K1 05/02/2018

enroll: damin1718
- Kuis: P1 & P10
- Ujian: Opensheet
- UTS 30%
  UAS 25%
  Praktikum 25%
  Tugas Akhir 20%
  - 5% laporan progress
    5% laporan akhir
    10% presentasi final
  - a. Clustering & deteksi anomali
    b. Teknik klasifikasi
    c. Association rule mining
    d. Teknik lanjutan: text, web, spatio-temporal data, sequential pattern } mining

Data: min 1000 sampel
10 atribut

Pengantar DATA MINING

Masalah eksplorasi data
→ menumpuknya data dalam database, data warehouse, dan penyimpanan lainnya.
→ Solusi: Data Warehousing & Data Mining
(& Online Analytical Processing (OLAP)) — ekstraksi pengetahuan (aturan, pola / kendala) dari basis data berukuran besar.
sistem penyimpanan terpusat dari 1/lebih sumber yang berbeda untuk reporting dan analisis data

Mengapa Menambang Data?
1. Sudut pandang komersil
   - telah byk data yg dikumpulkan — web data, e-commerce, e-banking, grocery stores, bank/credit card transactions
   - teknologi komputer mjd lbh murah dan powerful
   - tekanan kompetisi semakin kuat
     : menyediakan layanan yg lbh baik, eg: customer Relationship Management
2. Sudut pandang keilmuwan
   - data dikumpulkan dan disimpan dgn kecepatan tinggi (GB/hour)
   - teknik tradisional tdk cukup utk analisis data
   - Data Mining membantu ilmuwan dlm: klasifikasi & segmentasi data, pemodelan, clustering
   - remote sensor pada satellite
   - microarray pembangkit gene expression data
   - data spasial dalam GIS
   - simulasi keilmuwan

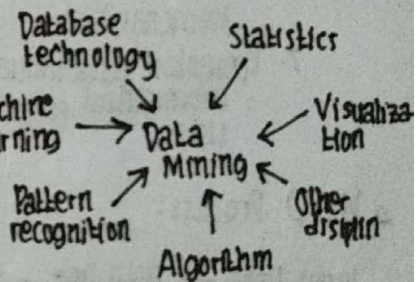
Evolution of Sciences: New Data Science Era
1990-now: Data Science
- data melimpah
- secara ekonomis: dapat disimpan dan di manage
- diakses banyak orang mel. internet
- meningkat secara linear

Pengertian Data Mining
Data Mining (knowledge discovery in database):
ekstraksi informasi/pola yg menarik (non-trivial, implicit, previously unknown dan potentially useful) dlm basis data berukuran besar.

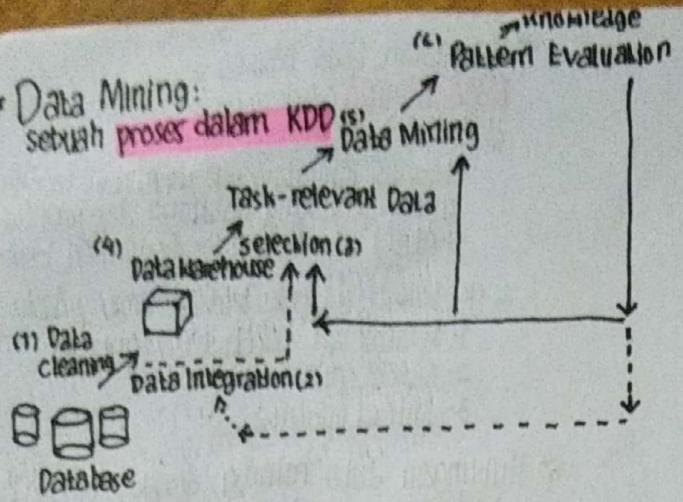
└ yg bukan termasuk data mining task:
1. Pemrosesan (deductive) query.
2. Sistem pakar
3. Sistem informasi

Hubungan data mining dengan bidang lain
└ berkaitan erat dengan bidang machine learning/AI, pattern recognition, statistika, dan sistem database.
└ Teknik tradisional mjd tdk sesuai karena:
- data besar
- tingginya dimensi data
- data heterogen, terdistribusi

Data Mining: Multi disiplin

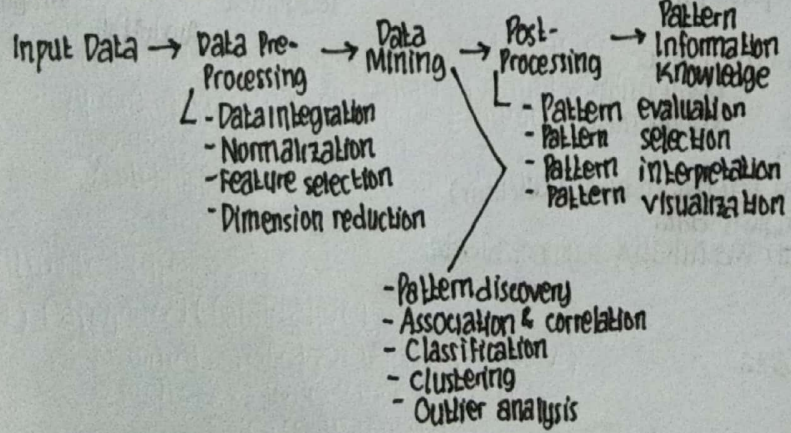


* Data Mining:
setuah proses dalam KDD

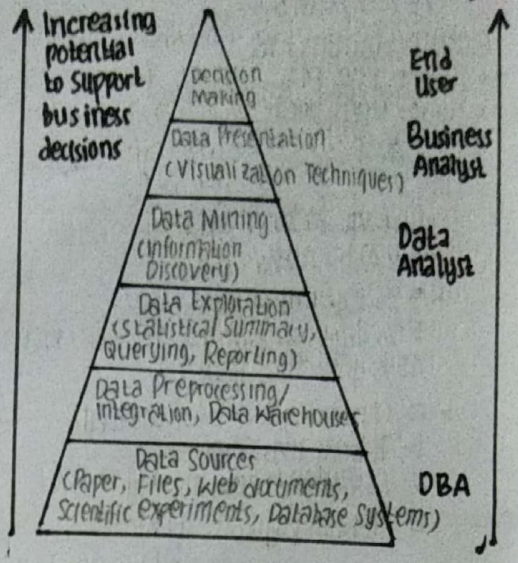


1. Pembersihan data
   : menghilangkan noise dan data yg tdk konsisten.
2. Pengintegrasian data
   : data digabungkan dari berbagai sumber
3. Seleksi data
   : data yg relevan dgn proses analisis diambil dari database.
4. Transformasi data
   : data ditransformasikan / digabungkan ke dlm bentuk yg sesuai utk di-mine dgn cara dilakukan peringkasan / operasi agregasi.
5. Data mining
   : proses penting dlm KDD dimana metode2x cerdas diaplikasikan utk mengekstrak pola2x data.
6. Evaluasi pola
   : mengidentifikasi pola2x yg menarik yg merepresentasikan pengetahuan berdasarkan suatu ukuran kemenarikan.
7. Presentasi pengetahuan
   : representasi pengetahuan yg telah digali kepada user.

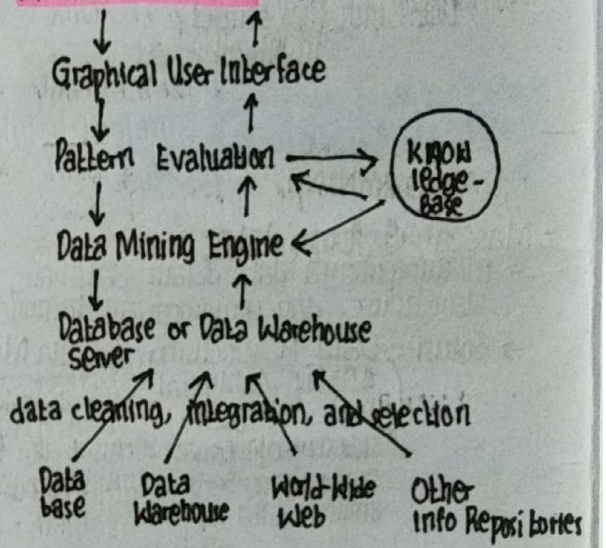
* KDD Process:



* Data Mining dalam Business Intelligence



* Arsitektur Sistem Data Mining



1. Basis data, data warehouse, / tempat penyimpanan informasi lainnya
2. Basis data dan data warehouse server
   : bertanggungjawab dlm pengambilan data yg relevan berdasar permintaan pengguna.
3. Basis pengetahuan
   : domain knowledge utk memandu pencarian/ evaluasi pola2x yg dihasilkan.
4. Data mining engine
   : terdiri dari modul2x fungsional data mining, seperti karakterisasi, asosiasi, klasifikasi, dan analisis cluster.
5. Modul evaluasi pola
   : mengg. ukuran2x kemenarikan dan berinteraksi dgn modul data mining dlm pencarian pola2x menarik
6. Antarmuka pengguna grafis
   : media komunikasi dgn pengguna dan sistem data mining.

Data yg bisa ditambang:
- basis data relasional
- Data warehouse
- Basis data transaksional (eg: kasir merekam transaksi)
- Tempat penyimpanan data lainnya:
  1. Basis data object-oriented dan basis data object-relational
  2. Basis data spatial
  3. Data time-series dan data temporal
  4. Sequence data (incl. bio-sequences)
  5. Data teks dan basis data multimedia
  6. WWW

Tugas2x dalam Data Mining
1. Metode Prediksi
   : menggu. beberapa variabel (atribut) utk memprediksi nilai yg blm diketahui/nilai yg akan datang dr variabel (atribut) lain.
2. Metode Deskripsi
   : menemukan pola2x (korelasi, trend, cluster, trayektori, dan anomali) yg meringkas hubungan dalam data.

Contoh:
1. Association (Correlation & Causality)
   - multi-dimensional vs. single-dimensional association
     - bagian sebab: ada beberapa operand / prasyarat utk melakukan aksi / gabungan x dan y
     - 1 kondisi yg jd prasyarat utk melakukan sesuatu (buy susu → buy roti)

   snack → soft drink [0.5%, 80%] (support, confidence)
   - confidence: peluang bersyarat (peluang ini diikuti ini)

2. Klasifikasi dan Prediksi
   : menemukan model (fungsi) yg menjelaskan dan membedakan kelas/konsep utk prediksi mendatang.
   contoh: klasifikasi negara berdasarkan iklim
   - Presentasi: decision tree, classification rule, neural network
   - Prediksi: prediksi nilai numerik yg blm diketahui / yg hilang
3. Analisis Cluster
   - label kelas tidak diketahui: mengelompokkan data utk membentuk kelas2x baru
   - Prinsip: memaksimumkan kemiripan intra-kelas meminimumkan kemiripan interkelas
4. Analisis Outlier
   : objek data yg tdk mengikuti perilaku umum dari data
   : noise / eksepsi, tetapi berguna dlm fraud detection, rare events' analysis.
5. Trend dan analisis evolusi
   - trend dan deviasi: analisis regresi
   - sequential pattern mining; contoh PC → Anti Virus

Pola yang menarik
Ukuran kemenarikan:
- mudah dimengerti oleh pengguna
- valid pd data baru / data tes dgn derajat kepastian (certainty)
- berguna
- novel (basisnya riset), atau
- memvalidasi hipotesis yg dicari oleh pengguna

Pendekatan: Human-centered, query-based, focused mining

Penerapan
- Analisis halaman web
- membangun sistem rekomendasi dan analisis kolaborasi
- analisis basket data utk target marketing
- analisis biologi dan medical data
- data mining dan software engineering
- from major dedicated data mining systems / tools to invisible data mining

K2 12/02/2018

Quiz
- Dividing the employee of a company according to their gender. (Bukan)
- Predicting future stock price using historical records. (Data Mining)
- Monitoring the heart rate of abnormalities. (Data Mining)
- Computing the total sales of a company. (Bukan)
- Characterize loyal buyers for a product. (Data Mining)

Data & Eksplorasi data

Data dan Tipe Data

Data: kumpulan dari objek dan atribut
Atribut: hal2x yg melekat pd objek yg diamati/ karakteristik dari objek. (property)
contoh: warnamata

	atribut
objek	

Atribut Values: bilangan/simbol yg diberikan pada suatu atribut.
contoh: huruf mutu (atribut)
A, AB, B, ... (atribut values)

Perbedaan: Atribut & Atribut Value
- Atribut yg sama dipetakan pada atribut value yg berbeda.
  Contoh: tinggi dpt diukur dlm feet / meter.
- Atribut yg berbeda dipetakan pd atribut value yg sama.
  Contoh: atribut value ID dan umur = integer
  - ID: tdk ada batasan
  - Umur: ada batasan min dan max

Atribute: Type
1.) Categorical (Qualitative)
  a. Nominal: perbedaan nama saja (=, ≠)
     Contoh: sex: male, female3, eye color, kode pos, ID pegawai
     Operasi: modus, entropy, contigency, correlation, χ^2 test
  b. Ordinal: informasi dapat diurutkan. (<, >)
     contoh: street number, grades, tinghat kekerasan material (good, better, best)
     Operass: median, persentil, rank correlation, run tests, sign tests.
2.) Numerical (Quantitative)
  a. Interval: perbedaan value memiliki makna. (+, -)
     Contoh: calendar dates, temperature in Celcius / Fahrenheit
  b. Ratio: perbedaan value dan rasio dari atribute value memiliki makna. (*, /)
     Contoh: temperature in Kelvin, monetary quantities, counts, age, mass, panjang, arus listrik.
     Operasi: geometric mean, harmonic mean, percent variation.

