Grammatical Evolution Tatami untuk Ekstraksi Fitur dengan Pengukuran Multi Fitness

Go Frendi Gunawan 1st Affiliation
Departemen Informatika, Fakultas Teknologi Informasi,
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
gofrendiasgard@gmail.com

Joko Lianto Buliali
Departemen Informatika, Fakultas Teknologi Informasi,
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
joko@its-sby.edu

Abstrak

Ekstraksi Fitur adalah salah satu topik signifikan dalam pemecahan masalah klasifikasi. Hingga saat ini, belum ada cara standar untuk menemukan fitur terbaik dari sebuah data. Pada penelitian ini, sebuah metode *grammatical evolution* dengan pengukuran *multi-fitness* (dinamai sebagai GE Tatami) telah dikembangkan untuk mengekstrak fitur-fitur terbaik dari sebuah data. Metode ini menghasilkan *n*-1 fitur yang sanggup memisahkan data secara hirarkikal di mana *n* adalah jumlah kelas.

Beberapa metode telah dievaluasi dalam penelitian ini, termasuk algoritma genetika, *grammatical evolution* dengan pengukuran *fitness* global (GE Global), *grammatical evolution* dengan pengukuran multi *fitness* (GE Multi), *grammatical evolution* Tatami (GE Tatami), dan *Grammatical Evolution* Gavrilis (GE Gavrilis)

Hasil percobaan menunjukkan bahwa metode GE Tatami menghasilkan hasil yang lebih baik dibandingkan denga keempat metode lain pada data-data sintesis yang terpisah secara hirarkikal menggunakan classifier *decision tree*. Adapun metode ini menunjukkan hasil yang buruk saat gagal menemukan fitur-figur ideal, atau saat digunakan *classifier* SVM

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

I. PENDAHULUAN

Ekstraksi fitur adalah proses untuk menemukan pemetaan dari fitur-fitur asli ke dalam fitur-fitur baru yang diharapkan dapat menghasilkan keterpisahan kelas secara lebih baik [5]. Ekstraksi fitur merupakan topik penting dalam klasifikasi, karena fitur-fitur yang baik akan sanggup meningkatkan tingkat akurasi, sementara fitur-fitur yang tidak baik cenderung memperburuk tingkat akurasi.

Beberapa metode berbasis algoritma genetika telah dibuat dalam penelitian-penelitian sebelumnya guna mencari fitur-fitur terbaik. Pada [3] dan [4], telah dikembangkan metode *grammatical evolution* untuk kepentingan ekstraksi fitur. Pada penelitian tersebut, Gavrilis dkk telah membuat metode yang berhasil membuat fitur-fitur baru dengan memanfaatkan akurasi *classifier* sebagai *fitness function*. Pada [5] dan [6], metode yang hampir sama juga digunakan untuk kasus yang berbeda. Adapun dalam penelitian-penelitian tersebut, terkadang fitur-fitur yang tidak relevan juga ikut tercipta dan digunakan sebagai dalam proses klasifikasi.

Pada [1], diciptkan pendekatan baru dengan mengukur akurasi setiap fitur guna meminimalisasi kemungkinan telibatnya fitur-fitur yang tak relevan dalam proses klasifikasi. Adapun pendekatan ini cenderung menghasilkan akurasi yang buruk, karena dalam metode tersebut hanya dihasilkan 1 fitur untuk memisahkan semua kelas.

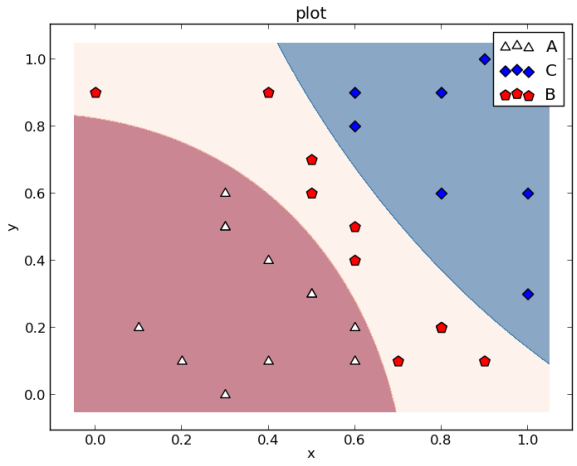
Dalam penelitian ini, diusulkan pendekatan multi *fitness* untuk memisahkan data secara hirarkikal. Pendekatan ini dinamai *Grammatical Evolution Tatami* (GE Tatami) karena dalam metode ini tampak bentuk yang menyerupai lantai tradisional Jepang (tatami). Sebagai pembanding, beberapa metode sebelumnya juga turut disertakan.

