BAB 2 DATA

Pendahuluan

Bab ini membahas beberapa isu-isu yang terkait dengan data yang penting untuk suksesnya *data mining*. Isu-isu tersebut meliputi

Tipe data; *Data set* berbeda dalam beberapa hal. Sebagai contoh, atribut-atribut digunakan untuk menjelaskan objek-objek data dari tipe-tipe yang berbeda, kualitatif atau kuantitatif. *Data set* juga dapat memiliki karakter khusus; misalnya beberapa *data set* mengandung deret waktu atau objek dengan hubungan eksplisit ke objek yang lain. Tipe data menentukan *tool* yang mana dan teknik apa yang akan digunakan untuk menganalisis data.

Kualitas data; Data seringkali jauh dari sempurna. Walaupun kebanyakan teknik *data mining* dapat mentoleransi beberapa tingkat ketidaksempurnaan dalam data, pemahaman dan peningkatan kualitas data secara khusus meningkatkan kualitas dari analisis yang dihasilkan. Isu kualitas data meliputi adanya *noise* dan *outlier*; data yang hilang, data yang tidak konsisten, atau data duplikat; dan data yang bias.

Langkah preprocessing untuk membuat data lebih sesuai untuk *data mining*; Seringkali data mentah perlu diproses agar data tersebut sesuai untuk analisis. Selain meningkatkan kualitas data, data seringkali dimodifikasi agar lebih cocok dengan teknik *data mining* tertentu. Sebagai contoh, atribut kontinu seperti panjang dapat ditranformasi ke dalam kategori diskret seperti pendek, sedang atau panjang, agar teknik tertentu dapat diaplikasikan. Selain itu, banyaknya atribut dalam *data set* sering kali dikurangi karena banyak teknik bekerja lebih efektif ketika data memiliki sejumlah atribut yang relatif lebih sedikit.

Menganalisis data dalam bentuk relasinya; Satu pendekatan untuk analisis data adalah menemukan hubungan antara objek-objek data dan kemudian melakukan analisis sisanya menggunakan hubungan-hubungan ini daripada menggunakan objek-objek data itu sendiri. Sebagai contoh, kita dapat menghitung kemiripan atau jarak antar sepasang objek dan kemudian melakukan analisis – *clustering*, klasifikasi, atau deteksi anomali– berdasarkan pada kemiripan dan jarak tersebut.

2.1 Tipe Data

Sebuah *data set* dapat dipandang sebagai sebuah koleksi dari objek- objek data. Nama lain dari sebuah objek data adalah *record*, titik, vektor, pola, *event*, case, *sample*, observasi atau entitas. Objek-objek data dijelaskan oleh sejumlah atribut yang menangkap karakteristik dasar dari sebuah objek, seperti massa dari sebuah objek fisik atau waktu pada saat sebuah kejadian terjadi. Nama-nama lain untuk atribut adalah variabel, karekteristik, *field*, fitur atau dimensi.

Contoh 2.1 (Informasi mahasiswa)

Seringkali, sebuah *data set* adalah sebuah file, dimana objek adalah *record-record* (atau baris) dalam file dan setiap *field* (atau kolom) berkaitan dengan sebuah atribut. Sebagai contoh, Tabel 2.1 menunjukkan sebuah *data set* yang terdiri dari informasi mahasiswa. Setiap baris berkaitan dengan seorang siswa dan setiap

kolom adalah sebuah atribut yang menjelaskan suatu aspek dari seorang mahasiswa seperti grade point average (GPA) atau identification number (ID).

Tabel 2.1 *Data set* mahasiswa

Student ID	Year	Grade Point Average (GPA)	…
…	…	…	…
1034262	Senior	3.24	…
1052663	Sophomore	3.51	…
1082246	Freshman	3.62	…
…	…	…	…

Selain *record-based*, *data set* juga bisa dalam bentuk *flat file* atau sistem basis data relasional.

2.1.1 Atribut dan Pengukuran

Definisi 2.1:

Sebuah **atribut** adalah sebuah sifat atau karakteristik dari sebuah objek yang dapat bervariasi, baik dari satu objek ke objek yang lain atau dari satu waktu ke waktu yang lain.

Sebagai contoh, warna mata bervariasi dari satu orang ke orang lain, sedangkan temperatur dari sebuah objek bervariasi sepanjang waktu. Perhatikan bahwa warna mata adalah sebuah atribut simbolik dengan sejumlah kecil nilai yang mungkin {coklat, hitam, biru, hijau, dan lain-lain), sementara temperatur adalah atribut numerik dengan banyaknya nilai yang tak terhingga.

Definisi 2.2:

Skala pengukuran adalah aturan (fungsi) yang menghubungkan nilai numerik atau simbolik dengan sebuah atribut dari sebuah objek.

Proses pengukuran adalah pengunaan skala pengukuran untuk menghubungkan sebuah nilai dengan sebuah atribut tertentu dari sebuah objek. Sebagai contoh, kita menghitung banyaknya kursi dalam sebuah ruangan untuk melihat apakah terdapat cukup empat duduk untuk semua orang yang akan datang pada sebuah pertemuan. Dalam kasus tersebut, nilai fisik dari sebuah atribut dari sebuah objek dipetakan ke sebuah nilai numerik atau simbolik.

Tipe dari atribut

Sifat dari sebuah atribut tidak perlu sama dengan sifat dari nilai yang digunakan untuk mengukur nilai tersebut. Dengan kata lain, nilai-nilai yang digunakan untuk merepresentasikan sebuah atribut dapat memiliki sifat yang bukan merupakan sifat dari atribut itu sendiri, dan sebaliknya.

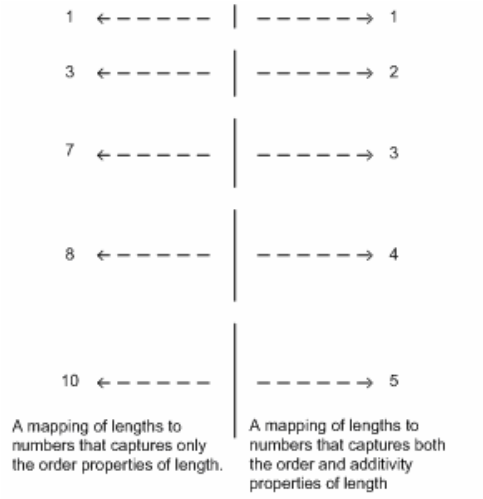
Contoh 2.2 (Umur karyawan dan ID Number):

Dua atribut yang terkait dengan karyawan adalah ID dan umur (dalam tahun). Kedua atribut terebut dapat dinyatakan sebagai integer. Kita mungkin saja ingin mengetahui rata-rata dari umur karayawan, tetapi tidak mungkin dan tidak berguna menginginkan rata-rata ID karayawan. Aspek dari karyawan yang bisa ditangkap dari atribut ID adalah bahwa ID tersebut nilainya berbeda. Sehingga operasi yang valid untuk ID karyawan adalah memeriksa apakah ID-ID tersebut

adalah sama. Tidak ada persyaratan dari pembatasan ini, ketika integer digunakan untuk merepresentasikan atribut ID karyawan. Untuk atribut umur, sifat dari integer yang digunakan untuk merepresentasikan umur adalah sifat-sifat atribut. Akan tetapi sifat-sifat integer tidak sepenuhnya dinyatakan sebagai sifat dari atribut umur, sebagai contoh umur memiliki sebuah nilai maksimum, sedangkan integer tidak.

Contoh 2.3 (Panjang dari segmen garis):

Perhatikan Gambar 2.1, yang menunjukkan beberapa objek, yaitu segmen-segmen garis, dan menunjukkan bagaimana atribut panjang dari objek-objek ini dapat dipetakan ke dalam bilangan dalam dua cara yang berbeda. Setiap segmen garis yang berurutan, mulai dari atas ke bawah, dibentuk dengan menambahkan segmen garis yang paling atas ke dirinya. Dengan demikian, segmen garis kedua dari atas dibentuk dengan menambahkan segmen garis yang paling atas kepada dirinya sebanyak dua kali. Segmen garis ketiga dari atas dibentuk dengan menambahkan segmen garis paling atas ke dirinya sebanyak tiga kali, demikian seterusnya. Dalam pengertian fisik, semua segmen garis adalah penggandaan dari garis pertama. Fakta ini dinyatakan oleh pengukuran pada ruas kanan dalam gambar. Skala pengukuran pada ruas kiri hanya menyatakan pengurutan dari panjang atribut, sedangkan sisi kanan menyatakan pengurutan dan sifat tambahan. Dengan demikian, seuah atribut dapat diukur dalam sebuah cara yang tidak mengambil semua sifat dari atribut.



Gambar 2.1 Pengukuran dari panjang segmen garis pada dua skala pengukuran yang berbeda

Tipe dari atribut penting untuk diketahui karena menyatakan sifat-sifat dari nilai yang terukur konsisten dengan sifat yang mendasari atribut. Seringkali tipe dari atribut dirujuk sebagai tipe dari sebuah skala pengukuran.

Cara yang sederhana untuk menentukan tipe dari sebuah atribut adalah mengidentifikasi sifat-sifat dari bilangan yang berkaitan dengan sifat mendasar dari atribut. Sebagai contoh, sebuah atribut seperti panjang memiliki banyak sifat-

sifat bilangan. Berikut adalah sifat-sifat bilangan yang sering digunakan untuk menjelaskan atribut:

- Kesamaan dan ketaksamaan: = dan ≠
- Urutan: <, ≤, > dan ≥
- Penambahan dan pengurangan: + dan –
- Pengandaan dan pembagian: * dan /

Terdapat empat tipe atribut seperti diberikan dalam Tabel 2.2.