Attribute: Discrette & Continuous

Data
1. Atribut Diskret
   - himpunan nilai terbatas
     contoh: kode pos, counts, sekumpulan kata pd koleksi dokumen
   - variabel integer
2. Atribut Kontinu
   - atribut value: real number
     contoh: temperatur, tinggi, berat
   - variabel float (floating-point data)

Tipe Dataset

1. Record: kumpulan record, setiap record terdapat himpunan atribut.
   - Data Matrix
   - Data Dokumen
   - Data Transaksi
     : setiap record / transaksi melibatkan sekumpulan items.
     contoh: TRANSAKSI

TID	Items
1	Bread, Tea, Milk
2	Coffee, Bread
3	Coffee, Tea, Sugar, Milk

2. Graph: ada edges, ada node
   - World Wide Web, Twitter, dll
   - Struktur Molekul, interaksi2x Protein
3. Ordered (bisa diurutkan)
   - Spasial Data: koordinat longitude, magnitude
   - Temporal Data: pengukuran yg dilakukan dari waktu ke waktu
     Time Series: sekuen / barisan pengukuran dari beberapa atribut / data dikumpulkan dlm periode waktu tertentu.
     contoh: hotspot data.
   - Sequential Data: set transaksi dan item, namun atribut waktu dan ID pembeli diasosiasikan thd masing2x transaksi.
   - Genetic Sequence Data

Kualitas Data

Masalah kualitas data:
- Noise dan outliers
- missing values
- duplicate data

1. Noise
: berubahnya nilai atribut yg asli
contoh: distorsi suara manusia ketika berbicara ditengah keramaian / telepon yg jelek / suara berisik dari televisi.

2. Outliers (pencilan)
: objek data yg memiliki karakteristik yg sgt berbeda dgn objek data pd umumnya.

3. Missing Values
Sebab: - informasi tsb tdk terkumpul, shg NA.
contoh: orang menolak memberikan info umur dan berat badan. (nilai kosong)
- Atribut tdk dpt diterapkan pd semua kasus.
eg: atribut pendapatan tahunan tdk bisa dijawab / tdk cocok dgn anak2x

Solusi: - eliminasi objek data
- estimasi missing values, melakukan interpolasi mengg. nilai2x yg ada pd atribut tsb.
- mengabaikan missing values selama tahap analisis.

4. Duplicate Data
Data cleaning: menghilangkan noise dan data yg tdk konsisten.

Statistika Ringkasan

: summarize property data, meliputi frequency, location, spread.
contoh: lokasi – mean, median
spread – standard deviation

1. Frequency and Mode
- Frequency: persentase kemunculan nilai tsb didalam dataset pada suatu waktu.
- Modus: nilai atribut yg memiliki frekuensi paling tinggi.

2. Ukuran Lokasi: Mean dan Median
sgt sensitif thd outlier.

$$\text{mean}(x) = \bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\text{median}(x) = \begin{cases} x_{(r+1)} ; & m = \text{odd}: 2r+1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) ; & m = \text{even}: 2r \end{cases}$$

3. Ukuran Penyebaran: Range and Variance
- Range: perbedaan antara nilai max dan min dr data.
- Variance atau Standard Deviation: ukuran paling umum utk penyebaran dr suatu himpunan titik.

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{x})^2$$

sensitif thd outliers, ada pengukuran lain:

absolute average deviation

$$AAD(x) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \bar{x}|$$

$$MAD(x) = \text{median}(\{|x_1 - \bar{x}|, \ldots, |x_m - \bar{x}|\})$$

median absolute deviation

$$\text{interquartile range}(x) = x_{75\%} - x_{25\%}$$

Visualisasi

: konversi data kedalam format visual / tabular shg karakteristik data dan keterhubungan antar item data / atribut dapat dianalisis.
: teknik yg powerful dan menarik utk eksplorasi data
- mendeteksi pola umum dan tren
- mendeteksi outlier dan pola yg tidak umum.

Representasi

: pemetaan informasi ke dalam format visual.
: objek data, atribut, keterhubungan antar objek data diterjemahkan ke dlm bentuk elemen grafik, seperti points, lines, shapes, and colors.

Arrangement

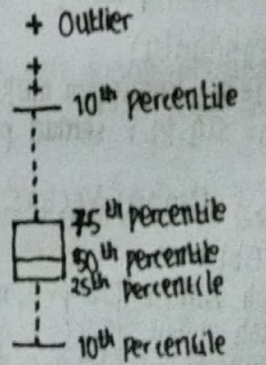
: penempatan elemen2x visual dlm suatu tampilan.
: dapat membuat perbedaan signifikan dlm membaca / memahami data.

Teknik Visualisasi

1. Histogram: dibagi2x kedlm bin
- distribusi nilai dari suatu variabel (single variable)
- tinggi bar: jumlah objek
- bentuk histogram tergantung jumlah bin.

2. 2-Dimensional Histogram
: joint distribution 2 atribut

3. Box Plots: membandingkan atribut2x



4. Scatter Plots
- atribut value: posisi

Kemiripan dan Ketidakmiripan

- Similarity
semakin tinggi nilainya, makin tinggi kemiripannya. [0,1]
- Ketidakmiripan
semakin besar jaraknya, semakin beda.
- nilai minimum = 0

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 ; p = q \\ 1 ; p \neq q \end{cases}$	$s = \begin{cases} 1 ; p = q \\ 0 ; p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (value mapped to integer 0 to n-1, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval / Ratio	$d = \lvert p-q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

- Minkowski Distance

$$dist = \left(\sum_{k=1}^{n} \lvert p_k - q_k \rvert^r \right)^{\frac{1}{r}}$$

- Mahalanobis Distance

$mahalanobis(p,q) = (p-q)\,\Sigma^{-1}(p-q)^T$; Σ : matriks kovarian data masukan X

$$\Sigma_{j,k} = \frac{1}{n-1} \sum_{i=1}^{n} (X_{ij} - \bar{X}_j)(X_{ik} - \bar{X}_k)$$

Karakteristik Umum

a. Jarak (Distance)

1. $d(p,q) \geq 0$; $d(p,q) = 0$ hanya jika $p = q$ (Positive definiteness)
2. $d(p,q) = d(q,p)$; semua p dan q → Symmetri
3. $d(p,r) \leq d(p,q) + d(q,r)$; semua titik p, q, r → Triangle inequality

; $d(p,q)$ = jarak ketidakmiripan antara titik^{2a} (objek data) p dan q

b. Kemiripan (Similarity)

1. $s(p,q) = 1$; kemiripan maks hanya jika $p = q$
2. $s(p,q) = s(q,p)$; semua p dan q → Simetri

Similarity antara Binary Vectors

→ p dan q = objek

→ M_{01} = jumlah atribut jika $p=0, q=1$
M_{10} = jumlah atribut jika $p=1, q=0$
M_{00} = jumlah atribut jika $p=0, q=0$
M_{11} = jumlah atribut jika $p=1, q=1$

- Simple Matching dan Koefisien Jaccard

$$SMC = \frac{\text{jumlah yg matches}}{\text{jumlah atribut}} = \frac{(M_{11} + M_{00})}{(M_{01} + M_{10} + M_{11} + M_{00})}$$

$$\text{Jaccard Coefficients } (J) = \frac{M_{11}}{(M_{01} + M_{10} + M_{11})}$$

contoh: p = 1 0 0 0 0 0 0 0 0 0
q = 0 0 0 0 0 0 1 0 0 1

- $M_{01} = 2$ $M_{00} = 7$
$M_{10} = 1$ $M_{11} = 0$

$SMC = \frac{(0+7)}{2+1+7+0} = 0,7$ //

$J = \frac{0}{2+1+0} = 0$ //

K3 19/02/2018

Pra-proses data

- Why Data praprocessing?
  - Data masih kurang baik (data dirty)
    1. Incomplete
       : tidak adanilai atribut / kelengkapan data tidak utuh.
       contoh, occupation = " "
    2. Noisy
       : error dan outliers.
       contoh, Salary = "-10"
    3. Inconsistent
       - sumber data yg berbeda
       - reduldansi data
  - kesalahan instrumen saat mengumpulkan data
  - human / computer error saat entry data
  - kesalahan saat transmisi data

- Why is Data Praprocessing important?
  "No quality data, no quality mining result!"
  → data berkualitas → keputusan berkualitas
  → Data warehouse: perlu integrasi yg konsisten dari data yg berkualitas
    ekstraksi data
    pembersihan data
    transformasi data

Major Task in Data Praprocessing

1. Data Cleaning
   : mengisi nilai yg kosong, smooth noisy data, identify / remove outliers, resolve inconsistencies.
2. Data Integration
   : integrasi multiple DB, data cubes, / files.
3. Data Transformation
   : normalization & aggregation
4. Data Reduction
   : menghs. jmh data yg lbh sedikit, namun hasil analisis sama dgn ukuran aslinya.
5. Data Discretization
   : bagian dari Data Reduction, tetapi dengan kepentingan tertentu, khususnya data numerik.

1. Data Cleaning

Task:
- mengisi nilai yg kosong
- identifikasi outliers & smooth out noisy data
- memperbaiki inkonsistensi data
- mengatasi redundansi yg disebabkan oleh integrasi data.

• Missing data

- data tidak selalu available
  eg: no record value for several attribute
- disebabkan oleh:
  1. kerusakan peralatan
  2. nilai atribut dihapus karena inkonsistensi dgn data lain yg telah tersimpan.
  3. data tdk diinput karena mis-understanding
  4. data tertentu dianggap tdk penting saat dientry
- missing data harus tetap diisi
- Solusi:
  1. mengabaikan record / sample tsb
  2. isi nilai kosong manual
  3. isi secara otomatis:
     - global constant
     - atribut mean
     - atribut mean dari seluruh sample pd kelas yg sama (lebih baik)
     - nilai yg plg mungkin: dugaan menggunakan formula Bayer / decision tree.

• Noisy data

: keragaman / error acak pd variabel yg diukur.

- disebabkan oleh:
  1. kesalahan instrument saat pengumpulan data
  2. masalah saat entry data
  3. masalah saat transmisi data
  4. keterbatasan teknologi
  5. inkonsistensi penamaan
- Solusi:
  1. Binning
     : data diurutkan, dibagi ke dalam bins (sesuai interval), smoothing (smooth by bin means, median, boundaries, etc)

2. Regresi
: smooth by fitting data into regression funct.
3. Clustering
: remove and detect outliers
4. Combined Computer & Human inspection
: deteksi nilai yg mencurigakan dan validasi oleh manusia.

Simple Discretization Methods: Binning

1. Equal-Width (distance) partitioning
: membagi nilai ke dalam range N interval dgn ukuran yg sama.

Jika A dan B: nilai atribut terkecil & terbesar, lebar interval $\Rightarrow W = \frac{(B-A)}{N}$

2. Equal-depth (frequency) partitioning
: membagi jumlah sampel yg sama ke dalam N interval.

contoh:

data: 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

- Partisi: equal-frequency (equal-depth) bins
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
- Smoothing by bin means
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
- Smoothing by bin boundaries
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

2. Data Integration

: gabung data dari berbagai sumber

1. Redundansi data: dideteksi dgn analisis korelasi
   - Object identification: atribut/objek yg sama memiliki nama yg berbeda di DB yg berbeda.
   - Derivable data: atribut merupakan atribut "turunan" dari tabel lainnya.

Analisis Korelasi

a. Data Numerik

$$r_{A,B} = \frac{\Sigma(A-\bar{A})(B-\bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\Sigma(AB)-n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

n = jumlah record
$\bar{A}, \bar{B}$ = rataan A dan B
σ_A, σ_B = standar deviasi dari A dan B
$\Sigma(AB)$ = jmh dr cross-product AB

- Jika $r_{A,B} > 0$ (A dan B korelasi ⊕) semakin tinggi nilai korelasi, semakin kuat korelasi A dan B. A↑, B↑
- $r_{A,B} = 0$ (independent)
- $r_{A,B} < 0$ (korelasi ⊖)

b. Categorical Data

X^2 (chi-square) test

$$X^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

contoh:

	play chess	not play	sum (row)
like science	250 (90)	200 (360)	450
not like	50 (210)	1000 (840)	1050
sum (col.)	300	1200	1500

$$X^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

∴ like science fiction dan play chess correlated in the group.

3. Data Transformation

- Smoothing: membuang noise dari data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization:
  1. Min-max normalization: [newMin_A, newMax_A]

$$v' = \frac{v - \text{min}_A}{\text{max}_A - \text{min}_A}(\text{new}\text{Max}_A - \text{new}\text{Min}_A) + \text{new}\text{Min}_A$$

eg: range pendekatan $12000 – $98000 dinormalisasi ke [0, 1]. maka $73000 akan dipetakan ke:

$$v' = \frac{(73000-12000)}{(98000-12000)}(1-0)+0 = 0{,}716$$

  2. Z-score normalization (μ: mean, σ: std. deviation)

$$v' = \frac{v - \mu_A}{\sigma_A}$$

eg: $\mu = 54000$, $\sigma = 16000$ $\Rightarrow v' = \frac{(73000-54000)}{16000} = 1{,}225$

  3. Normalization by decimal scaling

$$v' = \frac{v}{10^j}$$; j = integer terkecil $\Rightarrow$ Max ($|v'|$) < 1

4. Data Reduction

Strategi:
- Data cube aggregation
- Dimensionality reduction: membuang atribut yg tdk penting
- Data Compression
- Numerosity reduction: eg. fit data into model
- Discretization and concept hierarchy generation

K4 26/02/2018

5. Diskretisasi

- 3 tipe atribut:
  1. Nominal
     : unordered set. eg: color, profession
  2. Ordinal
     : ordered set. eg: military / academic rank
  3. Continuous
     : real number. eg: integer or real numbers.

- Discretization: → data kontinyu
  - membagi range atribut ke dalam interval
  - supervised vs. unsupervised
  - split (top-down) vs. merge (bottom-up)
  - dilerapkan dlm atribut secara rekursif.

- Concept Hierarchy Generation for Categorical Data
  - Spesifikasi of a partial / total ordering of attributes explicitly at the schema level by users or experts
    street < city < state < country
  - Specification of a hierarchy for a set of values by explicit data grouping
    {Urbana, Champaign, Chicago} < Illinois
  - Specification of only a partial set of attributes
    only street < city, not others
  - Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values.
    - for a set of attributes: {street, city, state, country}
    - attribute with the most distinct values is placed at the lowest of the hierarchy.
      eg: weekday, month, quarter, year

country (15 distinct values)
|
province or state (365 ...)
|
city (3567 ...)
|
street (674339 ...)