II. DATA DAN RUANG FITUR

Pada permasalahan klasifikasi, umumnya data terdiri dari sejumlah baris, kolom (fitur), dan kelas. Pada tabel 2.1, terdapat data yang terdiri dari 2 fitur (x dan y) serta 3 kelas (A, B, dan C).

Tabel 2.1 Contoh Data Numerik

Original Features		Class
x	**y**	
0.3	0.5	A
0.4	0.9	B
0.6	0.2	A
0.9	1.0	C
1.0	0.3	C
0.8	0.2	B
...	...	...



Gambar 2.1 Ruang Fitur yang Dibentuk dari Fitur Asli dan Garis Pemisah yang Dibentuk oleh SVM dengan Kernel RBF

Karena data tersebut terdiri dari 2 fitur asli, maka dimungkinkan untuk menggambarkannya dalam diagram cartesius untuk menghasilkan visualisasi yang lebih baik. Pada gambar 1.1 disajikan visualisasi dari data di tabel 1.1.

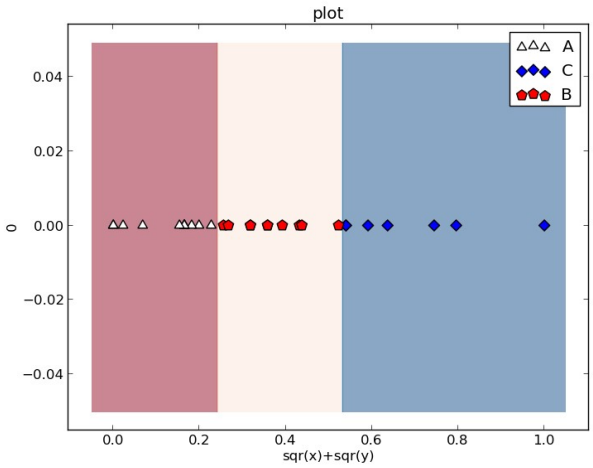
Seperti yang tampak pada gambar 2.1, mustahil untuk memisahkan data berdasarkan kelas hanya dengan menggunakan salah satu fitur asli saja (x atau y). Kedua fitur tersebut harus digunakan secara bersama-sama untuk menghasilkan ruang fitur yang memungkinkan terjadinya pemisahan data.

Classifier yang umum digunakan semisal SVM atau jaringan syaraf tiruan akan dapat memisahkan data berdasarkan kelas dengan menggunakan ruang fitur yang disajikan dalam gambar 2.1 dengan cukup baik. Adapun demikian, pemisahan data tidak bisa dilakukan secara linear. Garis-garis pemisah yang dihasilkan akan berbentuk kurva yang secara matematis lebih rumit dibandingkan sekedar garis linear.

III. EKSTRAKSI FITUR

Pada dasarnya ekstraksi fitur adalah pemetaan data dari fitur-fitur asli ke bentuk lain. Proses pemetaan tersebut mungkin membentuk dimensi yang lebih tinggi atau rendah daripada ruang fitur asli.

Pada permasalahan klasifikasi, adalah wajar untuk memetakan data yang tidak terpisahkan (atau sulit terpisahkan) ke dalam dimensi yang lebih tinggi. Dimensi yang lebih tinggi berkecenderungan untuk memberikan peluang yang lebih tinggi pula untuk memisahkan data. Adapun demikian, pemetaan ke dimensi yang lebih tinggi juga akan berimplikasi pada perhitungan yang lebih rumit.



Gambar 3.1 Ruang Fitur yang Dibentuk oleh Fitur Baru (sqr(x)+sqr(y))

Tujuan ekstraksi fitur adalah untuk membuat sesedikit fitur yang sanggup menciptakan pemisahan data dengan cara yang relatif sederhana (misalnya dengan menggunakan garis-garis linear).

