PENGKLASIFIKASI LINEAR

Budi Darmawan, 05947-TE
Jans Hendry, 05965-TE
Utis Sutisna, 06442-TE
Jurusan Teknik Elektro FT UGM,
Yogyakarta

3.1 PENDAHULUAN

Dalam beberapa kasus, diperlihatkan bahwa hasil pengklasifikasi berdasarkan pada densitas probabilitas atau fungsi-fungsi probabilitas ekivalen dengan sebuah himpunan fungsi-fungsi diskriminan linear. Pembahasan ini akan difokuskan pada perancangan pengklasifikasi linear, *terlepas dari distribusi-distribusi pokok yang menggambarkan data pelatihan*. Keuntungan utama dari pengklasifikasi linear adalah komputasinya yang sederhana dan menarik. Pembahasan ini dimulai dengan asumsi bahwa semua vektor ciri dari kelas-kelas yang ada dapat diklasifikasi dengan benar menggunakan sebuah pengklasifikasi linear, dan akan dikembangkan teknik-teknik untuk komputasi dari fungsi-fungsi linear yang berkaitan. Selanjutnya akan difokuskan pada beberapa masalah umum, dimana sebuah pengkalifikasi linear tidak dapat mengklasifikasi dengan benar seluruh vektor, namun akan dicari cara untuk merancang sebuah pengklasifikasi linear optimal dengan mengadopsi sebuah kriteria optimalitas yang cocok.

3.2 FUNGSI-FUNGSI DISKRIMINAN LINEAR DAN HYPERPLANES KEPUTUSAN

Subbab lebih difokuskan pada kasus dua-kelas dan mempertimbangkan fungsi-fungsi diskriminan linear. Kemudian *hypersurface* keputusan masing-masing dalam ruang ciri dimensi-*l* adalah *hyperplane* (dimensi ruang vektor), yaitu

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \ w_0 = 0 \tag{3.1}$$

Dimana $\boldsymbol{w} = [w_1, w_2, \dots, w_l]^T$ dikenal sebagai vektor bobot dan w_0 sebagai ambang. Jika x_1, x_2 adalah dua titik pada hyperplane, maka berikut ini adalah valid

$$0 = \boldsymbol{w}^T x_1 + w_0 = \boldsymbol{w}^T x_2 + w_0 \qquad \rightarrow \qquad \boldsymbol{w}^T(x_1 - x_2) = 0 \tag{3.2}$$

Karena vektor perbedaan $x_1 - x_2$ dengan jelas terletak pada *hyperplane* keputusan (untuk sembarang x_1, x_2), tampak pada Pers. (3.2) bahwa vektor $\boldsymbol{w}$ adalah ortogonal terhadap *hyperplane* keputusan.

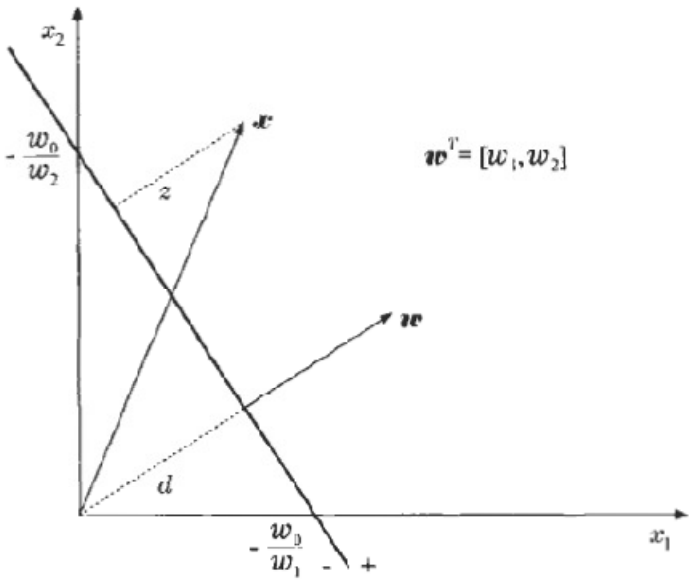
Gambar 3.1 menunjukkan hubungan geometri (untuk $w_1 > 0, w_2 > 0, w_0 < 0$). Mudah untuk melihat bahwa kuantitas yang masuk dalam gambar tersebut diberikan oleh

$$d = \frac{|w_0|}{\sqrt{w_1^2 + w_2^2}} \tag{3.3}$$

dan

$$z = \frac{|g(x)|}{\sqrt{w_1^2 + w_2^2}} \tag{3.4}$$

Dengan kata lain, $|g(x)|$ adalah sebuah pengukuran jarak Euclidean titik x dari *hyperplane* keputusan. Dalam kasus khusus dimana $w_0 = 0$ *hyperplane* melewati origin.



Gambar 3.1 Geometri untuk garis keputusan. Pada satu sisi dari garis adalah g(x) > 0(+) dan pada sisi lain adalah g(x) < 0(-).

3.3 ALGORITMA PERCEPTRON

Perhatian utama subbab ini adalah menghitung parameter-parameter yang tidak diketahui w_i, i = 0, …, l, yang mendefinisikan *hyperplane* keputusan. Dalam bagian ini diasumsikan bahwa dua kelas ω_1, ω_2 adalah dapat dipisahkan secara linear (*linearly separable*). Dengan kata lain diasumsikan bahwa ada sebuah *hyperplane*, yang didefinisikan oleh $\boldsymbol{w}^{*T}\boldsymbol{x} = 0$, sedemikian sehingga

$$\begin{aligned} \boldsymbol{w}^{*T}\boldsymbol{x} > 0 \quad & \forall \boldsymbol{x} \in \omega_1 \\ \boldsymbol{w}^{*T}\boldsymbol{x} < 0 \quad & \forall \boldsymbol{x} \in \omega_2 \end{aligned} \tag{3.5}$$

Rumusan di atas juga mencakup kasus dari sebuah *hyperplane* yang tidak melalui origin, yaitu, $\boldsymbol{w}^{*T}\boldsymbol{x} + w_0^* = 0$, karena ini dapat dibawa kepada rumus sebelumnya dengan mendefinisikan vektor-vektor dimensi-(*t* + 1) yang diperluas $\boldsymbol{x'} \equiv [\boldsymbol{x^T}.\boldsymbol{1}]^T.\boldsymbol{w'} \equiv [\boldsymbol{w}^{*T}.w_0^*]^T$**.** Maka $\boldsymbol{w}^{*T}\boldsymbol{x} + w_0^* = \boldsymbol{w'}^T\boldsymbol{x'}$.

Masalah ini didekati sebagai tugas optimisasi tipikal. Jadi perlu diadopsi (a) sebuah fungsi harga (*cost function*) yang cocok dan (b) sebuah pola algoritma untuk mengoptimasinya. Untuk yang bagian akhir ini, dipilih harga perceptron (*perceptron cost*) yang didefinisikan sebagai

$$J(\boldsymbol{w}) = \sum_{x \in Y}(\delta_x \boldsymbol{w}^T\boldsymbol{x}) \tag{3.6}$$

dimana Y adalah subset dari vektor pelatihan, yang diklasifikasi dengan salah oleh *hyperplane* yang didefinisikan oleh vektor bobot $\boldsymbol{w}$. Variabel δ_x dipilih sedemikian hingga $\delta_x = -1$ jika $\boldsymbol{x} \in \omega_1$ dan $\delta_x = +1$ jika $\boldsymbol{x} \in \omega_2$. Jelas, jumlah dalam (3.6) adalah selalu positif dan menjadi nol ketika Y menjadi himpunan kosong, yaitu, jika tidak ada vektor $\boldsymbol{x}$ yang diklasifikasikan dengan salah (*misclassified*). Tentu saja, jika $\boldsymbol{x} \in \omega_1$ dan diklasifikasikan dengan salah (*misclassified*), maka $\boldsymbol{w^T x} < 0$ dan $\delta_x < 0$, dan hasil kali adalah positif. Hasilnya adalah sama untuk vektor-vektor yang berasal dari kelas ω_2. Pada saat fungsi

harga (*cost function*) mengambil nilai minimumnya, 0, sebuah solusi telah diperoleh, karena semua vektor ciri pelatihan diklasifikasikan dengan benar.

Fungsi harga perceptron dalam (3.6) adalah kontinyu dan linear *piecewise*. Tentu saja, jika kita mengubah vektor bobot secara halus, harga $J(\boldsymbol{w})$ berubah secara linear sampai titik dimana ada perubahan dalam jumlah vektor-vektor *misclassified* (Problem 3.1). Pada titik-titik ini gradiennya tidak terdefinisi dan fungsi grdiennya tidak kontinyu.

