

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan dibahas beberapa teori dasar yang menunjang dalam pembuatan Tesis.

2.1 Ekstraksi Fitur

Ekstraksi fitur adalah suatu proses untuk mencari transformasi atau pemetaan dari fitur-fitur original ke ruang fitur baru yang dapat memperbesar keterpisahan antar kelas (Guo, Rivero, Dorado, Munteanu, & Pazos, 2011).

Banyak peneliti menyetujui bahwa ekstraksi fitur adalah proses terpenting dan tersulit pada masalah pengenalan pola dan klasifikasi. Pemilihan fitur yang paling tepat, mungkin merupakan tugas tersulit dalam pengenalan pola (Micheli-Tzanakou, 2000). Ekstraksi fitur yang ideal akan menghasilkan sebuah representasi yang sangat memudahkan pekerjaan *classifier* (Duda, Hart, & Stork, 2000). Dalam banyak kasus, ekstraksi fitur dilakukan oleh manusia, berdasarkan pengetahuan atau pengalaman, bahkan intuisi para peneliti (Guo, Rivero, Dorado, Munteanu, & Pazos, 2011).

Adapun fitur-fitur hasil ekstraksi bisa dikatakan baik, jika berhasil memisahkan data berdasarkan kelas yang diharapkan dengan tingkat kesalahan sekecil mungkin.

Untuk menjelaskan tujuan dari ekstraksi fitur, pada tabel 2.1 ditampilkan contoh data numerik. Data tersebut terdiri dari 2 fitur original, yakni x dan y . Masing-masing baris dalam tabel digolongkan dalam 3 buah kelas, yakni A, B dan C.

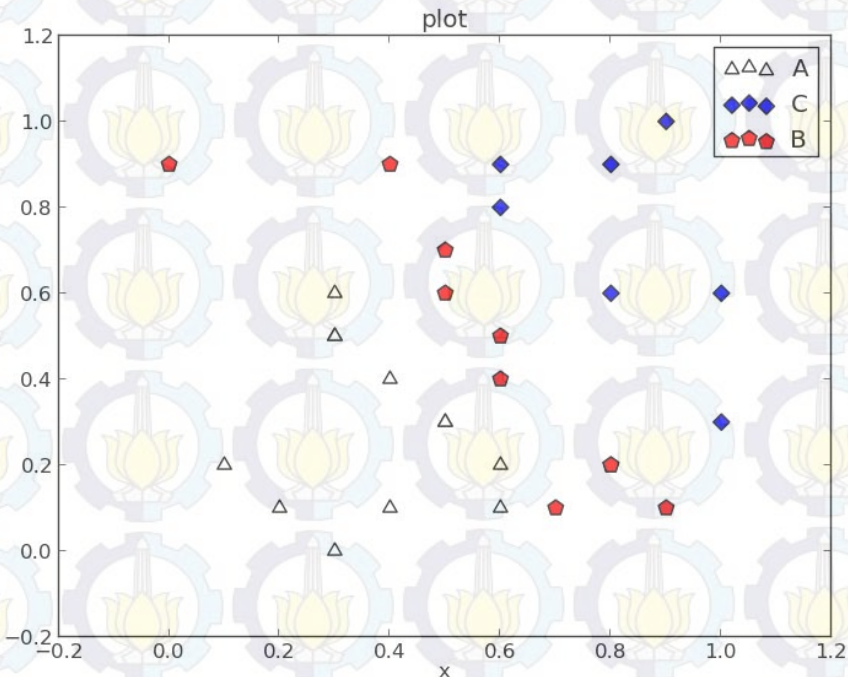
Jika data numerik pada tabel 2.1 direpresentasikan dalam bentuk grafis (ruang fitur) seperti yang disajikan pada gambar 2.1, dengan fitur x sebagai dimensi horizontal, dan y sebagai dimensi vertikal, maka akan tampak bahwa penggunaan dimensi x dan y dapat menciptakan ruang fitur yang sanggup memisahkan kelas A, B dan C.

