ISSN: 2089-1121

DATA TRANSFORMATION PADA DATA MINING

Hartarto Junaedi*), Herman Budianto), Indra Maryati**), dan Yuliana Melani**)**
*) Jurusan Sistem Informasi Bisnis
Sekolah Tinggi Teknik Surabaya
**) Jurusan Teknik Informatika
Sekolah Tinggi Teknik Surabaya
aikawa@stts.edu , herman@stts.edu , maryati@stts.edu , yulianamelani@gmail.com

ABSTRAK

Mempersiapkan data adalah tahapan *preprocessing* yang sangat penting pada data mining, alasan utamanya adalah karena kualitas dari *input* data sangat mempengaruhi kualitas *output* analisis yang dihasilkan. Ada beberapa teknik untuk *data preprocessing*, dimana salah satunya adalah *data transformation* yang mentransformasi atau mengkonsolidasi data ke bentuk yang cocok untuk *mining*. Penelitian ini membahas beberapa pendekatan yang terdapat pada data transformation dan melakukan analisis metode yang dimiliki untuk masing-masing pendekatan tersebut.

Selain itu juga dilakukan pembahasan dan analisis algoritma untuk data transformation dengan beberapa metode dan implementasinya pada *Weka*. Weka adalah tool data mining berbasis Java yang cukup populer dan banyak digunakan untuk melakukan eksperimen dan uji coba. Selain itu, ditunjukkan bagaimana cara mengintegrasikan metode baru yang dibuat sendiri ke dalam Weka melalui beberapa ketentuan.

Kata kunci : Data Mining, Data Transformation, Preprocessing, Weka

ABSTRACT

Preparing data is an important preprocessing step for data mining. The main reason is that the quality of the input data strongly influences the quality of the analysis result. There are a number of data preprocessing techniques, one of them is data transformation, which transforms or consolidates the data into forms appropriate for mining. This paper discusses the approaches of data transformation and some analysis methods to each of the approaches.

Several algorithms are obtained from Weka, which are comprehensive suite of Java class libraries that implement many-state-of-the-art machine learning and data mining algorithm. Its also shows how to integrate the new methods into Weka environment.

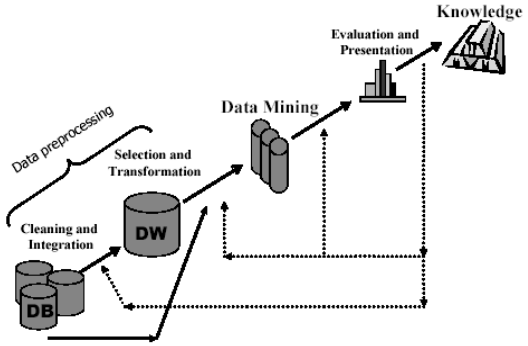
Keywords : Data Mining, Data Transformation, Preprocessing, Weka

1. DATA TRANSFORMATION

Ada 7 (tujuh) tahapan proses data mining, dimana 4 (empat) tahap pertama disebut juga dengan data preprocessing (terdiri dari *data cleaning*, *data integration*, *data selection*, dan data transformation), yang dalam implementasinya membutuhkan

ISSN: 2089-1121

waktu sekitar 60% dari keseluruhan proses. Dalam data transformation, terdapat beberapa pendekatan/teknik untuk melakukan transformasi data, yaitu smoothing, generalization, normalization, *aggregation*, dan attribute construction.



Gambar 1. Data Mining *Process*

A. Smoothing

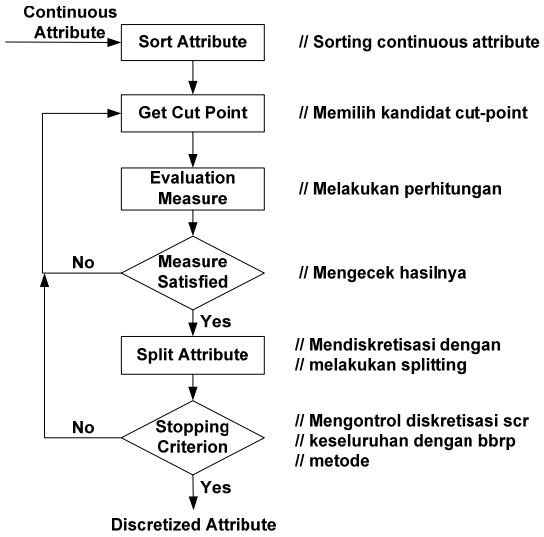
Smoothing dilakukan jika data mengandung noise/nilai yang tidak valid terhadap data yang di-mining. Untuk mengatasinya harus dilakukan smoothing (dengan memperhatikan nilai-nilai tetangga). Berikut teknik atau metode untuk smoothing:

