Proposal Tesis

Grammatical Evolution untuk Ekstraksi Fitur dengan Pengukuran *Multi Fitness*

Go Frendi Gunawan

NRP : 5111201033

DOSEN PEMBIMBING

Prof. Dr. Ir. Joko Lianto Buliali, Msc.

PROGRAM MAGISTER

BIDANG KEAHLIAN KOMPUTASI CERDAS VISUAL

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2012

LEMBAR PENGESAHAN PROPOSAL TESIS

Judul	:	*Grammatical Evolution* untuk Ekstraksi Fitur dengan Pengukuran *Multi Fitness*
Oleh	:	Go Frendi Gunawan
NRP	:	5111201033

Telah diseminarkan pada :

Hari	:	Rabu
Tanggal	:	17 Oktober 2012
Tempat	:	Lantai 2 Ruang Sidang Teknik Informatika ITS

Mengetahui/menyetujui:

Dosen Penguji:	Dosen Pembimbing:
1. Dr. Ir. R.V. Hari Ginardi, M.Kom	1. Prof. Dr. Ir. Joko Lianto Buliali, M.Sc
2. Bilqis Amaliah, S.Kom, M.Kom.	
3. Anny Yuniarti, S.Kom, M.Comp.Sc	

GRAMMATICAL EVOLUTION UNTUK EKSTRAKSI FITUR DENGAN PENGUKURAN MULTI FITNESS

Nama mahasiswa : Go Frendi Gunawan
NRP : 5111201033
Pembimbing : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc

ABSTRAK

Ekstraksi fitur merupakan salah satu topik yang cukup berpengaruh untuk menyelesaikan masalah klasifikasi. Sampai saat ini, tidak ada cara yang baku untuk menentukan fitur-fitur terbaik dari suatu data. Dalam proposal thesis ini, akan dicoba suatu pendekatan grammatical evolution dengan pengukuran multi fitness guna memperoleh fitur-fitur terbaik dari sebuah data.

Grammatical evolution digunakan untuk mengubah deret angka random menjadi sebuah fungsi berdasarkan *grammar* terdefinisi. Fungsi-fungsi yang terbentuk kemudian akan dijadikan fitur baru dan diukur tingkat kebaikannya untuk memisahkan satu kelas dengan kelas lain. Dalam hal ini, akan ada *n* nilai pengukuran untuk setiap fitur, di mana *n* adalah jumlah kelas

Kata Kunci: ekstraksi fitur, *grammatical evolution*, klasifikasi, multi-fitness.

GRAMMATICAL EVOLUTION FOR FEATURE EXTRACTION WITH MULTI FITNESS EVALUATION

Name : Go Frendi Gunawan
Student Identity Number: 5111201033
Supervisor : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc

ABSTRACT

Feature Extraction is a significant topic in classification problem solving. Until now, there is no such a standard way to determine the best features of a data. In this thesis proposal, grammatical evolution with multiple fitness evaluation approach will be used in order to extract best features of the data.

Grammatical Evolution is used to change random number set into a mathematic function based on defined grammar. The generated function is then become new features. The goodness of those features to separate a class with another classes is then measured. In this case, there will be *n* goodness value, with *n* is the count of the classes

Kata Kunci: feature extraction, grammatical evolution, classification, multi-fitness.

DAFTAR ISI

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Ekstraksi fitur merupakan salah satu hal yang paling berpengaruh dalam pemecahan masalah klasifikasi. Pemilihan fitur yang tidak baik akan mengakibatkan kesulitan dalam memisahkan kelas-kelas data. Kegagalan pemisahan kelas-kelas data akan berdampak pada turunnya akurasi dalam proses klasifikasi.

Dalam penelitian sebelumnya Gunawan et al, 2012, telah dicoba suatu pendekatan ekstraksi fitur dengan menggunakan *grammatical evolution*. Dalam penelitian tersebut, terdapat sebuah kelemahan dikarenakan hanya dilakukan 1 tolak ukur global untuk pengukuran *fitness value*. Hal ini mengakibatkan fitur-fitur yang sebenarnya cukup baik secara khusus, justru tersingkirkan karena nilai *fitness* globalnya rendah. Penelitian-penelitian lain seperti Gravilis et al, 2006 dan Gravilis et al, 2008 juga masih menggunakan satu nilai *fitness* saja. Guo et al, 2011 dan Li et al, 2011 menggunakan *classifier* sebagai bagian dalam pengukuran *fitneess*.