Techniques	Data cleaning	Data integration	Data Transfo..	Data Reduct..	Discretization
1. Binning	✓		✓	✓	✓
2. Regression	✓			✓	
3. Clustering	✓			✓	✓
4. Correlation analysis		✓			
5. Decision tree	✓			✓	
6. Normalization			✓		
7. Feature selection				✓	
8. Histogram				✓	✓
9. Sampling				✓	
10. Concept hierarchy generation			✓		✓

Analisis Cluster

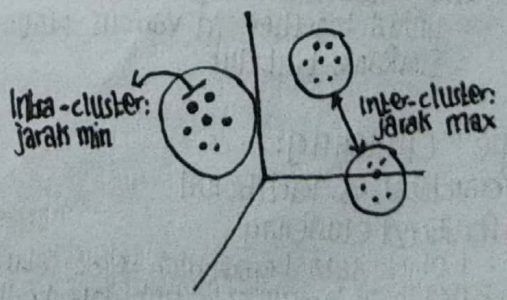
- Clustering -

: teknik unsupervised learning

: tidak ada proses learning, hanya berdasar informasi dari data yg akan diolah.

: Tujuan: Objek2x yg mirip dalam cluster yg sama
Objek2x yg berbeda dlm cluster yg sama.

: Ideal: similarity / homogenitas semakin besar di dalam 1 grup & dissimilarity semakin besar diantara grup → clustering semakin baik.

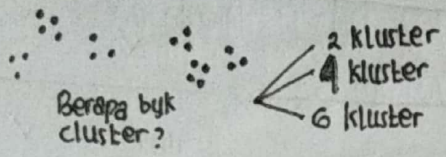


Aplikasi Analisis Cluster

1. Understanding
   : memahami pola / relasi antar objek data)
   eg: - groups related documents for browsing.
   - groups genes & proteins with similar functionality.
   - groups stocks with similar price fluctuations.

2. Summarization
   : reduce the size of large datasets.

Penentuan cluster terbaik tergantung dari kondisi data serta hasil yg diinginkan.
eg:



Major Clustering Approaches

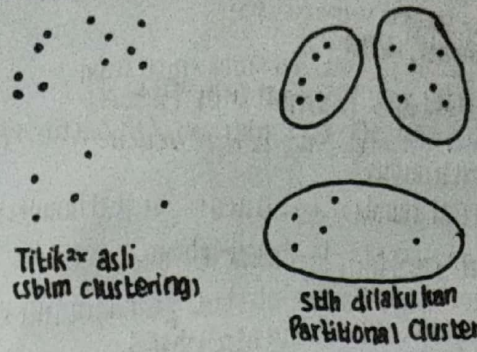
1. Partitioning approach
   : membuat beberapa cluster berdasar kriteria tertentu (sudah diketahui atau berapa cluster).
   : K-Means, K-Medoids, CLARANS.
2. Hierarchical approach
   : dipotong berdasar kriteria tertentu (dekomposisi hirarki)
   : Diana, Agnes, BIRCH, CAMELEON
3. Density-based approach
   : berdasar pada konektivitas dan density funct.
   : DBSACN, OPTICS, DenClue
4. Grid-based approach
   : berdasar pada multiple level granularity structure.
   : STING, WaveCluster, CLIQUE
5. Model-based
   : model hipotesis utk cluster dan dicari model terbaik (data baru langsung msk ke cluster manapun)
   : EM, SOM, COBWEB
6. Frequent pattern-based
   : analisis frequent patterns
   : p-Cluster
7. User-guided or constraint-based
   : by considering user specified/app specified constraint.
   : COD (obstacles), constrained clustering
8. Link-based clustering
   : linked together in various ways.
   : SimRank, LinkClus

Tipe2x Clustering:

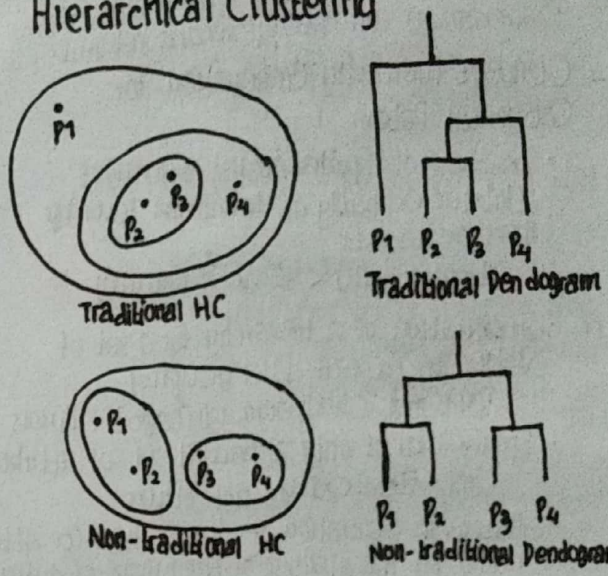
Hierarchical vs. Partitional

- Partitional Clustering
  : 1 objek data berada dlm tepat satu cluster
  : membagi himpunan objek data kedlm sub-himpunan (cluster) yg tdk overlap.
- Hierarchical clustering
  : cluster memiliki subcluster : himpunan cluster bersarang yg diatur dlm bentuk tree.

Partitional Clustering



Hierarchical Clustering



- Exclusive vs. Non-Exclusive
  └ 1 objek, 1 cluster
  └ 1 objek bisa byk cluster/ kelas
- Fuzzy vs. Non-Fuzzy
  : setiap objek mjd milik setiap cluster dgn nilai keanggotaan diantara 0 (mutlak bkn anggota cluster) dan 1 (mutlak anggota cluster)
  : penjumlahan bobot haruslah 1.
- Partial vs. Complete
  └ setiap objek tdk msk kedlm cluster, karena beberapa objek dlm dataset mungkin bukan anggota kelompok yg tlh didefinisikan dgn baik.
  └ setiap objek msk kedlm cluster

Data Structure

structure.

1. Data Matrix : object by variable
   : two modes → row: objek, column: variabel
   : N objects, p variables

$$\begin{bmatrix} x_{11} \dots x_{1f} \dots x_{1p} \\ \dots \dots \dots \\ x_{i1} \dots x_{if} \dots x_{ip} \\ \dots \dots \dots \\ x_{n1} \dots x_{nf} \dots x_{np} \end{bmatrix}$$

2. Dissimilarity Matrix : object by object structure
   : one mode.
   (△ atas dan ▽ bawah nilainya sama, pilih 1)

Type of data in Clustering Analysis

1. Interval-scaled variables
   : standarisasi data

Similarity and Dissimilarity Between Objects

└ ukuran kemiripan └ ukuran ketidakmiripan 2 objek.
: ukuran jarak (distance)

Metode: Minkowski Distance

$$d(i,j) = \sqrt[q]{(|x_{i1}-x_{j1}|^q + |x_{i2}-x_{j2}|^q + \dots + |x_{ip}-x_{jp}|^q)}$$

if $q=1 \rightarrow$ Manhattan Distance.
$q=2 \rightarrow$ Euclidean Distance.

Sifat: $d(i,j) \geq 0$
$d(i,i) = 0$
$d(i,j) = d(j,i)$
$d(i,j) \leq d(i,k) + d(k,j)$

2. Binary Variables
   : Tabel Kontingensi

		object j		
		1	0	sum
object i	1	a	b	a+b
	0	c	d	c+d
	sum	a+c	b+d	p

› Ukuran jarak utk distance variabel simetrik biner: (kemiripan, bukan jarak)

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

› Ukuran jarak utk distance variabel asimetrik biner: (ketidakmiripan)

$$d(i,j) = \frac{b+c}{a+b+c}$$

› Jaccad coefficient (ukuran similarity utk variabel asimetrik biner):

$$sim_{Jaccard}(i,j) = \frac{a}{a+b+c}$$

CONTOH:

Name	Gender	Fever	Cough	Test 1	Test 2	Test 3	Test 4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Gender: atribut simetrik, lainnya asimetrik.
- Ubah Y dan P=1, N=0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0,33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0,67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0,75$$

3. Nominal Variables
   : generalisasi dari variabel biner yaitu dapat mengambil lebih dari 2 nilai, contoh: red, yellow, blue, green.

Metode 1:
Simple Matching (pencocokan sederhana)

$$d(i,j) = \frac{p-m}{p}$$; m: #matches, p: total variabel

Metode 2:
Gunakan sejumlah besar variabel biner

4. Ordinal Variables
   : berupa nilai diskret / kontinu
   : urutan → penting

5. Ratio-Scaled Variables
   : ukuran positif pd skala non-linear, sekitar skala eksponensial.

"Ukuran Jarak Antar Cluster"

1. Complete Linkage Clustering
   : jarak maks antar elemen dlm cluster.

$$d(A,B) = \max\{S_{x,y}\}, x \in A, y \in A$$

2. Single Linkage Clustering
   : jarak minimum antar elemen setiap cluster

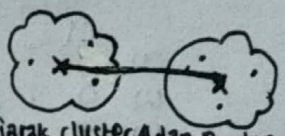
$$d(A,B) = \min\{S_{x,y}\}, x \in A, y \in A$$

3. Average Linkage Clustering
   : rata-rata jarak antar elemen setiap cluster.

$$d(A,B) = \frac{1}{n_A n_B} \sum_{x \in A} \sum_{y \in B} s(x,y)$$

4. Centroid Linkage

$$d(A,B) = s(\bar{x}, \bar{y})$$



5. Ward Linkage

$$d(A,B) = \frac{n_A n_B \delta^2_{AB}}{n_A + n_B}$$

jarak cluster A dan B dgn formula centroid linkage.

sensitif thd outliers.

K-Means Clustering

Complexity $O(n*K*I*d)$
n: # titik sampel
K: # cluster
I: # iterasi
d: # atribut

- Partitional clustering approach
- Setiap cluster berasosiasi dgn sebuah centroid (titik tengah)
- Setiap titik dikumpulkan pd cluster dgn centroid terdekat
- Jumlah cluster "k" hrs ditentukan
- Centroid awal ditentukan: random, dari nilai rata-rata (mean) dr titik dlm suatu cluster.
- Konvergen: menggunakan ukuran similaritas, ...
  closeness: menggunakan Euclidean dist, cosine similarity.

K-Medoids Clustering

Medoids: cari objek yg representatif didlm cluster.

PAM (Partitioning Around Medoids, 1987)

1. Pilih k objek yg representatif secara acak.
2. Utk tiap pasangan objek yg tidak dipilih h dan objek yg dipilih i, hitung total cost utk melakukan penukaran (swapping) TC_{ih}
3. Utk tiap pasangan dari i dan h,

   if $TC_{ih} < 0$, i digantikan h

   selanjutnya, tetapkan tiap^{ap} objek yg tdk dipilih ke objek representatif yg paling mirip.
4. Ulangi langkah 2-3 sampai tidak ada perubahan.

Contoh:

	A1	A2
(1)	2	6
(2)	3	4
(3)	3	8
(4)	4	7
(5)	6	2
(6)	6	4
(7)	7	3
(8)	7	4
(9)	8	5
(10)	7	6

K=2
- pilih acak 2 medoid
  eg: (2) = (3,4)
  (8) = (7,4)

→ Tetapkan setiap objek pd objek representatif terdekat ((Manhattan distance))

⇒ medoid (2) = (3,4)

$d(data_1, (2)) = |3-2| + |4-6| = 3$

$d(data_2, (2)) = |3-3| + |4-4| = 0$, dst.

	C1	C2
(1)	3	7
(2)	0	4
(3)	4	8
(4)	4	6
(5)	5	3
(6)	3	1
(7)	5	1
(8)	4	2
(9)	6	2
(10)	6	2

→ tentukan cluster 1 dan cluster 2; lihat jarak paling kecil masing^{ng} objek.

∴ Cluster 1 = {(1), (2), (3), (4)}
Cluster 2 = {(5), (6), (7), (8), (9), (10)}

→ Hitung kriteria absolute error (utk himpunan Medoids ((2), (8))) → nilai dari jarak yg sdh dihitung.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - O_i| = |O_1 - O_2| + |O_3 - O_2| + |O_4 - O_2| + |O_5 - O_8| + |O_6 - O_8| + |O_7 - O_8| + |O_9 - O_8| + |O_{10} - O_8|$$

$$= (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2)$$

$$= 20$$

→ Pilih objek acak: (7)

→ Tukar O_8 dan O_7

⇒ medoid (7) = (7,3)

$d(data_1, (7)) = |7-2| + |3-6| = 8$

$d(data_2, (7)) = |7-3| + |3-4| = 5$, dst.

Update tabel jarak C_2:

	C1	C2
(1)	3	8
(2)	0	5
(3)	4	9
(4)	4	7
(5)	5	2
(6)	3	2
(7)	5	0
(8)	4	1
(9)	6	3
(10)	6	3

∴ clusternya masih sama:
Cluster 1 = {1, 2, 3, 4}
Cluster 2 = {5, 6, 7, 8, 9, 10}

→ Hitung kriteria absolute error (utk himpunan Medoids (O_2, O_7))

$E = (3+4+4) + (2+2+1+3+3)$

$= 22$

→ Hitung nilai cost dari fs. Swap O_8 dan O_7

S = absolute error $[O_2, O_7]$ − absolute error $[O_2, O_8]$

$= 22 - 20$

$= 2 \rightarrow S > 0 \rightarrow$ bukan ide yg baik menggantikan O_8 dgn O_7.

Kesimpulan:

- PAM
  - lebih kokoh (robust) daripada K-Means (dgn adanya noise dan outlier) karena medoid kurang dipengaruhi oleh outlier atau nilai ekstrem lainnya.
  - efisien utk himpunan data kecil, tdk sesuai dgn himpunan data besar.
  - Kompleksitas: $O(k(n-k)^2)$ utk tiap iterasi (↓ #data, ↳ #cluster)
  - metode berbasis penarikan sampel (sampling): CLARA (Clustering LARge Applications).

Hierarchical Clustering

: himpunan cluster bersarang dalam bentuk tree.