- *Binning*
  Metode binning dilakukan dengan memeriksa "nilai tetangga", yaitu nilai-nilai yang ada disekelilingnya. Berikut adalah langkah-langkah metode binning:
  1. Data diurutkan dari yang terkecil sampai dengan yang terbesar.
  2. Data yang sudah urut kemudian dipartisi ke dalam beberapa bin. Teknik partisi ke dalam bin ada 2 (dua) cara: *equal-width* (*distance*) *partitioning* dan *equal-depth* (*frequency*) partitioning.
  3. Dilakukan smoothing dengan tiga macam teknik, yaitu: smoothing *by bin-means*, smoothing *by bin-medians*, dan smoothing *by bin-boundaries*.
- *Clustering*
  Digunakan untuk menyingkirkan *outliers* (keluar jauh-jauh dari cluster/*centroid*), data yang memiliki noise. Algoritma *k*-Means yang merupakan kategori metode partitioning dapat digunakan jika ukuran *database* tidak terlalu besar. Algoritma ini didasarkan pada nilai tengah dari objek yang ada dalam cluster. Algoritma *k*-Means meminta inputan parameter *k*, dan mempartisi satu set *n* objek ke dalam *k* cluster sehingga menghasilkan tingkat kemiripan yang tinggi antar objek dalam kelas yang sama (*intra-class similarity*) dan tingkat kemiripan yang paling rendah antar objek dalam kelas yang berbeda (*inter-class similarity*). Kemiripan cluster diukur dengan menghitung nilai tengah dari objek yang ada di dalam cluster.
- *Regression*
  Linear regression memodelkan sebuah random variable, *Y* (disebut *response variable*) sebagai sebuah fungsi linier dari random variable yang lain, *X* (disebut sebagai *predictor variable*), dengan persamaan empiris: $Y = \alpha + \beta X$, dimana α dan β adalah koefisien regresi. Koefisien ini dapat dihitung menggunakan metode *least squares* dengan persamaan sebagai berikut: $\beta = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$ dan $\alpha = \bar{y} - \beta\bar{x}$, dimana $\bar{x}$ adalah nilai rata-rata dari $x_1, x_2, \ldots, x_i$ dan $\bar{y}$ adalah nilai rata-rata dari $y_1, y_2, \ldots, y_i$.

ISSN: 2089-1121

B. Generalization

Generalization atau generalisasi adalah ketika data level rendah (*low-level data*) diganti dengan konsep yang lebih tinggi, yaitu dengan melakukan *diskretisasi*. Teknik diskretisasi dapat digunakan untuk mereduksi sekumpulan nilai yang terdapat pada atribut continuous, dengan membagi *range* dari atribut ke dalam *interval*.



Gambar 2. Proses Diskretisasi

Proses diskretisasi secara umum terdiri dari 4 tahapan (gambar 2), yaitu:

1. *Sorting*, melakukan sorting nilai atribut continuous yang mau didiskretisasi.
2. *Memilih "cut-point"*, banyak fungsi evaluasi yang dapat digunakan seperti binning dan pengukuran entropy.
3. *Splitting*, dilakukan evaluasi cut-point yang ada dan pilih satu yang terbaik dan lakukan split range nilai atribut continuous ke dalam dua partisi. Diskretisasi berlanjut untuk tiap partisi sampai kondisi berhenti tercapai.
4. *Stopping criterion*, diperlukan untuk menghentikan proses diskretisasi.