Untuk lebih menjelaskan tujuan dan kegunaan dari ekstraksi fitur, data pada tabel 2.1 yang terdiri dari 2 fitur asli (x dan y) ditransformasi ke dalam sebuah ruang fitur 1 dimensi yang hanya berisi 1 fitur baru (sqr(x)+sqr(y)). Ruang fitur yang terbentuk disajikan pada gambar 3.1

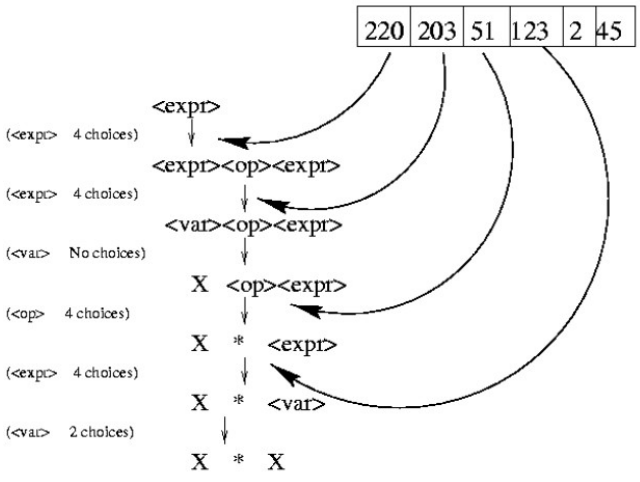
Pada hasil transformasi tersebut, dimungkinkan untuk memisahkan data dengan menggunakan dua buah garis linear yang secara matematika jauh lebih sederhana daripada sebelumnya (pada gambar 2.1).

IV. GRAMMATICAL EVOLUTION

Grammatical Evolution adalah sebuah algoritma evolusi berbasis algoritma genetik. Metode ini memanfaatkan context-free *grammar* terdefinisi untuk mengubah suatu individu menjadi bentuk apapun yang dimungkinkan oleh *grammar* tersebut.

Tabel 4.1 Contoh Grammar

Node	Production Rule	Index
<expr>	<expr><op><expr>	0
	(<expr><op><expr>)	1
	<pre-op><expr>	2
	<var>	3
<op>	+	0
	-	1
	*	2
	/	3
<pre-op>	Sin	0
	Cos	1
	Tan	2
var	x	0



Gambar 4.1 Proses Transformasi

Dalam *grammatical evolution*, setiap individu terdiri dari string biner atau integer yang disebut genotip. Dengan memanfaatkan *grammar* terdefinisi, genotip tersebut diubah menjadi fenotip. Fenotip yang terbentuk dapat berupa operasi matematika, sebuah fungsi, atau bahkan kode program komputer lengkap, tergantung dari *grammar* yang digunakan.

Sebagai contoh, didefinisikan *grammar* seperti pada tabel 4.1. Semisal genotip sebuah individu terdiri dari integer string 220, 203, 51, 123, 2, 45. Proses dimulai dari node <*expr*> sebagai start symbol. <*expr*> memiliki 4 kemungkinan aturan evolusi. Karena itu diambil segment pertama dari genotip (dalam kasus ini 220), dan dilakukan operasi modulo. Karena 220 mod 4 menghasilkan nilai nol, maka digunakanlah aturan ke nol. Hal ini mengakibatkan node <*expr*> berevolusi menjadi <*expr*><*op*><*expr*> sesuai aturan ke nol. Selanjutnya diambil segment kedua dari genotip (dalam kasus ini 203) dan node non-terminal pertama dari calon fenotip ***<expr>***<*op*><*expr*> (dalam kasus ini <*expr*>). Sekali lagi dilakukan operasi modulo. Karena <*expr*> memiliki 4 kemungkinan aturan evolusi, dan 203 mod 4 menghasilkan angka 3, maka diambillah aturan ketiga (dalam hal ini <*var*>). Sekarang calon fenotip berubah menjadi <*var*><*op*><*expr*>. Proses dilanjutkan sampai diperolah x*x sebagai fenotip individu. Gambar 4.2 menunjukkan proses transformasi secara lengkap.

Setelah mendapatkan fenotip, proses dilanjutkan seperti halnya pada algoritma genetika. Nilai *fitness* fenotip diukur dengan menggunakan *fitness function* tertentu, menghasilkan *fitness value*. *Fitness value* dari setiap individu digunakan untuk menentukan survival rate dari setiap individu, yang akan berkolerasi dengan keterpilihan individu terkait dalam generasi selanjutnya.

