

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi kajian pustaka dan pembahasan tentang teori dasar yang berhubungan dengan implementasi *Fuzzy Support Vector Machine* untuk pengklasifikasian genre musik. Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan

Kajian pustaka pada penelitian ini akan membahas penelitian sebelumnya yang berjudul '*Music Genre Classification Using Sparsity-Eager Support Vector Machine*' [AKS-12]. Teori dasar yang akan dibahas pada bab ini yaitu, Genre Musik, Data Mining, Ekstraksi Fitur, konsep dasar *Support Vector Machine*, Fungsi *Kernel*, serta konsep *Fuzzy Support Vector Machine*.

1.1 Kajian Pustaka

Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya yang berjudul '*Music Genre Classification Using Sparsity-Eager Support Vector Machines*'. Penelitian ini membahas tentang penggunaan *Sparsity-Eager Support Vector Machines* untuk melakukan klasifikasi genre musik. Untuk mengekstraksi fitur audio digunakan metode *Mel-Frequency Cepstral Coefficient* [AKS-12].

Pada penelitian sebelumnya klasifikasi permasalahan *multiclass* dilakukan dengan menggunakan dengan SVM yang disusun dalam sebuah *decision tree*. Dengan menggunakan metode ini akurasi dari SVM sangat dipengaruhi dengan penyusunan *decision tree*. Metode lainnya adalah dengan memasang-masangkan sebuah kelas dengan kelas lainnya, sehingga ditemukan *decision boundary* untuk kelas tersebut. Dengan metode ini terdapat sebuah area yang tidak dapat diklasifikasikan dengan dengan *decision boundary* yang ada.

Perbedaan yang dibuat penulis pada penelitian ini adalah penggunaan algoritma *Fuzzy Support Vector Machines*. Penelitian Sebelumnya yang berjudul “*Fuzzy Support Vector Machine for Multiclass Classifications*” menjelaskan bahwa FSVM dapat mengatasi Dengan algoritma ini proses klasifikasi dilakukan dengan memasang sebuah kelas dengan kelas lainya, serta mengaplikasikan fungsi keanggotaan sesuai dengan decision boundary yang telah ada pada area yang tidak dapat diklasifikasikan. Sehingga proses klasifikasi *multiclass* dapat dilakukan tanpa harus membuat decision tree untuk masing masing kelas target. *Fuzzy Support Vector Machines* memiliki langkah yang hampir sama dengan *Support Vector Machines* standar, akan tetapi FSVM membuat sebuah nilai keanggotaan pada data dengan decision function yang telah ada.

1.2 Genre Musik

Genre musik merupakan sebuah keterangan yang paling umum dari sebuah musik. Pada data digital genre merupakan sebuah metadata yang sangat krusial untuk mendeskripsikan sebuah musik [AJF-03]. Dengan demikian musik perlu didefinisikan kedalam sebuah genre agar dapat dideskripsikan dengan baik.

Genre musik tidak dapat dipisahkan dari pengertian genre itu sendiri. Kata *Genre* berasal dari kata dalam bahasa latin yaitu *genus* yang berarti jenis atau kelas. Genre dapat diartikan sebagai sebuah pola komunikasi yang dibuat berdasarkan penggabungan antara kemampuan kognitif, sosial, dan teknis yang implisit dari kejadian komunikatif yang berulang. Sebuah genre terbentuk dengan adanya sebuah ekspektasi yang sama terhadap bentuk dan konten pada sebuah interaksi.[KOK-02]. Dari penjabaran diatas sebuah genre sangat berhubungan dengan klasifikasi. Penentuan sebuah genre pada sebuah item merupakan cara yang sangat baik untuk menjelaskan apa yang menjadi persamaan dan perbedaan sebuah item dengan item yang lainnya.

Genre musik dapat diartikan sebagai sebuah kelas spesifik dari musik, yang memiliki sebuah persamaan. Dimana persamaan tersebut dapat ditemukan melalui pendengaran manusia [KOK-02]. Dengan demikian genre dapat dipahami dengan menganalisa musik berdasarkan tempo, *timbre* atau kalimat dari lirik

musik. Berdasarkan karakter tempo, *timbre* atau lirik dari musik ini dapat dilakukan pengklasifikasian berdasarkan genre. Klasifikasi ini dapat dilakukan dengan metode Data mining

1.3 Data Mining

Data mining merupakan sebuah analisa pada dataset observational yang umumnya berjumlah besar untuk menemukan sebuah relasi yang implisit serta untuk merangkum data kedalam sebuah representasi yang dapat dimengerti dan bermanfaat oleh pemilik data [HDH-01]. Secara singkat data mining dapat merupakan sebuah proses ekstraksi data untuk menghasilkan sebuah model atau pola yang mudah dimengerti.

