
- Perceptron menjamin bahwa Anda menemukan hyperplane jika ada.
- SVM menemukan margin maksimum yang memisahkan hyperplane.

Support Vector Machine



$g(x) = w^T x + b$

Maksimalkan k sedemikian rupa sehingga:

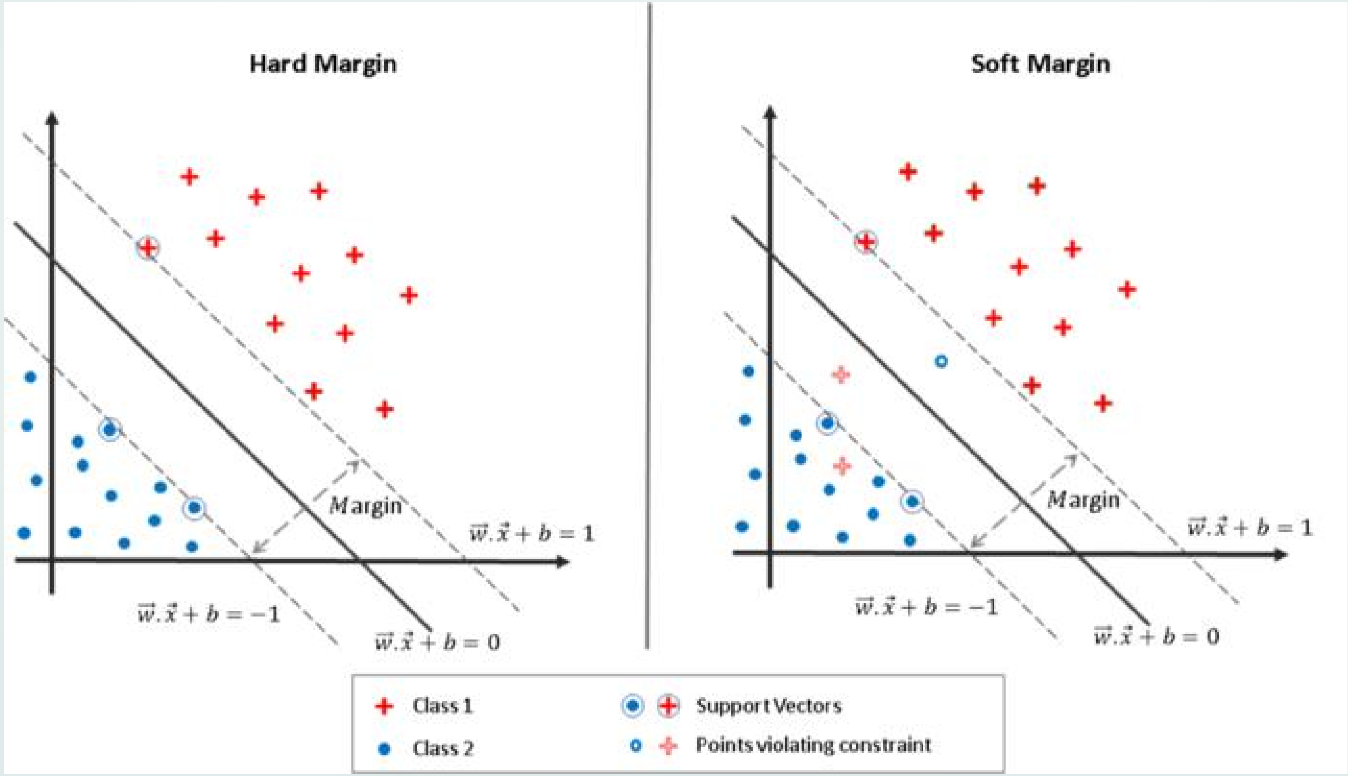
$-w^T x + b \geq k$ untuk $d_i == 1$

$-w^T x + b \leq k$ untuk $d_i == -1$

Nilai $g(x)$ tergantung pada $||w||$:

- Biarkan $||w|| = 1$ dan maksimalkan nilai $g(x)$, atau
- $g(x) \geq 1$ dan minimalkan nilai $||w||$

Support Vector Machine



Iterative Reweighted Least Squares

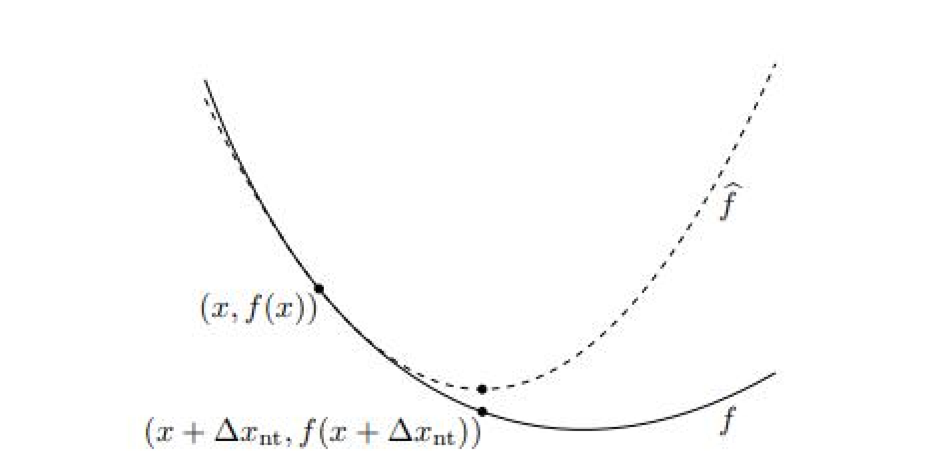
- Iterative reweighted least squares (IRLS) adalah metode penurunan seperti pada metode penurunan gradien.
- Kasus khusus dari metode **Newton-Raphson**
    - Fungsi Aproksimasi menggunakan ekspansi Taylor orde kedua:

$$\hat{f}(\boldsymbol{w}+\boldsymbol{v}) = f(\boldsymbol{w})+\nabla f(\boldsymbol{w})^T(\boldsymbol{v}-\boldsymbol{w})+\frac{1}{2}(\boldsymbol{v}-\boldsymbol{w})^T\nabla^2 f(\boldsymbol{w})(\boldsymbol{v}-\boldsymbol{w})$$

    - Solusi bentuk tertutup untuk meminimalkan ini adalah *straight-forward*: kuadratik, turunan linier

Iterative Reweighted Least Squares

Grafik Metode **Newton-Raphson**



Resources

http://vda.univie.ac.at/Teaching/ML/15s/LectureNotes/04_classification.pdf

https://www.ics.uci.edu/~xhx/courses/CS273P/05-linear-classification-273p.pdf

Terima Kasih