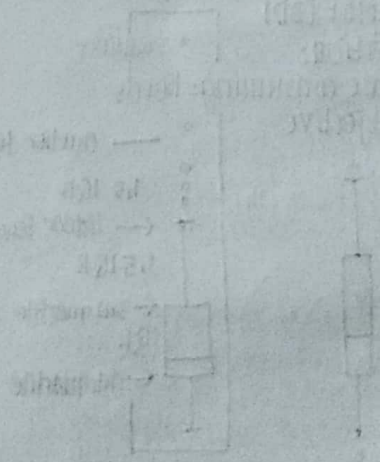
: visualisasi: Dendogram

⊕ 1. Tidak perlu mengasumsikan jmh cluster.
   : dpt memotong dendogram pd tingkat yg tepat utk mendapatkan cluster yg diinginkan.
2. Menghs taksonomi yg bermakna

2 tipe utama:

a. Agglomerative ——— strategi bottom-up:
   setiap objek dlm cluster kecil kemudian bergabung (merge) mjd 1 cluster yg lbh besar sng objek berada pd cluster tunggal/sampai kondisi tertentu yg diinginkan.
   - 1 objek, 1 cluster (mulai dr titik sbg kluster individu) : jumlah cluster awal sebanyak jmh titiknya.
   - setiap langkah, gabungkan pasangan yg terdekat sampai hnya tersisa 1 kluster (/ k cluster)

b. Divisive ——— strategi top-down:
   menempatkan semua objek dlm 1 cluster utama, kemudian dipecah mjd sub-divisi cluster yg lbh kecil sesuai kondisi tertentu.
   - semua titik dikelukan dlm 1 kluster (mulai dgn 1 cluster, all-inclusive cluster).
   - utk setiap tahapan, pisahkan suatu cluster sampai setiap cluster mengandung 1 titik (/ terdapat sebanyak k cluster).



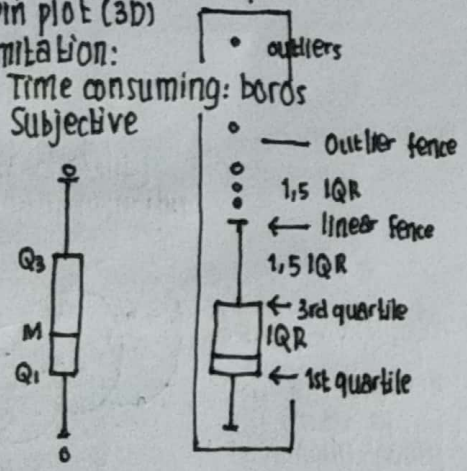
K 5 05/03/2018

Deteksi ANOMALI

- Anomali / Outlier
  : sekumpulan titik data yg sangat berbeda (Tan)/ sgt tidak mirip (dissimilar / Tan) dgn data lainnya.
- Anomaly Detection Schemes
  * General steps
    1. Buat profile dr kondisi "normal"
       : dpt berupa pola/ringkasan secara statistik dr seluruh populasi.
    2. Menggunakan profil "normal" utk deteksi anomali.
       : hasilnya adalah objek yg sgt berbeda dr profil "normal"
  > Tipe :
    1. Graphical & Statistical-based
    2. Distance-based
    3. Model-based

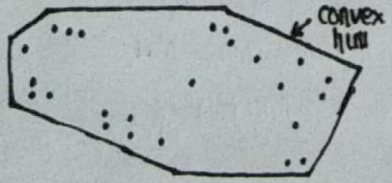
1. Graphical Approaches

- Boxplot (1D), Scatter-plot (2D), Spin plot (3D)
- Limitation:
  - Time consuming: boros
  - Subjective



Convex Hull Method

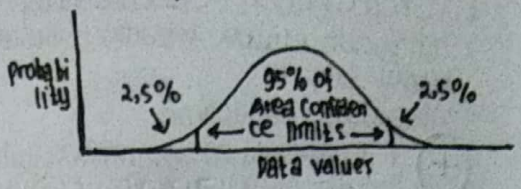
: titik ekstrim diasumsikan sbg outlier.
: untuk mendeteksi nilai ekstrim
: pd area border dataset



Statistical Approaches

: mengasumsikan model parametik yg mendeskripsikan distribusi data (eg: distribusi normal).
: Uji statistik sesuai dgn: distribusi data; parameter dr distribusi (eg: mean, variance); jmh outlier yg diharapkan.

(batas dr nilai* selang kepercayaan / confidence limit)



2. Distance-based Approaches

- mengatasi kelemahan/keterbatasan metode statistik.
- data direpresentasikan sbg vektor.
- mengukur jarak masing* objek.
- 3 Major Approaches:
  - Nearest-neighbor based
  - Density based: kerapatan sebaran data.
  - Clustering based

1. Nearest-Neighbor Based Approach

: jarak antara data ke data lain secara menyeluruh (Matrix Jarak antar pasangan objek)
: Mendefinisikan Pencilan:
- titik data hanya memiliki jumlah tetangga yg lbh sedikit dari nilai yg ditetapkan dalam jarak D. (#tetangga < k)
- n titik data dgn jarak terbesar ke tetangga terdekat ke-k. (jarak > jarak minimum)
- n titik data dgn jarak rata* terbesar ke k tetangga terdekat. (jarak rata*n objek > threshold)

2. Density-Based Local Outlier Detection

density
- tinggi: jarak antar objek dekat.
- rendah: jauh

- didasarkan pd global distance distribution
- mengamati kepadatan sekelompok populasi.
- Metode distance blm cukup maka perlu: Local outlier.

Local Outlier Factor (LOF)

: setiap objek: hitung densitas dari ketetanggaannya
: setiap titik punya LOF
: setiap titik hitung LOF = density objek p dan density tetangganya.
Jika LOF terbesar: outlier

Contoh:

Diketahui 4 titik data:
a(0,0), b(0,1), c(1,1), d(3,0)
Calculate the LOF for each points & show the top 1 outlier, set K=2 and use Manhattan Distance.

Jawab:

› Step 1: Calculate distance

	a	b	c	d
a	0	1	2	3
b		0	1	4
c			0	3
d				0

› Step 2: Calculate $dist_2(O)$
: jarak maksimum utk tetangga masing** objek. k=2 → $dist_2$

$dist_2(a) = dist(a,c) = 2$ (c is the 2nd nearest neighbor)
$dist_2(b) = dist(b,a) = 1$ (a/c ...)
$dist_2(c) = dist(c,a) = 2$ (a & ...)
$dist_2(d) = dist(d,a) = 3$ (a/c ...)

› Step 3: Calculate $N_k(O)$

$N_k(O) = \{O' \mid O' \text{ in } D, dist(O,O') \leq dist_k(O)\}$

$N_2(a) = \{b, c\}$
$N_2(b) = \{a, c\}$
$N_2(c) = \{b, a\}$
$N_2(d) = \{a, c\}$

› Step 4: Calculate $lrd_k(O)$: Local Reachability Density of O

$$lrd_k(O) = \frac{||N_k(O)||}{\sum_{O' \in N_k(O)} reachdist_k(O' \leftarrow O)}$$

$reachdist_k(O \leftarrow O') = \max\{dist_k(O), dist(O,O')\}$

- $lrd_2(a) = \frac{||N_2(a)||}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)}$

→ $reachdist_2(b \leftarrow a) = \max\{dist_2(b), dist(b,a)\} = \max\{1,1\} = 1$

→ $reachdist_2(c \leftarrow a) = \max\{dist_2(c), dist(c,a)\} = \max\{2,2\} = 2$

$lrd_2(a) = \frac{2}{1+2} = 0,667$

- $lrd_2(b) = \frac{2}{2+2} = 0,5$
- $lrd_2(c) = \frac{2}{1+2} = 0,667$
- $lrd_2(d) = \frac{2}{3+3} = 0,33$

› Step 5: Calculate $LOF_k(O)$

$$LOF_k(O) = \frac{\sum_{O' \in N_k(O)} \frac{lrd_k(O')}{lrd_k(O)}}{||N_k(O)||}$$

$$= \sum_{O' \in N_k(O)} lrd_k(O') \cdot \sum_{O' \in N_k(O)} reachdist_k(O' \leftarrow O)$$

- $LOF_2(a) = (lrd_2(b) + lrd_2(c)) * (reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a))$
$= (0,5 + 0,667) * (1 + 2) = 3,501$
- $LOF_2(b) = 5,336$
- $LOF_2(c) = 3,501$
- $LOF_2(d) = 8,004$

› Step 6: Sort all the $LOF_k(O)$

$LOF_2(d) = 8,004$
$LOF_2(b) = 5,336$
$LOF_2(a) = 3,501$
$LOF_2(c) = 3,501$

∴ Top 1 Outlier is point d.

— Outlier Discovery:
Deviation-Based Approach —
: identifikasi outlier dgn mengamati karakteristik dr tiap objek dlm cluster.
: Sequential Exception Technique
↳ cara agar manusia dpt membedakan objek tdk umum dari serangkaian objek** dgn karakteristik tertentu.
: OLAP data cube technique
↳ mengg. data cubes utk identifikasi daerah anomali dgn multidimensi yg besar.

— 3. Clustering-Based —
: mengelompokkan data ke dalam kepadatan (density) yg berbeda.
: kandidat outlier
↳ pilih titik** dlm cluster kecil
: Hitung jarak antara titik kandidat outlier dgn titik** kluster** yg tak mjd kandidat outlier.
↳ Jika titik kandidat jauh dari semua titik yg bukan kandidat, maka titik tsb adalah outlier.

DBSCAN
(Density-Based Spatial Clustering of Application with Noise)
: mengelompokkan titik dgn titik** yg memiliki byk tetangga dekat.
: Outlier: titik yg terletak sendirian / jauh dari titik** lainnya & density kecil.

K6 12/03/2018

CLASSIFICATION:

DECISION TREE

Definisi Kepadatan (Density)

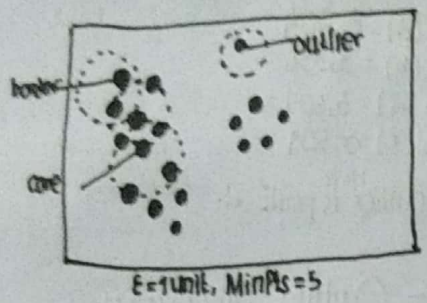
› ε-Neighborhood: objek dgn radius ε dari suatu objek.

$N_\varepsilon(p): \{q \mid d(p,q) \leq \varepsilon\}$

› High density: ε-Neighborhood of an object contains at least MinPts of object.

MinPts = 4
- Density of p: high
- Density of q: low

Core, Border, & Outlier



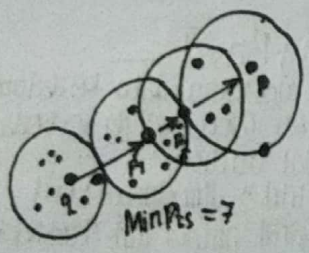
ε=1unit, MinPts=5

› Core Point: titik mempunyai lebih dari MinPts dlm ε (Eps)
: titik** ini berada pd bagian dlm (interior) dr cluster.

› Border Point: titik memiliki lbh sedikit dr MinPts, berada dlm lingkungan dr Core Point.

› Noise Point: bukan Core point & Border point.

Density-Reachability
: directly dan indirectly



› titik p directly density-reachable dari p_2
› p_2 directly density-reachable dari p_1
› p_1 directly density-reachable dari q

(rantai) $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$
↳ p (indirectly) density-reachable dari q
↳ q is not density-reachable from p.

⊕: 1. Tahan thd noise
2. dpt menangani kluster dari bentuk dan ukuran yg berbeda

⊖: 1. Tidak mampu menangani berbagai variasi kepadatan.
2. Sensitif thd parameter
: sulit menentukan himpunan parameter yg benar
Parameter.

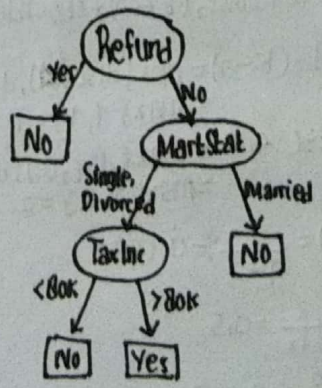
› Definisi: proses *training* + *testing*
↳ diberikan kumpulan data (training set) setiap record: ada informasi kelas
↳ dilakukan proses *training*, menghasilkan: model (bisa untuk memprediksi data baru)
↳ model diterapkan pada test set. dilakukan proses *testing*.

› Classification Techniques:
1. Decision Tree Based Methods
2. Rule-based Methods
: dari hasil (1) didapatkan aturan** utk pengelompokan kelas
3. Memory based reasoning
4. Neural Networks
5. Naive Bayes & Bayesian Belief Networks
6. Support Vector Machines

— Decision Tree —

eg:

Training set	Refund (categorical)	Marital status (categorical)	Taxable Income (continuous)	Cheat (class)
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



Test Data

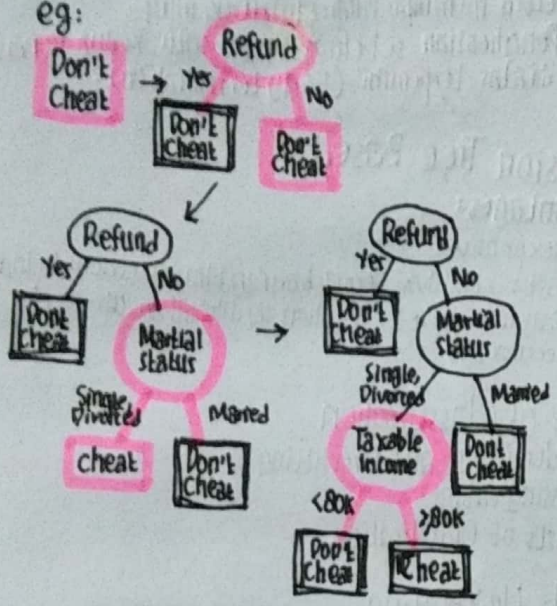
	Refund	Marital status	Taxable Income	Cheat
1.	No	Married	80K	? ((No))

> Algoritme:
1. Hunt's Algorithm (one of the earliest)
2. Classification and Regression Trees (CART)
3. Iterative Dichotomikser 3 (ID3), C4.5
4. SLIQ, SPRINT

1. Hunt's Algorithm

diketahui: D_t = training records
> if D_t memiliki kelas sama dgn y_t, maka t is a leaf node labeled as y_t.
> If D_t is empty set, then t is a leaf node labeled by default class y_d.
> if D_t memiliki >1 class, use an attribute test to split the data into smaller subset.

eg:



Tree Induction

> Greedy Strategy
: memecah sampel data berdasar atribut yg dipilih → diperoleh subset data training.