Data mining merupakan bagian dari *Knowledge Discovery from Data* atau KDD . Proses *knowledge discovery* terdiri dari beberapa langkah yaitu:

1. *Data cleaning* yaitu langkah untuk menghilangkan *noise* serta data yang tidak konsisten
2. *Data integration* merupakan langkah mengkombinasikan data sources jika data sources lebih dari satu
3. *Data selection* merupakan langkah pengambilan data yang relevan terhadap proses analisa data
4. *Data transformation* merupakan proses transformasi data agar sesuai dengan proses mining selanjutnya.
5. *Data mining* merupakan proses dimana proses ekstraksi pola dari data didapat
6. *Pattern evaluation* merupakan proses identifikasi pola untuk mendapatkan pola yang dapat merepresentasikan sebuah informasi (*knowledge*)
7. *Knowledge presentation* merupakan proses visualisasi informasi agar dapat mudah ditangkap oleh user.

Proses 1 hingga 4 merupakan proses preprocessing, dimana data dipersiapkan untuk *mining* [HJM-06]. Pada kasus data audio, *preprocessing* diimplementasikan sebagai proses ekstraksi fitur audio.

Data mining dapat dikelompokkan berdasarkan fungsinya. Secara umum data mining dapat dikategorikan menjadi beberapa kelompok besar yaitu [TDA-05].:

1. *Description* merupakan proses data mining dimana hasil dari proses mining berupa deskripsi dari pola dan tren yang ada pada data.
2. *Estimation* merupakan proses data mining yang meng-estimasi nilai sebuah target variable berdasarkan data yang ada.
3. *Prediction* merupakan proses data mining yang mirip dengan *estimation* tetapi target variable merupakan kondisi yang berada di masa depan.
4. *Classification* merupakan sebuah proses data mining yang mengelompokkan data input kedalam sejumlah besar data yang telah ada sebelumnya.
5. *Clustering* merupakan sebuah proses data mining yang mengelompokkan data yang telah ada kedalam sebuah kelompok yang memiliki kemiripan.
6. *Association* merupakan sebuah proses data mining yang meneliti hubungan antara atribut-atribut yang ada.

1.4 Ekstraksi Fitur Audio

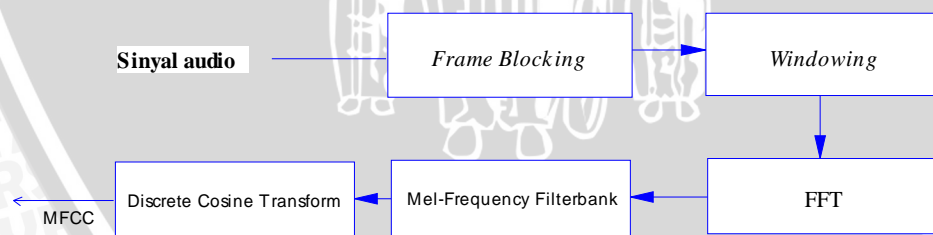
Salah satu tantangan dalam melakukan pengenalan genre adalah untuk mengetahui apa yang membedakan antara satu jenis musik dan jenis lainnya dengan menggunakan data audio. Data audio merupakan data yang tersusun atas sample-sample yang mewakili sebuah sinyal audio. Sehingga sample audio tidak dapat begitu saja dilakukan proses klasifikasi.

Untuk melakukan klasifikasi dibutuhkan sebuah proses *preprocessing* untuk memungkinkan perbandingan, dimana data yang ada harus di transformasi terlebih dahulu. *Preprocessing* untuk klasifikasi pada data sinyal audio adalah proses ekstraksi fitur audio. Ekstraksi fitur audio adalah proses komputasi untuk mendapatkan data numerik yang merepresentasikan karakteristik dari sebuah

audio segment [KOK-02]. Hasil ekstraksi fitur musik dapat berupa data yang menggambarkan nada, tempo dan lain sebagainya.

Dalam permasalahan sinyal audio ekstraksi fitur harus dapat disesuaikan dengan persepsi pendengaran manusia. Persepsi manusia terhadap frekuensi dari suara tidak mengikuti skala linear melainkan logaritmik. Oleh karena itu digunakan bentuk *Mel-Frequency Cepstral Coefficients* (MFCC) untuk merepresentasikan fitur audio sehingga sesuai dengan pendengaran manusia. MFCC didapatkan dari informasi spectrum frekuensi dari suara, tetapi dimodelkan untuk menangkap bagian-bagian relevan bagi pendengaran manusia [KOK-02].