Untuk penyelesaian masalah ekstraksi fitur, umumnya akurasi *classifier* digunakan sebagai *fitness value*. Oleh sebab itu *fitness value* dari setiap individu akan berkisar antara 0 sampai 1.

Seperti halnya dalam algoritma genetik, pada *grammatical evolution*, akan ditemukan individu terbaik setelah beberapa generasi. Individu terbaik ini kemudian diambil sebagai solusi optimal untuk masalah yang dipecahkan.

V. GE MULTI

Semisal sebuah data terdiri dari *n* kelas, adalah cukup logis untuk berpikir bahwa dapat digunakan *n* buah fitur untuk memisahkan data berdasarkan kelasnya. Masing-masing fitur akan bertugas untuk memisahkan satu kelas tertentu dengan semua kelas lainnya.

Untuk mencapai pendekatan ini, sebuah *grammatical evolution* sederhana dengan satu nilai *fitness* untuk setiap individu tidak akan dapat bekerja dengan baik. Oleh sebab itu, dibuatlah versi modifikasi dari *grammatical evolution* yang menggunakan sejumlah nilai *fitness* untuk setiap individu. Versi termodifikasi ini diberi nama GE Multi.

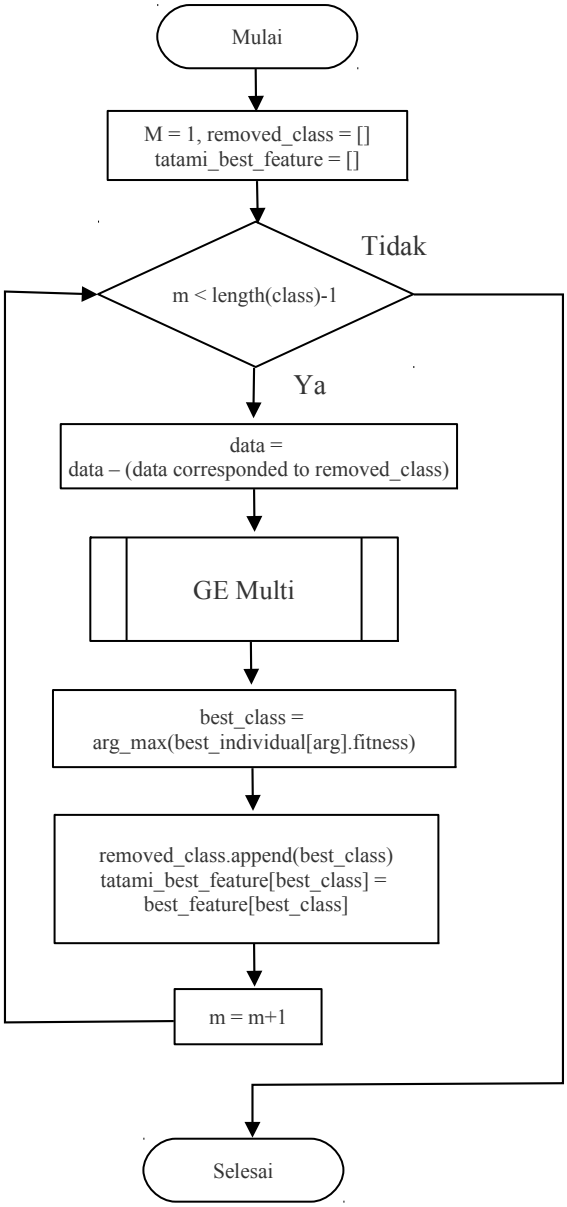
Semisal terdapat *n* kelas: $\{c_1, c_2, c_3, \ldots, c_n\}$. Dalam GE Multi, setiap individu akan memiliki *n fitness value*: $\{f_1, f_2, f_3, \ldots, f_n\}$. *Fitness value* pertama f_1 menggambarkan tingkat keberhasilan fenotip untuk memisahkan kelas c_1 dan semua kelas lainnya ($\{c_2, c_3, \ldots, c_n\}$). *Fitness value* f_2 menggambarkan tingkat keberhasilan fenotip untuk memisahkan c_2 dan semua kelas lainnya ($\{c_1, c_3, \ldots, c_n\}$). *Fitness value* ke *n*, f_n menggambarkan tingkat keberhasilan fenotip untuk memisahkan c_n dan semua kelas lainnya ($\{c_1, c_2, c_3, \ldots, c_{n-1}\}$).