Dalam penelitian ini, akan dibuat suatu standar baru dalam penilaian *fitness*. Penilaian *fitness* tersebut akan dilakukan dengan skenario 1 vs all untuk semua kelas. Pada akhir proses, diharapkan akan ditemukan sejumlah n fitur terbaik, di mana n sama dengan jumlah kelas dalam proses klasifikasi.

1.2 Perumusan Masalah

Dalam penelitian ini, masalah-masalah yang akan diselesaikan dirumuskan sebagai berikut:

1. Bagaimana menentukan formula untuk mengukur nilai *fitness* dari sebuah fitur
2. Bagaimana membuat skenario *1 vs all* untuk semua kelas

3.Bagaimana melakukan pengujian atas fitur-fitur yang sudah di *generate*

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1.Data yang diproses adalah data numerik

2.Data yang diproses harus lengkap dan bisa dipisahkan per kelas.

1.4 Tujuan Penelitian

Menciptakan dan menguji suatu metode baru dengan menggunakan prinsip *grammatical evolution* untuk mengekstrasi fitur pada data numerik.

1.5 Manfaat Penelitian

Hasil penelitian dapat digunakan sebagai salah satu langkah pre-processing sebelum melakukan proses klasifikasi.

Dengan metode ekstraksi fitur dalam tahapan pre-processing, diharapkan proses klasifikasi data yang tidak memiliki korelasi langsung terhadap kelas dapat dilakukan dengan lebih baik.

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

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

2.1 Ekstraksi Fitur

Ekstraksi Fitur merupakan sebuah teknik untuk memilih fitur-fitur terbaik yang bisa digunakan untuk mengklasifikasikan data secara paling sederhana.

Di dalam proses ektraksi fitur, terkadang dimunculkan fitur-fitur baru berdasarkan fitur-fitur original. Proses tersebut sering pula dikenal dengan sebutan feature construction. Namun ada kalanya, hanya digunakan sebagian dari fitur-fitur original, atau biasa disebut sebagai feature selection. Dalam berbagai kasus, seringkali keduanya dilakukan bersamaan, sehingga muncul fitur-fitur baru yang berdampingan dengan beberapa fitur original.

Adapun fitur-fitur hasil ekstraksi bisa dikatakan baik, jika berhasil memisahkan data berdasarkan kelas yang diharapkan dengan tingkat kesalahan sekecil mungkin. Walau demikian, tingkat kompleksitas masing-masing fitur dan tingkat kompleksitas operasi untuk melakukan pemisahan data juga perlu dipertimbangkan. Kerumitan pada saat pembuatan fitur baru maupun pada saat operasi klasifikasi akan berpengaruh buruk pada performa proses klasifikasi itu sendiri.

Untuk menggambarkan tujuan ekstraksi fitur secara lebih jelas, maka disediakan contoh data numerik pada tabel 2.1. Jika digambarkan dalam bentuk cartesian dengan fitur x dan fitur y sebagai aksis dan ordinat, maka kelas A, B dan C tidak bisa dipisahkan secara linear. Pengklasifikasian data seperti ini dengan menggunakan neural network misalnya, akan membutuhkan banyak hidden neuron yang berujung pada peningkatan kompleksitas proses klasifikasi.

Tabel 2.1. Contoh Data numerik

Fitur original		kelas
x	y	
0	0	A
1	1	A
-1	-1	A
1	-1	A
-1	1	A
2	2	B
-2	-2	B
-2	2	B
2	-2	B
3	3	C
-3	-3	C
3	-3	C
-3	3	C

Proses ektraksi fitur diharapkan dapat menyederhanakan proses klasifikasi. Proses ektraksi fitur dapat saja menghasilkan sebuah fitur tunggal x^2 + y^2 yang sanggup memisahkan data secara linear. Fitur tersebut cukup baik, namun bukan yang terbaik, karena memiliki kompleksitas yang cukup tinggi jika dibandingkan dengan fitur-fitur original.

Salah satu alternatif yang lebih baik adalah dengan menggunakan abs(x) +abs(y). Fitur ini memiliki kompleksitas yang lebih rendah daripada fitur sebelumnya, namun tetap mampu memisahkan data secara linear.