Tabel 2.2 Tipe-tipe atribut yang berbeda

Tipe atribut		Deskripsi	Contoh	Operasi
Kategori (kualitatif)	Nominal	Nilai dari atribut nominal adalah nama-nama yang berbeda, yaitu nilai nominal hanya menyediakan informasi yang cukup untuk membedakan satu objek dengan objek yang lain. (= dan ≠)	Kode pos, ID Number karyawan, warna mata, jenis kelamin	Mode, entropy, contingency correlation, uji χ^2
	Ordinal	Nilai dari atribut ordinal menyediakan informasi yang cukup mengurutkan objek. (<, >)	Kekerasan mineral {baik, lebih baik, sangat baik}, nomor jalan, grade	Median, presentil, rank correlation, run test, sign test
Numerik (Kuantitatif)	Interval	Untuk atribut interval, perbedaan antarnilai adalah sesuatu yang berarti, adanya unit pengukuran. (+,–)	Tanggal pada kalender, temperatur dalam Celcius atau Fahrenheit	Rataan, simpangan baku, korelasi Pearson, Uji t dan F
	Ratio	Untuk variabel rasio, perbedaan dan rasio merupakan hal yang berarti. (*, /)	Temperatur dalam Kelvin, kuantitas moneter, count, umur, panjang, arus listrik	Rataan geometri, rataan harmonik, variasi persen

Atribut nominal dan ordinal secara kolektif dinyatakan sebagai atribut kategori atau kualitatif. Atribut kualitatif tidak memiliki sifat dari bilangan. Jika atribut tersebut direpresentasikan oleh bilangan misalkan integer, maka integer tersebut harus diperlakukan sebagai simbol. Atribut interval dan rasio secara kolektif dinyatakan sebagai atribut kantitatif atau numerik. Atribut ini dinyatakan oleh bilangan dan memiliki sifat-sifat bilangan. Atribut kuantitatif dapat bernilai integer atau kontinyu.

Tipe atribut juga dapat dinyatakan dalam bentuk tranformasi yang tidak merubah arti dari atribut tersebut. Sebagai contoh, arti dari atribut panjang tidak akan berubah jika atribut tersebut diukur dalam meter daripada dalam feet. Tabel 2.3 menunjukkan transformasi yang diperbolehkan untuk keempat atribut pada Tabel 2.2.

Tabel 2.3 Transformasi yang diperbolehkan untuk tipe-tipe atribut

Tipe atribut		Transformasi	Keterangan
Kategori (kualitatif)	Nominal	Pemetaan satu-satu, contoh permutasi dari nilai	Jika semua nomor ID karyawan ditetapkan ulang, perubahan tersebut tidak membuat adanya perbedaan
	Ordinal	Perubahan dari nilai yang mempertahankan urutan dari nilai-nilai tersebut, yaitu nilai_baru = f(nilai_lama) dimana f adalah fungsi monotonik	Atribut yang meliputi notasi baik, lebih baik, sangat baik dapat direpresentasikan oleh nilai {1, 2, 3} atau oleh {0.5, 1, 10}
Numerik (Kuantitatif)	Interval	Nilai_baru = a*nilai_lama + b, a dan b: konstanta	Skala temperatur Celcius dan Fahrenheit berbeda dalam posisi angka nol-nya dan ukuran dari unit-nya.
	Ratio	Nilai_baru = a*nilai_lama	Atribut panjang dapat diukur dalam meter atau feet.

Salah satu cara untuk membedakan antarartibut adalah dengan nilai bilangan yang dapat diambil oleh atribut tersebut.

Dikret; Sebuah **atribut diskret** memiliki himpunan nilai berhingga atau tidak berhingga. Atribut demikian dapat berupa atribut kategori, seperti kode pos, Nomor ID karyawan, atau numerik seperti count. Atrbut diskret sering direpresetasikan menggunakan variabel integer. **Atribut biner** adalah kasus khusus dari atribut diskret dan hanya memiliki dua nilai, seperti yes/no, benar/salah, laki-laki/perempuan, atau 0/1. Atribut biner sering direpresentasikan dengan menggunakan veriabel Boolean, atau sebagai variabel integer yang hanya mempunyai nilai 0 atau 1.

Kontinu; **Atribut kontinu** adalah atribut yang memiliki nilai berupa bilangan real. Contoh atribut tersebut adalah temperatur, tinggi atau berat. Atribut kontinu secara khusus direpresentasikan sebagai variabel *floating-point*. Secara praktis, nilai-nilai real hanya dapat diukur dan direpresentasikan dengan presisi terbatas.

Secara khusus, atribut nominal dan ordinal adalah biner atau diskret, sedangkan atribut interval dan rasio adalah kontinu. Atribut count dapat berupa diskret atau juga atribut rasio.

Atribut asimetrik adalah atribut yang hanya memiliki nilai tak nol. Untuk *data set* dimana setiap objek adalah mahasiswa dan setiap atribut mencatat apakah mahasiswa mengambil mata kuliah tertentu atau tidak. Untuk mahasiswa tertentu, sebuah atribut memiliki nilai 1 jika mahasiswa tersebut mengambil mata kuliah yang terkait dengan atribut tersebut dan bernilai 0 untuk selainnya. Karena mahasiswa hanya mengambil sejumlah kecil dari mata kuliah yang ditawarkan, sebagian besar nilai dari *data set* adalah 0. Dengan demikian, analisis lebih bermakna dan lebih efisien bila difokuskan pada nilai tak nol.

Atribut biner yang hanya memiliki nilai tak nol dinamakan **atribut biner asimetrik**. Tipe atribut ini secara khusus penting untuk analisis asosiasi yang akan dibahas pada bab selanjutnya. Jika banyaknya kredit yang terkait dengan

setiap mata kuliah dicatat, maka *data set* yang dihasilkan akan mengandung **atribut diskret asimterik** atau **atribut kontinu asimetrik**.

2.1.2 Tipe-tipe *Data set*

Terdapat banyak tipe dari *data set*. Dengan berkembangkan bidang *data mining*, berbagai *data set* tersedia untuk analisis. Dalam bagian ini, *data set* akan dikelompokan ke dalam 3 kelompok, yaitu data *record*, data berbasis graf, dan data terurut.

Terdapat tiga karakteristik *data set* dan memiliki pengaruh penting pada teknik *data mining* yang digunakan. Karakteristik tersebut adalah:

1. **Dimensionalitas**. Dimensionalitas dari *data set* adalah banyaknya atribut yang dimiliki objek dalam *data set*. Data dengan jumlah dimensi yang kecil cenderung berbeda secara kualitatif dibandingkan dengan data berdimensi sedang atau tinggi. Untuk memudahkan analisis, pada data berdimensi tinggi seringkali dilakukan reduksi dimensi yaitu pada tahap preprocessing.
2. ***Sparsity***. Untuk beberapa *data set*, misal pada *data set* yang mengandug atribut asimetrik, kebanyakan atribut memiliki nilai 0. Dalam kebanyakan kasus, kurang dari 1% dari keseluruhan data yang memiliki nilai tak nol. Dalam praktik, sparsity adalah keuntungan karena biasanya hanya nilai-nilai tak nol yang perlu disimpan dan dimanipulasi. Hasil ini secara signifikan menghemat biaya komputasi dan tempat penyimpanan.
3. **Resolusi**. Data pada tingkat resolusi yang berbeda seringkali diperoleh, dan sering pula sifat-sifat dari data berbeda pada resolusi yang berbeda. Sebagai contoh, permukaan bumi terlihat sangat tidak rata pada resolusi tertentu (dari beberapa meter), tetapi terlihat halus jika terlihat dari puluhan kilo meter. Pola data juga tergantung pada level resolusi. Jika resolusi terlalu halus, pola tertentu dapat tidak tampak atau dapat terkubur dalam *noise*; jika resolusi terlalu kasar, pola dapat hilang. Sebagai contoh, variasi dalam tekanan atmosfer pada skala jam merefelksikan pergerakan badai dan sistem cuaca lainnya. Pada skala bulanan, fenomena tersebut tidak akan terdeteksi.

Berikut adalah tiga kategori dari *data set*:

Data *record*

Data set ini merupakan kumpulan *record* (objek data), masing-masing *record* mengandung sekumpulan *field* data (atribut) (Gambar 2.2 (a)). Untuk kebanyakan bentuk dasar dari data *record*, tidak ada hubungan yang eksplisit diantara *record* atau *field* data, dan setiap *record* (objek) memiliki himpunan atribut yang sama. Data *record* biasanya disimpan dalam *flat file* atau dalam basis data relasioanal. Basis data relasional lebih dari pada koleksi data, tetapi *data mining* seringkali tidak menggunakan informasi tambahan yang ada dalam basis data relasional. Beberapa bentuk data *record* diilustrasikan dalam Gambar 2.2.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(a) Data record

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

(b) Data transaksi

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

(c) matriks data

	team	coach	play	ball	score	game	win	lost	timeout	season

(d) *Document-term matrix*

Gambar 2.2 Variasi yang berbeda dari data record

Data transaksi atau data *market basket*. Data transaksi adalah bentuk khusus dari data *record*, dimana setiap *record* (transaksi) meliputi sekumpulan item. Sebagai contoh pada toko grosir, sekumpulan produk yang dibeli oleh seorang pelanggan selama satu kali perjalanan belanja merupakan sebuah transaksi, dengakan produk individual yang dibeli merupakan item. Tipe data ini dinamakan data *market basket* karena item-item dalam setiap *record* adalah produk-produk dalam keranjang belanja seorang pelanggan. Data transaksi adalah koleksi dari himpunan-himpunan item, tetapi data tersebut dapat dipandang sebagai sekumpulan *record* yang memiliki *field-field* berupa atribut asimetrik. Seringkali atribut yang ada adalah atribut biner, yang menunjukkan apakah ada atau tidak item yang dibeli. Tetapi secara umum, atribut dapat berupa atribut diskret atau kontinu seperti banyaknya item yang dibeli atau banyaknya uang yang dibayarkan untuk item-item tersebut. Gambar 2.2 (b) menunjukkan contoh dari data transaksi. Setiap baris menyatakan pembelian dari seorang pelanggan pada waktu tertentu.