Untuk mendapatkan MFCC sebuah sample audio harus melewati beberapa proses. Sample audio akan mengalami proses *frame blocking*, serta *windowing*. Hasil dari proses *frame blocking* dan *windowing* kemudian ditransformasi menggunakan FFT untuk mengubahnya ke dalam domain frekuensi. Hasil dari proses FFT kemudian dikalikan dengan *mel-frequency filterbank* dan kemudian dijumlahkan. Proses terakhir adalah mengubah hasil dari proses sebelumnya kembali ke domain waktu dengan menggunakan *Discrete Cosine Transform*. Hasil akhir dari proses ini adalah sejumlah koefisien yang disebut MFCC. Proses MFCC dijelaskan dengan block diagram pada Gambar 2.1



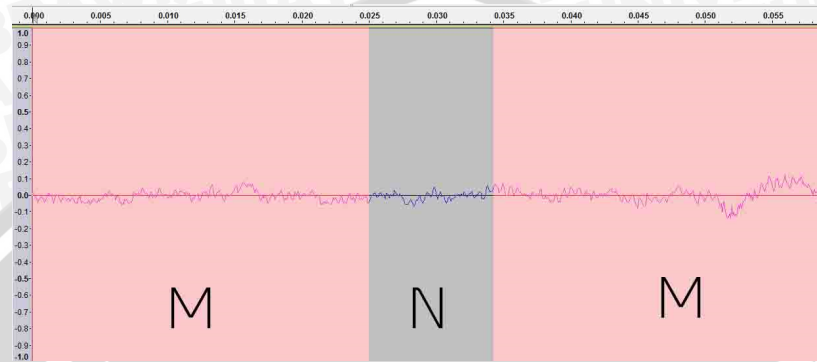
Gambar 2.1 Block Diagram Proses MFCC

Sumber: Dua, Sumeet; Sahni, Sartaj; Goyal, DP 2011

2.2.1. *Frame Blocking*

Pada Proses ini sinyal suara/musik dibagi menjadi beberapa blok. Sinyal dibagi menjadi sample sejumlah M sample dan dengan jarak antar frame

berikutnya sejumlah N [DSS-11]. Dengan demikian dari sebuah data musik dapat diperoleh sejumlah M sample seperti Gambar 2.2.



Gambar 2.2 Framing pada sebuah sinyal audio

Sumber: Audacity Music Software

2.2.2. Windowing

Windowing dilakukan untuk meminimalisir distorsi antar sample yang ada. Jenis *Window* yang umum digunakan adalah *Hamming Window*. Persamaan *Hamming Window* adalah sebagai berikut[DSS-12].

$$h(n) = 0.54 - 0.46 * \cos(2\pi n/N - 1) \quad (2.1)$$

Dimana n merupakan index dari sample pada sebuah frame dan N adalah jumlah sample pada sebuah Frame. Dengan mengkalikan sample dengan persamaan 2.1 maka sinyal akan ter filter dengan *Hamming Window*. Hasil dari windowing kemudian di transformasi kedalam domain frekuensi dengan Fast Fourier Transform

2.2.3. Transformasi Fourier

Transformasi *Fourier* merupakan sebuah transformasi yang digunakan untuk analisa pada fungsi non-periodic. Transformasi fourier memiliki pengaplikasian yang sangat luas. Akan tetapi pada saat melakukan komputasi dengan komputer, operasi integral pada Transformasi Fourier sulit dilakukan.

Oleh karena itu terdapat bentuk diskrit dari Transformasi *Fourier* yang disebut *Discrete Fourier Transform*.

Discrete Fourier Transform (DFT) merupakan proses yang dapat menganalisis data digital pada domain waktu untuk menentukan sejumlah gelombang sinus yang jika dijumlahkan akan menjadi fungsi originalnya [MMO-10]. persamaan untuk melakukan DFT ditunjukkan oleh persamaan 2.2

$$S_i(k) = \sum_{n=1}^N s_i(n) h(n) e^{-\frac{j2\pi kn}{N}} \quad 1 \leq k \leq K \quad (2.2)$$

Dimana $s_i(n)$ merupakan sinyal dengan domain waktu yang telah mengalami proses framing, $h(n)$ merupakan fungsi hamming window, dan, K merupakan panjang dari DFT [LJA-12]. Pada persamaan 2.2 $e^{-\frac{j2\pi kn}{N}}$ merupakan bilangan kompleks yang mengikuti transformasi euler seperti ditunjukkan dalam persamaan 2.3

$$e^{j\theta} = \cos \theta + j \sin \theta \quad (2.3)$$

Dengan j merupakan bilangan imajiner $\sqrt{-1}$. Nilai dari DFT akan berupa bilangan kompleks yang dapat direpresentasikan dengan menggunakan *Magnitude*. *Magnitude* dari DFT dapat direpresentasikan dengan menggunakan persamaan 2.4 [KWA-05]. Dengan $Re S_i(k)$ merupakan bagian real dari bilangan kompleks dan $Im S_i(k)$ merupakan bagian imajiner dari bilangan kompleks hasil DFT.