2.2 Grammatical Evolution

Dikarenakan setiap data akan memiliki karakteristik yang berbeda-beda. Oleh sebab itu, tidak ada rumus baku untuk menghasilkan fitur baru secara umum. Cara yang cukup masuk akal untuk menghasilkan fitur-fitur semacam itu adalah dengan *trial and error*.

Algoritma genetika dan turunan-turunannya terbukti cukup baik dalam men-simulasikan trial and error yang dilakukan oleh manusia. *Grammatical evolution* merupakan salah satu turunan algoritma genetika yang memiliki context

free *grammar*, sehingga sangat cocok digunakan untuk mengekstraksi fitur dari data-data numerik.

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

Representasi genotip di sini berupa sekumpulan angka sebagaimana layaknya pada algoritma genetika. Di sini representasi genotip akan digunakan untuk mengevolusikan sebuah start symbol pada sebuah *grammar* untuk menjadi sebuah kalimat. Kalimat tersebutlah yang menjadi representasi fenotip individu.

Representasi genotip pada *grammatical evolution* tidaklah berbeda dengan representasi individu pada algoritma genetika, yakni berupa sekumpulan angka. Angka yang dimaksud bisa berupa angka biner maupun desimal. Contoh representasi genotip pada *grammatical evolution* dalam bentuk biner adalah 11001001.

Representasi genotip perlu diubah ke dalam bentuk fenotip dengan menggunakan *grammar*. Representasi fenotip pada *grammatical evolution* dapat berupa fungsi matematika ataupun sebuah program komputer, tergantung pada *grammar* yang didefinisikan. Contoh representasi fenotip yang berupa fungsi matematika adalah X+1, (X+3)*2 dan seterusnya.

Untuk proses penentuan *fitness* value, yang digunakan adalah representasi fenotip, sedangkan untuk operasi genetika, yang digunakan adalah representasi genotip.

2.2.1. Grammar Pada Grammatical Evolution

Seperti yang telah disebutkan, bahwa 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:

Tabel 2.2. Contoh Grammar

Node Notation	Node	Aturan Produksi	Notasi Aturan
(A)	<expr>	<expr><op><expr>	(A1)
		<num>	(A2)
		<var>	(A3)
(B)	<op>	+	(B1)
		-	(B2)
		*	(B3)
		/	(B4)
(C)	<var>	x	(C1)
		y	(C2)
(D)	<num>	1	(D1)

- 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 T={+, -, *, /, x, y, 1}. Node-node yang menjadi anggota T, sudah tidak mungkin dapat dievolusikan. Sedangkan yang dimaksud N adalah {<expr>, <op>, <var>, <num>}. Node-node tersebut masih mungkin berevolusi. Semisal node <op>, dapat berevolusi menjadi +, -, *, ataupun /. Sementara itu, yang menjadi start symbol (S) adalah <expr>. Jadi jika ada sebuah genotip yang akan dicari representasi fenotipnya, maka akan digunakan node <expr> sebagai node awal.

2.2.2. Transformasi Genotip ke Fenotip pada Grammatical Evolution

Seandainya kita memiliki sederetan angka representasi genotip dalam bentuk biner 11.01.00.10.01, maka proses untuk mendapatkan fenotipnya dapat digambarkan secara lengkap pada tabel 2.3.

Tabel 2.3. Proses Transformasi Genotip ke Fenotip

Before	Gene	Rule	After Transformation
<expr>	11 -> 3	<expr><op><expr>	<expr><op><expr>
<expr>	01 -> 1	<num>	<num><op><expr>
<num>	-	1	1<op><expr>
<op>	00 -> 0	+	1+<expr>
<expr>	10 -> 2	<var>	1+<var>
<var>	01 -> 1	y	1+y

Proses transformasi diawali dengan start symbol (dalam hal ini <expr>). Selanjutnya diambil sebuah segmen dari genotiip (dalam hal ini 11). Segmen tersebut dapat pula dinyatakan dalam bilangan decimal (dalam hal ini 3). Node <expr> memiliki 3 kemungkinan perubahan (A0 : <expr><op><expr>, A1:<num>, dan A2:<var>). 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 mod 3 = 0, maka dipilihlah aturan A0, yakni <expr><op><expr>. Proses ini dilanjutkan terus sampai seluruh node telah bertransformasi menjadi anggota terminal set (T).