Untuk menurunkan algoritma untuk minimisasi iteratif dari fungsi harga (*cost function*), kita akan mengadopsi sebuah pola iteratif dalam jiwa dari metode *gradient descent*, yaitu,

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \rho_t \left.\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}(t)} \tag{3.7}$$

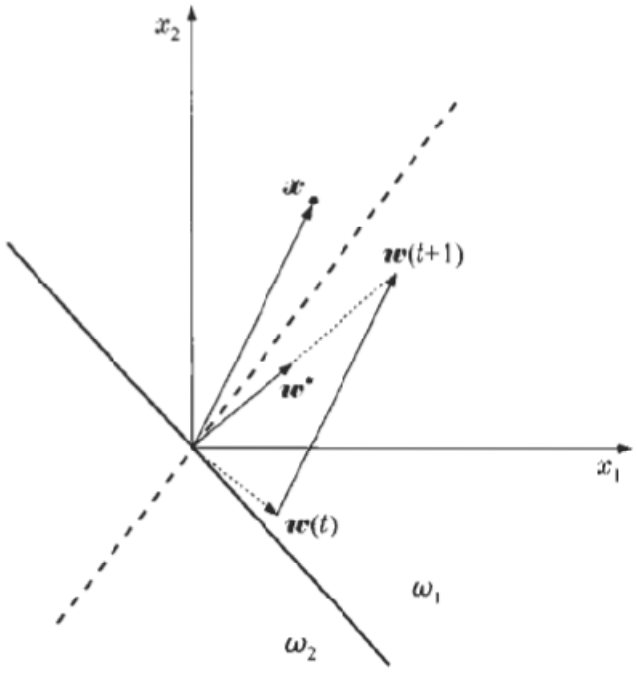
dimana $\boldsymbol{w}(t)$ adalah perkiraan vektor bobot pada langkah iterasi ke-*t*, dan ρ_t adalah runtun bilangan real positif. Akan tetapi, harus hati-hati disini. Ini tidak didefinisikan pada titik dikontinuitas. Dari definisi dalam (3.6), dan pada titik dimana ini adalah valid, didapatkan

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \sum_{x \in Y} \delta_x \boldsymbol{x} \tag{3.8}$$

Dengan mensubstitusikan (3.8) ke (3.7) diperoleh

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \rho_t \sum_{x \in Y} \delta_x \boldsymbol{x} \tag{3.9}$$

Algoritma tersebut dikenal sebagai algoritma perceptron dan sungguh sederhana dalam strukturnya. Perhtikan bahwa Pers. (3.9) terdefinisi dalam semua titik. Algoritma ini diinisialisasi dari sebuah vektor bobot sembarang $\boldsymbol{w}(0)$, dan vektor koreksi $\sum_{x \in Y} \delta_x \boldsymbol{x}$ dibentuk menggunakan fitur-fitur (ciri-ciri) *misclassified*. Vektor bobot kemudian dikoreksi berdasarkan aturan yang mendahuluinya. Hal ini diulangi sampai algoritma tersebut konvergen ke sebuah solusi, yaitu, semua ciri diklasifikasi secara benar.



Gambar 3.2 Interpretasi geometri dari algoritma perceptron.

Gambar 3.2 menyediakan interpretasi geometris dari algoritma tersebut. Telah diasumsikan bahwa pada langkah *t* hanya ada satu sampel yang diklasifikasi dengan salah, $\boldsymbol{x}$, dan $\rho_t = 1$. Algoritma perceptron mengoreksi vektor bobot dalam arah $\boldsymbol{x}$. efeknya adalah memutar *hyperplane* yang berhubungan sedemikian sehingga $\boldsymbol{x}$ diklasifikasikan dalam kelas yang benar ω_1.

Perhatikan bahwa agar mencapai ini, mungkin dilakukan lebih dari satu langkah iterasi, tergantung pada nilai ρ_t. Tidak diragukan lagi, urutan ini adalah kritis untuk konvergen. Sekarang akan ditunjukkan bahwa algoritma perceptron konvergen pada sebuah solusi dalam jumlah berhingga langkah iterasi, asalkan bahwa urutan ρ_t dipilih dengan tepat. Solusi ini tidak unik, karena ada lebih dari satu *hyperplane* yang memisahkan dua kelas yang dapat dipisahkan secara linear. Bukti konvergensi adalah penting karena algoritma ini adalah bukan sebuah algoritma *gradient descent* yang sebenarnya dan perangkat umum untuk konvergensi dari pola *gradient descent* tidak dapat diterapkan.

Varian-Varian dari Algoritma Perceptron

Algoritma yang telah disajikan hanya satu dari sejumlah varian yang telah ditawarkan untuk palatihan dari sebuah pengklasifikasi linear dalam kasus kelas-kelas yang dapat dipisahkan secara linear. Sekarang akan dibahas bentuk lain yang lebih simpel dan populer. *N* vektor pelatihan memasuki algoritma ini secara kritis, satu demi satu. Jika algoritma ini belum konvergen setelah penyajian dari semua sampel satu kali, maka prosedur akan tetap mengulang sampai konvergensi dicapai, yaitu, pada saat semua sampel pelatihan telah diklasifikasi dengan benar. Misal $\boldsymbol{w}(t)$ adalah perkiraan vektor bobot dam $\boldsymbol{x}_{(t)}$ vektor ciri yang berkaitan, disajikan pada langkah iterasi ke-*t*. Algoritmanya dinyatkan sebagai berikut:

$$\begin{aligned} \boldsymbol{w}(t+1) &= \boldsymbol{w}(t) + \rho \boldsymbol{x}_{(t)} \quad if\ \boldsymbol{x}_{(t)} \in \omega_1\ dan\ \boldsymbol{w}^{\boldsymbol{T}}(t)\boldsymbol{x}_{(t)} \leq 0 \\ \boldsymbol{w}(t+1) &= \boldsymbol{w}(t) - \rho \boldsymbol{x}_{(t)} \quad if\ \boldsymbol{x}_{(t)} \in \omega_2\ dan\ \boldsymbol{w}^{\boldsymbol{T}}(t)\boldsymbol{x}_{(t)} \geq 0 \\ \boldsymbol{w}(t+1) &= \boldsymbol{w}(t)\ \ jika\ selainya \end{aligned} \tag{3.21}$$

Dengan kata lain, jika sampel pelatihan sekarang terklasifikasi secara benar, tidak ada tindakan yang diambil. Sebaliknya, jika sampel tersebut tidak terklasifikasi secara benar, vektor bobot dikoreksi dengan menambahkan (mengurangkan) sejumlah proposional kepada $\boldsymbol{x}_{(t)}$. Algoritma ini termasuk sebuah keluarga algoritmis yang lebih umum yang dikenal sebagai pola *reward and punishment*. Jika pengklasifikasian benar, *reward*-nya adalah tidak ada tindakan yang diambil. Jika vektor sekarang adalah keliru diklasifikasi, *punishment*-nya adalah harga koreksi. Dapat ditunjukkan bahwa bentuk algoritma perceptron ini juga konvergen dalam sejumlah berhingga langkah iterasi.

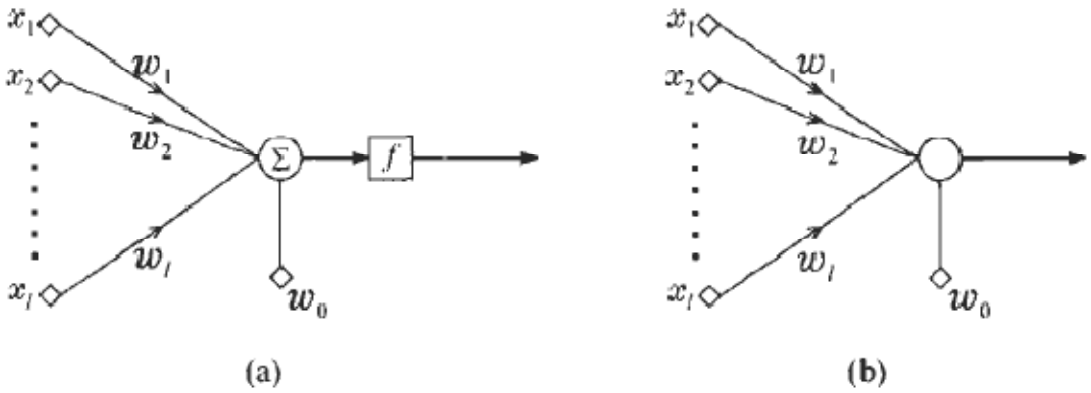
Algoritma perceptron pada asalnya ditawarkan oleh Rosenblatt di akhir 1950an. Algoritma ini dikembangkan untuk pelatihan perceptron, satuan dasar untuk pemodelan neuron-neuron otak. Ini dianggap pokok dalam pengembangan model-model yang powerful untuk pembelajaran mesin (*machine learning*) [Rose 58, Min 88].

Perceptron

Sekali algoritma perceptron telah konvergen ke sebuah vektor bobot $\boldsymbol{w}$ dan sebuah ambang w_0, tujan berikutnya adalah klasifikasi sebuah vetor ciri yang tidak diketahui ke salah satu dari dua kelas. Klasifikasi dicapai melalui aturan sederhana

$$\begin{aligned} &\text{Jika} \quad \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + w_0 > 0 \qquad \text{tempatkan } \boldsymbol{x} \text{ ke } \omega_1 \\ &\text{Jika} \quad \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + w_0 < 0 \qquad \text{tempatkan } \boldsymbol{x} \text{ ke } \omega_2 \end{aligned} \tag{3.22}$$

Sebuah unit jaringan dasar yang mengimplementasikan operasi ini diperlihatkan dalam Gambar 3.3(a).



Gambar 3.3 Model perceptron dasar.

Elemen-elemen vektor ciri x_1, x_2,..., x_3 digunakan pada node-node input dari jaringan tersebut. Masing-masing dikalikan dengan bobot-bobot w_i, i = 1, 2, …, l. Ini dikenal sebagai bobot synaptic atau sinapsis saja. Hasil kali-hasil kali ini dijumlahkan bersama dengan nilai ambang w_0. Hasilnya kemudian menuju sebuah alat nonlinear, yang mengimplementasikan fungsi aktivasi. Sebuah pilihan biasa adalah pembatas tegas, yaitu, $f(\cdot)$ adalah fungsi langkah [$f(x) = -1$ jika $x < 0$ dan $f(x) = 1$ jika $x > 0$]. Vektor ciri yang bersesuaian diklasifikasikan kedalam salah satu kelas tergantung pada tanda output. Disamping +1 dan -1, nilai-nilai lain (label-label kelas) untuk pembatas tegas adalah juga memungkinkan. Pilihan populer lainnya adalah 1 dan 0 dan ini dicapai dengan memilih dua level dari fungsi langkah dengan tepat.