Matriks data. Jika objek-objek data dalam koleksi dari data seluruhnya memiliki kumpulan atribut-atribut numerik yang sama, maka objek data tersebut dapat dipandang sebagai titik atau vektor dalam ruang multidimesi, dimana setiap dimensi menyatakan atribut yang berbeda yang menjelaskan objek. Himpunan objek data demikian dapat diiterpretasikan sebagai matriks berukuran m × n, dimana m adalah jumlah baris dan n adalah jumlah kolom. Setiap baris menyatakan objek, sedangkan setiap kolom menyatakan atribut. Matriks demikian dinamakan **matriks data** atau **matriks pola**. Matriks data adalah variasi dari data *record*, tetapi karena matriks data terdiri dari atribut-atribut numerik, maka operasi dasar matriks dapat diaplikasikan untuk mentranformasi atau memanipulasi data. Dengan demikian, matriks data merupakan format data standar untuk kebanyakan data statsitik. Gambar 2.2 (c) menunjukkan contoh matriks data.

Matriks Data Jarang. Matriks data jarang (*sparse data matrix*) adalah kasus khusus dari matriks data dimana atribut-atribut memiliki tipe yang sama dan merupakan atribut asimetrik (hanya nilai yang tak nol yang penting). Data transaksi adalah contoh dari matriks data jarang yang hanya memiliki entri 0 atau 1. Contoh lainnya dalah data dokumen. Secara khusus, jika urutan dari istilah (kata) dalam sebuah dokumen diabaikan, maka dokumen dapat direpresentasikan sebagai vektor istilah, dimana setiap istilah adalah sebuah komponen (atribut) dari vektor dan nilai dari setiap komponen adalah banyaknya kemunculan istilah dalam dokumen. Representasi dari koleksi dokumen sering dinamakan ***document-term matrix***. Gambar 2.2 (d) menunjukkan contoh dari document-term matrix. Dokumen adalah baris dari matriks, sedangkan istilah adalah kolom dari matriks tersebut. Dalam praktis, entri-entri tak nol dari matriks data jarang yang disimpan.

Data berbasis Graf

Graf dapat digunakan dalam merepresentasikan data. Graf dapat menangkap hubungan antar objek data, atau objek data itu sendiri direpresentasikan dalam graf.

Relasi antar objek seringkali menyampaikan informasi yang penting. Dalam kasus demikian, data seringkali direpresentasikan dalam bentuk graf. Secara khusus, objek data dipetakan ke *node* dari graf, sedangkan hubungan antar objek dinyatakan oleh *link* (*arc*) antar objek dan sifat-sifat *link*, seperti arah dan bobot. Sebagai contoh, halaman web pada WWW, yang mengandung teks dan *link* ke halaman lain. Untuk memproses kueri pencarian, mesin pencari web mengumpulkan dan memproses halaman web dan mengekstrak isinya. *Link* dari dan ke halaman web menyediakan informasi yang relevan tentang halaman web sehingga perlu diperhatikan dalam proses kueri.

Jika objek memiliki struktur, bahwa objek tersebut mengandung subobjek yang memiliki relasi, maka objek demikian seringkali direpresentasikan sebagai graf. Sebagai contoh, struktur dari campuran bahan kimia dapat direpresentasikan oleh sebuah graf, dimana *node* adalah atom dan *link* antar *node* adalah ikatan bahan kimia.

Data terurut

Untuk beberapa tipe data, atribut memiliki hubungan yang melibatkan urutan waktu atau ruang. Tipe-tipe yang berbeda dari data terurut dapat dilihat pada Gambar 2.3.

Data sekuensial. Data sekuensial jika dirujuk sebagai data temporal. Data tersebut dapat dipandang sebagai perluasan dari data *record*, dimana setiap *record* memiliki nilai waktu yang berkaitan dengan *record* tersebut. Perhatikan *data set* transaksi ritel yang juga menyimpan waktu kapan transaksi tersebut terjadi. Informasi waktu ini memungkinkan untuk menemukan pola seperti “penjualan permen mencapai puncaknya sebelum Halloween”. Waktu dapat diasosiasikan dengan setiap atribut. Sebagai contoh, setiap atribut dapat menjadi histori pembelian dari seorang pelanggan, dengan sebuah daftar item-item yang dibeli pada waktu yang berbeda. Dengan menggunakan infromasi ini, dimungkinkan untuk mendapat pola seperti “orang yang membeli DVD player cenderung segera

membeli DVD dalam periode tertentu mengikuti pembelian DVD player". Gambar 2.3 (a) menunjukkan contoh data transaksi sekuensial. Terdapat lima waktu yang berbeda, yaitu t1, t2, t3, t4 dan t5; tiga pelanggan yang berbeda, yaitu C1, C2, dan C3; dan lima item yang berbeda, yaitu A, B, C, D dan E. Dalam tabel yang atas, setiap baris berkaitan dengan item yang dibeli pada waktu tertentu oleh setiap pelanggan. Sebagai contoh, pada waktu t3, pelanggan C2 membeli item A dan D. Dalam tabel yang di bawah, informasi yang sama ditampilkan, tetapi setiap baris berkaitan dengan seorang pelanggan tertentu. Setiap baris mengandung informasi pada setiap transaksi yang melibatkan pelanggan, dimana sebuah transaksi dipandang sebagai sebuah himpunan dari item dan waktu dimana item tersebut dibeli. Sebagai contoh, pelanggan C3 membeli item A dan C pada waktu t2.

Waktu	Pelanggan	Item yang dibeli
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Pelanggan	Waktu dan item yang dibeli
C1	(t1: A, B) (t2: C, D) (t5: A, E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG

Gambar 2.3 Bentuk-bentuk data terurut

Data urutan (*Sequence Data*). Data urutan terdiri dari *data set* yang merupakan urutan dari entitas individual seperti urutan kata atau huruf. Data ini hampir mirip dengan data sekuesial, kecuali bahwa dalam data urutan tidak ada unsur waktu, akan tetapi terdapat posisi dalam rangkaian yang terurut. Sebagai contoh, informasi genetik dari tanaman dan binatang dapat direpresentasikan dalam bentuk rangkaian nucleotide yang dikenal sebagai gen. Gambar 2.3 (b) menunjukkan bagian dari kode genetik manusia yang dinyatakan menggunakan empat nucleotide: A, T, G dan C.

Data time series. Data time series adalah bentk khusus dari data sekuensial dimana setiap *record* adalah sebuah time series, yaitu sebuah rangkaian dari pengukuran yang diambil sepanjang waktu. Sebagai contoh, *data set* finansial dapat terdiri dari objek-objek yang merupakan time series dari harga harian dari berbagai stock.

Data spasial. Beberapa objek memiliki atribut-atribut spasial, seperti posisi atau area, juga tipe atribut lainnya. Salah satu contoh dari data spasial adalah data cuaca (curah hujan, temperatur, dan tekanan) yang dikumpulkan dari berbegai lokasi geografis.

Sebagian besar algoritme *data mining* dirancang untuk data *record* dan variasinya, seperti data transaksi dan matriks data. Teknik berorientasi *record*

dapat diaplikasikan ke data bukan *record* dengan mengekstrak fitur-fitur dari objek data dan menggunakan fitur-fitur ini untuk membuat *record* yang terkait dengan setiap objek data. Perhatikan struktur bahan kimia yang dijelaskan sebelumnya. Diberikan sekumpulan substruktur, setiap campuran dapat direpresentasikan sebagai sebuah *record* dengan atribut biner yang menunjukkan apakah campuran mengandung substruktur tertentu ataukah tidak. Representasi demikian merupakan *data set* transaksi, dimana transaksi adalah campuran dan itemnya adalah substruktur.

2.2 Kualitas Data

Data mining sering diaplikasikan pada data yang telah dikumpulkan untuk maksud yang lain, atau untuk masa mendatang. Untuk menjaga kualitas data, *data mining* memfokuskan ada (1) deteksi dan koreksi dari masalah kualitas data dan (2) menggunakan algoritme yang dapat menberikan tolerasi kualitas data yang jelek. Koreksi dan deteksi masalah kualitas data sering dikenal sebagai data cleaning.

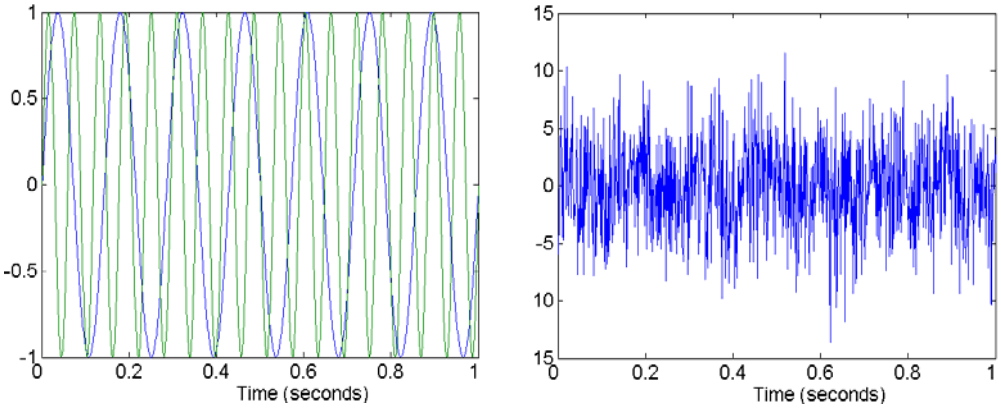
2.2.1 Pengukuran dan Isu Pengumpulan Data

Data yang sering kita jumpai adalah data yang tidak sempurna. Terdapat banyak masalah yang berkaitan dengan pengukuran data, seperti *human error*, keterbatasan pada alat pengukuran, atau kesalahan-kesalahan pada proses pengumpulan data. Nilai atau bahkan keseluruhan objek data dapat hilang. Dalam kasus lain sering dijumpai pula adanya duplikasi data, yaitu terdapat banyak objek data yang semuanya terkait dengan sebuah objek real. Sebagai contoh, dalam *data set* mungkin terdapat dua *record* untuk seseorang yang baru-baru ini tinggal pada dua alamat yang berbeda. Kadangkala data yang ada terlihat bagus padahal mengandung data yang tidak konsisten, misalnya seseorang dapat memiliki panjang 2 meter tetapi hanya memiliki bobot 2 kilogram.