Adapun demikian, penggunaan dimensi x dan y secara terpisah akan mengakibatkan data-data pada kelas A, B dan C saling overlap (menempati posisi

yang sama). Sebagai contoh, terdapat data dari kelas A, B dan C yang sama-sama memiliki nilai $x = 0,6$

Tabel 2.1 Contoh Data Numerik

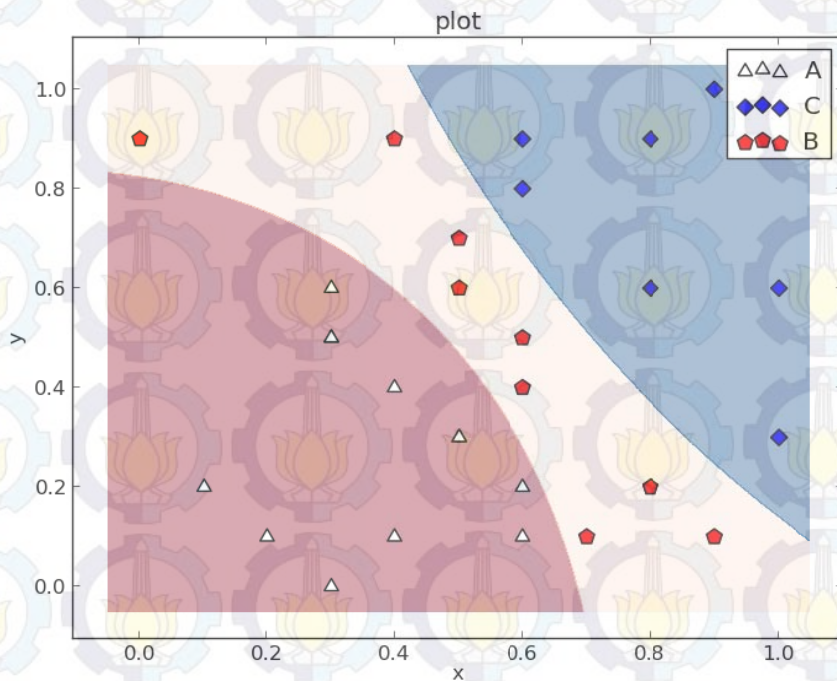
Fitur Asli		Kelas
x	y	
3	5	A
4	9	B
6	2	A
1	2	A
8	2	B
5	3	A
10	3	C
0	9	B
9	10	C
6	8	C
5	3	A
8	6	C
9	1	B
6	1	A
5	6	B
3	0	A
6	9	C
7	1	B
6	4	B
4	1	A
6	5	B
3	6	A
8	9	C
3	5	A
8	2	B
5	7	B
4	4	A
10	6	C
2	1	A



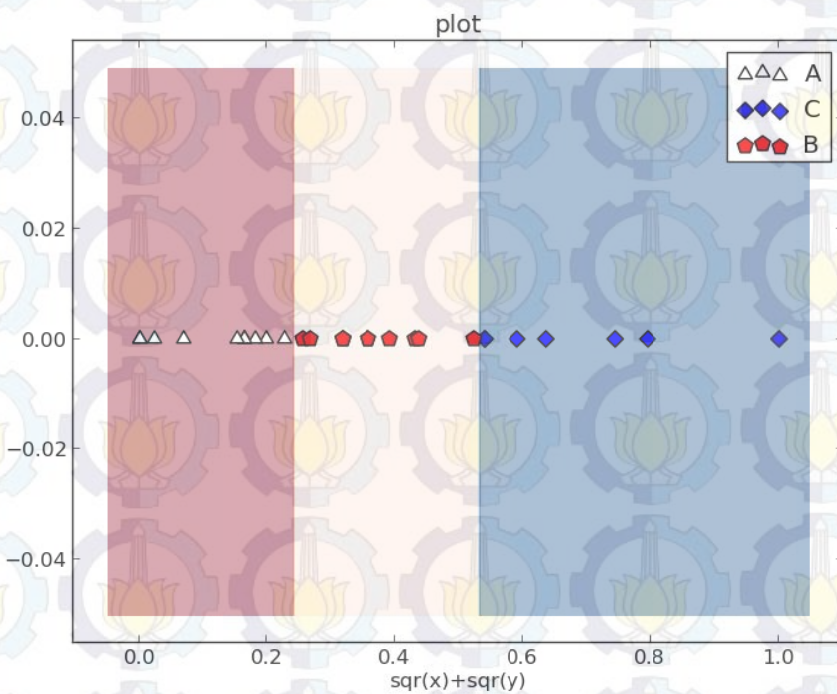
Gambar 2.1 Ruang Fitur yang Terbentuk Berdasarkan Data pada Tabel 2.1