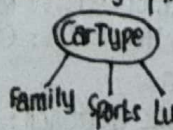
> Issues:
- Determine how to split the records.
  1. How to specify the attribute test condition?
  2. How to determine the Best split?
- Determine when to stop splitting.

1. Specify Test Condition

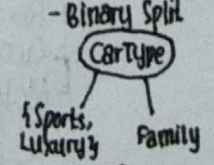
- bergantung tipe atribut: nominal, ordinal, continuous
- bergantung jumlah split: 2-way split (dicari optimal partitioning), multiway split.

eg: • Splitting Nominal Attribute

- Multiway Split

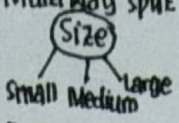


- Binary Split

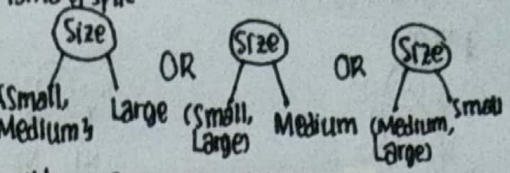


• Splitting Ordinal Attribute

- Multiway split



- Binary Split

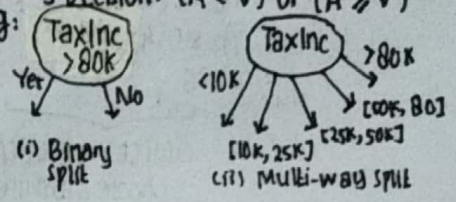


• Splitting Continuous Attribute

1. Proses Diskretisasi
   - Static: dikelompokkan diawal (eg: dibagi jd 5)
   - Dynamic: range by equal interval, equal frequency (percentile), or clustering

2. Binary Decision: $(A < V)$ or $(A \geq V)$

eg:



2. How to determine the Best Split?

Ideal: semua leaf berasal dari kelas yg sama.
> Greedy Approach: node with homogeneous class distribution lebih diutamakan.

Need a measure of node impurity:

C0: 5	C0: 9
C1: 5	C1: 1
Non-Homogeneous (High-degree of impurity)	Homogeneous (Low-degree of impurity)

Purity
semakin byk nilai pd variasi kelas, semakin tinggi purity

Measure of Node Impurity

1. Gini Index 2. Entropy 3. Misclassification error

1. GINI
: pilih nilai GINI terkecil.
: node t.

$$GINI(t) = 1 - \sum_j [P(j|t)]^2$$

$P(j|t)$: relative frequency of class j at node t

> Min (0.0): if all records belong to 1 class. most interesting information
> Maximum $(1 - 1/n_c)$: records are equally distributed among all classes. least interesting information (n_c = # split)

eg:

C1: 0, C2: 6 → $P(C1) = 0/6 = 0$, $P(C2) = 6/6 = 1$ → $GINI = 1 - P(C1)^2 - P(C2)^2 = 1 - 0^2 - 1^2 = 0$

C1: 1, C2: 5 → $P(C1) = 1/6$, $P(C2) = 5/6$ → $GINI = 1 - (1/6)^2 - (5/6)^2 = 0,278$

Splitting Based On GINI

: digunakan dalam CART, SLIQ, SPRINT
: node p is split into k partitions (children), quality of split:

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

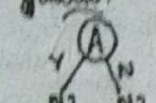
n_i → # data pd child i; n → # data pd node p

eg: Parent
C1: 6
C2: 6 > GINI = 0.5

Split: Binary partition (2 partisi)

Mana yg lbh baik?

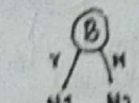
Jawab:



Gini(N1) = 1 − (4/7)² − (3/7)²
= 0,4898

Gini(N2) = 1 − (2/5)² − (3/5)²
= 0,48

Gini(children) = 7/12 * 0,4898 + 5/12 * 0,48
= 0,486



Gini(N1) = 0,32
Gini(N2) = 0,408
Gini(children)
= 5/12 * 0,32 + 7/12 * 0,408
= 0,375

Gini(B) < Gini(A)
∴ choose attribute B, karena lebih murni (pure)

Continuous Attribute: Computing GiniIndex

Efficient computing:

1. Sort attribute value: lower → higher
2. Linearly scan these values, each time updating the count matrix and computing Gini Index.
3. Choose the split position that has the least Gini Index.

eg: ((lihat slide hal. 18))

2. Entropy

: node t

$$\text{Entropy}(t) = -\sum_j P(j|t) \log_2 P(j|t)$$

> Maximum ($\log_2 n_c$)
> Minimum (0.0)
} = GINI

eg:
C1: 0
C2: 6
→ P(C1) = 0/6 = 0
P(C2) = 6/6 = 1
> Entropy = −0 * log₂0 − 1 * log₂1 = −0 − 0 = 0

Splitting Based On Entropy

• Information Gain

(−): menghasilkan partisi yg besar, @ partisi berukuran kecil, tapi homogen/pure.

$$GAIN_{split} = \text{Entropy}(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} \text{Entropy}(i)\right)$$

; p = parent node
k = # split

Measure reduction in Entropy
: choose split that achieves most reduction (maximizes GAIN)

• Gain Ratio

(+): mengatasi kelemahan information Gain.

$$GainRATIO_{split} = \frac{GAIN_{split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

3. Classification Error

: node t

$$\text{Error}(t) = 1 - \max P(i|t)$$

> Maximum ($1 - 1/n_c$)
> Minimum (0.0)
} = GINI

eg:
C1: 0
C2: 6
→ P(C1) = 0/6 = 0
P(C2) = 6/6 = 1
> Error = 1 − max(0,1) = 1 − 1 = 0

Determine When to Stop Splitting

Criteria Stop:

1. Membuat melakukan pembagian suatu node ketika semua record mjd anggota kelas yg sama.
2. Hentikan memperluas suatu node ketika semua record memiliki nilai atribut yg mirip.
3. Penghentian sebelum salah satu kedua kondisi diatas terpenuhi (Early termination).

Decision Tree Based

Advantages:

1. Inexpensive
2. Fast: klasifikasi record baru yg blm diketahui kelasnya
3. Easy to interpret: if pohon yg dihasilkan utk. kecil.
4. Accuracy

Issue of Classification

1. Underfitting and Overfitting
2. Missing Values
3. Costs of Classification

1. Underfitting

: model terlalu umum → tidak dapat membedakan kelas
: kedua error hasil training dan testing nilainya besar.

Overfitting

: dalam proses training, model yg dihasilkan terlalu ideal, karena misclassified diperhatikan → training akurasinya 100% → tidak mampu mengeneralisir (modelnya), shg ketika ada data test baru, error akan semakin naik.
: error ketika training set kecil, tapi error test set besar.
: disebabkan: sampel data training yg digunakan kurang

Estimating Generalization Errors

- Re-substitution errors: error on training ($\Sigma e(t)$)
- Generalization errors: error on testing ($\Sigma e'(t)$)
- Metodenya:
  1. Optimistic approach: $e'(t) = e(t)$
  2. Pessimistic approach:
     - For each leaf node: $e'(t) = e(t) + 0.5$
     - Total error: $e'(T) = e(T) + N \times 0.5$ (N = number of leaf node)

eg: For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances):
• Training error = 10/1000 = 1%
• Generalization error = $\frac{(10 + 30 \times 0.5)}{1000} = 2.5\%$

› Reduced error pruning (REP)
: mengg. data testing (validasi) utk estimasi generalization error.

Mengatasi Overfitting

1. Pre-Pruning (Early Stopping Rule)
   - Hentikan algoritme sebelum mjd "a fully-grown tree"
   - Kondisi berhenti (stopping) suatu node:
     - Hentikan jika semua instans (records) memiliki kelas label yg sama.
     - Hentikan ketika semua nilai atribut adalah sama.
   - Kondisi yg lebih ketat:
     - Hentikan ketika jmh records dlm suatu node lbh kecil dr threshold yg didefinisikan oleh user.
     - Hentikan ketika distribusi kelas dari instans/record2x ny a independen dr fitur2x yg ada. (eg: gunakan χ^2 test)
     - Hentikan ketika perluasan suatu node tdk memperbaiki derajat homogenitas (impurity). (gunakan Gini/Information Gain).

2. Post-pruning
   - Buat decision tree sampai mjd pohon lengkap.
   - Potong simpul2x/node2x pd decision tree dimulai dr bawah (bottom-up fashion).
   - Hitung generalization error.
     Jika generalization error mjd lbh baik stlh proses pemotongan (trimming), ganti sub-tree dgn a leaf node.
   - Label kelas dr leaf node ditentukan dr label kelas mayoritas dr instans2x/records pd sub-tree tsb
   - Dapat mengg. Minimum Description Length (MDL) utk post-pruning.

eg: Class: Yes 20
Class: No 10
Error = $10/30$

› Training Error (Before splitting) = $10/30$

› Pessimistic Error (Before splitting) = $\frac{(10 + 0.5)}{30} = 10.5/30$

› Training Error (After splitting) = $9/30$

› Pessimistic Error (After splitting) = $\frac{(9 + 4 * 0.5)}{30} = 11/30$

K7 19/03/2018

Klasifikasi

dengan SVM, Seleksi, dan Evaluasi Model

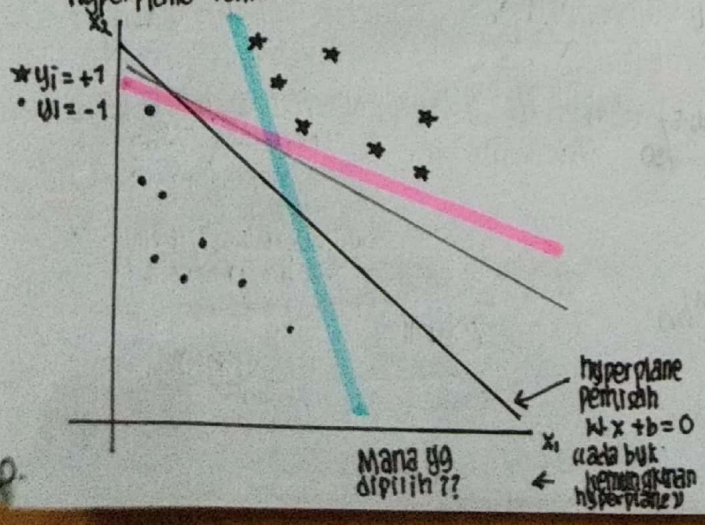
Pendekatan dengan *Machine Learning* Terminologi Dasar

1. Training set: N data $\{x_1, \ldots, x_n\}$
   - digunakan utk tuning parameter dari model
   - kategori dari data *training set* telah dikelahui sebelumnya
2. Target vector: vektor unik t utk tiap target
   - merepresentasikan identitas dari data yg bersesuaian.
3. Learned function: $y(x)$
   - Training phase (fase pelatihan): proses untuk menentukan $y(x)$ berdasarkan data latih (*training data*)
4. Data uji (Test set): data yg blm terdapat pd data latih (*training set*)
   - setelah model selesai dilatih, model dpt menentukan kategori dari data baru.
   - kemampuan utk mengkategorikan dgn benar data baru yg berbeda dgn data yg digunakan pd training set → GENERALISASI

Definisi Masalah

- diberikan sekumpulan n titik (vectors): $x_1, x_2, \ldots, x_n$ dimana x_i adh vektor dgn pjg m dan masing2x adalah anggota salah satu dari dua kelas yg memiliki label "+1" dan "-1".
- maka training set: $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$
  $\forall x_i \in \mathbb{R}^m$, $y_i \in \{+1, -1\}$
  Fungsi keputusan: $f(x) = sign(w \cdot x + b)$
- diinginkan untuk menemukan *hyperplane* $w \cdot x + b = 0$ yg memisahkan titik2x tsb kedlm 2 kelas: Positif (kelas "+1") dan Negatif (kelas "-1").
  (Asumsi: titik** dpt dipisahkan secara linier).

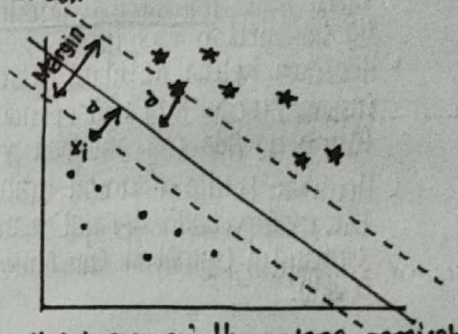
- Hyperplane Pemisah -



Memilih Hyperplane Pemisah

Pendekatan SVM: Kasus Linear Separable

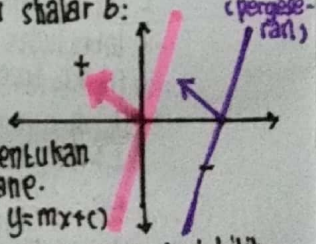
- Hyperplane: sebaiknya sejauh mungkin dari titik sampel (ideal)
  - sehingga data baru yg dekat dgn sampel data akan diklasifikasikan dgn benar.
    → Generalisasi Baik.
- Ide dasar dari SVM: memaksimalkan distance (jarak) antara *hyperplane* dan titik sampel terdekat.
  Pada optimal hyperplane: jarak ke titik negatif terdekat = jarak ke titik positif terdekat.
- Tujuan SVM: memaksimalkan margin yg besarnya 2x jarak "d" antara hyperplane pemisah dgn sampel terdekat.



Untuk mencari Hyperplane pemisah yg optimal, daerah yg dibentuk oleh hyperplane dan margin dpt: diputar ke kanan / ke kiri, dipersempit / diperlebar

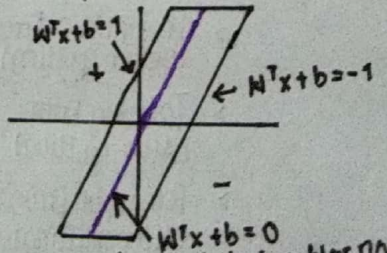
Tujuan: - mendapatkan hyperplane dgn margin yg maksimal.
- data pd kelas positif ($y=+1$) dan kelas negatif ($y=-1$) tetap terpisah.