Jaringan dasar ini dikenal sebagai sebuah perceptron atau neuron. Perceptron-perceptron adalah contoh sederhana dari mesin pembelajaran (*learning machine*), yaitu, struktur-struktur yang parameter bebasnya di-update dengan algoritma pembelajaran, seperti algoritma perceptron, supaya "belajar" sebuah tugas yang spesifik, didasarkan pada sebuah himpunan data pembelajaran. Pada bagian yang akan datang akan digunakan perceptron sebagai elemen pembangun dasar untuk jaringan-jaringan pembelajaran yang lebih kompleks. Gambar 3.3b adalah sebuah grafik yang disederhanakan dari neuron dimana perangkat penjumlah dan nonlinear telah digabung untuk penyederhanaan notasi. Kadang-kadang sebuah neuron dengan sebuah perangkat pembatas tegas diacu sebagai sebuah neuron McCulloch-Pitts.

Algoritma Pocket

Sebuah persyaratan dasar untuk konvergensi algoritma perceptron adalah kemampuan dipisah secara linear dari kelas-kelas. Jika ini benar, sebagaimana biasanya kasus dalam praktek, algoritma perceptron tidak konvergen. Sebuah varian dari algoritma perceptron diusulkan dalam [Gal 90] yang konvergen pada sebuah solusi optimal meskipun kondisi dapat dipisah secara linear tidak terpenuhi. Algoritma ini dikenal sebagai algoritma pocket dan terdiri dari dua langkah berikut

- Inisialisasi vektor bobot $\boldsymbol{w}(0)$ secara acak. Tetapkan sebuah vektor yang disimpan (dalam saku/*pocket*!). Set sebuah counter/pencacah history h_s dari $\boldsymbol{w_s}$ menjadi nol.
- Pada langkah iterasi ke-t hitung update $\boldsymbol{w}(t+1)$, sesuai dengan aturan perceptron. Gunakan vektor bobot yang di-update untuk menguji jumlah h vektor pelatihan yang diklasifikasikan secara benar. Jika $h > h_s$, gantikan $\boldsymbol{w_s}$ dengan $\boldsymbol{w}(t+1)$ dan h_s dengan h. Lanjutkan iterasi.

Dapat ditunjukkan bahwa algoritma ini konvergen dengan probabilitas satu ke solusi optimal, yaitu, yang menghasilkan jumlah minimum salah klasifikasi [Gal 90, Muse 97]. Algoritma terkait lainnya yang

menemukan solusi-solusi yang agak baik pada saat kelas-kelas tidak dapat dipisahkan secara linear adalah algoritma perceptron termal (*thermal perceptron algorithm*) [Frea 92], algoritma minimisasi kerugian (*loss minimization algorithm*) [Hryc 82] dan prosedur koreksi barycentric [Poul 95].

Konstruksi Kesler (*Kesler's Construction*)

Sejauh ini telah diuraikan kasus dua kelas. Generalisasi pada sebuah tugas *M*-kelas adalah pendekatan. Sebuah fungsi diskriminan linear $\boldsymbol{w}_i$, i = 1, 2, …, M, didefinisikan untuk masing-masing kelas. Sebuah vektor ciri $\boldsymbol{x}$ (dalam ruang (l + 1)-dimensi untuk menerangkan/menyebabkan ambang) diklasifikasi dalam kelas ω_i jika

$$\boldsymbol{w}_i^T x > \boldsymbol{w}_j^T x, \quad \forall j \neq i \tag{3.23}$$

Kondisi ini membawa kepada apa yang disebut konstruksi Kesler. Untuk setip vektor pelatihan dari kelas $\boldsymbol{w}_i$, i = 1, 2, …, M, dapat dikonstruksi M – 1 vektor $\boldsymbol{x}_{ij} = [\boldsymbol{0}^T, \boldsymbol{0}^T, \dots, \boldsymbol{x}^T, \dots, -\boldsymbol{x}^T, \dots, \boldsymbol{0}^T]^{\boldsymbol{T}}$ dimensi (l + 1)M x 1. Yaitu, mereka merupakan vektor-vektor blok (*block vectors*) yang mempunyai nol di semua tempat kecuali pada posisi blok ke–i dan ke-j, dimana mereka mempunyai masing-masing $\boldsymbol{x}$ dan $-\boldsymbol{x}$, untuk $j \neq i$. Kita juga mengkonstruksi vektor blok $\boldsymbol{w} = [\boldsymbol{w}_1^T, \dots, \boldsymbol{w}_M^T]^T$. Jika $\boldsymbol{x} \in \omega_{\boldsymbol{i}}$, ini menentukan kondisi/syarat bahwa $\boldsymbol{w}^T \boldsymbol{x}_{\boldsymbol{ij}} > 0, \forall j = 1,2, \dots, M, j \neq i$. Tugas sekarang adalah merancang sebuah pengklasifikasi linear, dalam ruang M-dimensi yang diperluas. Sehingga masing-masing (M – 1) N vektor pelatihan terletak dalam sisi positifnya. Algoritma perceotron tidak akan mempunyai kesulitan dalam memecahkan permasalahan ini untuk kita, asalkan bahwa sebuah solusi seperti itu adalah mungkin, yaitu, jika semua vektor pelatihan dapat diklasifikasi dengan benar menggunakan sebuah himpunan fungsi-fungsi diskriminan linear.

3.4 METODE KUADRAT TERKECIL (LEAST SQUARES METHODS)

Seperti yang telah ditunjukkan, daya tarik pengklasifikasi linear terletak pada kesederhanaannya. Dengan demikian, dalam banyak kasus, meskipun diketahui bahwa kelas-kelas yang tidak dipisahkan secara linear, masih tetap mengadopsi pengklasifikasi linier, meskipun fakta bahwa hal ini akan mengarah pada kinerja suboptimal dari probabilitas kesalahan klasifikasi pengamatan. Tujuannya sekarang adalah untuk menghitung vektor bobot yang sesuai di bawah kriteria optimalitas yang cocok.

3.4.1 Estimasi Galat Rerata Kuadrat

Subbabini memfokuskan pada masalah dua kelas. Pada bagian sebelumnya dilihat bahwa output perceptron adalah ±1, tergantung pada kepemilikan kelas x. Karena kelas yang linier terpisah, output ini adalah benar untuk semua vektor ciri pelatihan, tentu saja sesuai dengan sifat konvergensi algoritma perceptron itu. Pada bagian ini akan dicoba untuk merancang pengklasifikasi linear sehingga output yang diinginkan ± 1, tergantung pada kepemilikan kelas dari vektor input. Namun, akan terus didapatkan galat, yaitu output yang sebenarnya tidak akan selalu sama dengan yang diinginkan. Mengingat vektor x, output dari pengklasifikasi akan menjadi $\mathrm{w}^{\mathrm{T}}\mathrm{x}$ (ambang batas dapat diakomodasi dengan ekstensi vektor). Output yang diinginkan akan dinotasikan sebagai $\mathrm{y}(\mathrm{x}) \equiv \mathrm{y} = \pm 1$. Vektor bobot akan dihitung sehingga dapat meminimalkan *Mean Square Error* (MSE) antara output yang diinginkan dan output yang sebenarnya, yaitu:

$$\mathrm{J}(\mathrm{w}) = \mathrm{E}[|\mathrm{y} - \mathrm{x}^{\mathrm{T}}\mathrm{w}|^2] \tag{3.24}$$

$$\widehat{w} = \arg\min_{\mathrm{w}} J(w) \tag{3.25}$$

Dapat dengan mudah diperiksa bahwa J(w) sama dengan:

$$J(w) = P(\omega_1)\int(1 - x^T\omega)^2 p(x|\omega_1)dx + P(\omega_2)\int(1 + x^T w)^2 p(x|\omega_2)d\,x \qquad (3.26)$$

Dengan meminimalkan (3.24) dengan mudah menghasilkan:

$$\frac{\partial\, J(w)}{\partial w} = 2E[x(y - x^T w)] = 0 \qquad (3.27)$$

Kemudian

$$\widehat{w} = R_x^{-1}E[xy] \qquad (3.28)$$

Dimana

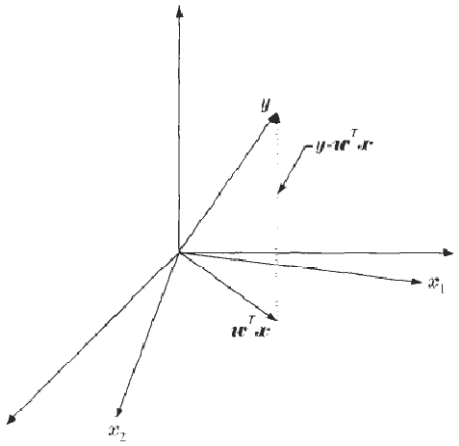
$$R_x \equiv E[xx^T] = \begin{bmatrix} E(x_1x_1) & \cdots & E(x_1x_l) \\ E(x_2x_1) & \dots & E(x_2x_l) \\ \vdots & \vdots & \vdots \\ E(x_lx_1) & \cdots & E(x_lx_l) \end{bmatrix} \qquad (3.29)$$

dikenal sebagai korelasi atau autokorelasi matriks dan sama dengan matriks kovarian, yang diperkenalkan pada bab sebelumnya, jika nilai rata-rata masing-masing adalah nol. Vektor

$$E[xy] = E\left[\begin{bmatrix} x_1y \\ \vdots \\ x_ly \end{bmatrix}\right] \qquad (3.30)$$

dikenal sebagai korelasi-silang antara output yang diinginkan dan (input) vektor ciri. Dengan demikian, hasil rerata kuadrat optimal vektor bobot sebagai solusi dari himpunan persamaan linier, tentu saja bahwa matriks korelasi adalah invertible.