Sekalipun ruang fitur pada gambar 2.1 telah memungkinkan data untuk dipisahkan berdasarkan kelas, namun garis pemisah antara ketiga fitur tersebut ternyata berbentuk garis non-linear yang secara matematis akan lebih kompleks dibandingkan garis linear. *Classifier* seperti SVM (Support Vector Machine) sebenarnya mampu menemukan kurva pemisah non-linear tersebut. Gambar 2.2 menunjukkan kurva pemisah yang dibentuk oleh SVM dengan kernel rbf.

Tujuan dari ekstraksi fitur adalah untuk menciptakan sesedikit mungkin fitur yang dapat memisahkan kelas-kelas secara cukup baik dan sederhana. Fitur yang dihasilkan dari proses ekstraksi dapat merupakan suatu fungsi matematika yang menggunakan subset dari fitur-fitur original sebagai komponennya, semisal x^2+y^2 . Gambar 2.3 menunjukkan bahwa penggunaan fitur x^2+y^2 ternyata telah cukup untuk memisahkan kelas A, B dan C secara sederhana (dengan menggunakan dua buah garis linear).



Gambar 2.2 Kurva yang Dibentuk oleh SVM untuk Memisahkan Ketiga Kelas



Gambar 2.3 Ruang Fitur yang Terbentuk dengan Memanfaatkan Fitur x^2+y^2

2.2 Grammatical Evolution

Grammatical Evolution adalah pengembangan dari *Genetics Programming* (yang merupakan pengembangan dari algoritma genetika), yang merupakan suatu algoritma untuk mendapatkan satu set program dalam bahasa tertentu. Dengan memanfaatkan *context-free grammar* yang ditulis dalam *Backus Naur form*, *grammatical evolution* mampu memisahkan representasi fenotip dan genotip dalam individu (Harper & Blair, 2006).

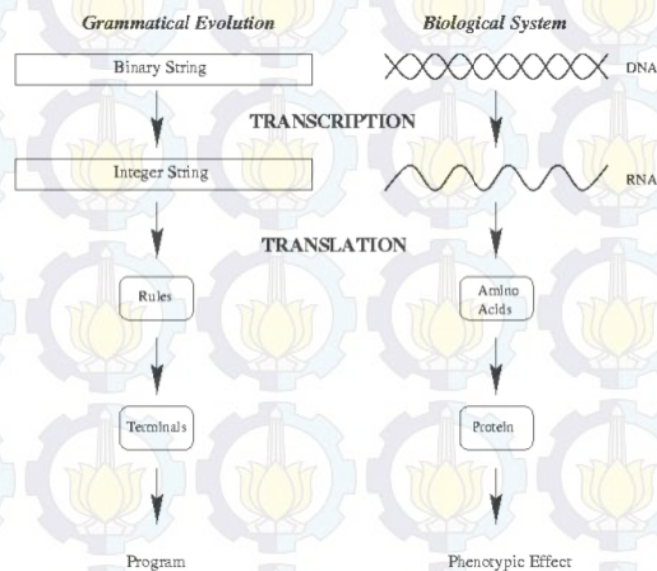
Pada *grammatical evolution*, sebuah individu memiliki dua buah representasi. Representasi yang pertama adalah representasi genotip, sedangkan representasi yang kedua adalah representasi fenotip.