$$Mag S_i(k) = \sqrt{Re S_i(k)^2 + Im S_i(k)^2} \quad (2.4)$$

Fast Fourier Transdorm (FFT) adalah bentuk pengembangan dari DFT. Transformasi ini mengubah sinyal dari domain waktu ke domain frekuensi. FFT ditunjukkan pada persamaan 2.5 Dimana $W_N = e^{-j2\pi/N}$ dari persamaan 2.5 dijabarkan menjadi persamaan 2.6 dimana operasi dibagi menjadi dua, yaitu

bagian ganjil dan bagian genap. Dimana $W_N^{2rk} = W_{N/2}^{rk}$ sehingga persamaan 2.7 dapat dituliskan dengan persamaan 2.7 [MMO-10].

$$X(k) = \sum_{n=0}^{N-1} x(n)W_n^{kn} \quad (2.5)$$

$$\begin{aligned} X(k) &= \sum_{n_{\text{Genap}}} W_N^{kn} x(n) + \sum_{n_{\text{Ganjil}}} W_N^{kn} x(n) \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1)W_N^{(2r+1)k} \end{aligned} \quad (2.6)$$

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_{\frac{N}{2}}^{rk} + W_N^k \sum_{r=0}^{(\frac{N}{2})-1} x(2r+1)W_{\frac{N}{2}}^{rk} \quad (2.7)$$

Dengan menggunakan persamaan 2.7 Discrete Fourier Transform dapat dilakukan dengan lebih cepat dikarenakan proses penjumlahan masing-masing suku dilakukan dengan metode *divide and conquer*.

2.2.4. Mel-Frequency Filterbanks

Penerimaan manusia terhadap konten frekuensi pada suara tidak mengikuti skala linier. *Mel-frequency scale* dianggap lebih mengakomodir penerimaan manusia dibandingkan skala frekuensi normal yang bersifat linier. *Mel-frequency scale* bersifat linear untuk frekuensi yang berada dibawah 1000Hz dan logaritmik untuk frekuensi diatas 1000Hz. Persamaan 2.8 merupakan Fungsi untuk menghitung *mel-frequency* [DSS-11].

$$mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right) \quad (2.8)$$

Untuk mendapatkan *Mel filterbank* kita membutuhkan sebuah titik acuan yaitu frekuensi terendah dan frekuensi tertinggi. Dengan persamaan 2.8 ditemukan sejumlah n acuan untuk n filterbank yang ingin dibuat. Filterbank dibuat dengan membuat sebuah filter berbentuk triangular berdasarkan acuan yang ada. Masing-masing frekuensi yang diperoleh dari FFT dikalikan dengan masing-masing

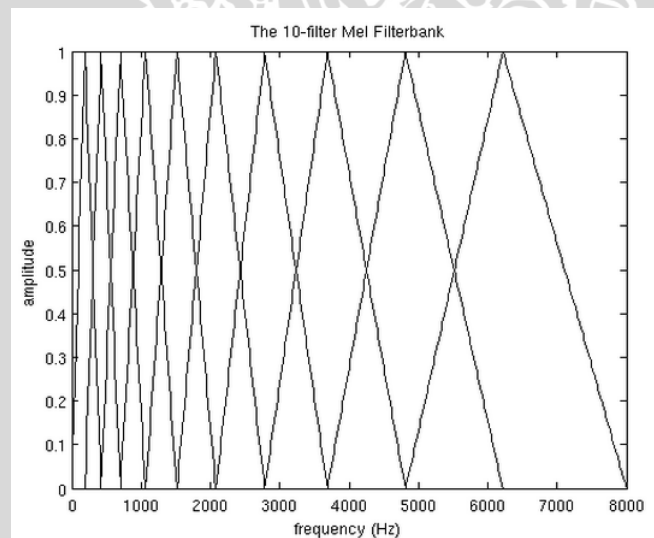
filterbank yang diperoleh serta kemudian dijumlahkan. Hasil Filterbank akan ditunjukkan pada Gambar 2.3 [DSS-11].

2.2.5. Cepstrum Coefficients

Pada tahap akhir, ekstraksi MFCC yang diperoleh dari *Mel-Filterbank* harus dikonversi kembali ke domain waktu. Dikarenakan spektrum mel merupakan bilangan real, maka untuk mengubah kembali ke domain waktu digunakan *Discrete Cosine Transform* (DCT). Persamaan 2.9 merupakan persamaan DCT.

$$c_n = \sum_{k=1}^K (\log S_k) \cos\left[n\left(k - 0.5\right)\frac{\pi}{K}\right] \quad (2.9)$$

K merupakan jumlah mel cepstrum coefficient yang ingin diekstraksi, dan n bernilai 1...K Nilai dari set ini disebut *acoustic vector* [DSS-12].



Gambar 2.3 10 Mel-filterbank dengan batas bawah 0Hz dan batas atas 8000Hz.