Dalam contoh transformasi di tabel 2.3, diperoleh representasi fenotip dari 11.01.00.10.01 adalah 1+Y

BAB 3
METODE PENELITIAN

3.1 Langkah-Langkah Penelitiann

Pada penelitian ini, terdapat beberapa tahapan penyelesaian yang akan dilakukan, yang masing-masing tahapan menggunakan suatu metode tertentu. Adapun tahapan dan metode yang digunakan adalah sebagai berikut (gambar 3.1):

1.Studi literatur dan pencarian dataset

Proses ini terdiri atas pencarian referensi-referensi pendukung yang sesuai, baik dari buku, jurnal, maupun artikel. Proses tersebut dilanjutkan dengan pencarian data-data numerik yang tersedia di internet sesuai dengan batasan permasalahan.

2.Menyusun *grammar* dan rancang bangun sistem

Proses ini terdiri atas perancangan formula *grammar* dan algoritma umum dalam proses ekstraksi fitur

3.Menyusun rancangan pengujian sistem

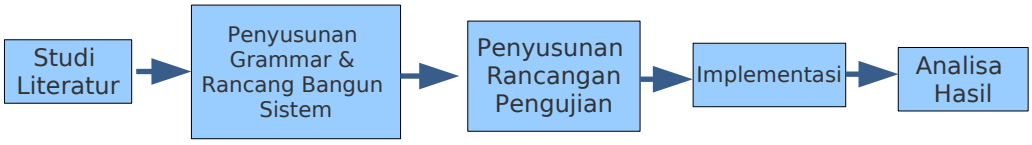
Dalam proses ini ditentukan skenario pengujian. Pengujian yang dimaksud dapat berupa pembandingan hasil klasifikasi dengan ekstraksi fitur dalam penelitian ini, ekstraksi fitur dalam penelitian sebelumnya, dan tanpa ekstraksi fitur. Performa klasifikasi (seperti kompleksitas) akan turut dihitung. Untuk proses klasifikasi sendiri akan digunakan soft SVM

4.Mengimplementasikan sistem

Sistem akan dibuat dalam bahasa pemrograman python yang umum digunakan dalam kepentingan penelitian.

5.Menganalisis hasil yang diperoleh untuk menghasilkan kesimpulan

Hasil ekstraksi fitur pada langkah nomor 4 akan diuji sesuai dengan rancangan pada langkah no 3. Selanjutnya akan disimpulkan apakah hasil penelitian lebih baik dari penelitian sebelumnya. Seandainya tidak lebih baik, maka akan dianalisa penyebab kegagalannya.



Gambar 3.1 Skema Metode Penelitian

3.2 Jadwal Kegiatan Penelitian

Pada Tabel 3.1 diperlihatkan jadwal kegiatan penelitian selama 4 bulan. Jadwal akan disajikan perminggu selama 4 bulan mulai dari November 2012 sampai dengan Februari 2013.

Tabel 3.1 Jadwal Rencana Kegiatan Penelitian

Kegiatan	Bulan															
	November 2012				Desember 2012				Januari 2013				Februari 2013			
Studi Literatur																
Analisa dan Desain Algoritma																
Pengembangan Perangkat Lunak																
Analisa Hasil Pengujian																
Penulisan Laporan																

3.3 Rancangan Sistem

Secara umum, algoritma yang akan digunakan adalah sebagai berikut:

Input dataset

1. Melakukan *grammatical evolution* untuk menciptakan fitur-fitur
2. Generate genotip
    1. Transform genotip menjadi fenotip (fitur-fitur baru), sesuai dengan aturan *grammar* yang disediakan
    2. Hitung nilai *fitness* dari setiap fenotip (fitur-fitur baru) terhadap setiap kelas.
    3. Pilih fitur-fitur terbaik

3. Membangun *feature space* berdasarkan fitur-fitur terbaik, dan melakukan proses klasifikasi.

Gambar 3.2. Skema algoritma

3.3.1. Pembuatan Fitur

Proses pembuatan fitur dilakukan dengan menggunakan *grammatical evolution*. Proses ini ditujukan untuk membuat sebanyak mungkin calon fitur yang akan dinilai tingkat *fitness* nya.

Proses ini dimulai dengan pendefinisian *grammar*. *Grammar* yang telah didefinisikan, kemudian akan digunakan untuk mentransformasi sejumlah genotip yang dihasilkan secara random menjadi sejumlah fenotip. Setiap fenotip akan dihitung nilai *fitness* nya. Selanjutnya semua fenotip akan diurutkan berdasarkan nilai *fitness*.