Sangat menarik untuk menunjukkan bahwa ada interpretasi geometris pada solusi ini. Variabel Acak dapat dianggap sebagai titik dalam ruang vektor. Hal ini mudah untuk melihat bahwa operasi harapan E [xy] antara dua variabel acak memenuhi sifat-sifat *inner product*. Memang, $\mathrm{E}[\mathrm{x}^2] \geq 0, \mathrm{E}[\mathrm{xy}] = \mathrm{E}[\mathrm{yx}], \mathrm{E}[\mathrm{x}\,(\mathrm{c}_1\mathrm{y} + \mathrm{c}_2\mathrm{z})] = \mathrm{c}_1\mathrm{E}\,[\mathrm{xy}] + \mathrm{c}_2\mathrm{E}[\mathrm{xz}]$, dalam ruang vektor $\mathrm{w}^{\mathrm{T}}\mathrm{x} = \mathrm{w}_1\mathrm{x}_1 + \cdots + \mathrm{w}_1\mathrm{x}_1$ adalah kombinasi linier dari vektor sehingga terletak pada subspace yang didefinisikan oleh $\mathrm{x_i}'\mathrm{s}$.



Gambar 3.5: Interpretasi perkiraan MSE sebagai proyeksi ortogonal pada *subspace* elemen vektor input.

Hal ini diilustrasikan oleh contoh pada Gambar 3.5. Kemudian, jika nilai y ingin dicari dengan kombinasi linear ini, galat yang dihasilkan adalah $y - w^T x$. Persamaan (3.27) menyatakan bahwa solusi MSE minimal yang didapat jika kesalahan tersebut ortogonal untuk setiap x_i. sehingga akan ortogonal terhadap vektor subspace terbentang oleh x_i, i = 1,2,. . . . l. Dengan kata lain, jika y didekati oleh proyeksi ortogonal terhadap subspace (Gambar 3.5). Persamaan (3.27) juga dikenal sebagai kondisi orthogonal.

Generalisasi Multikelas

Dalam kasus multikelas tugasnya adalah untuk merancang fungsi diskriminan linear M $g_i(x) = w_i^T x$ sesuai dengan kriteria MSE. Respon output yang diinginkan (misalnya, label kelas) dipilih sehingga $y_i = l\ jika\ x \in w_i$ dan $y_j = 0$ dan sebaliknya. Hal ini sesuai dengan kasus dua kelas. Memang, untuk semacam pilihan dan jika M = 2. desain *hyperplane* keputusan $w^T x \equiv (w_1 - w_2)^T x$ sesuai dengan ± 1 tanggapan yang diinginkan, tergantung pada kepemilikan kelas masing-masing.

Didefinisikan $y^T = [y_1, \ldots, y_m]$, untuk vektor **x** yang diberikan, dan $W = [w_1, \ldots w_m]$. Yaitu, matriks W sebagai vektor kolom bobot w_i. Kriteria MSE dalam (3.25) sekarang dapat digeneralisasi untuk meminimalkan *norm* dari vektor galat $y - W^T x$, yaitu,

$$\widehat{W} = \arg\min_w E[||y - W^T x||^2] = \arg\min_w E[\sum_{i=1}^{M} (y_i - w_i^T x)^2] \tag{3.31}$$

Hal ini ekivalen dengan M permasalahan minimisasi independen MSE dari tipe (3.25), dengan respon skalar yang diinginkan. Dengan kata lain, untuk desain fungsi diskriminan linier optimal MSE, cukup merancang salah satunya sehingga output yang diinginkannya adalah 1 untuk vektor yang memilik kelas yang sesuai dan 0 untuk yang lainnya.

3.4.2 Aproksimasi Stokastik dan Algoritma LMS

Solusi dari (3.28) membutuhkan perhitungan matriks korelasi dan vektor cross-korelasi. Hal ini mengandaikan pengetahuan tentang yang mendasari distribusi, yang pada umumnya tidak diketahui. Tujuan utamanya sekarang adalah untuk melihat apakah mungkin untuk memecahkan (3,27) tanpa tersedianya informasi statistik. Jawabannya telah disediakan oleh Robbins dan Monro [Robb 51] dalam konteks yang lebih umum dari teori pendekatan stokastik. Pertimbangkan persamaan dari bentuk $E\ [F\ (x_k, w)] = 0$, dimana x_k, k = 1,2,. . . , Adalah vektor urutan acak dari distribusi yang sama, F (., .) adalah sebuah fungsi, dan w adalah vektor parameter yang tidak diketahui. Lalu mengadopsi skema iteratif

$$\widehat{w}(k) = \widehat{w}(k-1) + \rho_k F(x_k, \widehat{w}(k-1)) \tag{3.32}$$

Dengan kata lain, tempat nilai rata-rata (yang tidak dapat dihitung karena kurangnya informasi) diambil oleh sampel dari variabel-variabel acak yang dihasilkan dari percobaan. Ternyata bahwa dalam kondisi ringan pola iteratif kovergen dalam probabilitas ke solusi w dari persamaan asli, asalkan urutan ρ_k memenuhi dua kondisi

$$\sum_{k=1}^{\infty} \rho_k \to \infty \tag{3.33}$$

$$\sum_{k=1}^{\infty} \rho_k^2 < \infty \tag{3.34}$$

dan yang menyiratkan bahwa

$$\rho_k \to 0 \tag{3.35}$$

Yaitu

$$\lim_{k\to 0} prob\{\hat{w}(k) = w\} = 1 \tag{3.36}$$

Semakin kuat, dalam rerata kuadrat, konvergensi juga adalah benar

$$\lim_{k\to 0} E[||\hat{w}(k) = w||^2] = 0 \tag{3.37}$$

Kondisi (3.33), (3.34) telah dipenuhi sebelumnya dan menjamin bahwa koreksi dari estimasi dalam iterasinya cenderung nol. Jadi, selama nilai k yang besar (dalam teori infinitif) iterasinya terhenti. Namun, ini tidak boleh terjadi terlalu awal (kondisi awal) untuk memastikan bahwa iterasi tidak berhenti jauh dari solusi. Kondisi kedua bahwa derau yang terakumulasi, karena sifat stokastik dari variabel, tetap terbatas dan algoritma tersebut dapat mengatasinya[Fuku 90]. Buktinya adalah di luar lingkup dari teks ini. Namun, akan ditunjukkan kebenarannya melalui contoh. Pertimbangkanlah persamaan sederhana $E[x_k - w] = 0$. Untuk $\rho_k = {}^1/_k$ iterasi menjadi

$$\hat{w}(k) = \hat{w}(k-1) + \frac{1}{k}[x_k - \hat{w}(k-1)] = \frac{k-1}{k}\hat{w}(k-1) + \frac{1}{k}x_k$$

Untuk nilai k yang besar adalah mudah untuk melihat bahwa

$$\hat{w}(k) = \frac{1}{k}\sum_{r=1}^{k} x_r$$

Tampak bahwa solusinya adalah mean sampel pengukuran. Paling Natural!

Sekarang kembali ke masalah asal dan menerapkan iterasi tersebut untuk memecahkan (3.27). Kemudian (3.32) menjadi

$$\hat{w}(k) = \hat{w}(k-1) + \rho_k x_k(y_k - x_k^T \hat{w}(k-1) \tag{3.38}$$

dimana (y_k, x_k) adalah output yang diinginkan (± 1) - input pasangan sampel pelatihan, berturut-turut diberikan kepada algoritma. Algoritma ini dikenal sebagai *least mean squares* (LMS) atau algoritma Widrow-Hoff. Algoritma ini konvergen secara asimtotik menuju solusi MSE.

Sejumlah varian dari algoritma LMS telah diusulkan dan digunakan. Sebagai contoh[Hayk 96, Kalou 93], sebuah varian yang umum adalah dengan menggunakan ρ konstan dalam ρ_k. Namun, dalam hal ini algoritma tidak konvergen ke solusi MSE. Hal ini dapat ditunjukkan, misalnya[Hayk 96], bahwa jika $0 < \rho < 2/trace(R_x)$ maka

$$E[\hat{w}(k)] \to w_{MSE} \quad dan \quad E[||\hat{w}(k) = w||^2] \to konstant \tag{3.39}$$

Dimana w_{MSE} menunjukkan estimasi optimal MSE dan trace{.} trace matriks. Artinya, nilai rata-rata estimasi LMS adalah sama dengan solusi MSE dan juga varian yang sesuai masih terbatas. Ternyata bahwa semakin kecil ρ, semakin kecil varians di sekitar solusi MSE yang diinginkan. Namun, semakin kecil ρ,

semakin lambat konvergensi dari algoritma LMS. Alasan untuk menggunakan konstanta ρ di tempat urutan hilang adalah untuk menjaga algoritma "alert" untuk melacak variasi apabila statistik tersebut tidak stasioner tetapi perlahan-lahan bervariasi, yaitu, ketika distribusi yang mendasari adalah terikat waktu.