Sumber: Lyons, James 2012

1.5 Support Vector Machine

Support Vector Machine (SVM) adalah sebuah *classifier* linear dengan *feature set* yang telah ditentukan sebelumnya [AKS-12]. SVM memberikan hasil yang maksimal ketika jumlah data training sedikit dan tidak ada *overlap* antara kelas yang ada [AST-02]. SVM akan mencari sebuah hyperplane linier untuk

memisahkan kelas yang ada. SVM linier dapat dikembangkan menjadi SVM linier dengan *Soft margin* serta SVM untuk permasalahan nonlinear.

Pada permasalahan nonlinear SVM menggunakan sebuah pemetaan nonlinear yang disebut fungsi kernel untuk mengubah data training asli menjadi berdimensi lebih luas. Di dalam dimensi baru ini SVM mencari sebuah *hyperplane* linier optimal yang memisahkan sebuah kelas dengan kelas lainnya [HJM-06].

1.5.1 Support Vector Machine Linier

Data Training pada klasifikasi biner dapat di gambarkan pada persamaan 2.10

$$(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l), x \in \mathcal{R}^n, y \in \{+1, -1\} \quad (2.10)$$

Dimana pada kasus data dua dimensi ($x \in \mathcal{R}^2$), data yang dapat dipisahkan secara linier memiliki berbagai macam *hyperplanes* [WLI-05]. *Hyperplanes* yang dapat melakukan pemisahan ditunjukkan pada Gambar 2.4

SVM akan mencari sebuah *hyperplane* dengan margin terbesar yang disebut *maximum marginal hyperplane* (MMH). Margin yang dimiliki MMH ini akan memberikan jarak terbesar antar kelas. Jarak antara *hyperplane* dengan sebuah sisi dari margin tersebut sama dengan jarak antara *hyperplane* dengan margin pada sisi lainnya [HJM-06]. Gambar 2.5 menggambarkan sebuah MMH

Sebuah fungsi penentu atau *decision function* $d(x, w, b)$ untuk menentukan MMH dapat dinyatakan dengan persamaan 2.11

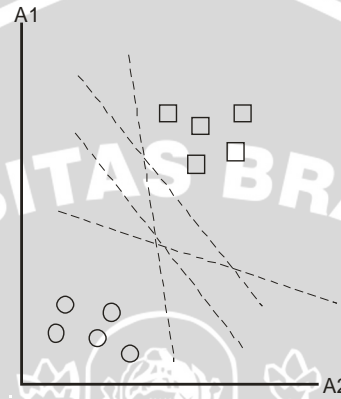
$$d(x, w, b) = W^T \cdot X + b = \sum_{i=1}^n w_i x_i + b \quad (2.11)$$

Dimana W merupakan vector bobot, dimana $W = \{w_1, w_2, \dots, w_n\}^T$, dan b adalah sebuah bilangan scalar yang menyatakan nilai bias. Dengan menggunakan persamaan 2.11 dapat diketahui *decision rule* yang menentukan hasil klasifikasi adalah:

- Jika $d(x_p, w, b) > 0$ x_p merupakan anggota class 1 ($y_p = +1$),
- Jika $d(x_p, w, b) < 0$ x_p merupakan anggota class 2 ($y_p = -1$).

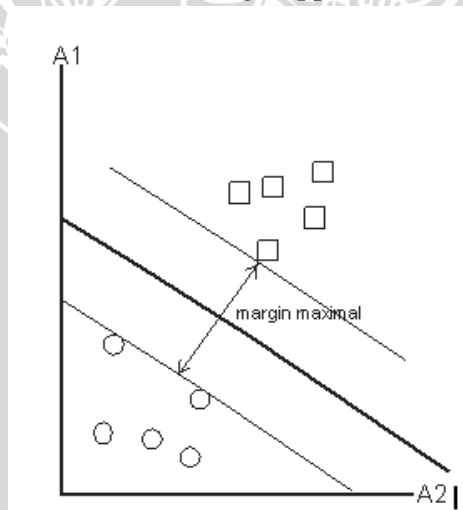
Dari decision rule dan persamaan 2.11 dapat diperoleh persamaan 2.12 [WLI-05]. Dimana X_i adalah data, y_i adalah kelas dari X_i , dan l adalah jumlah data training yang ada.

$$y_i(W^T X_i + b) \geq 1, i = 1, \dots, l \quad (2.12)$$



Gambar 2.4 Data dua dimensi yang dapat dipisahkan secara linier memiliki hyperplanes yang tidak terHINGGA.

Sumber: Wang, Lippo 2005



Gambar 2.5 SVM bertujuan menemukan *maximum marginal hyperplane*.