Ada 5 metode untuk melakukan diskretisasi pada atribut continuous, yaitu: binning, cluster analysis, *histogram analysis*, *entropy-based discretization*, dan *segmentation by "natural partitioning"*. Dua metode pertama telah dibahas pada data smoothing, pada subbab ini akan dibahas 3 metode yang lainnya.

- Histogram Analysis
  Seperti binning sebelumnya, pertama data harus diurutkan dahulu kemudian membagi data ke dalam keranjang dan menyimpan nilai rata-rata (total) tiap keranjang. Untuk menentukan jumlah keranjang dan nilai atribut yang dipartisi, ada beberapa aturan partisi yaitu: *equal-width*, *equal-depth*, *V-Optimal*, dan *MaxDiff*. V-Optimal dan MaxDiff histogram cenderung lebih akurat dan praktis.
- Entropy-Based Discretization
  Diskretisasi berdasarkan nilai entropy merupakan metode diskretisasi secara *supervised*. Seperti metode diskretisasi lainnya, atribut yang mau didiskretisasi diurutkan dahulu. Algoritma *supervised* ini menggunakan *class information entropy* dari partisi untuk memilih batas bin dalam melakukan diskretisasi. Jika diberikan sebuah data set *S*, sebuah atribut *A*, dan sebuah batas partisi *T*, *class information entropy* dari partisi dengan threshold *T* adalah:

ISSN: 2089-1121

$E(A,T;S) = \frac{|S_1|}{|S|}Ent(S_1) + \frac{|S_2|}{|S|}Ent(S_2)$ dimana S_1 dan S_2 adalah data yang ada pada *S* secara berturut-turut untuk kondisi $A < T$ dan $A \geq T$. Fungsi entropy *Ent* untuk sekumpulan data yang diberikan, dihitung berdasarkan distribusi *class* dari data pada kumpulan tersebut. Contohnya diberikan *m* class, entropy dari S_1 adalah: $Ent(S_1) = -\sum_{i=1}^{m} p_i \log_2(p_i)$ dimana p_i adalah probabilitas class *i* pada S_1. Nilai dari *Ent(*S_2*)* dihitung dengan cara yang sama.

Untuk atribut *A*, batas T_{min} yang meminimalkan fungsi entropy (atau memaksimalkan information gain) dari semua batas partisi yang mungkin, dipilih sebagai batas partisi. Metode ini dapat digunakan secara rekursif untuk kedua partisi sampai kondisi berhenti tercapai, yaitu: $Gain(A,T;S) < \delta$ dimana δ adalah metode *stopping criterion* yang mau diimplementasikan.

- Segmentation by Natural Partitioning
  Secara umum, aturan partisi terhadap data dibagi menjadi 3, 4, atau 5 interval dengan range yang sama (equal-width interval) secara rekursif dan perlevel, berdasarkan range nilai pada most significant digit. Aturannya adalah sebagai berikut:
  - Jika sebuah interval meliputi 3, 6, 7, atau 9 distinct value pada most significant digit (*msd*), maka mempartisi range tersebut ke dalam 3 interval.
  - Jika sebuah interval meliputi 2, 4, atau 8 distinct value pada *msd*, maka mempartisi range tersebut ke dalam 4 equal-width interval.
  - Jika sebuah interval meliputi 1, 5, atau 10 distinct value pada *msd*, maka mempartisi range tersebut ke dalam 5 equal-width interval.

C. Normalization

Normalization atau normalisasi adalah proses transformasi dimana sebuah atribut numerik diskalakan dalam range yang lebih kecil seperti -1.0 sampai 1.0, atau 0.0 sampai 1.0. Ada beberapa metode/teknik yang diterapkan untuk normalisasi data, diantaranya:

- *Min-max Normalization*
  Min-max normalization memetakan sebuah value *v* dari atribut *A* menjadi v' ke dalam range [new_min_A, new_max_A] berdasarkan rumus: $v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$
- *Z-Score Normalization*
  Disebut juga zero-mean normalization, dimana value dari sebuah atribut *A* dinormalisasi berdasarkan nilai rata-rata dan standar deviasi dari atribut *A*. Sebuah value *v* dari atribut *A* dinormalisasi menjadi v' dengan rumus: $v' = \frac{v - \overline{A}}{\sigma_A}$ dimana $\overline{A}$ dan σ_A adalah nilai rata-rata dan standar deviasi dari atribut *A*.
- *Normalization by Decimal Scaling*

ISSN: 2089-1121

Normalisasi yang diperoleh dengan melakukan penggeseran titik desimal dari value sebuah atribut *A*. Jumlah titik desimal yang digeser tergantung dari nilai absolut maksimum dari atribut *A*.