* Hyperplane dpt direpresentasikan oleh vektor normal w dan suatu skalar b:
  $w^T x + b = 0$



- Vektor normal w: menentukan orientasi dari hyperplane.
  (ingat: gradien (m) pd $y = mx + c$)
- Bias b: menentukan pergeseran dari titik asal/pusat (origin)

* Margin dpt direpresentasikan dgn 2 buah hyperplane: $w^T x + b = 1$ (kelas positif)
  $w^T x + b = -1$ (kelas negatif)



- Dengan mengubah sudut dari vektor normal w, margin dpt dirotasikan.
- Menggeser margin: dgn memperbesar / memperkecil bias b.

- Lebar Margin (m) = 2 dibagi panjang vektor normal ($\|w\|$)

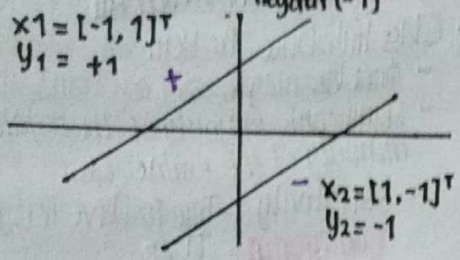
$$m = \frac{2}{\|w\|}$$

Berarti: lebar margin m berbanding terbalik dgn panjang vektor normal.

Makin besar/panjang vektor normal, makin sempit margin (m), dan sebaliknya.

Memecahkan Masalah Optimasi:
Memaksimalkan Lebar Margin (m)

misal: x_1 dan x_2, label: y_1 dan y_2
kelas: positif (+1)
negatif (-1)



Lebar Margin (m) = $\frac{2}{\|w\|}$

MAX m → $\max\left(\frac{2}{\|w\|}\right)$ (memaksimalkan m), artinya: panjang vektor normal w harus diminimalkan, maka:

$$\max\left(\frac{2}{\|w\|}\right) \rightarrow \min\left(\frac{\|w\|}{2}\right)$$

Untuk menghilangkan akar kuadrat, vektor normal harus dikuadratkan.

$$\max\left(\frac{2}{\|w\|}\right) \rightarrow \min\left(\frac{\|w\|}{2}\right) \rightarrow \min\left(\frac{\|w\|^2}{2}\right)$$

Konstrain:
- kelas positif: $w^T x_1 + b \geq 1$
- kelas negatif: $w^T x_2 + b \leq -1$

Selanjutnya, kalikan tiap konstrain dgn label yg sesuai.

$$y_1(w^T x_1 + b) \geq 1 \text{ dan } y_2(w^T x_2 + b) \leq -1$$

Sehingga diperoleh:

$$\min\left(\frac{\|w\|^2}{2}\right) \text{ dgn constraint: } y_i(w^T x_i + b) - 1 \geq 0;$$

masalah OPTIMASI

dikalikan dgn Lagrange Multipliers utk membentuk The Primal-Dual Form:

$$\text{minimize } L_p(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(x_i \cdot w + b) - 1)$$

s.t. $\alpha_i \geq 0$

Selanjutnya: $\frac{\partial L_p}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$

$$\frac{\partial L_p}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

Mendapatkan Lagrangian Dual Problem

Langrangian Dual Problem:

$$\text{maximize } L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

s.t. $\alpha_i \geq 0$ dan $\sum_{i=1}^{n} \alpha_i y_i = 0$

// $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$: variabel utk tiap sampel/point x_i.

Latihan:-

data: $x_1 = [-2, 1]^T$, $y_1 = 1$; $x_2 = [1, -2]^T$, $y_2 = -1$

((nanti dibelakang))

Karakteristik Solusi

- Banyak α_i yg bernilai nol.
  - w = kombinasi linear dari sejumlah kecil data
  - representasi sparse
- x_i non-zero α_i disebut support vectors (SV)
  - decision boundary ditentukan oleh SV
  - jika t_j ($j = 1, \ldots, s$): indeks dari support, maka:

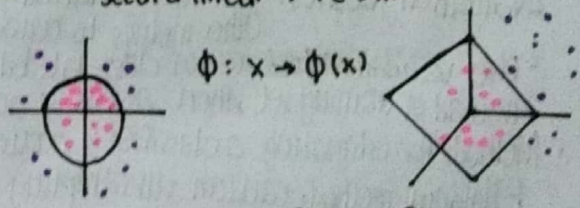
$$w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} x_{t_j}$$

- Testing data baru z
  - Hitung $w^T z + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (x_{t_j}^T z) + b$

dan klasifikasikan z sebagai kelas 1 jika hasil penjumlahannya positive, kelas 2 jika negatif.

Non-linear SVMs:
Feature Spaces

Ide umum: original feature space dpt selalu dipetakan ke beberapa higher-dimensional feature space dimana training set dpt dipisahkan secara linear → VC Dimension



Perluasan ke Non-linear Decision Boundary
- Ide: transformasikan x_i kesebuah ruang dimensi yg lebih tinggi
  - Input space: ruang x_i berada
  - Feature space: ruang dr $\Phi(x_i)$ hasil transformasi
- Transformasi: perlu
  - Linear operation pada feature space ekuivalen dgn non-linear operation pd input space.
  - Klasifikasi dpt dilakukan dgn lbh mudah dgn suatu transformasi yg sesuai, eg: XOR
  - Masalah yg muncul: komputasi tinggi dan sulit mendapatkan estimasi yg baik.

SVM memecahkan 2 masalah tsb secara simultan.
- Kernel tricks: komputasi yg efisien
- Meminimalkan $\|w\|^2$ utk mendapatkan a "good" classifier.

The Kernel Tricks

- Problem:

$$\max L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{\Phi(x_i^T)\Phi(x_j)}$$

$$\text{St. } \alpha_i \geq 0, \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

fungsi keputusan:

$$f(\Phi(x)) = \text{sign}(w^T \Phi(x) + b)$$
$$= \text{sign}\left(\sum_{i=1}^{s} \alpha_i y_i \boxed{\Phi(x_i^T)\Phi(x)} + b\right)$$

- Solusi:
  - Tidak perlu memetakkan secara eksplisit / mengetahui dimensi dr space baru, karena hanya perlu mengg. *dot product* dari vector cirinya, baik pada *training* maupun *test*.
  - Hitung: $K(x_i, x_j)$ dan tidak perlu melakukan perhitungan dalam ruang berdimensi tinggi secara eksplisit. ("Kernel Trick")

- Kernel Functions:
  - Polynomial Kernel with degree d
    $K(x,y) = (x^T y + 1)^d$
  - Radial Basic Function Kernel with degree σ
    $K(x,y) = \exp(-\|x-y\|^2 / (2\sigma^2))$
    : closely related to radial basis function neural networks
  - Sigmoid with Parameter k dan θ
    $K(x,y) = \tanh(kx^T y + \theta)$
    : it doesn't satisfy the Mercer condition on all k and θ
  - Research on different kernel functions in different applications is very active.

— Model Evaluation and Selection —

- Evaluation Metrics: How can we measured accuracy? Other metrics to consider?
  - Use validation test set of class-labeled tuples instead of training set when assessing accuracy.
- Methods for estimating a classifier's accuracy:
  1. Holdout method, random subsampling
  2. Cross-validation
  3. Bootstrap
- Comparing classifiers:
  - Confidence intervals

Classifier Evaluation Metrics:

Confusion Matrix

actual class \ predicted class	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

Accuracy, Error Rate, Sensitivity and Specificity

Actual \ Predicted	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

- Classifier Accuracy / Recognition rate
  : percentage of test set tuples that are correctly classified.

  Accuracy = (TP + TN) / All

- Error Rate: 1 − accuracy, or

  Error Rate = (FP + FN) / All

- Class Imbalance Problem
  - One class may be rare, eg: fraud, HIV-positive
  - Significant majority of the negative class and minority of the positive class
  - Sensitivity: True Positive recognition rate

    Sensitivity = TP/P
  - Specificity: True Negative recognition rate

    Specificity = TN/N

Precision and Recall, F-measures

- **Precision:** exactness – what % of tuples that the classifier labeled as positive are actually positive

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Perfect score = 1.0
- Inverse relationship between precision & recall
- **F-measure** (F_1 or F-score): harmonic mean of precision & recall

$$F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

F_β: weighted measure of precision and recall
assigns β times as much weight to recall as to precision

$$F_\beta = \frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$$

latihan:

Actual class \ Predicted class	cancer=yes	cancer=no	Total	Recognition(%)
cancer=yes	90	210	300	Sensitivity
cancer=no	140	9560	9700	Specificity
Total	230	9770	10000	Accuracy

- Sensitivity = TP/P = 90/300 = 30%
- Specificity = TN/N = 9560/9700 = 98.56%
- Accuracy = (TP + TN)/All = (90 + 9560)/10000 = 96.5%

- Precision = $\frac{TP}{TP+FP} = \frac{90}{(90+140)} = 39.13\%$
- Recall = $\frac{TP}{TP+FN} = \frac{90}{90+210} = 30\%$

1. Holdout Method

- Given data is randomly partitioned into 2 independent sets
  - Training set (eg: 2/3) for model construction
  - Test set (eg: 1/3) for accuracy estimation
- Random sampling: a variation of holdout
  - Repeat holdout k times, accuracy = avg. of the accuracies obtained

2. Cross-Validation (k-fold, where k=10 is most popular)

- Randomly partition the data into k mutually exclusive subsets, each approximately equal size.
- At i-th iteration, use D_i as test set and others as training set.
- Leave-one-out: k folds where k = # of tuples, for small sized data.
- Stratified Cross-Validation: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data.

3. Bootstrap

- Works well with small data sets
- Samples the given training tuples uniformly with replacement.
  eg: each time a tuple is selected, it is equally likely to be selected again and re-added to the training set.

Estimating Confidence Intervals:

- Classifier Models M_1 vs M_2
  - Suppose we have 2 classifiers, M_1 and M_2, which one is better?
  - Use 10-fold cross-validation to obtain $\overline{err}(M_1)$ | $\overline{err}(M_2)$
  - These mean error rates are just *estimates* of error on the true population of *future* data cases.
  - What if the difference between the 2 error rates is just attributed to chance?
    - Use a test of statistical significance
    - Obtain confidence limits for our error estimates
- Null Hypothesis
  - Perform 10 fold cross-validation
  - Assume samples follow a t distribution with k-1 degrees of freedom (here, k=10)
  - Use t-test (or Student's test)
  - Null Hypothesis: M_1 & M_2 are the same
  - If we can reject null hypothesis, then:
    - we conclude that the difference between M_1 & M_2 is statistically significant.
    - chose model with lower error rate.

Issues Affecting Model Selection

1. Accuracy
   + classifier accuracy: predicting class label
2. Speed
   - time to construct the model (training time)
   - time to use the model (classification/prediction time)
3. Robustness
   : handling noise and missing values
4. Scalability
   : efficiency in disk-resident databases
5. Interpretability
   : understanding and insight provided by the model
6. Other measures, eg: goodness of rules, such as decision tree size or compactness of classification rules.

latihan

~~1. Binning~~
~~2. K-Medoids~~
~~3. LOF~~
~~4. GINI & Decision Tree~~
~~5. Hyperplane~~
6. Entropy.

~~Sheet: hyperplane~~

K8 16/04/2018

Analisis

ASOSIASI:

Konsep Dasar dan Algoritme

- Clustering: SSE kecil
- Klasifikasi: akurasi bagus / sensitivity
  - tinggi
  - for imbalance class

ARM (Association Rule Mining)

: diberikan himpunan transaksi, cari aturan-x yang memprediksi kemunculan sebuah item berdasarkan kemunculan-an item-x lain dalam transaksi.

eg: Transaksi Market-Basket

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

{Diaper} → {Juice}
{Milk, Bread} → {Eggs, Coke}
{Juice, Bread} → {Milk}

Implikasi: kemunculan sama-an, bukan hub. sebab-akibat

Definition:

1. Itemset: sekumpulan 1 atau lebih items
   eg: {Milk, Bread, Diaper}
   - k-itemset: itemset yg mengandung k items.
2. Support count (σ): frekuensi kemunculan sebuah itemset. → support absoulute.
   eg: σ({Milk, Bread, Diaper}) = 2.
3. Support: fraksi transaksi yg mengandung sebuah itemset. → support relative.
   eg: s({Milk, Bread, Diaper}) = 2/5.
4. Frequent Itemset: sebuah itemset dikatakan *support* jika ≥ minsup threshold.
5. Association Rule: ekspresi implikasi dari bentuk $X \rightarrow Y$, dimana X dan Y = itemsets.
   eg: {Milk, Diaper} → {Juice}
6. **Rule Evaluation Metrics**
   - Support (s): fraction of transactions that contain both X and Y.
   - Confidence (c): measures how often items in Y appear in transactions that contain X.

eg: {Milk, Diaper} ⇒ Juice

$$s = \frac{\sigma(\text{Milk, Diaper, Juice})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk, Diaper, Juice})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Mining Task

- given a set of transactions T
- goal: find all rules having
  support ≥ minsup threshold
  confidence ≥ minconf threshold
- Brute-force approach:
  1. List all possible association rule
  2. Compute the support and confidence for each rule
  3. Prune rules that fail the *minsup* and *minconf* thresholds.

Mining Association Rules

- Two-step approach:
  1. Frequent Itemset Generation
     : generate all itemsets whose support ≥ minsup
  2. Rule Generation
     : generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset.

1. Frequent Itemset Generation

Given d unique items:

- Total number of itemsets = 2^d
- Total number of possible association rules =

$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right] = 3^d - 2^{d+1} + 1$$

eg: $d=6 \rightarrow R = 3^6 - 2^{6+1} + 1 = 602$ rules

- Complexity ~ O(NMw) → expensive since $M = 2^d$

Solusi:

1. Mengurangi jumlah kandidat (M)
   - Complete search: $M = 2^d$
   - Use pruning techniques to reduce M
2. Mengurangi jumlah transaksi (N)
   - Reduce size N as the size of itemset increases
   - Use vertical-partitioning of the data to apply the mining algorithms
3. Mengurangi jumlah comparisons (NM)
   - Use efficient data structures to store the candidates or transactions
   - No need to match every candidate against every transaction.