Sumber: Wang, Lippo 2005

Dari persamaan diatas, dapat diformulasikan besar dari margin yang maksimal. Jarak dari *hyperplane* ke sebuah titik di salah satu kelas adalah $\frac{1}{\|W\|}$, dimana $\|W\|$ merupakan *Euclidean norm* dari W yang dijabarkan pada persamaan 2.13 [HJM-06].

$$\|W\| = \sqrt{W^T \cdot W} = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2} \quad (2.13)$$

Seperti yang telah dijelaskan diatas jarak hyperplane ke sebuah kelas dan jarak hyperplane ke kelas lainnya memiliki nilai yang sama. Oleh sebab itu margin maximal dirumuskan oleh persamaan 2.14.

$$M = \frac{2}{\|W\|} \quad (2.14)$$

Dengan nilai M merupakan nilai margin yang harus dimaksimalkan. Dikarenakan minimalisasi nilai dari \sqrt{f} ekuivalen dengan minimalisasi dari f , maka minimalisasi dari $\|W\|$ sama dengan minimalisasi dari $W^T W$ dengan meminimalisasi persamaan 2.15 maka akan mengoptimalkan margin M[WLI-05].

$$\text{minimize } \frac{1}{2} W^T W \quad (2.15)$$

Minimalisasi persamaan 2.15 dilakukan dengan constrain yang ditunjukkan oleh pertidaksamaan 2.146

$$y_i [W^T x_i + b] \geq 1, \quad i = 1, \dots, l \quad (2.16)$$

Untuk memaksimalkan margin sebuah *learning machine* harus meminimalkan nilai $\|W\|^2$ sesuai dengan constrain pertidaksamaan 2.16. Permasalahan ini merupakan permasalahan optimasi quadratic dengan constrain pertidaksamaan yang dapat diselesaikan dengan mengubah persamaan 2.15 dan 2.16 menjadi fungsi Lagrange pada persamaan 2.17.

$$L(w, b, \alpha) = \frac{1}{2} W^T W - \sum_{i=1}^l \alpha_i \{y_i [W^T x_i + b] - 1\}, \quad (2.17)$$

dimana α_i non-negatif merupakan Lagrange multipliers. Fungsi ini harus diminimalkan terhadap variable w dan b dan harus dioptimalkan terhadap variable α . Untuk mendapatkan kondisi optimal persamaan 2.17 diturunkan terhadap w dan b seperti peada persamaan 2.18 dan 2.19

$$\frac{\partial L}{\partial W} = 0, \text{ atau } W = \sum_{i=1}^l \alpha_i y_i x_i \quad (2.18)$$

$$\frac{\partial L}{\partial b} = 0, \text{ atau } \sum_{i=1}^l \alpha_i y_i = 0 \quad (2.19)$$

Dengan mensubstitusikan persamaan 2.18 dan 2.19 kedalam penjabaran persamaan 2.17 diperoleh persamaan 2.20

$$L_d(\alpha) = \frac{1}{2} W^T W - \sum_{i=1}^l \alpha_i y_i W^T x_i - b \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i$$

$$L_d(\alpha) = \frac{1}{2} \sum_{i=1}^l \alpha_i y_i W^T x_i - \sum_{i=1}^l \alpha_i y_i W^T x_i - b \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i$$

$$L_d(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (2.20)$$

Pada persamaan 2.20 terlihat bahwa $x_i^T x_j$ adalah produk scalar (*dot product*) dari input. Ketergantungan $L_d(\alpha)$ pada produk scalar dari input nantinya akan sangat memudahkan saat melakukan klasifikasi dengan decision boundaries non-linier. Permasalahan optimasi kuadrat standar dapat ditunjukkan dalam notasi matrik dan diformulasikan dalam persamaan 2.21

$$L_d(\alpha) = -0.5 \alpha^T H \alpha + f^T \alpha \quad (2.21)$$

Dengan constraint

$$y^T \alpha = 0 \quad (2.22)$$

$$\alpha_i \geq 0, i = 1, \dots, l, \quad (2.23)$$

dimana $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_l]^T$, H merupakan notasi untuk *Hessian matrix* dimana $H_{ij} = y_i y_j x_i^T x_j$ dan f merupakan sebuah (1,l) vector dimana $f = [1 \ 1 \ \dots \ 1]^T$. Dengan memperhatikan persamaan 2.14 diperoleh persamaan 2.24 dan 2.25 untuk mendapatkan nilai w dan b yang optimal.

$$w_o = \sum_{i=1}^l \alpha_{oi} y_i x_i \quad (2.24)$$

$$b_o = \frac{1}{N_{sv}} \left(\sum_{s=1}^{N_{sv}} \left(\frac{1}{y_s} - x_s^T w_o \right) \right) \quad (2.25)$$

Dimana pada persamaan 2.25 hanya digunakan data yang merupakan support vector ($\alpha > 0$) [WLI-05].