Error Pengukuran

Error pengukuran merujuk pada masalah yang dihasilkan dari proses pengukuran, dimana nilai yang dicatat berbeda dengan nilai sebenarnya. Untuk atribut kontinu, beda numerik dari nilai yang diukur dan nilai sebenarnya dinamakan *error*. Istilah data collection *error* merujuk ke *error* seperti penghilangan objek data atau nilai atribut atau memasukan sebuah objek data secara tidak tepat. Sebagai contoh, studi seekor binatang dari spesies tertentu dapat melibatkan binatang dari spesies terkait yang penampilannya mirip dengan spesies yang sedang dipelajari.

Noise adalah komponen acak dari *error* pengukuran. *Noise* dapat melibatkan distorsi dari sebuah nilai atau tambahan dari objek yang palsu. Gambar 2.4 menunjukkan dua gelombang sinus sebelum dan setelah kena *noise*.



(a) Dua gelombang Sinus (b) Dua gelombang Sinus +*noise*

Gambar 2.4 Dua gelombang Sinus sebelum dan setelah kena *noise*

Outlier

Outlier adalah (1) objek data yang memiliki karakteristik yang berbeda dari kebanyakan objek data lainnya dalam *data set*, atau (2) nilai dari atribut yang tidak biasa terhadap nilai khas untuk atribut tersebut. Perlu dibedakan antara *noise* dengan *outlier*. *Outlier* merupakan objek data atau nilai yang sah. Tidak seperti *noise*, *outlier* kadang-kadang menarik untuk dianalisis seperti dalam *fraud detection* atau *network intrusion detection*.

Nilai yang Hilang

Seringkali ditemui sebuah objek yang kehilangan satu atau lebih nilai atributnya. Dalam beberapa kasus, informasi tidak dikumpulkan; sebagai contoh beberapa orang menolak memberikan data umur dan berat badannya. Dalam kasus lain, beberapa atribut tidak digunakan untuk semua objek data; sebagai contoh formulir yang memiliki bagian kondisional yang akan diisi jika seseorang menjawab pertanyaan sebelumnya, tetapi untuk kemudahan semua *field* tersebut disimpan. Terdapat beberapa strategi untuk menangani data yang hilang, yaitu

1. Mengeliminasi objek data atau atribut

   Cara sederhana dan efektif adalah menghilangkan objek yang memiliki nilai yang hilang. Walaupun beberapa objek data tertentu mengandung informasi, tetapi jika banyak objek yang memiliki nilai yang hilang, maka analisis akan sulit dilakukan. Penghilangan objek data atau atribut harus dilakukan dengan hati-hati, karena mungkin saja atribut atau objek data yang dibuang merupakan salah satu bagian penting dalam analisis.

2. Mengeliminasi nilai yang hilang

   Kadang-kadang data yang hilang dapat diestimasi. Sebagai contoh, perhatikan sebuah time series yang berubah dalam mode yang halus, tetapi memiliki sedikit nilai yang hilang yang tersebar secara luas. Dalam kasus demikian, nilai yang hilang dapat diestimasi (diinterpolasi) dengan menggunakan nilai-nilai yang ada. Contoh lain adalah pada *data set* yang memiliki banyak titik data yang mirip. Dalam kasus ini, nilai atribut dari titik terdekat ke titik yang

memiliki nilai yang hilang sering digunakan untuk mengestimasi nilai yang hilang tersebut. Jika atribut adalah kontinu, maka digunakan rataan dari nilai atribut dari data terdekat. Sedangkan jika data adalah kategori, maka diambil nilai atribut yang paling banyak muncul.

3. Mengabaikan nilai yang hilang selama analisis

   Banyak pendekatan *data mining* yang dapat dimodifikasi untuk mengabaikan nilai yang hilang. Sebagai contoh, anggaplah objek-objek sedang di-cluster-kan dan kemiripan antara pasangan objek data perlu diperhitungkan. Jika satu atau kedua objek dari pasangan tersebut memiliki nilai yang hilang untuk beberapa atribut, maka kemiripan dapat diperhitungkan dengan hanya menggunakan atribut yang tidak mengandung nilai yang hilang.

Nilai yang tidak Konsisten

Data dapat mengandung nilai yang tidak konsisten. Perhatikan *field* alamat, dimana didalamnya terdapat kode pos dan kota, tetapi area kode pos tertentu tidak terdapat dalam kota tersebut. Hal ini mungkin saja terjadi dikarenakan pemasukan data atau kesalahan dalam membaca formulir yang berisi data tersebut. Hal yang lebih penting daripada mencari penyebab ketidakkonsistenan data adalah mendeteksi, dan jika mungkin memperbaikan ketidakkonsistenan tersebut. Beberapa bentuk ketidakkonsistenan cukup mudah untuk dideteksi. Sebagai contoh, tinggi badan seseorang tidak mungkin negatif.

Data Duplikat

Sebuah *data set* mungkin meliputi objek data yang merupakan duplikat, atau hampir, dari data yang lain. Banyak orang menerima surat yang sama (duplikat) karena mereka muncul berkali-kali dalam basis data dengan nama-nama yang agak berbeda. Untuk mendeteksi dan menghilangkan duplikasi demikian, perlu diperhatikan hal-hal berikut:

1. Jika terdapat dua objek yang secara aktual merepresentasikan sebuah objek, maka nilai dari atribut yang terkait dapat berbeda, dan nilai yang tidak konsisten tersebut harus diatasi.
2. Diperlukan langkah yang hati-hati untuk menghindari penggabungan secara tidak sengaja dari data objek yang mirip, tetapi bukan duplikat, misalnya dua orang yang berbeda dengan nama yang identik.

2.3 *Preprocessing* Data

Langkah preprocessing perlu dilakukan agar data dapat sesuai untuk *data mining*. Strategi atau pendekatan yang sering digunakan adalah agregasi, *sampling*, reduksi dimensional, *feature subset selection*, pembuatan fitur, diskretisasi dan binerisasi, dan transformasi variabel. Pendekatan-pendekatan ini dapat dikelompokan ke dalam dua kategori, yaitu seleksi objek-objek data dan atribut-atribut untuk analisis atau pembuatan/perubahan atribut. Tujuan dari kedua kategori tersebut adalah untuk meningkatkan analisis *data mining* terhadap waktu, biaya dan kualitas.

2.3.1 Agregasi

Agregasi adalah mengkombinasikan dua atau lebih objek ke dalam sebuah objek tunggal. Sebagai contoh, *data set* yang berisi transaksi (objek data) yang mencatat penjualan produk harian di berbagai lokasi toko (Tabel 2.4). Salah satu cara untuk meng-agregasi transaksi untuk *data set* ini adalah mengganti semua transaksi dari toko-toko dengan sebuah transaksi tunggal. Agregasi ini mengurangi ratusan atau ribuan transaksi yang terjadi setiap hari pada satu toko tertentu ke transaksi harian tunggal, dan banyaknya objek data direduksi menjadi banyaknya toko.

Tabel 2.4 *Data set* yang berisi informasi tentang penjualan

Transaction ID	Item	Store Location	Date	Price	...
...	...	...	...	...	...
101123	Watch	Chicago	09/06/04	$25.99	
101123	Baterry	Chicago	09/06/04	$5.99	
101124	Shoes	Minneapolis	09/06/04	$75.00	
...	...	...	...	...	...

Atribut kualitatif seperti harga biasanya diagregasi dengan menentukan penjumlahannya atau nilai rata-ratanya. Atribut kualitatif seperti item, dapat dihilangkan atau diringkas sebagai himpunan dari semua item yang terjual pada lokasi tertentu.

Data dalam Tabel 2.4 dapat dipandang sebagai array multidimensi, dimana setiap atribut adalah sebuah dimensi. Dari sudut pandang ini, agregasi adalah proses mengeliminasi atribut seperti tipe item, atau mereduksi banyaknya nilai untuk atribut tertentu; misalnya mereduksi nilai yang mungkin untuk tanggal dari 365 hari ke 12 bulan. Tipe agregasi demikian umumnya digunakan dalam *Online Analytical Processing* (OLAP).

Agregasi dilakukan juga karena beberapa alasan, yaitu

1. *Data set* yang lebih kecil yang dihasilkan dari reduksi data memerlukan memory yang lebih sedikit dan waktu pemrosesan yang lebih cepat.
2. Agregasi dapat bertindak sebagai sebuah perubahan skala dengan memberikan sudut pandang data pada level tinggi daripada pada level yang rendah. Dalam contoh sebelumnya, agregasi pada bulan memberikan sudut pandang data bulanan daripada harian.
3. Perilaku kelompok objek atau atribut seringkali lebih stabil daripada objek atau atribut individual. Sebagai contoh, nilai rataan atau total dari nilai-nilai atribut kuantitatif memiliki variabilitas yang lebih kecil dibandingkan dengan nilai individual dari atribut tersebut.

Agregasi yang dilakukan memiliki kerugian yaitu potensial kehilangan detail data yang menarik.

2.3.2 *Sampling*

Sampling adalah pendekatan yang umum digunakan untuk menyeleksi sebuah *subset* dati objek data untuk dianalisis. Prinsip utama untuk *sampling* yang efektif dalam *data mining* adalah sebagai berikut: penggunaan *sample* akan

bekerja hampir seperti menggunakan keseluruhan *data set* jika *sample* adalah representatif. Sebuah *sample* adalah representatif jika *sample* tersebut memiliki sifat (dari kemenarikan) yang sama secara pendekatan seperti himpunan data awal. Jika rata-rata dari objek data adalah sifat yang menarik, maka sebuah *sample* adalah representatif jika *sample* tersbut memiliki rataan yang dekat dengan data awal.

