APLIKASI MINING DATA MAHASISWA DENGAN METODE KLASIFIKASI DECISION TREE

Sunjana
Universitas Widyatama
sunjana@widyatama.ac.id

ABSTRAKS

Data mining merupakan proses analisa data untuk menemukan suatu pola dari kumpulan data tersebut. Data mining mampu menganalisa data yang besar menjadi informasi berupa pola yang mempunyai arti bagi pendukung keputusan.

Salah satu teknik yang ada pada data mining adalah klasifikasi. Pada paper ini akan dibahas teknik klasifikasi yang diterapkan untuk menemukan pola yang terjadi pada data mata kuliah mahasiswa. Teknik klasifikasi yang akan digunakan adalah Decision tree , yaitu algoritma C4.5.

Kata kunci : *Data mining, Decision tree, C4.5*

1. PENDAHULUAN

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan. *Data mining* mampu menganalisa data yang besar menjadi informasi berupa pola yang mempunyai arti bagi pendukung keputusan.

Hasil dari aplikasi d*ata mining* tersebut dievaluasi untuk menemukan suatu informasi/pengetahuan baru yang menarik dan bernilai bagi perusahaan, dan kemudian divisualisasikan agar mempermudah bagi user memilih informasi-informasi yang mempunyai arti bagi pendukung keputusan.

Salah satu proses dalam *data mining* adalah klasifikasi, pada klasifikasi diberikan sejumlah *record* yang dinamakan *training set*, yang terdiri dari beberapa atribut, salah satu atribut menunjukkan kelas untuk *record*. Tujuan dari klasifikasi adalah untuk menemukan model dari *training set* yang membedakan *record* kedalam kategori atau kelas yang sesuai, model tersebut kemudian digunakan untuk mengklasifikasikan *record* yang kelasnya belum diketahui sebelumnya. Salah satu metode yang digunakan di dalam klasifikasi adalah pengklasifikasian dengan menggunakan *decision tre*e (pohon keputusan). Salah satu algoritma *decision tree* yang umum dipakai adalah C4.5.

Pemabahasan paper ini adalah sebagai berikut : 1.pendahuluan membahas pengertian-pengertian dasar dari topic yang diteliti, 2. Landasan teori membahas dasar-dasar teori yang digunakan dalam penelitian, 3. Implementasi membahas implementasi teori dan hasilnya, dan terkhir membahas kesimpulan.

2. LANDASAN TEORI

2.1 Pengertian Data Mining

Data mining adalah sebuah proses untuk menemukan pola atau pengetahuan yang bermanfaat secara otomatis dari sekumpulan data yang berjumlah banyak, data mining sering dianggap sebagai bagian dari *Knowledge Discovery in Database* (KDD) yaitu sebuah proses mencari pengetahuan yang bermanfaat dari data, proses KDD secara garis besar dapat dijelaskan sebagai berikut :

1. Data Selection

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. Pre-processing/ Cleaning

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Selain itu dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. Transformation

Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *Data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. Interpretation/ Evaluation

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap

ini merupakan bagian dari proses KDD yang disebut dengan *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

2.2 Decision tree

Decision tree adalah flow-chart seperti struktur tree, dimana tiap internal node menunjukkan sebuah test pada sebuah atribut, tiap cabang menunjukkan hasil dari test, dan leaf node menunjukkan class-class atau class distribution [6].

Algoritma ID3 dan Algoritma C4.5

Sebelum membahas algoritma C4.5 perlu dijelaskan terlebih dahulu algoritma ID3 karena C4.5 adalah ekstensi dari algoritma decision-tree ID3. Algoritma ID3/C4.5 ini secara rekursif membuat sebuah *decision tree* berdasarkan *training* data yang telah disiapkan. Algoritma ini mempunyai inputan berupa *training samples* dan *samples*. *Training samples* berupa data contoh yang akan digunakan untuk membangun sebuah tree yang telah diuji kebenaranya. Sedangkan *samples* merupakan field-field data yang nantinya akan kita gunakan sebagai parameter dalam melakukan klasifikasi data. Berikut adalah algoritma dasar dari ID3 dan C4.5

Algoritma ID3 [6]

Input : *Training* samples, samples
Output : *Decision tree*
Method :