Klasifikasi dimana data training masing-masing kelas benar-benar terpisah sangat jarang ditemui. Seringkali dataset memiliki data yang nilainya berada di kelas yang bukan kelas dari data tersebut. Akan tetapi solusi pemrograman kuadrat dengan batasan pada persamaan 2.15 tidak dapat dipenuhi. Pada data yang *overlap* (berada pada kelas yang tidak semestinya), nilai α akan cenderung bernilai tak-hingga.

Dalam permasalahan ini, untuk mengetahui *classifier* dengan nilai margin maximal, algoritma SVM linier diatas harus diubah untuk mengakomodasi data yang *overlap*. Oleh karena itu dibutuhkan sebuah *soft margin* dimana data didalam margin tersebut diabaikan. Lebar dari *soft margin* dapat dikontrol dengan menggunakan parameter C [WLI-05]. Sehingga $L_1(\alpha)$ harus dimaksimalkan sesuai dengan nilai nilai α_i yang lebih besar atau sama dengan nol dan dengan constraint $C \geq \alpha_i \geq 0$ dan $\sum_{i=1}^l \alpha_i y_i = 0$

1.5.2 Support Vector Machine Nonlinier

Klasifikasi secara linier sangat terbatas. Seringkali kelas-kelas tidak memiliki sebuah hyperplane tetapi memiliki pemisah berupa yang nonlinier. Kebutuhan untuk membuat sebuah SVM nonlinear ditunjukkan oleh Gambar 2.6 dimana batas pemisah yang sebenarnya adalah garis quadratic [WLI-05].

Ide dasar dari nonlinear SVM adalah dengan memetakan vector input $x \in \mathcal{R}^n$ kedalam vector $\Phi(x)$ yang memiliki *feature space* F berdimensi lebih banyak seperti pada persamaan 2.26.

$$x \in \mathcal{R}^n \rightarrow \Phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_1(x)]^T \in \mathcal{R}^f \quad (2.26)$$

Sehingga permasalahan nonlinear dapat diselesaikan dengan menggunakan klasifikasi linier pada *feature space* baru. Dengan menggantikan nilai dot pada matrix hessian, pemetaan pada data kedalam dimensi yang lebih tinggi tidak perlu dilakukan. Beberapa fungsi kernel yang sering digunakan adalah[WLI-05]:

1. Linear atau dot product yang didefinisikan dengan persamaan 2.27

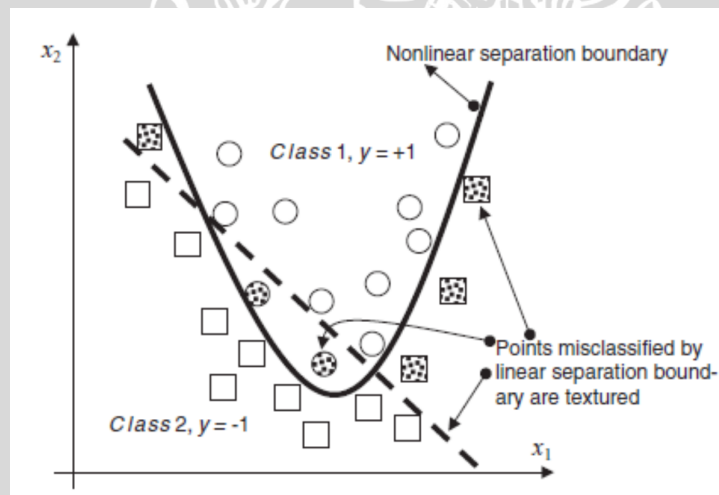
$$K(x_i, x_j) = (x_i^T x_j) \quad (2.27)$$

2. *Complete Polynomial* dengan derajat d didefinisikan dengan persamaan 2.28

$$K(x_i, x_j) = [(x_i^T x_j) + 1]^d \quad (2.28)$$

3. Gaussian RBF yang didefinisikan dengan persamaan 2.29.

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2} \quad (2.29)$$



Gambar 2.6 Sebuah permasalahan dimana *decision boundary* tidak linier, dan penyelesaian dengan batas linier menyebabkan miss-klasifikasi

Sumber: Wang, Lipo 2005.

1.6 Fuzzy Support Vector Machine

Fuzzy Support Vector Machine (FSVM) merupakan pengembangan Support Vector Machine untuk permasalahan multiclass. Dengan menggunakan *decision function* yang diperoleh dari SVM untuk sebuah pasangan kelas, untuk setiap kelas didefinisikan sebuah polyhedral pyramidal fungsi keanggotaan [AST-02].