Terdapat banyak teknik *sampling*. Bentuk sederhana dari *sampling* adalah simple random *sampling*. Untuk tipe *sampling* ini, terdapat peluang yang sama dalam menyeleksi item tertentu. Terdapat dua variasi dalam random *sampling*: (1) *sampling* tanpa pergantian, selama setiap item dipilih, item dipindahkan dari himpunan semua objek yang menyusun populasi, dan (2) *sampling* dengan pergantian. Dalam *sampling* dengan pergantian, objek tidak dipindahkan dari populasi selama mereka dipilih untuk *sample*. Dalam *sampling* dengan pergantian, objek yang sama dapat dipilih lebih dari satu kali. *Sample* yang dihasilkan dari kedua metode ini banyak begitu berbeda ketika ukuran *sample* relatif kecil dibandingkan dengan *data set*. Tetapi *sampling* dengan pergantian lebih sederhana dianalisis karena peluang dari seleksi suatu objek tetap konstan selama proses *sampling*.

Ketika populasi terdiri dari tipe-tipe yang berbeda dari objek, dengan jumlah yang besar dari objek, simple random *sampling* mungkin tidak dapat merepresentasikan tipe-tipe objek yang jarang muncul. Hal ini akan menimbulkan masalah ketika analisis memerlukan representasi dari seluruh tipe objek. Sebagai contoh, ketika membangun model klasifikasi untuk kelas yang jarang, kelas yang jarang tersebut perlu direpresentasikan dalam *sample*. Dengan demikian skema *sampling* harus mengakomodasi perbedaan frekuensi untuk item-item yang diperlukan. ***Stratified sampling*** adalah pendekatan yang dapat digunakan dalam kasus tersebut. Pendekatan ini mulai dengan kelompok objek yang telah ditetapkan. Dalam versi yang sederhana banyaknya objek digambarkan dari setiap grup walaupun grup-grup tersebut berbeda ukuran. Dalam variasi yang lain, banyaknya objek yang digambarkan dari setiap grup adalah proporsional terhadap ukuran dari grup tersebut.

Contoh 2.4 (*Sampling* dan Kehilangan Informasi)

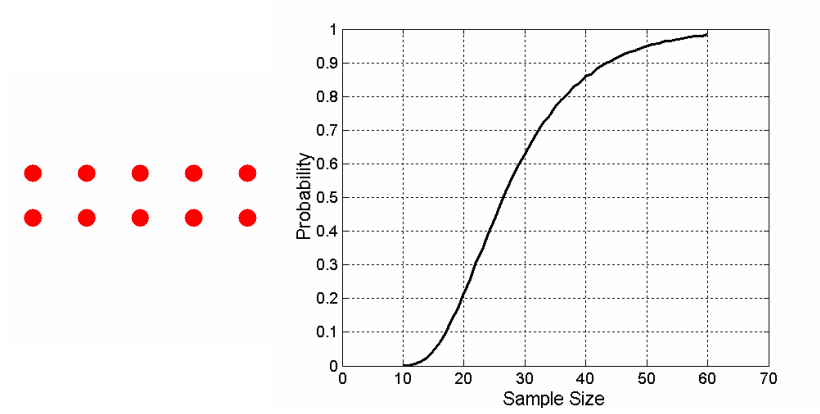
Ukuran *sample* perlu dipilih setelah teknik *sampling* ditetapkan. Ukuran *sample* yang terlalu besar meningkatkan peluang bahwa sebuah *sample* akan representatif, tetapi dapat menghilangkan keuntungan dari *sampling*. Sebaliknya jika ukuran *sample* terlalu kecil, maka pola dapat hilang atau pola keliru dapat terdeteksi. Gambar 2.5(a) menunjukkan *data set* yang berisi 8000 titik dua dimensi, sedangkan Gambar 2.5(b) dan (c) menunjukkan *sample* dari *data set* ini dengan ukuran berturut-turut 2000 dan 500. Hampir semua struktur dari *data set* ada dalam *sample* dengan ukuran 2000 titik, tetapi kebanyakan struktur hilang dalam *sample* dengan 500 titik.

(a) 8000 points (b) 2000 Points (c) 500 Points

Gambar 2.5 Contoh kehilangan struktur dalam *sampling*

Contoh 2.5 (Menentukan ukuran *sample* yang tepat)

Diberikan sebuah *data set* yang terdiri dari sejummlah kecil grup yang hampir sama ukurannya, tentukan sedikitnya satu titik yang representatif untuk setiap grup ini. Asumsikan bahwa objek dalam setiap grup sangat mirip dengan objek lain dalam grup yang sama, tetapi sangat tidak mirip dengan objek pada grup lain. Asumsikan pula bahwa terdapat sejumlah kecil grup, misalkan ada 10 grup. Gambar 2.6 menunjukkan sebuah himpunan dari cluster (grup) dimana titik-titik ini digambarkan.



Gambar 2.6 Menemukan titik-titik yang representatif dari 10 grup

Masalah ini dapat diselesaikan dengan menggunakan *sampling*. Salah satu pendekatan untuk menentukan sebuah *sample* kecil dari titik-titik data, hitung kemiripan antar titik-titik, kemudian bentuk kelompok titik-titik yang memiliki kemiripan yang tinggi. Himpunan titik yang diinginkan kemudian diperoleh dengan mengambil satu titik dari setiap grup-grup ini. Ukuran *sample* perlu ditentukan, dengan peluang yang tinggi, yang akan menjamin output yang diinginkan; bahwa sedikitnya satu titik akan diperoleh dari setiap cluster. Gambar 2.6 (b) menunjukkan peluang dari memperoleh satu objek dari 10 grup sebagaimana *sample* bergerak dari 10 ke 60. Dengan ukuran *sample* 20, terdapat kemungkinan kecil (20%) memperoleh sebuah *sample* yang meliputi ke-10 cluster. Pada saat ukuran *sample* 30, terdapat kemungkinan hampir 40% memperoleh *sample* yang tidak mengandung objek-objek dari ke-10 cluster.

Progressive Sampling

Adaptive atau *progressive sampling* digunakan ketika ukuran *sample* yang tepat cukup sulit ditentukan. Pendekatan ini bermula dari *sample* yang kecil, kemudian meningkatkan ukuran *sample* sampai *sample* dengan ukuran yang cukup diperoleh.

2.3.3 Reduksi Dimensionalitas

Data set dapat memiliki sejumlah besar fitur. Perhatikan sekumpulan dokumen dimana setiap dokumen direpresentasikan oleh sebuah vektor yang memiliki komponen berupa frekuensi kata yang muncul dalam dokumen. Dalam kasus demikian, terdapat ribuan atau bahkan puluhan ribu atribut (komponen), satu komponen untuk setiap kata dalam vocabulary.

Terdapat beberapa keuntungan dalam reduksi dimensionalitas, yaitu:

1. Keuntungan utama adalah banyak algoritme *data mining* bekerja lebih baik jika dimensi – jumlah atribut dalam data – lebih rendah. Hal ini dikarenakan reduksi dimensinalitas dapat mengeleiminati fitur-fitur yang tidak relevan dan mengurangi *noise*.
2. Reduksi dimensionalitas dapat memberikan model yang lebih mudah dimengerti karena model tersebut melibatkan lebih sedikit atribut.
3. Memungkinkan data lebih mudah di-visualisasi.
4. Jumlah waktu dan memory yang dibutuhkan oleh algoritme *data mining* berkurang dengan berkurangnya dimensi dari *data set*.

Istilah reduksi dimensionalitas sering dipakai untuk teknik-teknik yang mereduksi dimensi dari *data set* dengan membuat atribut-atribut baru yang merupakan kombinasi dari atribut-atribut lama. Reduksi dimensionalitas dengan memilih atribut baru yang merupakan *subset* dari atribut lama dikenal sebagai ***feature subset selection*** atau ***feature selection***.

Curse of dimensionality terjadi ketika dimensi meningkat, data menjadi lebih jarang dalam ruang yang ditempatinya. Untuk klasifikasi, *curse of dimensionality* dapat berarti bahwa tidak terdapat objek data yang cukup untuk membuat model yang menugaskan sebuah kelas untuk semua kemungkinan objek.

Terdapat beberapa pendekatan yang dapat digunakan dalam reduksi dimensionalitas, khususnya untuk data kontinu, Teknik-teknik ini menggunakan aljabar linier untuk memproyeksikan data dari ruang berdimensi tinggi ke ruang dengan dimensi yang lebih rendah. Teknik tersebut adalah:

1. *Principal Component Analysis* (PCA). PCA adalah teknik aljabar linier untuk atribut kontinu yang menemukan atribut baru (komponen prinsip) yang merupakan
   - Kombinasi linier dari atribut-atribut awal.
   - Ortogonal (tegak lurus) satu sama lain.
   - Mengambil jumlah variasi maksimum dalam data.

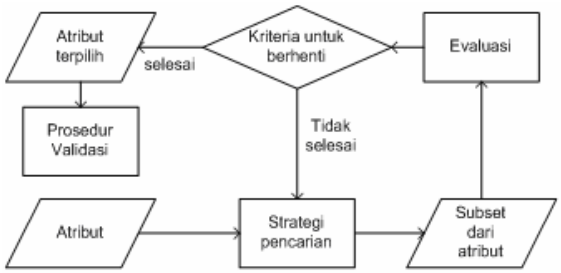
2. Singular Value Decomposition (SVD). SVD adalah teknik aljabar linier yang terkait dengan PCA (penjelasan lebih lanjut dapat dilihat pada Apendix B, Tan et al (2005)).