Untuk memastikan bahwa hasil ekstraksi fitur pasti lebih baik dari fitur original, maka dalam proses ini, fitur-fitur original pun secara elit akan ikut disertakan bersama-sama dengan fitur-fitur yang telah dihasilkan, untuk dinilai *fitness* nya.

3.3.2. Penilaian Fitness

Dalam menentukan nilai *fitness*, akan digunakan skenario *1 vs all* untuk masing-masing kelas klasifikasi. Maka, setiap fitur yang dihasilkan akan memiliki n buah nilai *fitness*, di mana n adalah jumlah kelas dalam proses klasifikasi.

Setiap nilai *fitness* dalam sebuah fitur akan menggambarkan seberapa baik fitur tersebut memisahkan sebuah kelas dengan kelas-kelas lain.

Secara umum, sebuah fitur akan dikatakan baik apabila:

- Proyeksi data terhadap fitur tidak tumpang tindih antara kelas-kelas yang berbeda

- Dalam proyeksi data terhadap fitur tidak ada kelas lain di antara kelas yang diharapkan
- Jarak intra-class dari kelas yang diharapkan cukup kecil, sedangkan jarak inter-class terhadap kelas-kelas yang tidak diharapkan cukup besar.
- Fitur relatif tidak kompleks

Dalam proses ini akan dipilih sejumlah n fitur terbaik yang dianggap paling bisa memisahkan satu kelas dengan kelas lain. Pemilihan n fitur tersebut disesuaikan dengan jumlah kelas dalam proses klasfikasi.

Semisal ada kelas A, B, C, dan D dalam proses klasifikasi, maka nantinya akan muncul maksimal 4 buah fitur. Fitur pertama memisahkan kelas A dan bukan A. Fitur kedua memisahkan kelas B dan bukan B, demikian selanjutnya

3.3.3. Pengukuran Performa Fitur

Untuk klasifikasi, akan digunakan metode *soft SVM* yang sudah umum digunakan. Dalam proses ini, fitur-fitur original akan dilakukan berbagai skenario pembandingan. Diharapkan nantinya akan diperoleh peningkatan kualitas klasifikasi yang signifikan. Dalam pengujian, akan dibandingkan hal-hal berikut:

1. Kompleksitas proses ekstraksi fitur.
2. Kompleksitas proses klasifikasi
3. Akurasi hasil klasifikasi

Proses ekstraksi fitur yang dibandingkan adalah sebagai berikut:

1. *Genetics Programming*, dengan akurasi *classifier* sebagai *fitness function*
2. *Grammatical Evolution*, dengan akurasi *classifier* sebagai *fitness function*
3. *Grammatical Evolution*, dengan separabilitas data secara global sebagai *fitness function*
4. *Grammatical Evolution*, dengan pengukuran multi-fitness sebagai *fitness function*

DAFTAR PUSTAKA

[1] Gunawan G. F., Gosaria S, Arifin A. Z. (2012). "*Grammatical Evolution For Feature Extraction In Local Thresholding Problem*", Jurnal Ilmu Komputer dan Informasi, Vol 5, No 2 (2012)

[2] Harper R., Blair A. (2006). "*Dynamically Define Functions in Grammatical Evolution*", IEEE Congress of Evolutionary Computation, July 16-21, 2006

[3] Gavrilis D., Tsoulous I. G., Georgoulas G., Glavas E. (2005). "*Classification of Fetal Heart Rate Using Grammatical Evolution*", IEEE Workshop on Signal Processing Systems Design and Implementation, 2005.

[4] Gavrilis D., Tsoulous I. G., Dermatas E. (2008). "*Selecting and Constructing Features Using Grammatical Evolution*", Journal Pattern Recognition Letters Volume 29 Issue 9, July, 2008 Pages 1358-1365 .

[5] Guo L., Rivero D., Dorado J., Munteanu C. R., Pazos A. (2011). "*Automatic feature extraction using genetic programming: An application to epileptic EEG classification* ", Expert Systems with Applications 38 Pages 10425-10436

[6] Li B., Zhang P.Y., Tian H., Mi S.S., Liu D.S., Ruo G.Q. (2011). "*A new feature extraction and selection scheme for hybrid fault diagnosis of gearbox*", Expert Systems with Applications 38 Pages 10000-10009