Jika klasifikasi dilakukan secara berpasang-pasangan *decision function* untuk kelas i dan kelas j dengan margin maksimal diformulasikan pada persamaan 2.30 Akan tetapi jika dilakukan klasifikasi akan terdapat data yang tidak dapat diklasifikasikan seperti yang ditunjukkan pada Gambar 2.7

$$D_{ij}(x) = w_{ij}^t x + b_{ij} \quad (2.30)$$

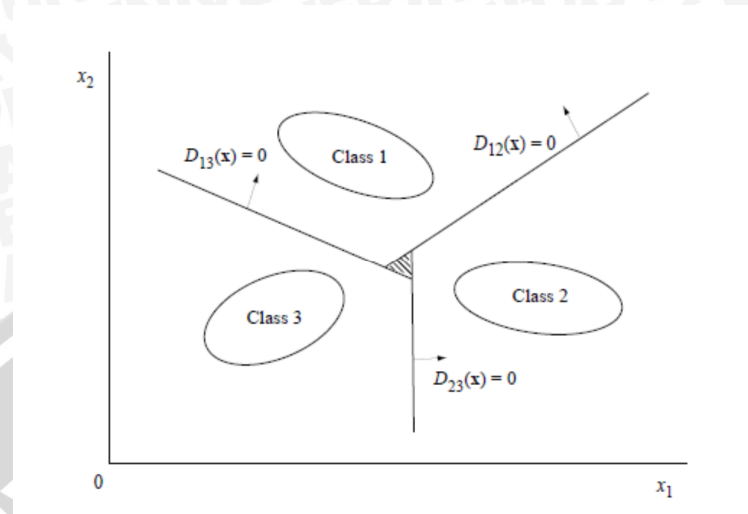
FSVM menggunakan fungsi keanggotaan untuk mengklasifikasikan daerah yang tidak dapat diklasifikasikan oleh *decision function*. Persamaan 2.31 menunjukkan fungsi keanggotaan m_{ij}

$$m_{ij} = \begin{cases} 1 & \text{untuk } D_{ij}(x) \geq 1, \\ D_{ij}(x) & \text{untuk lainnya} \end{cases} \quad (2.31)$$

Dengan menggunakan $m_{ij}(x)$, dapat didefinisikan fungsi keanggotaan x terhadap kelas i sesuai persamaan 2.32

$$\begin{aligned} m_i(x) &= \min_{j=1, \dots, n} m_{ij}(x) \\ m_i(x) &= \min (1, \min_{j \neq i, j=1, \dots, n} D_{ij}(x)) \\ m_i(x) &= \min_{j \neq i, j=1, \dots, n} D_{ij}(x) \end{aligned} \quad (2.32)$$

Sehingga data x yang diklasifikasikan berdasarkan nilai keanggotaan yang paling tinggi.



Gambar 2.7 daerah yang diarsir merupakan daerah yang tidak dapat diklasifikasikan.

Sumber: Shigeo Abe; Inoue Takuya 2002

1.7 Pengujian

Pengujian atau Evaluasi merupakan kegiatan yang membandingkan antara hasil yang diperoleh oleh implementasi dengan kriteria dan standar yang telah ditetapkan untuk melihat keberhasilannya. Dari evaluasi akan tersedia informasi mengenai sejauh mana suatu kegiatan tertentu telah dicapai. Dengan demikian kita dapat mengetahui selisih antara standar yang telah ditetapkan dengan hasil yang dicapai. Pada proses pengujian dikenal beberapa istilah yang mengidentifikasi kebenaran dari sebuah hasil klasifikasi:

- TP: *true positives* merupakan jumlah nilai positif yang diklasifikasikan sebagai positif.
- FP: *false positive* merupakan jumlah nilai negatif yang diklasifikasikan sebagai positif.
- FN: *false negative* merupakan jumlah nilai positive yang diklasifikasikan sebagai negative.
- $P= TP+FN$ merupakan total jumlah nilai positif.
- $N=FP+TN$ merupakan jumlah total nilai negative.

Dalam proses pengujian sebuah sistem klasifikasi terdapat beberapa parameter untuk mengetahui tingkat akurasi dari sebuah mesin klasifikasi [BMA-07]. Parameter-parameter itu adalah *Recall* atau *True Positive Rate*, *Precision*, *F1 Score*, serta *Accuracy*. *True Positive Rate* merupakan proporsi nilai positif yang secara tepat diklasifikasikan sebagai positif. *True Positive Rate* formulasikan dengan persamaan 2.33.

$$\text{True Positive Rate (recall)} = TP/P \quad (2.33)$$

Precision merupakan proporsi dari nilai yang diklasifikasikan positif dan benar-benar positif. *Precision* digormulasikan dengan persamaan 2.34

$$\text{Precision} = TP/(TP + P) \quad (2.34)$$

F1 Score merupakan perhitungan yang mengkombinasikan *Precision* dan *Recall*, diformulasikan pada persamaan 2.35

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.35)$$

Accuracy merupakan proporsi dari nilai yang diklasifikasikan secara tepat, diformulasikan dengan persamaan 2.37

$$\text{Accuracy} = \frac{TP + TN}{P + N} \quad (2.35)$$