2.3.4 *Feature Subset Selection*

Terdapat tiga pendekatan dalam *feature subset selection*, yaitu:

- Pendekatan *embedded*. *Feature selection* muncul sebagai bagian dari algoritme *data mining*. Selama operasi dari algortime *data mining*, algoritme dengan sendirinya menentukan atribut yang mana yang akan digunakan dan atribut yang mana yang diabaikan. Algoritme yang bekerja dengan pendekatan ini adalah algoritme klasifikasi.
- Pendekatan filter. Fitur diseleksi sebelum algoritme *data mining* bekerja, dengan menggunakan pendekatan yang tidak tergantung pada pekerjaan *data mining*.
- Pendekatan *wrapper*. Medote ini menggunakan algortime *data mining* target sebagai *black box* untuk menentukan *subset* atribut yang paling baik, tanpa menghitung semua *subset* yang mungkin.

Proses *feature selection* terdiri dari empat bagian: ukuran untuk mengevaluasi *subset*, strategi pencarian yang mengontrol pembangunan dari *subset* baru, kriteria berhenti dan prosedur validasi. Keterkaitan antarbagian tersebut dapat dilihat pada Gambar 2.7.



Gambar 2.7 Diagram alir proses *feature selection*

2.3.5 Pembuatan Fitur (*Feature Creation*)

Seringkali diperlukan membuat sekumpulan atribut baru dari atribut-atribut awal yang menangkap informasi penting dalam *data set* dengan lebih efektif. Terdapat tiga metodologi yang terkait dengan pembuatan atribut baru, yaitu:

1. *Feature extraction*: pembuatan sekumpulan fitur yang baru dari data mentah awal.
2. Pemetaan data ke ruang yang baru.
3. Konstruksi fitur. Kadangkala fitur dalam *data set* awal memiliki informasi yang penting, tetapi tidak sesuai untuk algoritme *data mining*. Dalam kasus ini, satu atau lebih fitur yang dikonstruksi dari fitur awal dapat lebih berguna daripada fitur awal.

2.3.6 Diskretisasi dan Binerisasi

Kadangkala algoritme *data mining*, khususnya algoritme kalsifikasi tertentu, memerlukan data dalam bentuk atribut kategori. Sedangkan algoritme untuk menemukan pola asosiasi memerlukan data dalam atribut biner. Dengan demikian diperlukan transformasi dari atribut kontinu ke atribut kategori (diskretisasi), dan atribut kontinu juga atribut diskret perlu ditransformasikan ke dalam atribut biner (binerisasi). Di samping itu, jika atribut kategori memiliki sejumlah besar nilai (kategori), atau memiliki nilai yang jarang muncul, maka diperlukan reduksi jumlah kategori dengan mengkombinasikan beberapa nilai.

Binerisasi

Teknik sederhana untuk mentransformasi atribut kategori ke atribut biner adalah sebagai berikut: jika terdapat m nilai kategori, maka tetapkan setiap nilai awal untuk sebuah integer dalam interval [0, m−1]. Jika atribut adalah ordinal, maka urutan harus dijaga oleh penugasan tersebut. Jika atribut awal direpresentasikan dengan integer, binerisasi diperlukan jika integer tersebut tidak di dalam interval [0, m−1]. Selanjutnya konversikan setiap m integer ini ke bilangan biner. Karena $n = \lceil \log_2(m) \rceil$ digit biner diperlukan untuk merepresentasikan integer-integer ini, representasikan bilanga-bilangan biner ini dengan menggunakan n atribut biner. Sebagai ilustrasi, sebuah variabel kategori yang memiliki lima nilai {awful, poor, OK, good, great} akan memerlukan tiga variabel biner x1, x2, dan x3. Konversi ditunjukkan dalam Tabel 2.5.

Tabel 2.5 Konversi atribut kategori ke tiga atribut biner

Nilai Kategori	Nilai integer	x1	x2	x3
Awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

Transformasi demikian dapat menyebabkan komplikasi seperti adanya hubungan yang tidak diinginkan diantara atribut-atribut yang ditransformasi. Sebagai contoh dalam Tabel 2.5, atribut x2 dan x3 dikolerasikan karena nilai good dikodekan menggunakan kedua atribut tersebut.

Analisis asosiasi memerlukan atribut biner yang asimetrik, dimana hanya adanya atribut (nilai = 1) yang penting. Untuk masalah tersebut, diperlukan satu atribut biner untuk setiap nilai kategori sepeti dalam Tabel 2.6.

Tabel 2.6 Konversi atribut kategori ke lima atribut biner asimetrik

Nilai Kategori	Nilai integer	x1	x2	x3	x4	x5
Awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

Jika jumlah atribut yang dihasilkan terlalu besar, maka teknik yang dijelasakan sebelumnya untuk mereduksi banyaknya nilai kategori dapat digunakan sebelum binerisasi.

Diskretisasi Atribut Kontinu

Transformasi dari atribut kontinu ke atribut kategori melibatkan dua pekerjaan utama, yaitu menentukan berapa banyaknya kategori yang akan dimiliki dan menentukan bagaimana memetakan nilai-nilai dari variabel kontinu ke kategori-kategori ini. Pada langkah pertama, setelah semua nilai atribut kontinu disimpan, mereka dibagi kedalam n interval dengan menentukan n−1 ***split point***. Selanjutnya semua nilai-nilai dalam satu interval dipetakan ke dalam nilai kategori yang sama. Dengan demikian, masalah disktetisasi adalah menentukan berapa banyak *split point* dipilih dan dimana menempatkannya. Hasilnya dapat direpresentasikan juga dalam bentuk interval $\{(x_0, x_1], (x_1, x_2], ..., (x_{n-1}, x_n)\}$, dimana x_0 dapat berupa $+\infty$ dan x_n dapat berupa $-\infty$, atau $x_0 < x_1 < x_2 < .. < x_{n-1} < x_n$.

2.3.7 Transformasi Variabel

Transformasi variabel diaplikasikan pada seluruh nilai pada sebuah variabel. Dengan kata lain, untuk setiap objek, transformasi dilakukan untuk nilai dari variabel untuk objek tersebut. Sebagai contoh, jika hanya besaran dari variabel yang penting, maka nilai variabel dapat ditransformasikan dengan menetapkan nilai absolut. Terdapat dua bentuk transformasi variabel, yaitu:

1. Transformasi fungsi-fungsi sederhana

   Dalam transformasi ini, fungsi-fungsi matematika sederhana dapat diaplikasikan untuk setiap nilai individual. Jika x adalah sebuah variabel, maka contoh dari transformasi ini adalah x^k, log x, e^x, sin x, 1/x, $x^{0.5}$ dan |x|.

   Transformasi perlu dilakukan secara hati-hati karena dapat merubah sifat dari data. Sebagai contoh, transformasi 1/x akan mengurangi besaran nilai yang sama dengan 1 atau lebih besar, tetapi meningkatkan besaran nilai yang berada pada range 0 sampai dengan 1. Sebagai ilustrasi, {1, 2, 3} ditansformasikan ke {1, ½, 1/3} sedangkan {1, ½, 1/3} ditansformasikan ke {1, 2, 3}. Dengan demikian, untuk semua himpunan nilai, hasil transformasi 1/x membalik urutan nilai.

   Untuk membantu penjelaskan pengaruh dari sebuah transformasi, beberapa hal perlu diperhatikan, yaitu:

   - Apakah urutan dari data perlu tetap dijaga?
   - Apakah tranformasi akan diaplikasikan ke semua nilai, kecuali bilangan negatif atau 0?
   - Apa pengaruh transformasi pada nilai diantara 0 dan 1?

2. Normalisasi atau Standarisasi

   Bentuk lain dari transformasi variabel adalah standarisasi atau normalisasi variabel. Tujuannya adalah untuk membuat sebuah himpunan dari nilai yang memiliki sifat tertentu. Salah satu contoh adalah standarisasi variabel dalam

statistika. Jika $\bar{x}$ adalah mean atau rataan dari nilai atribut dan s_x adalah deviasi standarnya, maka transformasi $x' = (x - \bar{x})/ s_x$ membuat variabel baru yang memiliki rataan 0 dan deviasi standar 1.

2.4 Ukuran Kemiripan dan Ketidakmiripan

Kemiripan dan ketidakmiripan merupakan hal yang penting, karena sejumlah teknik *data mining* seperti clustering, *nearest neighbor classification*, dan deteksi anomali. Istilah ***proximity*** digunakan untuk merujuk kemiripan atau ketidakmiripan.

Secara formal, kemiripan antara dua objek adalah ukuran numerik dari derajat dimana dua objek adalah serupa. Kemiripan adalah paling tinggi untuk pasangan objek yang paling serupa. Kemiripan biasanya dinyatakan oleh bilangan tak negatif dan umumnya berada diantara 0 (tidak ada kemiripan) dan 1 (sangat mirip). Ketakmiripan antara dua objek adalah ukuran numerik dari derajat dimana kedua objek tersebut adalah berbeda. Ketakmiripan adalah paling rendah untuk pasangan objek yang lebih mirip. Biasanya istilah jarak (distance) digunakan sebagai sinonim untuk ketakmiripan. Ketakmiripan kadang-kadang berada dalam interval [0, 1], tapi umumnya berada dalam range dari 0 sampai ∞.

Transformasi sering digunakan untuk mengkonversi kemiripan ke ketakmiripan, dan sebaliknya, atau mentransformasi ukuran *proximity* ke dalam range tertentu, misalnya [0, 1]. Sebagai contoh, nilai kemiripan berada pada range 1 sampai dengan 10, sedangkan algoritme atau *tool* tertentu hanya bekerja pada range [0, 1].

Seringkali ukuran *proximity*, khususnya kemiripan, didefinisikan dan ditransformasikan ke sebua nilai pada interval [0, 1]. Jika kemiripan objek berada pada range 1 (tidak mirip) ke 10 (sangat mirip), kita dapat membuat nilai-nilai tersebut berada dalam interval [0, 1] dengan menggunakan transformasi s' = (s–1)/9, dimana s dan s'berturut-turut adalah nilai kemiripan awal dan nilai kemiripan baru. Secara umum, transformasi kemiripan ke interval [0, 1] diberikan oleh pernyataan s'= (s – min_s)/(max_s – min_s), dimana max_s dan min_s berturut-turut adalah nilai kemiripan maksimum dan minimum. Sedikian juga, ukuran ketakmiripan dengan range berhingga dapat dipetakan ke interval [0, 1] dengan menggunakan formula d'= (d – min_d)/(max_d – min_d).