(1) Create node N;
(2) **If** samples are all of the same class, C **then**
(3) Return N as a leaf node labeled with the class C;
(4) **if** atribute-list is empty **then**
(5) Return N as a leaf node labeled with the most common class in samples; // majority voting
(6) select test-atribute, atribute among atribute-list with the highest information gain;
(7) label node N with test-atribute;
(8) for each known value a_i of test-atribute // partition the samples
(9) grow a branch from node N for the condition test-atribute = a_i;
(10) let s_i be the set of samples in samples for which test-atribute = a_i; // a partition
(11) **if** s_i is empty then
(12) attach a leaf labeled with the ,most common class in samples;
(13) **else** attach the node returned by Generate_decision_tree(s_i, attribute-list-test-atribute);

Algoritma C4.5 [15]

1. Build the *decision tree* form the *training* set (conventional ID3).
2. Convert the resulting tree into an equivalent set of rules. The number of rules is equivalent to the number of possible paths from the root to a leaf node.
3. Prune (generalize) each rule by removing preconditions that increase *classification* accuracy.
4. Sort pruned rules by their accuracy, and use them in this order when classifying future test examples.

Information Gain

Information gain adalah salah satu a*tribute selection measure* yang digunakan untuk memilih test atribute tiap node pada tree. Atribut dengan information gain tertinggi dipilih sebagai test atribut dari suatu node [6]. Ada 2 kasus berbeda pada saat penghitungan Information Gain, pertama untuk kasus penghitungan atribut tanpa *missing value* dan kedua, penghitungan atribut dengan *missing value*.

Penghitungan Information Gain tanpa missing value

Misalkan S berisi *s* data samples. Anggap atribut untuk class memiliki *m* nilai yang berbeda, C_i (untuk i = 1, …,I). anggap s_i menjadi jumlah samples S pada class C_i. Maka besar information-nya dapat dihitung dengan :

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i * \log_2(p_i)$$

Dimana $p_i = \frac{si}{s}$ adalah probabilitas dari sample yang mempunyai class C_i.

Misalkan atribut A mempunyai *v* nilai yang berbeda, $\{a_1, a_2, ..., a_v\}$. Atribut A dapat digunakan untuk mempartisi S menjadi *v* subset, $\{S_1, S_2, ..., S_v\}$, dimana S_j berisi samples pada S yang mempunyai nilai a_j dari A. Jika A terpilih menjadi test atribut (yaitu, best atribut untuk splitting), maka subset-subset akan berhubungan dengan pertumbuhan node-node cabang yang berisi S. Anggap s_{ij} sebagai jumlah samples class C*i* pada subset S_j. Entropy, atau nilai information dari subset A adalah :

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_1, s_2, \ldots, s_m)$$

$\frac{s_{1j} + \ldots + s_{mj}}{s}$ adalah bobot dari subset *j*th dan jumlah samples pada subset (yang mempunyai nilai a_j dari A) dibagi dengan jumlah total samples pada S. Untuk subset S_j,

$$I(s_{1j}, s_{2j}, \ldots, s_{mj}) = -\sum_{i=1}^{m} p_{ij} * \log_2(p_{ij})$$

Dimana $p_{ij} = \frac{s_{ij}}{|s_j|}$ adalah probabilitas sample S_j yang mempunyai class C*i*.

Maka nilai information gain atribut A pada subset S adalah

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A)$$

Sebagai contoh : kita ingin mencari apakah pegolf akan masuk class play atau don't play berdasarkan data berikut:

Tabel 2.1. Contoh data play tennis

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	Don't Play
Sunny	80	90	True	Don't Play
Overcast	83	78	False	Play
Rain	70	96	False	Play
Rain	68	80	False	Play
Rain	65	70	True	Don't Play
Overcast	64	65	True	Play
Sunny	72	95	False	Don't Play
Sunny	69	70	False	Play
Rain	75	80	False	Play
Sunny	75	70	True	Play
Overcast	72	90	True	Play
Overcast	81	75	False	Play
Rain	71	80	True	Don't Play

Dari data-data pada tabel kita akan mencoba untuk membangun sebuah classifier yang berdasarkan atribut Outlook, Temperature, Humidity dan Windy. Disana ada dua kelas yaitu Play dan Don't play. Dan ada 14 examples, 5 examples menyatakan Don't Play dan 9 examples menyatakan Play. Maka,

$$I(s_1,s_2) = I(9,5) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0{,}940$$