Untuk menghitung nilai *fitness* dari setiap individu, data harus ditransformasi menjadi 2 kelas. Untuk mengukur nilai *fitness* pertama *f1*, maka kelas-kelas $\{c_2, c_3, \ldots, c_n\}$ harus digabungkan menjadi satu kelas (dimisalkan $c_{\sim 1}$). Kemudian *classifier* akan bertugas untuk memisahkan c_1 and $c_{\sim 1}$. Akurasi *classifier* kemudian digunakan sebagai *fitness value* f_1.

Dengan menggunakan pendekatan ini, maka akan terbentu *n* atau kurang individu-individu terbaik. Setiap individu memiliki nilai *fitness* terbaik untuk setiap kelas (akan ada individu dengan nilai f_1 terbaik, individu dengan nilait f_2 terbaik, dan seterusnya). Ada kemungkinan pula, bahwa sebuah individu memiliki beberapa nilai *fitness* terbaik. Dengan demikian, penggunaan GE Multi sebagai *feature extractor* akan menghasilkan maksimum *n* fitur baru dimana *n* adalah jumlah kelas dalam data.

VI. GE TATAMI

GE Tatami merupakan perkembangan dari GE Multi yang didesain untuk data yang terpisah secara hirarkikal. Untuk data yang terpisah secara hirarkikal, maka pemisahannya pun akan dapat dilakukan secara hirarkikal. Sebagai contoh, semisal ada *n* kelas, $\{c_1, c_2, c_3, \ldots, c_n\}$. Sekali c_1 terpisah dari kelas-kelas lainnya, maka dalam proses selanjutnya c_1 dapat diabaikan, sehingga proses pemisahan hanya akan terfokus pada $\{c_2, c_3, \ldots, c_n\}$.

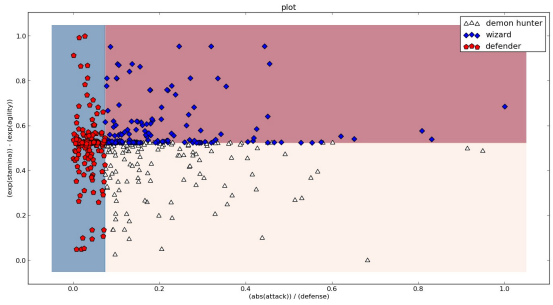


Gambar 6.1 Flowchart GE Tatami

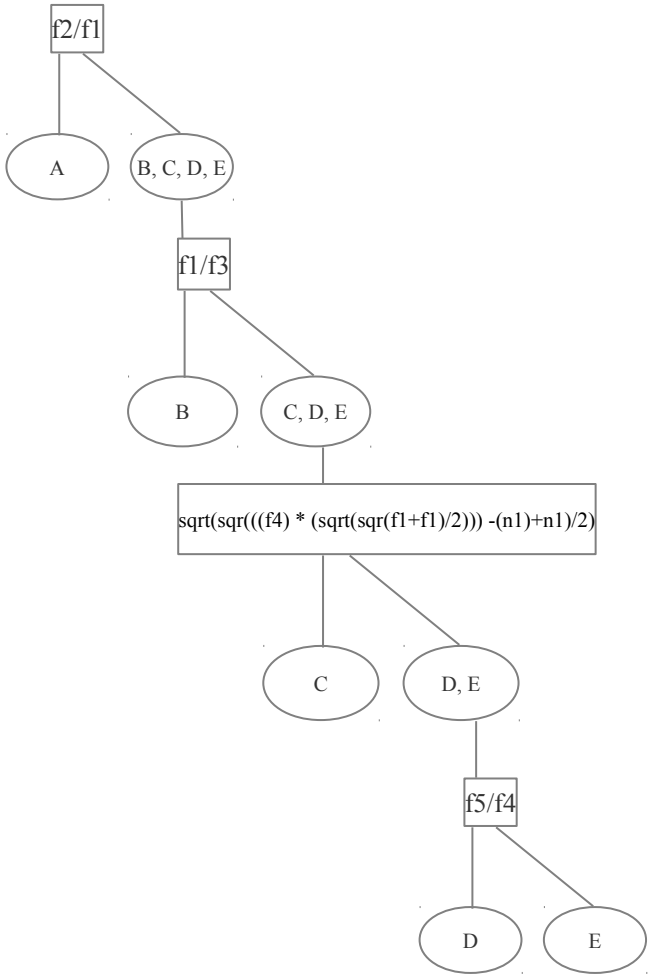
GE Tatami terdiri dari *n-1* iterasi di mana *n*=jumlah kelas. Pada setiap iterasi, GE Multi digunakan untuk menentukan individu-individu terbaik. Nilai *fitness* maksimum dari semua nilai *fitness* terbaik dari GE Multi digunakan untuk menentukan kelas mana yang paling mungkin terpisah dari semua kelas lain. Individu yang terkai dengan kelas yang paling terpisah kemudian dipilih dan dimasukkan ke dalam list fitur baru. Kelas yang sudah terpisah itu selanjutnya dihilangkan dari data. Proses iterasi selanjutnya dilanjutkan dengan kelas yang telah dihilangkan. Dalam setiap iterasi, jumlah data akan terus berkurang sesuai dengan kelas-kelas yang telah terpisah. Pada iterasi terakhir, hanya akan tersisa 2 kelas. Proses berakhir saat telah ditemukan individu terbaik untuk memisahkan kedua kelas terakhir tersebut. Gambar 6.1 menunjukkan flowchart global dari GE Tatami.