3.4.3 Sum of Error Squares Estimation (Jumlah Kuadrat Kesalahan Estimasi)

Sebuah kriteria yang terkait erat dengan MSE adalah jumlah kriteria galat kuadrat didefinisikan sebagai

$$J(w) = \sum_{i=1}^{N} (y_i - x_i^T w)^2 \equiv \sum_{i=1}^{N} e_i^2 \tag{3.40}$$

Dengan kata lain, galat antara output yang diinginkan dari pengklasifikasi (± 1 dalam kasus dua kelas) dan output yang sebenarnya dijumlahkan kepada semua vektor ciri pelatihan yang tersedia, bukannya merata-ratakan vektor ciri tersebut. Dengan demikian kebutuhan informasi yang eksplisit dari pdf (*probability density function*) dapat dipenuhi. Dengan meminimalkan (3,40) yang berkenaan dengan w menghasilkan

$$\sum_{i=1}^{N} x_i((y_i - x_i^T \hat{w}) = 0 \rightarrow \left(\sum_{i=1}^{N} x_i x_i^T \right) \hat{w} = \sum_{i=1}^{N} (x_i y_i) \tag{3.41}$$

Untuk formulasi matematika, didefinisikan:

$$x = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \quad y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} \tag{3.42}$$

Artinya, *X* adalah matriks *N* x *l* yang barisnya adalah vektor ciri pelatihan yang tersedia, dan y adalah vektor yang terdiri dari respon yang diinginkan. Kemudian $\sum_{i=1}^{N} x_i x_i^T = X^T X$ dan juga $\sum_{i=1}^{N} x_i y_i = X^T y$. Oleh karena itu, (3.41) sekarang dapat ditulis sebagai

$$(X^T X)\hat{w} = X^T y \Rightarrow \hat{w} = (X^T X)^{-1} X^T y \tag{3.43}$$

Dengan demikian, vektor bobot optimal diberikan sebagai solusi dari himpunan persamaan linier. Matrix $X^T X$ dikenal sebagai matriks korelasi sampel. Matrix $X^+ \equiv (X^T X)^{-1} X^T$ dikenal sebagai *pseudoinverse* dari X, dan ini hanya berarti jika $X^T X$ adalah invertible, X dimensi *l*. X^+ adalah generalisasi dari invers matriks bujur sangkar yang invertible. Memang, jika X adalah matrik bujur sangkar dan invertible berdimensi *l x l*, maka mudah untuk melihat bahwa $X^+ = X^{-1}$. Dalam kasus seperti vektor bobot diperkirakan adalah solusi dari sistem linier $X\hat{w} = y$. Namun, jika ada persamaan lebih dari variabel yang tidak diketahui, N> *l*, seperti halnya yang biasa dalam pengenalan pola, pada umumnya tidak memiliki solusi. Solusi yang diperoleh dengan *pseudoinverse* adalah vektor yang meminimalkan jumlah kuadrat kesalahan.

3.5 ESTIMASI MEAN SQUARE REVISITED

3.5.1 Regresi Mean Square Error

Dalam subbagian ini akan didekati tugas MSE dari perspektif yang sedikit berbeda dan dalam kerangka yang lebih umum

Biarkan y, x menjadi dua vektor acak dari dimensi M x l dan l x 1, berturut-turut, dan mengasumsikan bahwa mereka digambarkan oleh pdf bersama p(y, x). Tugas yang menarik adalah untuk memperkirakan nilai dari y, yang diberi nilai x, yang diperoleh dari percobaan. Tidak diragukan lagi tugas klasifikasi jatuh di bawah formulasi yang lebih umum ini. Sebagai contoh, ketika diberi vektor ciri x, tujuannya adalah untuk memperkirakan nilai dari label kelas y, yang merupakan ± 1 dalam kasus dua kelas.

Estimasi rerata kuadrat $\hat{y}$ dari vektor y acak, diberi nilai x, didefinisikan sebagai

$$\hat{y} = \arg\min_{\hat{y}} E[\|y - \hat{y}\|^2] \tag{3.44}$$

Perhatikan bahwa nilai rata-rata di sini adalah berkenaan dengan pdf bersyarat $p(y|x)$. Akan ditunjukkan bahwa estimasi optimal adalah nilai rata-rata dari y, yaitu,

$$\hat{y} = E[y|x] \equiv \int_{-\infty}^{\infty} yp(y|x)\, dy \tag{3.45}$$

3.5.2 Estimasi MSE Probabilitas Kelas Posterior

Akan dipertimbangkan kasus multikelas. Diberikan x, diinginkan untuk mengestimasi label kelasnya. Biarkan $g_i(x)$ adalah fungsi diskriminan yang harus dirancang. *Cost funtion* dalam Persamaan (3.31) sekarang menjadi

$$J = E\left[\sum_{i=1}^{M}(g_i(x) - y_i)^2\right] \equiv E[\|g(x) - y\|^2] \tag{3.49}$$

dimana vektor y terdiri dari nol dan 1 tunggal pada tempat yang tepat. Perhatikan bahwa setiap $g_i(x)$ hanya bergantung pada x, sedangkan y_i tergantung pada kelas w_j yang milik x. Biarkan p(x, w_i) menjadi probabilitas densitas gabungan dari vektor ciri milik kelas i. Kemudian (3.49) ditulis sebagai

$$J = \int_{-\infty}^{\infty}\sum_{i=1}^{M}\left\{\sum_{i=1}^{M}(g_i(x) - y_i)^2\right\}\mathrm{p}(\mathrm{x}, w_j)\,\mathrm{dx} \tag{3.50}$$

Dengan mempertimbangkan bahwa $\mathrm{p}(\mathrm{x}, w_j)$ = P (w_j|x) p (x), (3.50) menjadi

$$\mathrm{J} = \int_{-\infty}^{\infty}\left\{\sum_{i=1}^{M}\sum_{i=1}^{M}(g_i(x) - y_i)^2\ \mathrm{P}\left(w_j|\mathrm{x}\right)\right\}\ \mathrm{p(x)dx}$$

$$= E\left[\sum_{i=1}^{M}\sum_{i=1}^{M}(g_i(x) - y_i)^2\ \mathrm{P}\left(w_j|\mathrm{x}\right)\right] \tag{3.51}$$

Yang mana rerata diambil sehubungan dengan x. Dengan memperluas ini, didapatkan

$$j = E\left[\sum_{i=1}^{M}\sum_{i=1}^{M}\left({g_i}^2(x)\mathrm{P}\left(w_j|\mathrm{x}\right) - 2g_i(x)y_i\mathrm{P}\left(w_j|\mathrm{x}\right) + \mathrm{y_i^2P}\left(w_j|\mathrm{x}\right)\right)\right] \tag{3.52}$$

Dengan memanfaatkan fakta bahwa g_i (x) adalah fungsi dari x saja dan $\sum_{j=i}^{M}\mathrm{P}\left(w_j|\mathrm{x}\right) = 1$, (3.52) menjadi

$$j = E\left[\sum_{i=1}^{M}\left({g_i}^2(x) - 2g_i(x)\sum_{\mathrm{i=1}}^{\mathrm{M}}y_i\mathrm{P}\left(w_j|\mathrm{x}\right) + \sum_{\mathrm{i=1}}^{\mathrm{M}}\mathrm{y_i^2P}\left(w_j|\mathrm{x}\right)\right)\right]$$

$$= \sum_{i=1}^{M} \left({g_i}^2(x) - 2g_i(x)_i \mathrm{E}\,(y_i|\mathrm{x}) + \mathrm{E}\,({y_i}^2|\mathrm{x}) \right) \tag{3.53}$$

dimana E $(y_i|\mathrm{x})$ dan E $({y_i}^2|\mathrm{x})$ adalah nilai rata-rata masing-masing dikondisikan pada x. Dengan menambah dan mengurangkan $\left(\mathrm{E}\,(y_i|\mathrm{x})\right)^2$. Persamaan (3.53) menjadi

$$J = E\left[\sum_{i=1}^{M} (g_i(x) - y_i)^2\right] + E\left[\sum_{i=1}^{M} \left(E[y_i^2|x] - (E[y_i|x)^2\right)\right] \tag{3.54}$$

Syarat kedua dalam (3.54) tidak tergantung pada fungsi $g_i(x)$. i = 1,2,. . . , M. Dengan demikian, minimisasi J berkenaan dengan (parameter) $g_i(.)$ hanya mempengaruhi yang pertama dari dua syarat. Mari berkonsentrasi dan melihatnya lebih hati-hati. Setiap M *summands* melibatkan dua syarat: fungsi diskriminan yang tidak diketahui g_i (.) dan rata-rata bersyarat dari respon yang diinginkan yang sesuai. Selanjutnya dnulis g_i (.) = g_i $(\cdot; w_i)$, untuk menyatakan secara eksplisit bahwa fungsi yang didefinisikan dalam bentuk satu set parameter, akan ditentukan secara optimal selama pelatihan. Meminimalkan J sehubungan dengan w_i, i = I, 2,. . . , M, menghasilkan perkiraan kuadrat rata-rata dari parameter yang tidak diketahui, w_i, sehingga fungsi diskriminan mengestimasi secara optimal rata-rata bersyarat yang bersesuaian - yaitu, regresi dari yi dikondisikan pada x. Selain itu, untuk masalah M-kelas dan definisi sebelumnya telah dimiliki

$$E[y_i|x] \equiv \sum_{j=1}^{M} y_i P(w_j x) \tag{3.56}$$

Namun $y_i = I(0)$ jika $x \in w_i$($x \in w_j, j \neq i$). Karenanya

$$g_i(x, \widehat{w}_i) adalah\ perkiraan\ MSE\ dari\ P(w_i|x) \tag{3.56}$$

Ini adalah hasil yang penting. Pelatihan fungsi diskriminan g_i dengan output yang diinginkan 1 atau 0 dalam arti MSE, Persamaan. (3.49) adalah ekivalen untuk memperoleh estimasi MSE dari probabilitas posterior kelas, tanpa menggunakan informasi statistik atau pemodelan pdf! Ini sudah cukup untuk mengatakan bahwa estimasi ini pada gilirannya dapat digunakan untuk klasifikasi Bayesian. Suatu hal yang penting di sini adalah untuk menilai seberapa baik perkiraan yang dihasilkan. Itu semua tergantung pada seberapa baik fungsi yang diadopsi $g_i(\cdot; w_i)$ dapat memodelkan (secara umum) fungsi nonlinear yang diinginkan $P(w_i|x)$. Jika, misalnya, diadopsi model linier, seperti yang terjadi pada Persamaan(3.31), dan $P(w_i|x)$ sangat nonlinear, maka aproksimasi optimal MSE yang dihasilkan akan menjadi buruk. Fokus dalam bab berikutnya akan pada pengembangan teknik pemodelan untuk fungsi nonlinier.