Entropy untuk atribut Outlook adalah :

$$E(\text{Outlook}) = \frac{5}{14} * I(2,3) + \frac{4}{14} * I(4,0) + \frac{5}{14} * I(3,2)$$

$$= 0{,}694$$

Dengan nilai Gain(Outlook) yaitu:

$$Gain(\text{Outlook}) = I(s_1, s_2) - E(\text{Outlook})$$
$$= 0{,}94 - 0{,}694$$
$$= 0{,}246$$

dengan menggunakan cara yang sama, Gain dari semua atribut dapat dicari.

Gain (Outlook) = 0,246
Gain (Humidity) = 0,151
Gain (Windy) = 0,048
Gain (Temperature) = 0,029

Setelah nilai information gain pada semua atribut dihitung, maka atribut yang mempunyai nilai information gain terbesar yang dipilih menjadi test atribut.

Penghitungan Information Gain dengan missing value

Untuk atribut dengan *missing value* penghitungan information gain-nya diselesaikan dengan Gain Ratio. Sebelum menghitung gain ratio terlebih dahulu dihitung $I(s_1, s_2, \ldots, s_m)$ dan E(A).

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i * \log_2(p_i)$$

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_1, s_2, \ldots, s_m)$$

Dimana penghitungan $I(s_1, s_2, \ldots, s_m)$ dan E(A) hanya dilakukan pada atribut yang ada nilainya.

Kemudian untuk mencari gain dari atribut A dihitung dengan rumus sebagai berikut :

Gain (A) = Prob S yang diketahui * E(A)

Dimana,

A = atribut dengan missing value yang sedang dicari nilai gain-nya,

S = jumlah samples pada subset A yang diketahui nilainya.

Sedangkan nilai split pada atribut A dinyatakan dengan :

$$Split(A) = -u * \log_2 u - \sum_{i=1}^{m} p_j * \log_2(p_j)$$

Dimana,

u adalah prob samples pada atribut A yang merupakan *missing values*.

$p_j = \frac{s_j}{|s|}$ adalah probabilitas sample S_j yang diketahui nilainya.

Nilai Gain Ratio pada atribut A :

Gain Ratio(A) = Gain(A) / Split(A)

Sebagai contoh : kita ingin mencari apakah pegolf akan masuk class play atau don't play berdasarkan data berikut:

Tabel 2.2. data play tennis dengan missing value

Outlook	Temp	Humidity	Windy	Class
sunny	75	70	yes	play
sunny	80	90	yes	don't play
sunny	85	85	no	don't play
sunny	72	95	no	don't play
sunny	69	70	no	play
?	72	90	yes	play
cloudy	83	78	no	play
cloudy	64	65	yes	play
cloudy	81	75	no	play
rain	71	80	yes	don't play
rain	65	70	yes	don't play
rain	75	80	no	play
rain	68	80	no	play
rain	70	96	no	play

Pertama, kita menghitung frekuensi pada OUTLOOK sebagai berikut

	Play	Don't play	Total
Sunny	2	3	5
Cloudy	3	0	3
Rain	3	2	5
Total	8	5	13

Untuk data pada tabel 2.2 maka penghitungan information gainnya adalah sebagai berikut :

$$I(s_1,s_2) = I(8,5) = -\frac{8}{13}\log_2\frac{8}{13} - \frac{5}{13}\log_2\frac{5}{13} = 0.961$$

$$I(\text{outlook}) = \frac{5}{13} * (-\frac{2}{5} * \log_2\frac{2}{5} - \frac{3}{5} * \log_2\frac{3}{5}) + \frac{3}{13} * (-\frac{3}{3} * \log_2\frac{3}{3} - \frac{0}{3} * \log_2\frac{0}{3}) + \frac{5}{13} * (-\frac{3}{5} * \log_2\frac{3}{5} - \frac{2}{5} * \log_2\frac{2}{5}) = 0.747$$

$$\text{Gain (Outlook)} = \frac{13}{14} * (0.961-0.747) = 0.199$$

$$\text{Split} = -\frac{5}{14} * \log_2\frac{5}{14} - \frac{3}{14} * \log_2\frac{3}{14} - \frac{5}{14} * \log_2\frac{5}{14} - \frac{1}{14} * \log_2\frac{1}{14} = 1.809$$

$$\text{Gain ratio} = \frac{0,199}{1,809} = 0.110$$

Setelah semua Gain diketahui maka dapat ditentukan atribut mana yang layak menjadi root. Atribut yang layak adalah atribut yang mempunyai Gain terbesar

Penanganan Atribut Kontinyu

Algoritma C4.5 juga menangani masalah atribut kontinyu. Salah satu cara adalah dengan *Entropy-Based Discretization* yang melibatkan penghitungan class entropy.