Dengan pendekatan ini, maksimum akan didapatkan *n*-1 fitur baru. Figure 6.2 menunjukkan bagaimana dua buah fitur yang dihasilkan oleh GE Tatami sanggup memisahkan tiga kelas dengan sangat baik.

GE Tatami sangat cocok sebagai langkah pre-processing pada *classifier decision tree*, karena keduanya sama-sama dirancang untuk memisahkan data secara hirarkikal. *Decision Tree* yang terbentuk dengan menggunakan fitur-fitur yang dihasilkan oleh GE Tatami ditunjukkan pada gambar 6.3.



Gambar 6.2 Ruang Fitur yang Dihasilkan GE Tatami



Gambar 6.3 Decision Tree yang Dihasilkan dengan Menggunakan Fitur-Fitur yang Dihasilkan GE Tatami

VII. HASIL DAN PEMBAHASAN

Untuk membandingkan efektifitas GE Multi dan GE Tatami dengan metode-metode sebelumnya, telah dibuat beberapa percobaan yang melibatkan sejumlah dataset. Dataset yang digunakan dalam percobaan terdiri dari 3 buah dataset sintesis yang terpisah secara hirarkikal serta 3 dataset lain yang umum diambil dari website UCI Machine Learning (iris, balanced scale, dan E-Coli).

Pada dataset sintesis 03, fitur pemisah sengaja disembunyikan dalam beberapa fitur tampak. Ini bertujuan untuk menguji kemampuan GE Tatami dalam memisahkan data yang secara eksplisit tidak tampak terpisah secara hirarkikal, namun sebenarnya terdapat cara untuk memisahkan data tersebut secara hirarkikal. Fitur-fitur pemisah dapat diperoleh kembali dengan melakukan otak-atik menggunakan operasi matematika pada fitur-fitur tampak.

Tabel 7.1 Rata-Rata Hasil Percobaan

Experiment		GA Select Feature		GE Global		GE Multi		GE Tatami		GE Gavrilis	
		Accuracy (%)	**Features**	**Accuracy (%)**	**Features**	**Accuracy (%)**	**Features**	**Accuracy (%)**	**Features**	**Accuracy (%)**	**Features**
Iris	**Train**	96.2	2	98.9	1	98.8	3	98.9	2	**99.0**	1
	Test	93.7		92.5		94.7		92.5		**97.5**	
	Total	95.7		97.6		98.0		97.6		**98.7**	
B. Scale	**Train**	71.3	3	88.6	1	**96.2**	1	93.3	2	82.7	25
	Test	69.8		79.0		**81.6**		81.5		75.4	
	Total	71.0		86.7		**93.3**		91.0		81.2	
E. Coli	**Train**	97.0	5	86.9	1	**98.2**	8	97.2	7	98.0	15
	Test	**78.2**		65.1		63.5		50.6		67.2	
	Total	**93.4**		82.7		91.5		88.2		92.0	
Syn-01	**Train**	73.9	3	78.8	1	99.8	3	**100**	2	87.9	20
	Test	71.8		42.4		**84.8**		82.2		80.7	
	Total	73.5		71.7		**96.9**		96.5		86.5	
Syn-02	**Train**	78.3	4	71.4	1	99.8	3	**100**	3	89.4	12
	Test	74.4		39.7		73.0		**79.3**		78.0	
	Total	77.5		65.2		94.5		**95.9**		87.2	
Syn-03	**Train**	72.7	5	67.8	1	99.1	5	**100**	4	81.9	19
	Test	40.5		36.9		63.0		**77.6**		50.8	
	Total	66.3		61.7		92.0		**95.5**		75.8	
AVG	**Train**	81.6	3	82.1	1	**98.6**	3	98.2	3	89.8	15
	Test	71.4		59.3		76.8		**77.3**		74.9	
	Total	79.6		77.6		**94.4**		94.1		86.9	