Akhirnya, harus ditekankan bahwa kesimpulan di atas merupakan implikasi *cost function* itu sendiri dan bukan fungsi model spesifik yang digunakan. Yang terakhir memegang peranan pada saat isu akurasi aproksimasi datang ke tempat kejadian. Harga MSE hanyalah salah satu dari harga yang memiliki sifat penting. *Cost function* lainnya berbagi sifat ini juga, untuk contoh bisa dilihat, [Rich 91, Bish 95, Pear 90, Cid 99].

3.5.3 Dilema Bias-Varians

Sejauh ini telah dibahas beberapa masalah yang sangat penting mengenai interpretasi output dari sebuah pengklasifikasi yang dirancang secara optimal. Dilihat bahwa pengklasifikasi yang dapat dilihat sebagai

sebuah mesin belajar mewujudkan satu set fungsi g(x), yang mencoba memperkirakan kelas label y yang sesuai dan membuat keputusan berdasarkan pada perkiraan ini. Dalam prakteknya, fungsi g(.) diperkirakan menggunakan data set pelatihan yang berhingga $D = \{(y_i, x_i), i = 1,2, \dots, N\}$ dan metodologi yang sesuai (misalnya, jumlah kuadrat kesalahan, LMS, maksimum likelihood). Untuk menekankan ketergantungan eksplisit pada D dinulis g (x; D). Sub bagian ini difokuskan pada kemampuan g (x; D) untuk memperkirakan regressor MSE yang optimal E [y|x] dan tentang bagaimana hal ini dipengaruhi oleh ukuran data N.

Faktor kunci di sini adalah ketergantungan pendekatan pada D. Pendekatan tersebut mungkin sangat baik untuk data set pelatihan tertentu tetapi sangat buruk bagi lain. Efektivitas estimator dapat dievaluasi dengan menghitung deviasi rerata kuadrat dari nilai optimal yang diinginkan. Hal ini dapat dicapai dengan merata-ratakan semua set D yang mungkin dengan ukuran N, yaitu,

$$E_D[(g(x;D) - E[y|x])^2] \tag{3.57}$$

Jika kita menambah dan mengurangi ED g [(x; 271 dan ikuti prosedur yang sama dengan yang di bukti (3.43), kita mudah memperoleh

$$E_D[(g(x;D) - E(y|x)^2 = (E_D[(x;D)] - E[y|x])^2 + E_D[(g(x;D) - E_D[g(x;D)])^2] \tag{3.58}$$

Syarat pertama adalah kontribusi bias dan yang kedua adalah varian. Dengan kata lain, jika estimator tidak berbias, masih bisa menghasilkan MSE yang besar karena syarat varians yang besar. Untuk sebuah data set yang terbatas, ternyata ada trade-off antara kedua syarat ini. Meningkatkan bias mengurangi varians dan sebaliknya. Hal ini dikenal sebagai dilema bias-variance. Perilaku ini adalah wajar. Permasalahannya mirip seperti kurva fitting melalui serangkaian data yang diberikan. Jika model yang diadopsi adalah kompleks (banyak parameter yang terlibat) dengan memperhatikan jumlah N, model akan sesuai dengan keistimewaan dari data set tertentu. Dengan demikian, akan menghasilkan bias rendah tetapi akan menghasilkan varians yang tinggi, seperti yang dirubah dari satu data set yang lain. Masalah utama sekarang adalah untuk mencari cara untuk membuat kedua bias dan varians menjadi rendah pada saat yang sama. Ternyata bahwa ini mungkin hanya asimtotik, sebagai nila N yang tumbuh menuju tak terhingga. Selain itu, N telah tumbuh sedemikian rupa untuk memungkinkan model yang lebih kompleks, g, untuk dipasang (yang mengurangi bias) dan pada saat yang sama untuk memastikan nilai varians rendah. Namun, dalam praktiknya N adalah terbatas dan satu harus bertujuan pada kompromi terbaik. Jika, di sisi lain, beberapa pengetahuan *priori* tersedia, ini harus dieksploitasi dalam bentuk kendala yang pengklasifikasi harus memuaskan. Hal ini dapat menyebabkan nilai lebih rendah dari varians dan bias, dibandingkan dengan jenis pengklasifikasi yang lebih umum. Hal ini wajar, karena mengambil keuntungan dari informasi yang tersedia dan membantu proses optimasi. Sebuah perlakuan yang sederhana dan sangat baik dari topik dapat ditemukan di [Gema 92].

3.6 MESIN VEKTOR PENDUKUNG (*SUPPORT VECTOR MACHINE*)

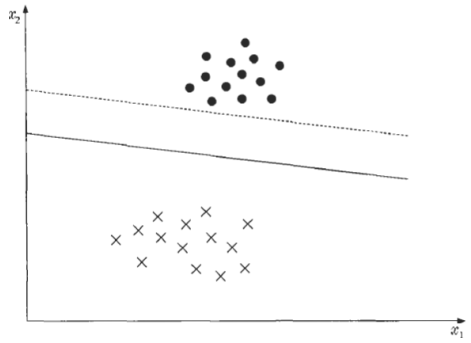
3.6.1 Kelas-Kelas yang Bisa Dipisah (*Separable Classes*)

Pada subbab ini akan dibahas sebuah alternatif yang lebih rasional dalam merancang pengklasifikasi linear. Bagian ini akan dimulai dengan pemisahan secara linear menjadi dua kelas, lalu dilanjutkan dengan kasus data yang tidak bisa dipisahkan secara linear.

Dimisalkan sebuah vektor x_i, i=1,2,…,N merupakan vektor ciri dari himpunan pelatihan, X. Vektor ini merupakan bagian dari dua kelas, yakni ω_1 dan ω_2, yang diasumsikan terpisah secara linear. Tujuan dari klasifikasi ini adalah merancang sebuah *hyperplane* yang dirumuskan dengan:

$$g(x) = w^T x + w_0 = 0 \tag{3.59}$$

yang dapat mengklasifikasikan dengan tepat semua vektor pelatihan. Seperti yang telah dibahas pada subbab 3.3, *hyperplane* seperti ini tidak unik (bukan hal khusus). Algoritma Perceptron dapat menemukan salah satu penyelesaian yang mungkin. Gambar 3.7 menggambarkan klasifikasi dengan dua penyelesaian *hyperplane* yang mungkin. *Hyperplane* ini digambarkan sebagai garis lurus. Tampak bahwa kedua *hyperplane* dapat memisahkan kedua data dengan tepat. Namun, pengklasifikasi yang tepat adalah garis solid karena *hyperplane* ini memberikan ruang yang banyak pada kedua kelas sehingga data baru dapat mnempati salah satu kelas dengan bebas dan resiko kesalahan menjadi kecil. *Hyperplane* ini dapat diandalkan bila dihadapkan dengan data yang belum dikenal. Hal tersebut merupakan yang terpenting ketika merancang sebuah pengklasifikasi. Sifat ini dikenal dengan *generalization performance of the pengklasifikasi* (sifat generalisasi dari sebuah pengklasifikasi). Artinya adalah kemampuan sebuah pengklasifikasi ketika telah dilatih menggunakan data pelatihan tertentu, akan memberikan hasil yang benar ketika diberi data yang berbeda dari sebelumnya. Dapt disimpulkan bahwa *hyperplane* yang paling baik sebagai pengklasifikasi adalah yang dapat memberikan batas maksimum (ruang lebih) untu kedua kelas.



Gambar 3.7: Sebuah contoh masalah dua kelas yang terpisah secara linear dengan dua pengklasifikasi linear yang mungkin

Lalu akan diukur besar "margin" yang dihasilkan oleh *hyperplane* dari kedua kelas tersebut. Setiap *hyperplane* dicirikan oleh arah (disimbolkan dengan w) dan posisi nya dalam ruang (disimbolkan dengan w_0). Tiap-tiap arah (w) memilih hyperplane yang memiliki jarak sama dari titik terdekat dalam kelas ω_1 maupun ω_2. Hal tersebut ditunjukkan oleh Gambar 3.8. *Hyperplane* yang ditunjukkan oleh garis solid adalah yang dipilih dari sekumpulan *hyperplane* lainnya. Margin untuk arah "1" adalah $2z_1$ dan margin untuk arah "2" adalah $2z_2$. Tujuannya adalah mencari arah yang memberikan nilai maksimum yang mungkin. Akan tetapi, tiap *hyperplane* ditentukan dalam sebuah faktor skala. Untuk saat ini, hal tersebut diabaikan dulu. Dari subbab 3.2 jarak sebuah titik dari *hyperplane* dihitung dengan:

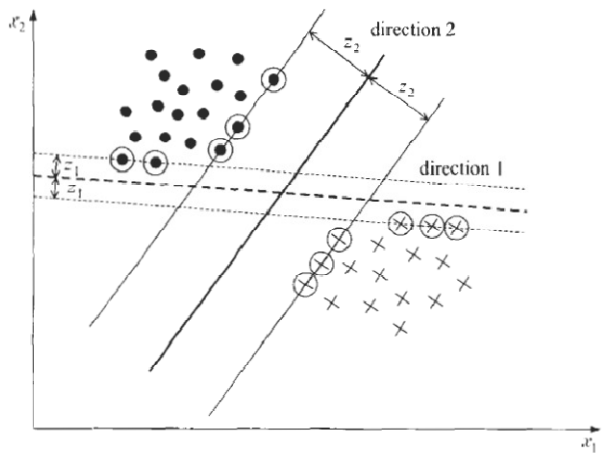
$$z = \frac{|g(x)|}{\|w\|}$$

Sekarang dapat dihitung skala w, w_0 sehingga nilai dari $g(x)$ berada pada titik terdekat dalam ω_1 dan ω_2 (dilingkari pada Gambar 3.8), yang mana bernilai 1 untuk ω_1 dan -1 untuk ω_2. Ini ekuivalen dengan:

1. Memiliki batas $\frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|}$
2. Dengan syarat

$$w^T x + w_0 \geq 1, \quad \forall x \in \omega_1$$

$$w^T x + w_0 \leq -1, \quad \forall x \in \omega_2$$



Gambar 3.8: Batas untuk arah 2 lebih besar dari batas untuk arah 1

Sekarang kita telah sampai pada titik dimana langkah berikut nya dilakukan dengan proses matematis. Untuk tiap x_i, kita menunjukkan kelas yang sesuai dengan input tersebut melalui y_i (+1 untuk ω_1, -1 untuk ω_2). Sehingga langkah-langkahnya dapat disederhanakan: Hitung parameter-parameter w, w_0 dari *hyperplan* sehingga dapat:

$$\text{Meminimalkan} \qquad J(w) \equiv \frac{1}{2}\|w\|^2 \tag{3.60}$$

$$\text{Memenuhi} \qquad y_i(w^T x_i + w_0) \geq 1, \qquad i = 1,2,\dots,N \tag{3.61}$$

Jelas bahwa dengan meminimalkan norm membuat batas jadi maksimum. Ini merupakan teknik optimisasi tidak linear (kuadratik) yang memenuhi syarat ketidaksamaan linear. Kondisi Karush-Kuhn-Ticker (KKT) (Apendik C) yang memperkecil nilai (3.60) dan (3.61) harus memenuhi:

$$\frac{\delta}{\delta w}\mathcal{L}(w, w_0, \lambda) = 0 \tag{3.62}$$

$$\frac{\delta}{\delta w0}\mathcal{L}(w, w_0, \lambda) = 0 \tag{3.63}$$

$$\lambda_i \geq 0, \quad i = 1,2,\dots,N \tag{3.64}$$

$$\lambda_i\,[y_i(w^T x_i + w_0) - 1] = 0, \quad i = 1,2,\dots,N \tag{3.65}$$

dimana λ adalah vektor pengali Lagrange, λ_i, dan $\mathcal{L}(w, w_0, \lambda)$ adalah fungsi Lagrange yang didefinisikan sebagai:

$$\mathcal{L}(w, w_0, \lambda) = \frac{1}{2}w^T w - \sum_{i=1}^{N}\lambda_i[y_i(w^T x_i + w_0) - 1] \tag{3.66}$$

dengan mengkombinasikan (3.66) dengan (3.62) dan (3.63) menghasilkan

$$w = \sum_{i=1}^{N}\lambda_i y_i x_i \tag{3.67}$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0 \tag{3.68}$$

Keterangan

- Faktor pengali Lagrange bisa bernilai nol atau positif (Appendix C). Maka, vektor parameter w dari solusi optimal adalah kombinasi linear dari $N_s \leq N$ vektor yang diasosiasikan dengan $\lambda_i \neq 0$. Menjadi,

$$w = \sum_{i=1}^{N_s} \lambda_i y_i x_i \tag{3.69}$$

  Yang dikenal sebagai *support vectors* dan pengklasifikasi *hyperplane* optimum sebagai sebuah *support vector machine* (SVM). Sebagaimana telah ditunjukkan pada Appendix C, faktor pengali tidak nol Lagrange bersesuaian dengan sebuah kekangan aktif. Sehingga, karena sekelompok kekangan pada (3.65) ditujukan untuk $\lambda_i \neq 0$, *support vectors* terletak pada kedua *hyperplane*, dengan kata lain.,

$$w^T x + w_0 = \pm 1 \tag{3.70}$$

  Mereka adalah vektor-vektor pelatihan yang terdekat ke pengklasifikasi linear, dan mereka merupakan elemen kritis dari set pelatihan. Vektor-vektor utama yang sesuai dengan $\lambda_i = 0$ bisa berada diluar dari "lapisan pemisah kelas," didefinisikan sebagai daerah antara dua *hyperplane* dalam (3.70), atau mereka bisa juga berada pada salah satu dari *hyperplane* ini (kasus degenarasi, Appendix C). pengklasifikasi *hyperplane* yang dihasilkan tidak sensitif terhadap nilai dan posisi dari vektor-vektor utama, dengan ketentuan mereka tidak melintasi lapisan pemisah kelas.
- Meskipun w diberikan secara eksplisit, w_0 dapat diambil secara implisit oleh kondisi (3.65), memenuhi syarat pelengkap (dengan kata lain, $\lambda_i \neq 0$, Appendix C). Dalam praktek, w_0 dihitung sebagai sebuah nilai rerata yang diambil menggunakan semua kondisi dari tipe ini.
- *Cost function* pada (3.60) adalah syarat konveks (Appendix C), sebuah sifat yang dijamin dengan fakta bahwa matrix Hessian adalah positif terhingga[Flet 87]. Selanjutnya, kekangan ketidaksamaan terdiri dari fungsi-fungsi linear. Sebagaimana yang didiskusikan pada Appendix C, kedua kondisi ini menjamin bahwa semua nilai lokal yang minimum adalah juga global dan unik. Hal ini dapat diterima. Pengklasifikasi *hyperplane* optimal dari sebuah SVM adalah unik.

Setelah mengenal semua sifat yang sangat menarik dari *hyperplane* SVM optimal, langkah berikutnya adalah menghitung semua parameter yang digunakan. Penghitungan yang akan dilakukan tidak mudah karena harus menggunakan beberapa algoritma misalnya [Baza 97]. Langkah berikutnya akan dibahas persamaan (3.60) dan (3.61). Persamaan ini termasuk masalah pemrograman konveks, karena *cost function* adalah konveks dan sekumpulan kekangan nya adalah linear dan menentukan sekumpulan solusi yang mungkin. Seperti yang telah kita diskusikan dalam Appendix C, masalah tersebut dapat diselesaikan dengan menggunakan dualitas Lagrange dan masalah itu dapat dinyatakan secara ekivalen oleh bentuk persamaan Wolfe, dengan kata lain:

$$\text{Maksimalkan} \qquad \mathcal{L}(w, w_0, \lambda) \tag{3.71}$$

$$\text{Memenuhi} \quad w = \sum_{i=1}^{N} \lambda_i y_i x_i \tag{3.72}$$

$$\sum_{i=1}^{N} \lambda_i y_i \tag{3.73}$$

$$\lambda \geq 0 \tag{3.74}$$

Kedua syarat persamaan merupakan hasil dari menyamakan dengan nol gradient dari Lagrange berkenaan dengan w,w_0. Vektor ciri pelatihan masuk ke dalam masalah melalui kekangan persamaan dan bukan kekangan pertidaksamaan sehingga lebih mudah untuk ditangani. Dengan substitusi persamaan (3.72) dan (3.73) ke dalam (3.71) menghasilkan persamaan yang ekivalen dengan persamaan optimisasi:

$$\max_{\lambda} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i^T x_j \right) \tag{3.75}$$

$$Memenuhi \sum_{i=1}^{N} \lambda_i y_i = 0 \tag{3.76}$$

$$\lambda \geq 0 \tag{3.77}$$

Bila pengali optimal Lagrange telah dihitung, dengan memaksimalkan (3.75), *hyperplane* optimal dapat dihitung melalui (3.72), dan w_0 melalui kondisi kelambanan komplemen.

Keterangan

- Selain kekangan pada (3.75), (3.76), masih ada alasan penting yang membuat persamaan ini populer. Vektor pelatihan yang dimasukkan berpasang-pasangan, dalam bentuk *inner product*. Fakta menarik nya adalah *cost function* tidak tergantung secara eksplisit pada dimensi dari inputnya. Sifat ini membuat generalisasi menjadi lebih efektif untuk kelas yang terpisah secara tidak linear.
- Walaupun *hyperplane* optimal yang dihasilkan unik, belum tentu pengali Lagrange λ_i juga unik. Dengan kata lain, ekspansi w sebagai bagian dari *support vectors* pada (3.72) bisa tidak unik, walaupun hasil akhirnya unik.

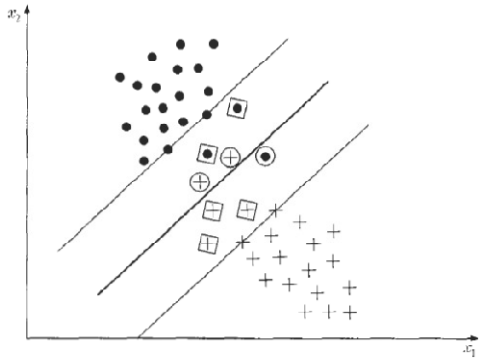
3.6.2 Nonseparable Classes

Pada kasus kelas yang tidak bisa dipisahkan, langkah-langkah sebelum nya tidak lagi berlaku. Gambar 3.9 menunjukkan kasus tersebut. Kedua kelas tersebut tidak bisa dipisahkan secara linear. Apapun usaha yang dilakukan dalam menggambar *hyperplane* agar kedua kelas tersebut terpisahkan secara linear tidak akan pernah bisa. Mengingat bahwa margin didefinisikan sebagai jarak antara pasangan *hyperplane* parallel yang dirumuskan dengan

$$w^T x + w_0 = \pm 1$$

Pelatihan vektor utama mengikuti aturan sebagai berikut:

- Vektor yang berada diluar pita dan terklasifikasi dengan tepat. Vektor ini mengikuti kekangan pada (3.61).