Misalkan T membagi S example menjadi subset S1 dan S2. Umpakan ada *k* class C1, C2, …,Ck. Misal P(C*i*, S*j*) menjadi perbandingan dari example pada S*j* yang mempunyai class *i*.
Maka class entropy dari subset S*j* didefinisikan dengan :

$$\text{Ent}(S) = -\sum_{i=1}^{k} P(Ci, S)\log(P(Ci, S))$$

Dan class information entropy E(A, T;S)

$$E(A, T;S) = \frac{|S1|}{|S|}\text{Ent}(S1) + \frac{|S2|}{|S|}\text{Ent}(S2)$$

Dimana Ent(S*j*) = class entropy dari subset S*j*

S*j* = subset dari S
C*i* = class *i*
P(C*i*, S*j*) = perbandingan instance dari S*j* yang berada pada class C*i*
E(A, TA;S) = class information entropy partisi dengan cut point T_A di A
A = atribut
|S*k*| = jumlah instance di S*k*

Cut point yang terbaik adalah yang memberikan class information entropy yang paling kecil diantara semua kandidat cut point.

Pruning Tree

Pruning tree adalah melakukan suatu kegiatan untuk mengganti suatu subtree dengan suatu leaf. Penggantian dilakukan jika error rate pada subtree lebih besar jika dibandingkan dengan single leaf.

Pada C4.5 perkiraan error untuk satu node dihitung dengan :

$$e = \left(f + \frac{z2}{2N} + z\sqrt{\frac{f}{N} - \frac{f2}{N} + \frac{z2}{4N2}} \right) \Big/ \left(1 + \frac{z2}{N} \right)$$

Jika c = 25% (default untuk C4.5) maka z = 0,69 (dari distribusi normal)
f : error pada data *training*
N : jumlah instance pada satu leaf

3. IMPLEMENTASI

Pada penerapan algoritma C4.5 disini, kita ingin melihat apakah IPK seorang mahasiswa dapat diperkirakan berdasarkan nilai beberapa mata kuliah yang dianggap paling siginifikan dalam menentukan IPK seorang mahasiswa . Nilai IPK merupakan atribut *class*, kelas IPK dibagi menjadi 4 yaitu:

1. A : Sangat baik (≥ 3)
2. B : Baik (≥ 2,5)
3. C : Kurang (≥ 2)
4. D : Sangat Kurang (< 2)

Untuk Matakuliah, penulis hanya mengambil 9 jenis matakuliah sebagai atribut, yaitu :

1. Algoritma I (Sem. I)
2. Kalkulus I (Sem. I)
3. Algoritma II (Sem. II)
4. Kalkulus II (Sem. II)
5. Matriks & Ruang Vektor I (Sem. III)
6. Statistika (Sem. III)
7. Struktur Data & Algoritma Lanjut (Sem. III)
8. Basis Data (Sem. IV)
9. Matriks & Ruang Vektor II (Sem. IV)

Matakuliah diatas merupakan matakuliah yang wajib diambil oleh setiap mahasiswa ditiap semesternya. Penulis mengambil matakuliah hanya sampai semester IV dan rata-rata 2 matakuliah, sehingga dosen dapat melihat IPK dan menentukan matakuliah apa yang harus diulang atau diambil agar IPK seorang mahasiswa dapat meningkat. Matakuliah-matakuliah yang penulis ambil merupakan matakuliah-matakuliah yang saling berhubungan satu dengan yang lainnya atau dengan

 ISSN: 1907-5022

kata lain sebagai matakuliah prasyarat, misal untuk dapat mengambil matakuliah Algoritma II seorang mahasiswa harus lulus terlebih dahulu dalam matakuliah Algoritma I. Untuk lebih jelasnya kita lihat alur berikut :