Representasi genotip berupa sekumpulan angka sebagaimana layaknya pada algoritma genetika. Genotip dapat berupa angka biner maupun desimal. Representasi genotip pada *grammatical evolution* akan ditransformasikan menjadi representasi fenotip.

Representasi fenotip pada *grammatical evolution* dapat berupa fungsi matematika, kode program komputer, atau apapun, tergantung pada *grammar* yang digunakan.

Sama halnya seperti dalam algoritma genetika, pada *grammatical evolution* juga terdapat *fitness function* untuk mengukur kebaikan dari setiap individu. Untuk kasus ekstraksi fitur, umumnya tingkat akurasi *classifier* digunakan sebagai *fitness function*.

Alur umum *grammatical evolution* dan perbandingannya dengan transformasi genotip menjadi fenotip dalam biologi disajikan pada gambar 2.4.



Gambar 2.4 Alur Umum Transformasi Genotip Menjadi Fenotip pada Grammatical Evolution dan Sistem Biologi. (Conor, 2006)

2.2.1 Grammar

Untuk mentransformasikan representasi genotip menjadi representasi fenotip dibutuhkan sebuah *grammar*. *Grammar* di sini sebenarnya mirip dengan *grammar* dalam bahasa natural. Hanya saja, direpresentasikan dalam bentuk *backus naur form* (BNF). Dalam sebuah *grammar* terdapat beberapa bagian penting, antara lain:

- T : Terminal set. Merupakan node-node yang sudah tidak mungkin dievolusikan
- N : Non-terminal set. Merupakan node-node yang masih mungkin dievolusikan
- P : Production rules. Merupakan keseluruhan *grammar*
- S :Start symbol. Merupakan salah satu anggota N yang digunakan sebagai node

Semisal, didefinisikan production rules (P) seperti pada tabel 2.2, maka +, -, *, /, x, y, 1 merupakan anggota dari himpunan Terminal Set (T). Node-node yang menjadi anggota T, merupakan node-node yang sudah tidak mungkin dapat dievolusikan. Sementara itu <expr>, <op>, <var>, <num> digolongkan sebagai

Non-terminal Set (N). Node-node tersebut masih mungkin berevolusi menjadi node lain. Node $\langle \text{expr} \rangle$ berfungsi sebagai start symbol (S), artinya node $\langle \text{expr} \rangle$ merupakan node awal.

Tabel 2.2 Contoh *Grammar*

Notasi Node	Node	Aturan Produksi	Notasi Aturan
A	$\langle \text{expr} \rangle$	$\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$	A0
		$\langle \text{num} \rangle$	A1
		$\langle \text{var} \rangle$	A2
B	$\langle \text{op} \rangle$	+	B0
		-	B1
		*	B2
		/	B3
C	$\langle \text{var} \rangle$	x	C0
		y	C1
D	$\langle \text{num} \rangle$	1	D0

2.2.2 Transformasi Genotip Menjadi Fenotip

Transformasi genotip ke fenotip memanfaatkan *grammar* yang ada dan operasi modulo (sisir bagi) untuk memilih aturan transformasi.

Semisal terdapat representasi genotip 11.01.00.10.01, Genotip tersebut dapat dibagi dalam beberapa segmen sesuai kebutuhan. Dalam contoh transformasi ini, setiap segmen terdiri dari dua digit biner. Pembagian genotip dalam segmen-segmen disajikan dalam tabel 2.3

Tabel 2.3 Segmen-Segmen Genotip

Indeks Segmen	Segmen
1	11
2	01
3	00
4	10
5	01

Proses transformasi diawali dengan start symbol (dalam hal ini $\langle \text{expr} \rangle$). Selanjutnya diambil segmen dari genotip (dalam hal ini 11). Segmen tersebut

dapat pula dinyatakan dalam bilangan decimal (dalam hal ini 3). Node $\langle \text{expr} \rangle$ memiliki 3 kemungkinan perubahan ($A0 : \langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$, $A1 : \langle \text{num} \rangle$, dan $A2 : \langle \text{var} \rangle$). Untuk menentukan aturan mana yang akan digunakan, maka dilakukan operasi modulo (sisa bagi), di mana segmen genotip terpilih akan dibagi dengan jumlah kemungkinan evolusi. Karena $3 \bmod 3 = 0$, maka dipilihlah aturan $A0$, yakni $\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$. Dari langkah ini, diperoleh, $\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$ sebagai calon fenotip yang baru.