Gambar 3.9: Dalam kasus kelas yang tidak bisa dipisahkan, titik-titiknya Terletak di dalam pita pemisah

- Vektor yang berada didalam pita dan terklasifikasi dengan tepat. Mereka adalah titik yang ditempatkan pada kotak kecil yang ada di gambar 3.9 dan mereka memenuhi pertidaksamaan

$$0 \leq y_i(w^T x + w_0) < 1$$

- Vektor yang bukan diantara tetapan diatas. Mereka adalah titik yang diberi lingkaran dan mengikuti pertidaksamaan

$$y_i(w^T x + w_0) < 0$$

Ketiga kasus diatas dapat disederhanakan menjadi satu kekangan saja dengan menyatakan variabel baru yakni

$$y_i[w^T x + w_0] \geq 1 - \xi_i \tag{3.78}$$

Kasus yang pertama bersesuaian dengan $\xi_i = 0$, kasus kedua $0 < \xi_i \leq 1$, dan kasus ketiga $\xi_i > 1$. Variabel ξ_i dikenal juga dengan *slack variable*. Tujuan nya sekarang adalah membuat batas nya sebesar mungkin tapi tetap menjaga jumlah titik-titiknya dengan $\xi_i > 0$ sekecil mungkin. Secara matematis, sama saja dengan meminimalkan *cost function*

$$J(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} I(\xi_i) \tag{3.79}$$

dimana ξ adalah vektor dari parameter ξ_i dan

$$I(\xi_i) = \begin{cases} 1, & \xi_i > 0 \\ 0, & \xi_i = 0 \end{cases} \tag{3.80}$$

parameter C adalah konstanta positif yang mengendalikan pengaruh relatif dari kedua kelas. Namun, optimisasi susah untuk dilakukan karena melibatkan fungsi diskontinu I(.). Karena hal ini sudah umum dalam kasus tersebut, kita memilih untuk mengoptimalkan *cost function* yang dekat dengan itu, sehingga tujuan kita menjadi

$$Minimalkan \quad J(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \tag{3.81}$$

$$Memenuhi \quad y_i[w^T x_i + w_0] \geq 1 - \xi_i, \quad i = 1,2,\dots,N \tag{3.82}$$

$$\xi_i \geq 0, \quad i = 1,2,\dots,N \tag{3.83}$$

Permasalahan nya adalah bahwa ini merupakan pemrograman konveks, dan Lagrangenya adalah

$$\mathcal{L}(w, w_0, \xi, \lambda, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\lambda_i\,[y_i(w^T x_i + w_0) - 1 + \xi_i] \tag{3.84}$$

Kondisi Karush – Kuhn – Tucker yang sesuai adalah

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad atau \quad w = \sum_{i=1}^{N}\lambda_i y_i x_i \tag{3.85}$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 0 \quad atau \quad \sum_{i=1}^{N}\lambda_i y_i = 0 \tag{3.86}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \quad atau \quad C - \mu_i - \lambda_i = 0, \quad i = 1,2,\dots,N \tag{3.87}$$

$$\lambda_i[y_i(w^T x_i + w_0) - 1 + \xi_i] = 0, \quad i = 1,2,\dots,N \tag{3.88}$$

$$\mu_i\xi_i = 0, \quad i = 1,2,\dots,N \tag{3.89}$$

$$\mu_i \geq 0, \quad \lambda_i \geq 0, \quad i = 1,2,\dots,N \tag{3.90}$$

Bentuk persamaan Wolfe yang bersesuain menjadi

$$Maksimalkan \quad \mathcal{L}(w, w_0, \lambda, \xi, \mu)$$

$$Memenuhi \quad w = \sum_{i=1}^{N}\lambda_i y_i x_i$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

$$C - \mu_i - \lambda_i = 0, \quad i = 1,2,\dots,N$$

$$\lambda_i \geq 0, \qquad \mu_i \geq 0, \quad i = 1,2,\dots,N$$

Dengan mensubstitusi persamaan diatas ke dalam Lagrange menghasilkan

$$\max_{\lambda} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i^T x_j \right) \tag{3.91}$$

$$Memenuhi \quad 0 \leq \lambda_i \leq C, \qquad i = 1,2,\dots,N \tag{3.92}$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0 \tag{3.93}$$

Keterangan

- Satu-satu nya perbedaan dengan kasus kelas yang terpisah secara linear pada bagian yang sebelumnya ada pada dua kekangan pertama, dimana pengali Lagrange harus dibatasi di atas C. Untuk kasus kelas yang terpisah secara linear, nilai C → ∞. *Slack variable,* ξ_i, dan pengali Lagrange yang bersesuaian, μ_i, tidak secara eksplisit ada. Variabel-variabel tersebut secara tidak langsung dicerminkan pada C.
- Sejauh ini, kita telah membahas tentang klasifikasi dua buah kelas. Untuk kelas berjumlah M, kita bisa melihatnya sebagai permasalahan dua kelas. Untuk tiap-tiap kelas, kita mencari desain fungsi diskriminan optimal, $g_i(x),\ i = 1,2,\dots,M$, sehingga $g_i(x) > g_j(x), \forall j \neq i, jika\ x \in \omega_i$. Dengan mengadopsi metodologi SVM, kita dapat membuat fungsi diskriminan sehingga $g_i(x) = 0$ menjadi *hyperplane* optimal yang memisahkan kelas ω_i dari kelas lainnya, tentu saja dengan asumsi bahwa hal ini mungkin. Maka, fungsi linear yang dihasilkan memberikan $g_i(x) > 0$ untuk $x \in \omega_i$ dan $g_i(x) < 0$. Klasifikasi ditentukan menurut aturan berikut:

$$assign\ x\ in\ \omega_i \quad if\ i = \arg \max_{k} \{g_k(x)\}$$

Teknik ini, menghasilkan solusi yang banyak, dimana lebih dari satu $g_i(x)$ adalah positif. Pendekatan lainnya adalah dengan memperluan persamaan matematis SVM dua kelas menjadi masalah M kelas, contoh [Vapn 98].

REFERENCES

[Baza 791 Bazaraa M.S., Shetty C.M. ***Nonlinear Programming,*** John Wiley & Sons, 1979.

[Bish 951 Bishop C. ***Neural Networks for Pattern Recognition,*** Oxford University Press, 1995.

[Cid 991 Cid-Sueiro J., Anibas J.I., Urban-Munoz S., Figuieras-Vidal **A.R.** "Cost functions to estimate a-posteriori probabilities in multiclass problems," ***ZEEE Transactions onNeural Networks,*** Vol. 10(3), pp. 645456, 1999.

[Flet 871 Fletcher R. ***Practical Methods of Optimization,*** 2nd edition, John Wiley & Sons, 1987.

[Frea 921 Frean M., "A thermal perceptron learning rule," ***Neural Computation,*** Vol. **4,** pp. 946957,1992.

[Fuku 903 Fukunaga K. ***Introduction to Statistical Pattern Recognition,*** 2nd ed., Academic Press, 1990.

[Gal 901 Gallant S.I. "Perceptron based learning algorithms," ***IEEE Transactions*** on ***Neural Networks,*** **Vol.** 1(2), pp. 179-191, 1990.

[Gema 921 Geman **S.,** Bienenstock E., Doursat **R.** "Neural networks and the biaslvariance dilemma," ***Neural Computation,*** Vol. 4, pp. 1-58, 1992.

[Hayk 961 Haykin S. ***Adaptive Filter Theory,*** 3rd ed., Prentice Hall, 1996.

[Ho **651** Ho Y.H., Kashyap R.L. "An algorithm for linear inequalities and its applications," ***IEEE Transactions*** on ***Electronic Computers,*** Vol. 14(5), pp. 683-688, 1965.

[Hryc 921 Hrycej T., ***Modular learning in neural networks,*** New York: Wiley, 1992.

[Kalou 931 Kalouptsidis N., Theodoridis S. ***Adaptive System Identification and Signal Processing Algorithms,*** Prentice Hall, 1993.

[Min 881 Minsky M.L., Papert **S.A.** ***Perceptrons,*** expanded edition, MIT Press, Cambridge, MA, 1988.

[Muse 971 Muselli M. "On convergence properties of pocket algorithm," IEEE ***Transactions on Neural Networks,*** Vol. 8(3), pp. 623-629, 1997.

[Papo 911 Papoulis A. ***Probability, Random Variables and Stochastic Processes,*** 3rd ed., McGraw-Hill, 199 1.

[Pear 901 Pearlmutter B., Hampshire J. "Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function," ***Proceedings Connectionists Models Summer School,*** San Diego, CA:Morgan Kauffman, 1990.

[Poul95] Poulard H., "Barycentric correction procedure: A fast method of learning threshold units," ***Proc. WCNN '95,*** Vol. **1,** Washington, D.C., pp. 710-713, July, **1995.**

[Rich 911 Richard M.D., Lippmann R.P. "Neural network classifiers estimate Bayesian a posteriori probabilities," ***Neural Computation,*** Vol. 3, pp. 461-483, 1991.

[Robb 5 I] Robbins H., Monro ***S .*** "A stochastic approximation method," ***Annals o/*** *Mathematical Statistics,* Vol. 22, pp. 400-407, 195 1 .

[Rose 581 Rosenblatt E "The perceptron: A probabilistic model for information storage and organiution in the brain," *Psychological Review,* Vol. 65, pp. **386408,** 1958.

[Tou 741 Tou J., Gonzalez R.C. *Pattern Recognition Principles,* Addison-Wesley, 1974.

[Vapn 981 Vapnik **V.N.** *Statistical Learning Theory,* John Wiley & Sons, 1998.

[Widr 601 Widrow **B.,** Hoff M.E., **Jr.** "Adaptive switching circuits." ***IRE WESCON*** *Convention Record,* pp. 96-1 04, 1960.

[Widr 901 Widrow B., Lehr M.A. "30 years of adaptive neural networks: Perceptron, madaline. and backpropagation," *Proceedings ofthe IEEE,* Vol. 78(9), pp. **14 15-1** 442, 1990.