1. Algoritma I → Algoritma II → Struktur Data & Algoritma Lanjut → Basis Data
2. Kalkulus I → Kalkulus II → Matriks & Ruang Vektor I → Matriks & Ruang Vektor II
3. Kalkulus I → Statistika

Pada tabel matakuliah tersebut terdiri dari 210 *record*, dengan 160 *record* sebagai data *training* dan 50 *record* sebagai data *testing*. Hasil uji yang dilakukan adalah sebagai berikut :

Tabel 4.1 Hasil Uji

Uji Ke -	Matakuliah	Traing	Testing
		Error Rate	Error Rate
1	Algoritma I Algoritma II Basis Data Statistika Struktur Data	18,75 %	26,00 %
2	Algoritma I Algoritma II Kalkulus I Kalkulus II Matriks I Matriks II Basis Data Statistika Struktur Data	5,00 %	46,00 %
3	Algoritma I Algoritma II Kalkulus I Kalkulus II	21,25 %	48,00 %
4	Algoritma I Algoritma II Matriks I Matriks II Basis Data	15,63 %	36,00 %
5	Algoritma I Algoritma II Kalkulus I Kalkulus II Statistika	15,00 %	44,00 %
6	Algoritma I Kalkulus I Matriks I Basis Data	13,75 %	46,00 %

Uji Ke -	Matakuliah	Traing	Testing
		Error Rate	Error Rate
1	Algoritma I Algoritma II Basis Data Statistika Struktur Data	18,75 %	26,00 %
2	Algoritma I Algoritma II Kalkulus I Kalkulus II Matriks I Matriks II Basis Data Statistika Struktur Data	5,00 %	46,00 %
3	Algoritma I Algoritma II Kalkulus I Kalkulus II Statistika	21,25 %	48,00 %

Dari hasil uji diatas dapat kita lihat prosentase dari beberapa matakuliah terhadap nilai IPK seorang mahasiswa. Prosentase *error rate* yang dihasilkan pada hasil testing rata-rata adalah dibawah 50%, bahkan ada yang 26 %. Itu menandakan bahwa *rule* yang dihasilkan sudah cukup baik. Hasil tersebut diperoleh dari data *training* pada matakuliah Algoritma I, Algoritma II, Basis Data, Statistika, dan Struktur Data. Dari data *training* pada matakuliah tersebut, ternyata menghasilkan *rule* yang digunakan untuk data *testing* dengan prosentase *error rate* yang sangat kecil. Semakin besar prosentase nilai *error rate* yang dihasilkan pada data *testing*, maka *rule* yang dihasilkan pun tidak baik. Begitu pula sebaliknya, semakin kecil prosentase *error rate* yang dihasilkan pada data *testing*, maka akan menghasilkan *rule* yang baik pula.

4. KESIMPULAN

Berdasarkan training dan pengujian kemudian dilakukan analisis maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penentuan data *training* sangat menentukan tingkat akurasi *tree* yang dibuat.
2. Besar prosentase kebenaran *tree* sangat dipengaruhi oleh data *training* yang digunakan untuk membangun model *tree* tersebut.
3. Nilai IPK seorang mahasiswa terlihat sangat terpengaruh dengan 9(Sembilan) mata kuliah yang dianggap pokok.

PUSTAKA

[1] Andrew W. Moore, "*Decision Trees*", Carnegie Mellon University. (http://www.cs.cmu.edu/~awm)

[2] Berry Linoff, *Mastering Data Mining*, John Wiley & Son, Inc, 2000

[3] Frank, Vanden Berghen, "*Classification Trees : C4.5*", Universit Libre de Bruxelles, 2003.

[4] Han,Jiawei and Khamber, Micheline. "*Data Mining : Concepts and Techniques*", Morgan Kaufmann Publishers, San Francisco,USA, 2001.

[5] Long, William, "*Classification Tress*", Harvard-MIT Division of Health Sciences and Technology.

[6] Malerba Donato, Floriana Esposito, dan Giovanni Semeraro, "*A Comparative Analysis of Methods for Pruning Decision Tress*", IEEE, Italy, 1997

[7] Quinlan, J.R. "C4.5: Programs For Machine Learning". San Mateo, CA: Morgan Kaufmann,