Kemiripan juga dapat ditranformasikan ke ketakmiripan. Jika kemiripan berada pada interval [0, 1] maka ketakmiripan dapat didefinisikan sebagai d = 1 – s (s = 1 – d). Pendekatan lainnya adalah mendefinisikan kemiripan sebagai negasi dari ketidakmiripan, atau sebaliknya. Sebagai ilustrasi, ketakmripan 0, 1, 10 dan 100 berturut-turut dapat ditransformasikan ke dalam kemiripan 0, –1, –10 dan –100.

Kemiripan yang dihasilkan dari transformasi negasi tidak dibatasi hanya pada interval [0, 1], jika diinginkan, transformasi seperti $s = \frac{1}{d+1}$, $s = e^{-d}$ atau $s = 1 - \frac{d - \min\_d}{\max\_d - \min\_d}$ dapat digunakan. Untuk transformasi $s = \frac{1}{d+1}$, ketidakmiripan 0, 1, 10 dan 100 ditransformasikan ke 1, 0.5, 0.09, 0.01; untuk

tranformasi $s = e^{-d}$ ditranformsikan ke 1.00, 0.37, 0.00, 0.00; untuk transformasi $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$ ditransformasikan ke 1.00, 0.99, 0.00, 0.00.

2.4.1 Kemiripan dan Ketidakmiripan antara Atribut Sederhana

Proximity dari objek dengan sebuah nilai dari atribut didefinisikan dengan mengkombinasikan *proximity* dari atribut-atribut individual. Perhatikan sebuah objek yang dijelaksan oleh satu atribut nominal. Karena atribut nominal hanya memberikan informasi tentang perbedaan dari objek, maka dapat dinyatakan bahwa dua objek memiliki nilai yang sama atau tidak. Dalam hal ini, kemiripan hanya didefinisikan sebagai 1 jika nilai-nilai atribut tersebut sama atau 0 untuk selainnya. Sedangkan ketakmiripan didefinisikan dengan cara yang berlawanan, 0 jika nilai-nilai atribut tersebut sama dan 1 selainnya.

Untuk objek dengan atribut ordinal tunggal, perlu diperhatikan informasi mengenai urutan. Perhatikan atribut yang mengukur kualitas dari sebuah produk, misalkan permen, pada skala {poor, fair, OK, good, wonderful}. Dapat dinyatakan bahwa produk P1, yang diberi nilai wonderful, akan dikatakan lebih dekat ke produk P2, yang diberi nilai good, daripada ke produk P3, yang diberi nilai OK. Dalam hal ini, nilai atribut ordinal sering dipetakan ke integer yang berurutan, dimulai dari 0 atau 1, sebagai contoh {poor = 0, fair = 1, OK = 2, good = 3, wonderful = 4}. Maka d(P1, P2) = 3 – 2 = 1 atau, jika diinginkan ketakmiripan jatuh pada interval [0, 1], ditetapkan d(P1, P2) = (3 – 2)/4 = 0.25. Kemiripan untuk atribut ordinal dapat diperoleh dari pernyataan s = 1 – d.

Untuk atribut interval atau rasio, ketakmiripan antar dua objek adalah nilai absolut dari beda nilai-nilai tersebut. Dalam hal ini ketakmiripan berada dalam interval 0 sampai ∞. Kemiripan dari atribut interval atau rasio dapat ditentukan dengan transformsi dari kemiripan ke ketidakmiripan. Tabel 2.7 meringkas cara menentukan nilai kemiripan dan ketidakmiripan untuk beberapa tipe atribut.

Tabel 2.7 Kemiripan dan ketakmiripan untuk atribut sederhana

Tipe atribut	Ketakmiripan	Kemiripan
Nominal	$d = \begin{cases} 0, \text{if } x = y \\ 1, \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1, \text{if } x = y \\ 0, \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n - 1)$ (nilai dipetakan ke integer 0 sampai dengan n – 1, dimana n adalah banyaknya nilai.	$s = 1 - d$
Interval atau rasio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

2.4.2 Ketakmiripan dan Kemiripan antara Objek Data

Salah satu jenis ketidakmiripan adalah jarak (*distance*). **Jarak Euclidean**, d, antara dua titik, **x** dan **y**, dalam ruang dimensi satu, dua, tiga, atau lebih tinggi, diberikan oleh formula berikut:

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}\ , \qquad (2.1)$$

dimana n adalah banyaknya dimensi, x_k dan y_k berturut-turut adalah atribut (komponen) ke-k dari x dan y. Jika d(**x**, **y**) adalah jarak antara dua titik, **x** dan **y**, maka sifat berikut dipenuhi:

1. *Positivity*
   a. d(**x**, **y**) ≥ 0 untuk semua **x** dan **y**,
   b. d(**x**, **y**) = 0 untuk semua **x** = **y**.
2. *Symmetry*
   d(**x**, **y**) = d(**y**, **x**) untuk semua **x** dan **y**.
3. *Triangle Inequality*
   d(**x**, **z**) ≤ d(**x**, **y**) + d(**y**, **z**), untuk semua **x**, **y** dan **z**.

Ukuran-ukuran yang memenuhi ketiga sifat ini dikenal sebagai **metrics**.

Untuk kemiripan, sifat *triangle inequality* tidak dipenuhi. Jika s(**x**, **y**) adalah kemiripan antara titik **x** dan **y**, maka sifat kemiripan adalah sebagai berikut:

1. s(**x**, **y**) = 1 hanya jika **x** = **y**. (0 ≤ s ≤ 1)
2. s(**x**, **y**) = s(**y**, **x**) untuk semua **x** dan **y** (*symmetry*).

2.4.3 Contoh-contoh Ukuran *Proximity*

Berikut adalah contoh-contoh spesifik dari ukuran-ukuran kemiripan dan ketidakmiripan.

Ukuran Kemiripan untuk Data Biner

Ukuran kemiripan antara objek yang hanya mengandung atribut-atribut biner dinamakan koefisien kemiripan, dan biasanya memiliki nilai antara 0 dan 1. Nilai 1 menunjukkan bahwa kedua objek tersebut adalah sangat mirip, sedangkan nilai 0 menunjukkan bahwa kedua objek tersebut sangat tidak mirip.

Misalkan **x** dan **y** adalah dua objek yang terdiri dari n atribut biner. Perbandingan dari kedua objek tersebut, yaitu dua vektor biner, memberikan empat kuantitas (frekuensi) berikut:

f_{00} = banyaknya atribut dimana **x** adalah 0 dan **y** adalah 0

f_{01} = banyaknya atribut dimana **x** adalah 0 dan **y** adalah 1

f_{10} = banyaknya atribut dimana **x** adalah 1 dan **y** adalah 0

f_1 = banyaknya atribut dimana **x** adalah 1 dan **y** adalah 1

Simple Matching Coefficient. Salah satu koefisien kemiripan yang banyak digunakan adalah *simple matching coefficient* (SMC), yang didefinisikan sebagai

$$\text{SMC} = \frac{\text{banyaknya nilai atribut yang sesuai (match)}}{\text{banyaknya nilai atribut}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}} \qquad (2.2)$$

SMC dapat digunakan untuk menentukan mahasiswa-mahasiswa yang memiliki jawaban yang mirip pada sebuah ujian yang hanya berisi pertanyaan benar/salah. Hal ini dikarenakan SMC menghitung baik **x** dan **y** sama-sama 0 (ketidakhadiran) dan **x** dan **y** sama-sama 1 (kehadiran).

Koefisien Jaccard. Anggap bahwa **x** dan **y** adalah objek data yang merepresentasikan dua baris (dua transaksi) dari sebuah matriks transaksi. Jika setiap atribut biner asimetrik berkaitan dengan sebuah item dalam sebuah toko, maka 1 menunjukkan bahwa item tersebut dibeli, sedangkan 0 menunjukkan bahwa item tersebut tidak dibeli. Karena banyaknya produk yang tidak dibeli oleh pelanggan jauh lebih sedikit dari banyaknya produk yang dibeli, maka ukuran kemiripan seperti SMC akan menyatakan bahwa semua transaksi sangat mirip. Koefisien Jaccard seringkali digunakan untuk menangani objek-objek yang terdiri dari atribut-atribut biner asimetrik. Koefisien Jaccard, seringkali dinyatakan oleh simbol J, diberikan oleh formula berikut

$$J = \frac{\text{banyaknya kehadiran yang sesuai (match)}}{\text{banyaknya nilai atribut yang tidak memasukkan 00}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \quad (2.3)$$

Contoh 2.6 (*Simple Matching Coefficient* dan Koefisien Jaccard)

Contoh ini menilustrasikan perbedaan antara pengukuran *Simple Matching Coefficient* dan koefisien Jaccard. Tentukan nilai SMC dan J untuk dua vektor biner berikut:

x = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)

y = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)

Berdasarkan kedua vektor tersebut diperoleh:

f_{01} = 2, banyaknya atribut dimana **x** adalah 0 dan **y** adalah 1

f_{10} = 1, banyaknya atribut dimana **x** adalah 1 dan **y** adalah 0

f_{00} = 7, banyaknya atribut dimana **x** adalah 0 dan **y** adalah 0

f_{11} = 0, banyaknya atribut dimana **x** adalah 1 dan **y** adalah 1

Sehingga nilai SMC = 0.7, sedangkan nilai J = 0.