Prinsip Apriori: jika itemset: frequent, maka semua subsets harus frequent.

$$\forall X, Y: (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

Method:

- Let k=1
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
  - Generate length (k+1) candidate itemsets from length k frequent itemsets.
  - Prune candidate itemsets containing subsets of length k that are infrequent.
  - Count the support of each candidate by scanning the DB
  - Eliminate candidates that are infrequent, leaving only those that are frequent.

eg: minsup = 2

TID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan →

C1

itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L1

itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

2nd scan →

C2

itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L2

itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

3rd scan →

C3

itemset	sup
{B, C, E}	2

L3

itemset	sup
{B, C, E}	2

sebenarnya:
{A, B, C} : tereliminasi,
{A, C, E} shg tidak diambil
{B, C, E}

Latihan:

TID	Items
10	Coke, Nuts, Diaper
20	Coke, Coffee, Diaper
30	Coke, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

minsup = 50%
minconf = 50%

Jawab:

C1

itemset	sup
{Coke}	3/5 = 60%
{Nuts}	3/5 = 60%
{Diaper}	4/5 = 80%
{Cofee}	2/5 = 40%
{Eggs}	3/5 = 60%
{Milk}	2/5 = 40%

→ L1

itemset	sup
{Coke}	60%
{Nuts}	60%
{Diaper}	80%
{Eggs}	60%

C2

itemset	sup
{Coke, Nuts}	1/5 = 20%
{Coke, Diaper}	3/5 = 60%
{Coke, Eggs}	1/5 = 20%
{Nuts, Diaper}	2/5 = 40%
{Nuts, Eggs}	2/5 = 40%
{Diaper, Eggs}	2/5 = 40%

→ L2

itemset	sup
{Coke, Diaper}	60%

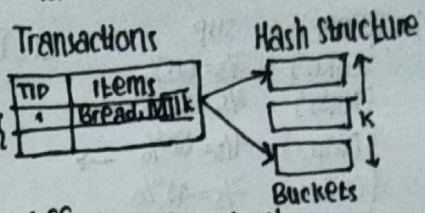
if minsup = 20%, how = ...?

- Horizontal Data layout

TID	Items
1	A, B, E
2	B, C, D
3	C, E
4	A, C, D
5	A, B, C, D
6	A, E
7	A, B
8	A, B, C
9	A, C, D
10	B

Reducing Number of Comparisons

- Candidate counting:
  1. Scan the database of transactions to determine the support of each candidate itemset.
  2. To reduce the number of comparisons, store the candidates in a hash structure.
     - Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets.



Factors Affecting Complexity

1. Choice of minimum support threshold
   - semakin kecil support threshold, makin banyak frequent itemsets
   - dpt meningkatkan jmh kandidat dan memaksimumkan length of frequent itemset.
2. Dimensionality (number of items) of the dataset
   - butuh banyak space utk menyimpan support count setiap item
   - jika jmh frequent items naik, maka biaya komputasi dan I/O juga meningkat.
3. Size of database
   - jumlah transaksi meningkat, run time algoritme meningkat
4. Average transaction width
   - transaction width increases with denser data sets.
   - menyebabkan peningkatan max length of frequent itemsets dan traversals of hash tree (number of subsets in a transaction increases with its width)

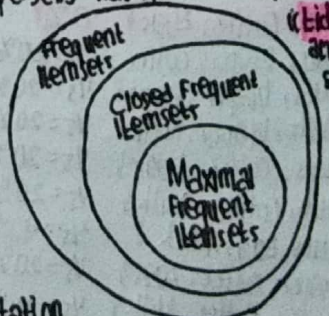
Maximal Frequent Itemset

: itemset dikatakan maximal frequent jika none of its immediate supersets is frequent.

("tidak ada item yg frequent dgn parentnya")

Closed Itemset

: itemset dikatakan closed jika none of its immediate supersets has the same support as the itemset.



("tidak ada item anaknya yg supportnya sama dgn parent))

Representation of Database →

- vertical Data Layout

A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

Jawaban PR:

minsup = 20%

TID	Items
10	Coke, Nuts, Diaper
20	Coke, Coffee, Diaper
30	Coke, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

C1

itemsets	sup
{Coke}	3/5 = 60%
{Nuts}	3/5 = 60%
{Diaper}	4/5 = 80%
{Coffee}	2/5 = 40%
{Eggs}	3/5 = 60%
{Milk}	2/5 = 40%

→ **L1**

itemset	sup
{Coke}	60%
{Nuts}	60%
{Diaper}	80%
{Coffee}	40%
{Eggs}	60%
{Milk}	40%

C2

itemset	sup
{Coke, Nuts}	1/5 = 20%
{Coke, Diaper}	3/5 = 60%
{Coke, Coffee}	1/5 = 20%
{Coke, Eggs}	1/5 = 20%
{Coke, Milk}	0%
{Nuts, Diaper}	2/5 = 40%
{Nuts, Coffee}	1/5 = 20%
{Nuts, Eggs}	2/5 = 40%
{Nuts, Milk}	2/5 = 40%
{Diaper, Coffee}	2/5 = 40%
{Diaper, Eggs}	2/5 = 40%
{Diaper, Milk}	1/5 = 20%
{Coffee, Eggs}	1/5 = 20%
{Coffee, Milk}	1/5 = 20%
{Eggs, Milk}	2/5 = 40%

→ **L2**

itemset	sup
{Coke, Nuts}	20%
{Coke, Diaper}	60%
{Coke, Coffee}	20%
{Coke, Eggs}	20%
{Nuts, Diaper}	40%
{Nuts, Coffee}	20%
{Nuts, Eggs}	40%
{Nuts, Milk}	40%
{Diaper, Coffee}	40%
{Diaper, Eggs}	40%
{Diaper, Milk}	20%
{Coffee, Eggs}	20%
{Coffee, Milk}	20%
{Eggs, Milk}	40%

C3

itemset	sup
{Coke, Nuts, Diaper}	1/5 = 20%
{Coke, Nuts, Coffee}	0%
{Coke, Nuts, Eggs}	0%
{Coke, Diaper, Coffee}	1/5 = 20%
{Coke, Diaper, Eggs}	1/5 = 20%
{Coke, Coffee, Eggs}	0%
{Nuts, Diaper, Coffee}	1/5 = 20%
{Nuts, Diaper, Eggs}	1/5 = 20%
{Nuts, Diaper, Milk}	1/5 = 20%
{Nuts, Coffee, Eggs}	1/5 = 20%
{Nuts, Coffee, Milk}	1/5 = 20%
{Nuts, Eggs, Milk}	2/5 = 40%
{Diaper, Coffee, Eggs}	1/5 = 20%
{Diaper, Coffee, Milk}	1/5 = 20%
{Diaper, Eggs, Milk}	1/5 = 20%
{Coffee, Eggs, Milk}	1/5 = 20%

→ **L3**

itemset	sup
{Coke, Nuts, Diaper}	20%
{Coke, Diaper, Coffee}	20%
{Coke, Diaper, Eggs}	20%
{Nuts, Diaper, Coffee}	20%
{Nuts, Diaper, Eggs}	20%
{Nuts, Diaper, Milk}	20%
{Nuts, Coffee, Eggs}	20%
{Nuts, Coffee, Milk}	20%
{Nuts, Eggs, Milk}	40%
{Diaper, Coffee, Eggs}	20%
{Diaper, Coffee, Milk}	20%
{Diaper, Eggs, Milk}	20%
{Coffee, Eggs, Milk}	20%

C4

itemset	sup
{Nuts, Diaper, Coffee, Eggs}	1/5 = 20%
{Nuts, Diaper, Coffee, Milk}	1/5 = 20%
~~{Nuts, Diaper, Coffee, Eggs, Milk}~~	~~1/5 = 20%~~
{Nuts, Diaper, Eggs, Milk}	1/5 = 20%
{Nuts, Coffee, Eggs, Milk}	1/5 = 20%
{Diaper, Coffee, Eggs, Milk}	1/5 = 20%

↓

L4

itemset	sup
{Nuts, Diaper, Coffee, Eggs}	20%
{Nuts, Diaper, Coffee, Milk}	20%
~~{Nuts, Diaper, Coffee, Eggs, Milk}~~	~~20%~~
{Nuts, Diaper, Eggs, Milk}	20%
{Nuts, Coffee, Eggs, Milk}	20%
{Diaper, Coffee, Eggs, Milk}	20%

C5

itemset	sup
{Nuts, Diaper, Coffee, Eggs, Milk}	1/5 = 20%

↓

L5

itemset	sup
{Nut, Diaper, Coffee, Eggs, Milk}	20%

K9 23/04/2018

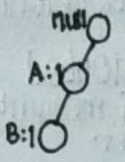
FP-Growth

- menggunakan FP-tree sbg representasi terkompresi dr database.
- buat tree → pake pendekatan divide and conquer utk dapetin frequent & infrequent itemset.
- Caranya:
  1. Baca per transaksi
  2. Mulai dgn node *null*, lalu buat node item baru. Jika item sudah ada, pakai yg sama.
  3. Node item akan bertambah kemunculannya.
  4. Bikin pointer.

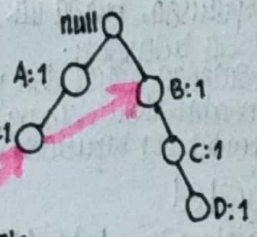
eg:

TID	Items
1	A, B
2	B, C, D
3	A, C, D, E
4	A, D, E
5	A, B, C
6	A, B, C, D
7	B, C
8	A, B, C
9	A, B, D
10	B, C, E

read: TID=1



read: TID=2

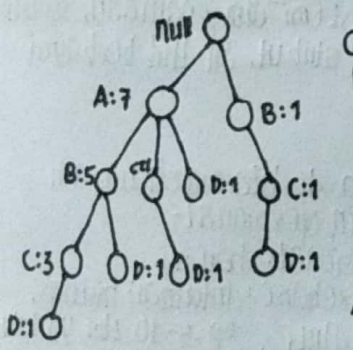


Header Table

Item	Pointer
A	
B	
C	

Pointer: membangun frequent itemset

((lihat di slide))



Conditional Pattern base for D:

P = {(A:1, B:1, C:1), (A:1 B:1), (A:1), (A:1, C:1), (B:1, C:1)}

// nilainya 1: karena muncul bersama D hny 1 transaksi

Frequent Itemset (sup > 1):

AD, BD, CD, ACD, BCD

4 3 3 2 2

ECLAT

(lihat di slide)

Pembangunan Rule

- Itemset frequent: L
- find seluruh subset yg tidak kosong fc L ∋ f → L − f memenuhi conf-min.

eg: {A, B, C, D} adalah frequent itemset, kandidat rules:

ABC → D, ABD → C, ACD → B, BCD → A
A → BCD, B → ACD, C → ABD, D → ABC
AB → CD, AC → BD, AD → BC, BC → AD,
BD → AC, CD → AB

- if |L| = k, maka ada $2^k - 2$ kandidat association rules (dgn mengabaikan L → ∅ dan ∅ → L)

Pruning Berbasis Confidence

if X → Y tidak memenuhi confidence threshold, maka sembarang rule X' → Y − X'; dimana X' = subset dari X, juga hrs tdk dpt memenuhi confidence threshold.

$$c(R_1) = \frac{|ABCD|}{|AB|} \quad c(R_2) = \frac{|ABCD|}{|ABC|} \quad \Big\rangle \; c(R_1) < c(R_2)$$

: R_2 (banyak transaksi) tidak memenuhi, apalagi R_1 (transaksi) nya dikit)

- Confidence: tidak memiliki sifat anti-monotone

eg: c(ABC → D) bisa > atau < dari c(AB → D)

tapi, confidence rules dari itemset sama punya sifat anti-monotone.

eg: L = {A, B, C, D}

c(ABC → D) ≥ c(AB → CD) ≥ c(A → BCD)

TID	Items
10	ABCD
20	BCE
30	ABCE
40	ADE

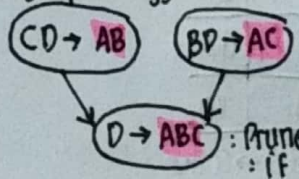
$c(ABC \to D) = \frac{\sigma(ABCD)}{\sigma(ABC)} = \frac{1}{2} = 0.5$

$c(AB \to CD) = \frac{\sigma(ABCD)}{\sigma(AB)} = \frac{1}{3} = 0.33$

$c(A \to BCD) = \frac{\sigma(ABCD)}{\sigma(A)} = \frac{1}{3} = 0.33$

Pembangunan Rule: Apriori

- kandidat rule dibangun dgn cara menggabungkan 2 rules yg punya prefix yg sama dibagian rule consequent.



: Prune rule D → ABC
: if subset AD → BC tidak punya high confidence.

Effect of Support Distribution

- if minsup = tinggi → interesting items kadang tidak muncul, eg. produk harga mahal
- if minsup = kecil → komputasi tinggi, jmh itemset besar

Solusi: multiple minimum supports
└ beda2x tiap barang.

eg: MS(i): minimum support for item i
MS(Milk) = 5%, MS(Coke) = 3%,
MS(Broccoli) = 0,1%, MS(Salmon) = 0.5%

MS({Milk, Broccoli}) = min(MS(Milk), MS(Broccoli))
= min(5%, 0.1%)
= 0.1%

Menghitung Interestingness Measure

Tabel Kontingensi X → Y

	Y	$\bar{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\bar{X}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	|T|

f_{11}: support of X dan Y
f_{10}: support of X dan $\bar{Y}$
f_{01}: support of $\bar{X}$ dan Y
f_{11}: support of $\bar{X}$ dan $\bar{Y}$

utk menentukan various measures:
support, confidence, lift, Gini, J-measure, etc.