Proses transformasi dilanjutkan dengan mengambil segmen kedua dalam genotip, yaitu 01. Segmen tersebut dapat dinyatakan dalam bentuk decimal 1. Selanjutnya, diambil node non-terminal pertama dari calon fenotip yang didapat dalam langkah sebelumnya ($\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$), yakni $\langle \text{expr} \rangle$. Node tersebut memiliki 3 kemungkinan perubahan ($A0 : \langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$, $A1 : \langle \text{num} \rangle$, dan $A2 : \langle \text{var} \rangle$). Sama seperti pada langkah sebelumnya segment genotip terpilih di modulo kan dengan jumlah kemungkinan perubahan. Karena $1 \bmod 3 = 1$, maka dipilihlah aturan $A1$, yakni $\langle \text{var} \rangle$. Maka calon fenotip $\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$ berubah menjadi $\langle \text{var} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$.

Node non-terminal pertama dari calon fenotip yang baru ($\langle \text{var} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$) adalah $\langle \text{var} \rangle$, sedangkan segmen ketiga genotip adalah 00 yang bisa direpresentasikan sebagai 0 dalam basis desimal. $\langle \text{var} \rangle$ memiliki 2 kemungkinan perubahan ($C0 : x$ dan $C1 : y$). Karena $0 \bmod 2 = 0$, maka dipilih aturan $C0$, sehingga $\langle \text{var} \rangle$ berubah menjadi x . Maka kini calon fenotip berubah menjadi $x \langle \text{op} \rangle \langle \text{var} \rangle$.

Sekarang node non-terminal pertama dari calon fenotip yang baru ($x \langle \text{op} \rangle \langle \text{var} \rangle$) adalah $\langle \text{op} \rangle$. Segmen keempat genotip adalah 10, yang bisa direpresentasikan sebagai 2 dalam basis desimal. $\langle \text{op} \rangle$ memiliki 4 aturan perubahan ($B0 : +$, $B1 : -$, $B2 : *$, $B3 : /$). Karena $2 \bmod 4 = 2$, maka dipilih aturan $B1$, sehingga $\langle \text{op} \rangle$ berubah menjadi $-$. Dengan demikian calon fenotip bertransformasi menjadi $x - \langle \text{expr} \rangle$.

Proses ini dilanjutkan terus sampai seluruh node telah bertransformasi menjadi anggota terminal set (T). Seandainya, dalam proses ini, segmen genotip telah habis terpakai sebelum semua node berubah menjadi terminal node, maka akan kembali digunakan segmen pertama.

Proses transformasi secara ringkas disajikan dalam tabel 2.4. Di sini diperoleh bahwa representasi fenotip dari 11.01.00.10.01 adalah 1+Y

2.3 Decision Tree

Decision Trees (DTs) merupakan salah satu metode yang umum digunakan untuk masalah regresi dan klasifikasi. Metode ini akan menghasilkan sebuah model yang dapat digunakan untuk menebak nilai variabel target. Model yang dihasilkan *decision tree* sebenarnya merupakan aturan inferensi sederhana yang didapat dari fitur-fitur yang ada (Pedregosa, et al., 2011)

Decision tree memiliki kecenderungan overfit yang tinggi, selain itu korelasi tiap atribut terhadap keterpisahan kelas memegang peranan yang sangat penting. Karakteristik ini menyebabkan ekstraksi fitur menjadi salah satu hal yang berdampak pada akurasi *decision tree*.

Halaman ini sengaja dikosongkan