Kemiripan Kosinus

Dokumen seringkali direpresentasikan sebagai vektor, dimana setiap atribut merepresentasikan frekuensi dimana istilah atau kata tertentu muncul dalam dokumen. Walaupun dokumen-dokumen memiliki ribuan atau bahkan puluhan ribu atribut (istilah), setiap dokumen adalah *sparse* (jarang) karena memiliki atribut tak nol yang relatif sedikit. Dengan demikian, seperti dalam data transaksi, kemiripan tidak tergantung pada banyaknya nilai 0. Jika kecocokan 0-0 dihitung, sebagian besar dokumen akan sangat mirip dengan dokumen-dokumen lain. Dengan demikian, ukuran kemiripan dokumen perlu mengabaikan kecocokan 0-0 seperti halnya dalam koefisien Jaccard, tetapi ukuran tersebut tentunya harus mempu menangani vektor-vektor non-biner.

Kemiripan kosinus adalah salah satu ukuran yang paling banyak digunakan untuk mengukur kemiripan dokumen. Jika **x** dan **y** adalah dua vektor dokumen, maka

$$\cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}, \qquad (2.3)$$

Dimana . menunjukkan perkalian titik pada vektor, $\mathbf{x}.\mathbf{y} = \sum_{k=1}^{n} x_k y_k$, dan $\|\mathbf{x}\|$ adalah panjang dari vektor **x**, $\|\mathbf{x}\| = \sqrt{\sum_{k=1}^{n} x_k{}^2} = \sqrt{\mathbf{x}.\mathbf{x}}$.

Contoh 2.7 (Kemiripan kosinus dari dua vektor dokumen)

Diberikan dua vektor dokumen berikut:

x = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)

y = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)

$\mathbf{x}.\mathbf{y}$ = 5, $\|\mathbf{x}\|$ = 6.48, $\|\mathbf{y}\|$ = 2.24, sehingga cos(**x**,**y**) = 0.31.

Perluasan Koefisien Jaccard (Koefisien Tanimoto)

Perluasan koefisien Jaccard dapat digunakan untuk data dokumen. Perluasan ini dikenal sebagai koefisien Tanimoto. Koefisien ini dinotasikan dengan EJ, didefinisikan oleh persamaan berikut:

$$\mathrm{EJ}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x}.\mathbf{y}} \qquad (2.4)$$

Korelasi

Korelasi antara dua objek data yang memiliki variabel biner atau kontinu adalah ukuran dari hubungan linier antara atribut-atribut dari objek. Koefisien korelasi Pearson antara dua objek data, **x** dan **y**, didefinisikan oleh persamaan berikut:

$$\mathrm{corr}(\mathbf{x},\mathbf{y}) = \frac{\mathrm{covariance}(\mathbf{x},\mathbf{y})}{\mathrm{standard\_deviation}(\mathbf{x}) * \mathrm{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x s_y} \qquad (2.5)$$

dimana *covariance* dan *standard deviation* diberikan oleh formula berikut:

$$\mathrm{covariance}(\mathbf{x},\mathbf{y}) = s_{xy} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})(y_k - \bar{y}) \qquad (2.6)$$

$$\mathrm{Standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})^2}$$

$$\mathrm{Standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k - \bar{y})^2}$$

$\bar{x}$ dan $\bar{y}$ berturut-turut adalah rataan dari **x** dan **y**.

2.4.4 Isu-isu dalam Perhitungan *Proximity*

Berikut adalah isu-isu penting yang terkait dalam perhitungan ukuran-ukuran *proximity*:

Standarisasi dan korelasi untuk ukuran-ukuran jarak

Isu penting dalam pengukuran jarak adalah bagaimana menangani situasi ketika atribut-atribut tidak memiliki range nilai yang sama (situasi ini seringkali dikatakan sebagai variabel-variabel yang memiliki skala yang berbeda). Isu terkait lainnya adalah bagaimana menghitung jarak ketika terdapat korelasi antara beberapa atribut. Generalisasi dari jarak Euclidean, jarak Mahalanobis, berguna ketika atribut-atribut berkorelasi, memiliki range nilai yang berbeda dan distribusi dari data adalah mendekati Gaussian (normal). Secara khusus, jarak Mahalonobis antara dua objek (vektor) **x** dan **y** didefiniskan sebagai

$$\text{mahalanobis}(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})\sum^{-1}(\mathbf{x}-\mathbf{y})^{\mathrm{T}} \qquad (2.7)$$

dimana $\sum^{-1}$ adalah inverse dari matriks covariance dari data.

Mengkombinasikan kemiripan untuk atribut heterogen

Dalam definisi-definisi kemiripan yang telah dibahas, diasumsikan bahwa semua atribut memiliki tipe yang sama. Pendekatan yang lebih umum diperlukan ketika atribut-atribut yang ada memiliki tipe yang berbeda. Salah satu pendekatan tersebut adalah dengan menghitung kemiripan masing-masing atribut secara terpisah menggunakan Tabel 2.5, kemudian mengkombinasikan kemiripan-kemiripan ini menggunakan metode yang menghasilkan kemiripan antara 0 dan 1. Secara khusus, kemiripan keseluruhan didefinisikan sebagai rataan dari semua kemiripan-kemiripan atribut individual.

Sayangnya pendekatan ini tidak bekerja dengan baik jika beberapa atribut adalah atribut asimetrik. Cara yang paling mudah untuk menyelesaikan permasalahan ini adalah dengan menghilangkan atribut asimetrik dari perhitungan kemiripan ketika nilainya adalah 0 untuk kedua objek yang kemiripannya sedang dihitung. Pendekatan serupa juga dapat digunakan untuk menangani nilai-nilai yang hilang.

Algoritme 2.1 adalah algoritme yang efektif untuk perhitungan seluruh kemiripan antara dua objek, **x** dan **y**, dengan tipe atribut yang berbeda.

Algoritme 2.1 Kemiripan objek heterogen

1. Untuk atribut ke-k, hitung kemiripan, $s_k(\mathbf{x},\mathbf{y})$, dalam range [0, 1].
2. Definisikan sebuah variabel indikator, δ_k, untuk atribut ke-k sebagai berikut
   - $\delta_k = 0$, jika atribut ke-k adalah atribut asimetrik dan kedua objek memiliki nilai 0, atau jika salah satu dari objek-objek tersebut memiliki sebuah nilai yang hilang untuk atribut ke-k.
   - $\delta_k = 1$, selainnya.
3. Hitung kemiripan keseluruhan diantara dua objek menggunakan formula berikut:

$$\text{similarity}(\mathbf{x},\mathbf{y}) = \frac{\sum_{k=1}^{n} \delta_k s_k(\mathbf{x},\mathbf{y})}{\sum_{k=1}^{n} \delta_k} \quad (2.8)$$

Dalam persamaan (2.8), semua atribut diperlakukan sama ketika menghitung kemiripan. Forluma tersebut tidak dapat digunakan ketika terdapat beberapa atribut yang lebih penting untuk definisi kemiripan (atau ketidakmiripan) daripada atribut yang lain. Untuk itu formula *proximity* dapat dimodifikasi dengan memberikan membobot kontribusi setiap atribut. Jika bobot tersebut dinyatakan sebagai w_k , dimana penjumlahan w_k untuk semua k adalah 1, maka formula (2.8) menjadi

$$\text{similarity}(\mathbf{x},\mathbf{y}) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k(\mathbf{x},\mathbf{y})}{\sum_{k=1}^{n} \delta_k} \quad (2.9)$$

Definisi jarak Euclidean juga dapat dimodifikasi sebagai berikut:

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{n} w_k (x_k - y_k)^2} \quad (2.10)$$

Penutup – Soal Latihan

Tugas Individu

Jawablah pertanyaan berikut secara singkat dan jelas.

1. Sebutkan dan jelaskan tipe-tipe atribut
2. Sebutkan dan jelaskan tipe-tipe *data set*
3. Sebutkan dan jelaskan strategi-strategi untuk menangani data yang hilang
4. Apa yang dimaksud dengan agregasi, apa keuntungan melakukan agregasi
5. Jelaskan perbedaan *noise* dan *outlier*
6. Apa yang dimaksud ukuran kemiripan dan ketidakmiripan antara atribut dan ukuran kemiripan dan ketidakmiripan antara objek data
7. Sebutkan ukuran-ukuran kemiripan untuk data biner
8. Untuk vektor-vektor berikut, **x** dan **y**, hitunglah ukuran kemiripan atau ukuran jarak yang ditetapkan.
   a. **x** = (1, 1, 1, 1), **y** = (2, 2, 2, 2), kosinus, korelasi, Euclidean
   b. **x** = (0, 1, 0, 1), **y** = (1, 0, 1, 0), kosinus, korelasi, Euclidean, Jaccard

c. **x** = (0, –1, 0, 1), **y** = (1, 0, –1, 0), kosinus, korelasi, Euclidean

d. **x** = (1, 1, 0, 1, 0, 1), **y** = (1, 1, 1, 0, 0, 1), kosinus, korelasi, Jaccard

e. **x** = (2, –1, 0, 2, 0, –3), **y** = (–1, 1, –1, 0, 0, –1), kosinus, korelasi

Tugas Kelompok

Diskusikan dengan kelompok anda jawaban untuk pertanyaan-pernyataan berikut.

1. Klasifikasikan atribut-atribut berikut
   - sebagai atribut biner, diskret, atau kontinu
   - sebagai atribut kualitatif (nominal atau ordinal) atau kuantitatif (interval atau rasio)

   Sebagai contoh: umur dalam tahun, jawaban: diskret, kuantitatif, rasio

   a. Waktu dalam AM atau PM
   b. Kecerahan yang diukur oleh *light meter*
   c. Kecerahan yang diukur oleh pendapat orang
   d. Sudut yang diukur dalam derajar antara 0 dan 360
   e. Medali emas, perak dan perunggu sebagai penghargaan dalam olimpiade
   f. Banyaknya pasien di rumah sakit
   g. Nomor ISBN dari buku
   h. Jarak dari pusat kota ke kampus

   Berilah penjelasan terhadap jawaban kelompok anda.

2. Berikan sedikitnya 3 contoh atribut yang termasuk ke dalam
   - atribut biner, diskret, atau kontinu
   - atribut kualitatif (nominal atau ordinal) atau kuantitatif (interval atau rasio)

   Berilah penjelasan terhadap jawaban kelompok anda.