D. Aggregation

Adalah operasi *summary* (peringkasan) diaplikasikan pada data numerik. Misalnya pada data penjualan harian digabungkan untuk menghitung pendapatan perbulan dan pertahun dengan dirata-rata atau ditotal. Langkah ini dilakukan dengan memanfaatkan operator *data cube* (operasi *roll up*/meringkas).

E. Attribute/Feature Construction

Pada attribute construction, atribut baru dibentuk dari atribut yang sudah ada dan ditambahkan bersama atribut lainnya untuk membantu meningkatkan ketelitian/ketepatan dan pemahaman struktur dalam *high-dimensional* data. Contohnya, mau menambahkan atribut luas berdasarkan atribut tinggi dan lebar. Atau, atribut lama kerja jadi dosen dan usia bisa digantikan dengan senioritas, yunioritas, dan orientasi.

2. ALGORITMA DATA TRANSFORMATION PADA WEKA

Dalam pelaksanaan penelitian ini dibutuhkan beberapa tahap proses yang harus dilakukan. Tahapan-tahapan yang dilakukan adalah sebagai berikut:

A. Smoothing

- EmptyAttributeFilter: menghapus semua atribut yang hanya mempunyai satu nilai, mempunyai satu nilai dan ada nilai yang tidak terisi, atau semua nilai atributnya tidak terisi.

B. Generalization

- DiscretizeFilter: mendiskretisasi range dari atribut numerik yang ada pada data set ke dalam atribut yang bertipe nominal.
- MakeIndicatorFilter: menciptakan sebuah data set baru dengan sebuah atribut boolean menggantikan sebuah atribut nominal.
- MergeTwoValuesFilter: mengga-bungkan dua buah nilai dari sebuah atribut nominal ke dalam satu kategori.
- NumericToBinaryFilter: mengu-bah semua atribut numerik pada data set ke dalam atribut yang bertipe biner.

C. Normalization

- NormalizationFilter: menormalisasi semua nilai atribut numerik pada data set dengan interval [0..1].

D. Attribute/Feature Construction

- AddFilter: menambahkan sebuah atribut baru ke dalam data set, atribut baru tersebut akan berisi missing value.
- AttributeExpressionFilter: menciptakan sebuah atribut baru dengan menggunakan ekspresi matematika pada atribut yang sudah ada.

ISSN: 2089-1121

E. Algoritma Tambahan

- NominalToBinaryFilter: mengubah semua atribut yang bertipe nominal yang ada pada data set ke dalam atribut biner.
- NonSparseToSparseFilter: mengubah semua instance yang ada pada data set ke dalam format sparse.
- NumericTransformFilter: mengubah atribut numerik berdasarkan fungsi transformasi yang digunakan.
- ObfuscateFilter: mengganti nama relasi, semua nama atribut yang ada, dan semua nilai atribut yang bertipe nominal dan string menjadi nama lain.
- SparseToNonSparseFilter: mengubah semua sparse instance yang terdapat pada data set ke dalam format non-sparse.
- StringToNominalFilter: mengubah atribut bertipe string menjadi atribut bertipe nominal.

3. MENGINTEGRASIKAN CLASS KE DALAM WEKA

Class baru yang dibuat harus mengikuti beberapa ketentuan agar dapat dijalankan seperti class-class data transformation yang lainnya. Berikut adalah beberapa persyaratan yang harus diperhatikan dalam menulis sebuah filter:

- Harus merupakan turunan dari superclass Filter. Menulis sebuah filter baru pada dasarnya meng-*override*, hanya tiga *method* yang perlu dioverride/diubah untuk mengimplementasikan sebuah filter dengan fungsionalitas yang baru, yaitu setInputFormat(), input(), dan batchFinished(). Jika filter dapat memproses setiap instancenya secara langsung, batchFinished() tidak perlu diubah.
- Apabila metode yang ditambahkan membutuhkan *parameter option* yang diinputkan oleh user, maka class yang dibuat harus mengimplementasikan *interface* OptionHandler dan kemudian meyediakan code untuk methodnya. Interface OptionHandler mendefinisikan method yang diimplementasikan oleh semua class yang dapat memproses option pada baris perintah (*command line option*).
- Tidak diperbolehkan melakukan perubahan data input, tidak diijinkan menambahkan instance baru ke dalam data set. Dihindari dengan menyimpan sebuah *copy*-an data set yang masih kosong ke dalam m_InputFormat.
- Setiap class yang meng-override method setInputFormat() harus memanggil method setInputFormat() milik class induk, hal ini dilakukan dengan code: super.setInputFormat (Instances).
- Instance yang diinputkan ke dalam filter tidak boleh di-*push*-kan secara langsung ke dalam *queue* output melainkan harus digantikan dengan sebuah objek baru dari class Instance. Misalnya dengan contoh sederhana ingin melewatkan semua instance pada data set dengan format *non-sparse* tanpa terjadi perubahan, maka dilakukan dengan memanggil method copy() dari class Instance sebelum menambahkannya ke dalam queue output.

4. PENUTUP

Beberapa kesimpulan yang didapat dari penelitian ini adalah pemilihan algoritma yang digunakan tergantung perlu atau tidaknya melihat semua data set sebelum melakukan proses pada setiap instancenya, apakah menggunakan algoritma

ISSN: 2089-1121

yang memproses setiap instance secara langsung, atau algoritma yang membaca semua training instance. Metode yang sama dapat digunakan untuk pendekatan data transformation yang berbeda, contohnya adalah metode binning dan clustering yang dipakai pada data smoothing juga digunakan untuk generalization. Hal yang sama juga terjadi pada tahapan yang lain pada data preprocessing. Algoritma data transformation yang diimplementasikan dapat bersifat *supervised* atau *unsupervised*. Semua algoritma filtering pada Weka khususnya algoritma yang mengimplementasikan data transformation, menjalankan method utama yang sama untuk melakukan preprocessing pada data set. Tentunya beberapa method perlu dilakukan override untuk implementasi filter dengan fungsionalitas yang berbeda. Pemanfaatan Weka untuk preprocessing, dapat ditambahkan metode baru untuk melakukan transformasi data dan mengintegrasikannya ke dalam lingkungan Weka dengan memperhatikan beberapa ketentuan yang harus dipatuhi.

Namun Weka selalu berkembang seiring dengan waktu, telah mempunyai beberapa versi dan terus diupdate sampai saat ini. Algoritma data transformation yang dimplementasikan pada Weka merupakan sebagian kecil dari keseluruhan sistem yang ada pada Weka. Apabila bermaksud untuk membuat sebuah aplikasi data mining, dapat dilakukan dengan mengakses program yang terdapat pada Weka dengan menggunakan bahasa sendiri, karena class library yang dimiliki Weka dapat diakses dari program Java yang dibuat sendiri.

5. DAFTAR PUSTAKA

Cios, Krzysztof J. dan Lukasz A. Kurgan. *Trends in Data Mining and Knowledge Discovery*. 2002.

Ding, Qin. *Introduction to Data Mining*. 2004.

Dougherty, J., R. Kohavi, dan M. Sahami. *Supervised and Unsupervised Discretization of Continuous Features*. 1995.

Han, Jiawei dan Micheline Kamber. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann. 2001.

Hortmann, Cay S. dan Garry Cornell. *Core Java 2: Volume I - Fundamentals*. California: Sun Microsystems, Inc. 2001.

Hortmann, Cay S. dan Garry Cornell. *Core Java 2: Volume II - Advanced Features*. California: Sun Microsystems, Inc. 2000.

Liu, H., F. Hussain, C.L. Tan, dan M. Dash. *Discretization: An Enabling Technique*. 12 Juli 2000.

Seldi, Thomas. *Data Mining Algorithm*. 2004.

Witten, Ian H. dan Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* San Francisco: Morgan Kaufmann. 1999.