Untuk setiap dataset, dilakukan 5 fold cross-validation dan sebuah test lain yang melibatkan semua data sebagai training sekaligus testing. Hasil lengkap percobaan dapat diakses secara bebas pada alamat
https://github.com/goFrendiAsgard/feature-extractor/

Hasil percobaan dirangkum dengan menggunakan perhitungan rata-rata untuk setiap dataset, menghasilkan tabel 7.1. Dengan melihat pada tabel tersebut, tampak bahwa GE Tatami menunjukkan hasil yang sangat baik pada data sintesis 02 dan sintesis 03, serta hasil yang cukup baik pada dataset sintesis 01. Adapun demikian, GE Multi menunjukkan akurasi yang lebih baik pada data-data non-sintesis.

Keunggulan mutlak GE Tatami pada dataset sintesis 02 dan sintesis 03 terjadi karena dataset tersebut terpisah secara hirarkikal, dan GE Tatami berhasil menemukan fitur space ideal untuk *classifier decision tree* (seperti yang digambarkan dalam gambar 6.2 dan gambar 6.3). Di sisi lain, GE Tatami tidak menunjukkan keunggulan mutlak untuk dataset sintesis 01. Walaupun dataset sintesis 01 juga terpisah secara hirarkikal, namun data tersebut hanya memiliki 3 kelas sehingga tidak memiliki struktur hirarkikal yang cukup dalam, dan karenanya menjadi kurang signifikan dalam proses.

VIII. KESIMPULAN DAN SARAN

Hasil percobaan menunjukkan bahwa GE Tatami menunjukkan performa yang baik pada data yang terpisah secara hirarkikal dengan menggunakan *classifier decision tree*, khususnya data yang terdiri dari banyak kelas (seperti data sintesis 02 dan sintesis 03). Adapun demikian, menggunakan *classifier* yang sama, GE Multi menunjukkan hasil yang relatif baik dan seimbang untuk semua dataset.

Dari hasil tersebut, dimunculkan hipotesa bahwa menggabungkan fitur-fitur yang dibentuk oleh GE Multi dan GE Tatami mungkin akan menghasilkan akurasi yang baik untuk data dengan cakupan karakteristik yang lebih luas (tidak hanya yang terpisah secara hirarkikal)

Dalam percobaan yang dilakukan, akurasi *classifier* digunakan untuk menentukan nilai *fitness* dari setiap individu. Penggunaan mekanisme yang lebih baik dan sederhana dalam pengukuran *fitness value* mungkin akan mempercepat proses ekstraksi fitur.

UCAPAN TERIMAKASIH

Beberapa orang telah berjasa dalam membantu penulis utama untuk menyelesaikan penelitian ini. Salah satu yang paling berperan penting adalah Bapak Joko Lianto, sebagai pembimbing dalam penelitian ini. Berbagai saran yang beliau berikan telah menginspirasi penulis utama untuk menyelesaikan berbagai macam permasalahan dan menyelesaikan penelitian ini tepat waktu.

Beberapa teman, seperti Bapak Mukhlis Amien dan Bapak Hendra Suprayogi, juga turut berperan dalam memberikan saran-saran yang bersifat out-of-the-box dan terkadang menjadi sangat berguna dalam penyelesaian masalah.

REFERENCES

[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

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V. , Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P. , Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research Vol. 12 Pages 2825-2830 Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)