Statistical Independence

- Population of 1000 students
  - 600 students : swim (S)
  - 700 students : bike (B)
  - 420 students : swim and bike (S, B)

$\rightarrow P(S \wedge B) = \frac{420}{1000} = 0.42$ //

$P(S) \times P(B) = \frac{600}{1000} \times \frac{700}{1000} = 0.42$ //

Sehingga:

$P(S \wedge B) = P(S) \times P(B)$: statistical independence
$P(S \wedge B) > P(S) \times P(B)$: korelasi positif
$P(S \wedge B) < P(S) \times P(B)$: korelasi negatif

- $Lift = \frac{P(Y|X)}{P(Y)}$. $Interest = \frac{P(X,Y)}{P(X)P(Y)}$
- $PS = P(X,Y) - P(X)P(Y)$
- $\phi\text{-coefficient} = \frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)[1-P(X)]P(Y)[1-P(Y)]}}$

eg:

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: Tea → Coffee

- Confidence = P(Coffee | Tea)
  $= \frac{P(\text{Coffee} \wedge \text{Tea})}{P(\text{Tea})}$
  $= \frac{15}{20} = 0.75$
- $P(\text{Coffee}) = \frac{90}{100} = 0.9$

maka, $lift = \frac{0,75}{0,9} = 0,8333$ (< 1, asosiasi negatif)

K10 Data Warehousing

and On-line Analytical Processing

Data Warehouse

- dikelola terpisah dari database operasional
- pemrosesan informasi dgn menyediakan data historis untuk dianalisis.
- merupakan koleksi data yg punya karakteristik: subject-oriented, integrated, time-variant, non volatile; yg digunakan utk pengambilan keputusan.
- Data Warehousing: proses membangun dan mengg. data warehouse.

1. Subject-Oriented

- diorganisasi menurut subjek penting, eg: customer, product, sales.
- fokus: pemodelan dan analisis data utk pembuatan keputusan, bukan utk operasional harian/pemrosesan transaksi.
- tampilan simpel dan padat thd subjek dgn mengabaikan data yg tidak berguna utk proses pendukung keputusan.

2. Integrated

- terintegrasi dr beberapa database.
  - relational database, flat files, on-line transaction records.
- menerapkan: Data Cleaning & Data Integration
  - memastikan konsistensi dlm penamaan, struktur encoding, satuan atribut, dll. thd berbagai sumber data.

3. Time-Variant

- periode waktu data di data warehouse lbh panjang dari sistem operasional.
  - database operasional: hari ini
    data di data warehouse: informasi historis, eg: 5-10 thn yg lalu.
- setiap struktur key dlm data warehouse:
  - berisi elemen waktu: eksplisit & implisit.

4. Non-Volatile

- tempat penyimpanan terpisah
- pembaruan data operasional tidak dilakukan: update data, recovery data; yg ada hanya 2 operasi akses data: initial loading of data, access of data.

Data Warehouse terpisah:

1. kinerja tinggi utk kedua sistem:
   DBMS - optimasi utk OLTP
   Warehouse - optimasi utk OLAP
2. fungsi yg berbeda dari data yang berbeda
   - Data yg tidak ada: perlu data historis yg tidak dikelola oleh DB operasional.
   - konsolidasi data: dr sumber data yg berbeda.
   - kualitas data: format yg tidak sama / inkonsisten, shg perlu rekonsiliasi.

A Multi-dimensional data model

From Tables & Spreadsheets → Data Cubes
↳ multidimensional data model

- Dimension tables: item (item_name, brand, type), time (day, week, month, quarter, year)
- Fact table: berisi measures dan key ke setiap dimensi tabel yg berhub.

Modeling data warehouses: dimensions & measures

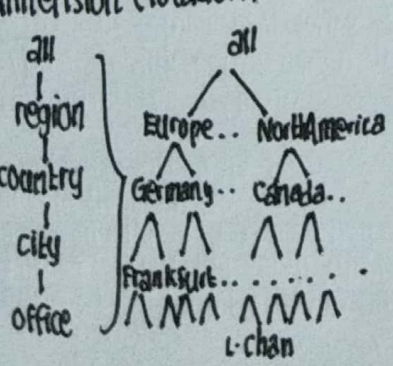
1. Star schema: tabel fakta di tengah yg terhubung sejumlah tabel dimensi.
2. Snowflake schema: perbaikan star schema, tabel fakta 1, terhubung dgn tabel dimensi yg punya anak.
3. Fact constellations: beberapa tabel fakta yg berbagi tabel dimensi. ((koleksi bintang)) (a galaxy))

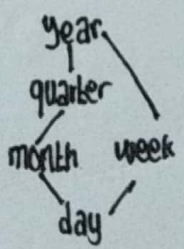
Measures of Data Cube:

three categories

1. Distributive: if hasil turunan dr n aggregate values = turunan dr fungsi tanpa partitioning.
   eg: count(), sum(), min(), max()
2. Algebraic: if dpt dihitung mengg. algebraic funct dgn M argumen (M = bil. int yg sgt besar), setiap perhitungan mengg. distributive aggregate funct.
   eg: avg(), min_N(), std_deviation()
3. Holistic: if there is no constant bound on the storage size needed to describe a subaggregate.
   eg: median(), mode(), rank()

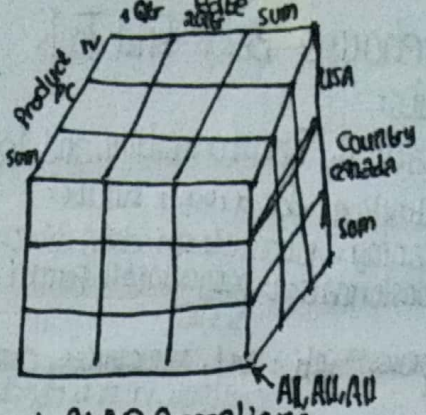
A Concept Hierarchy: dimension (location)





A Sample Data Cube

((lihat slide))

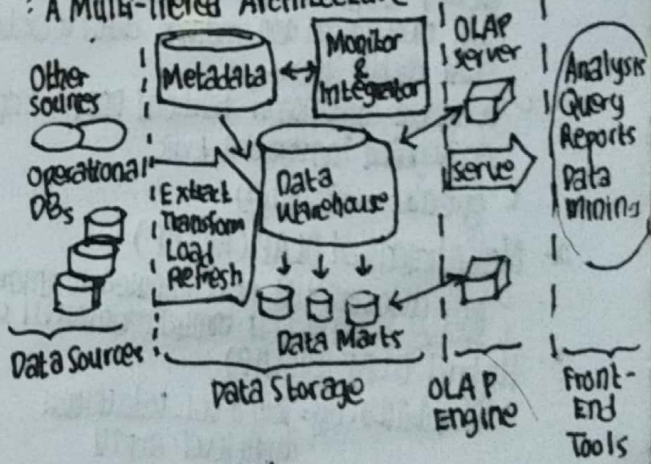


Typical OLAP Operations

1. Roll up (drill-up): summarize data by climbing up hierarchy or by dimension reduction
2. Drill down (roll-down): reverse of roll-up from higher level summary to lower level summary or detailed data, or introducing new dimensions.
3. Slice and dice: project and select.
4. Pivot (rotate): reorient the cube, visualization, 3D to series of 2D planes.
5. Other operations:
   - drill across: involving (across) more than one fact table.
   - drill through: through the bottom level of the cube to its back-end relational tables (using SQL)

Data Warehouse Architecture

: A Multi-Tiered Architecture



Three Data Warehouse Models

1. Enterprise warehouse
   : mencakup seluruh organisasi
2. Data Mart
   : kelompok pengguna terkendu / terbatas.
   eg. marketing data mart.
3. Virtual warehouse
   : sekumpulan view dari hasil query utk database operasional.

→

Data Warehouse Back-End Tools and Utilities:

ETL (Extraction, Transformation, and Loading)

1. Data extraction: dari berbagai sumber
2. Data cleaning: inkonsistensi data diperbaiki.
3. Data transformation: penyesuaian format agar sesuai.
4. Load: proses2x ini : sort, summarize, consolidate, compute views, check integrity, dan build indicies and partitions.
5. Refresh: update data sources ke warehouse.

Metadata Repository

: data yg mendefinisikan warehouse objek. Berisi:

1. deskripsi struktur data warehouse: schema, view, dimensions, hierarchies, data definition turunan, lokasi isi data mart.
2. Operational meta-data: siapa saja yang make, kapan saja, pake cara apa
3. Algoritme utk summarization
4. Mapping

OLAP Server Architectures

1. Relational OLAP (ROLAP)
   - menggu. DBMS relational atau extended-relational utk menyimpan dan manage data warehouse dan OLAP middle ware.
   - termasuk optimisasi backend DBMS, implementasi aggregation navigation logic
   - greater scallability.
2. Multidimensial OLAP (MOLAP)
   - fast indexing utk pre-computed summarized data.
   - sparse array-based multidimensional storage engine.
3. Hybrid OLAP (HOLAP)
   - flexibility, eg: low level: relational
     high level: array
   - eg: Microsoft SQLServer
4. Specialized SQL Servers
   - specialized support for SQL queries over star/snowflake schemas.
   - eg: Redbricks.

Data Warehouse Implementation

Indexing OLAP Data:

1. Bitmap Index

eg: Base Table

Cust	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retail

Index on Region

RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1

Index on Type

RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0

2. Join Indices

From Data Warehousing to Data Mining

Data Warehouse Usage

1. Information processing
2. Analytical processing
3. Data mining

GINI

C1: play tennis
C2: no played

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Outlook

Sunny: C1: 2, C2: 3
Overcast: C1: 4, C2: 0
Rain: C1: 3, C2: 2

- GINI(Sunny) = $1 - P(C1)^2 - P(C2)^2$
  $= 1 - (2/5)^2 - (3/5)^2 = 0.48$
- GINI(Overcast) = $1 - (4/4)^2 - (0/4)^2$
  $= 1 - 1 = 0 = 0$
- GINI(Rain) = $1 - (3/5)^2 - (2/5)^2 = 0.48$

GINI(Outlook) = $5/14 \cdot (0.48) + 4/14(0) + 5/14 \cdot (0.48)$
$= 0.34$ ((root))

Temperature

Hot: C1: 2, C2: 2
Mild: C1: 4, C2: 2
Cool: C1: 3, C2: 1

- GINI(Hot) = $1 - (2/4)^2 - (2/4)^2 = 0.5$
- GINI(Mild) = $1 - (4/6)^2 - (2/6)^2 = 0.444$
- GINI(Cool) = $1 - (3/4)^2 - (1/4)^2 = 0.375$

GINI(Temperature) = $4/14(0.5) + 6/14(0.444) + 4/14(0.375)$
$= 0.44$

HUMIDITY

High: C1: 3, C2: 4
Normal: C1: 6, C2: 1

- GINI(High) = $1 - (3/7)^2 - (4/7)^2 = 0.49$
- GINI(Normal) = $1 - (6/7)^2 - (1/7)^2 = 0.25$

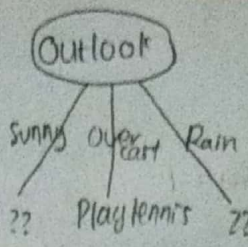
GINI(Humidity) = $7/14(0.49) + 7/14 \cdot (0.25) = 0.37$

WIND

Weak: C1: 6, C2: 2
Strong: C1: 3, C2: 3

- GINI(Weak) = $1 - (6/8)^2 - (2/8)^2 = 0.375$
- GINI(Strong) = $1 - (3/6)^2 - (3/6)^2 = 0.5$

GINI(Wind) = $8/14(0.375) + 6/14(0.5) = 0.43$



a) ((Sunny, Temperature))

(Sunny, Hot): C1: 0, C2: 2
(Sunny, Mild): C1: 1, C2: 1
(Sunny, Cool): C1: 1, C2: 0

- Gain(Sunny, Hot) = $1 - (0/2)^2 - (2/2)^2 = 0$
- Gain(Sunny, Mild) = $1 - (1/2)^2 - (1/2)^2 = 0.5$
- Gain(Sunny, Cool) = $1 - (1/1)^2 - (0/1)^2 = 0$

GAIN(Sunny, Temperature) = $2/5(0) + (2/5)(0.5) + 1/5(0)$
$= 0.2$

b) ((Sunny, Humidity))

(Sunny, High): C1: 0, C2: 3
(Sunny, Normal): C1: 2, C2: 0

- Gain(Sunny, High) = $1 - (0/3)^2 - (3/3)^2 = 0$
- Gain(Sunny, Normal) = $1 - (2/2)^2 - (0/2)^2 = 0$

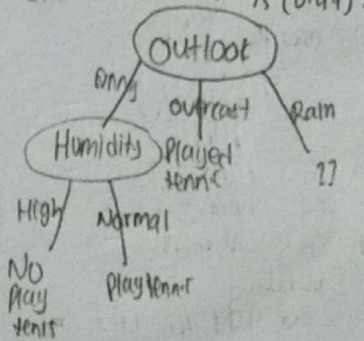
GAIN(Sunny, Humidity) = $3/5(0) + (2/5)(0) = 0$ ((leaf node for Sunny))

c) ((Sunny, Wind))

(Sunny, Weak): C1: 1, C2: 2
(Sunny, Strong): C1: 1, C2: 1

- Gain(Sunny, Weak) = $1 - (1/3)^2 - (2/3)^2 = 0.44$
- Gain(Sunny, Strong) = $1 - (1/2)^2 - (1/2)^2 = 0.5$

GAIN(Sunny, Wind) = $3/5(0.44) + 2/5(0.5) = 0.464$



d) ((rain, temperature))

(rain, hot): C1: 0, C2: 0
(rain, mild): C1: 2, C2: 1
(rain, cool): C1: 1, C2: 1

- GINI(rain, hot) = $1 - (0/0)^2 - (0/0)^2 =$ #
- GINI(rain, Mild) = $1 - (2/3)^2 - (1/3)^2 = 0.44$
- GINI(rain, cool) = $1 - (1/2)^2 - (1/2)^2 = 0.5$

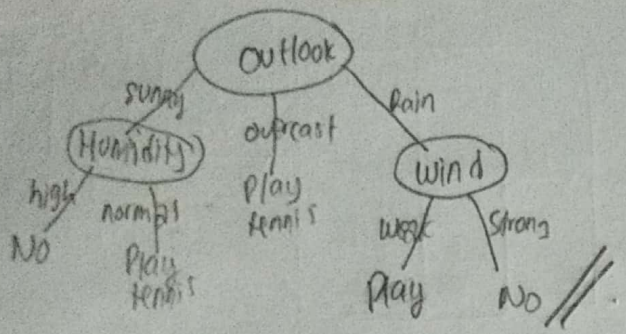
GINI(rain, temperature) = $0/5(1) + (3/5)(0.44) + 2/5(0.5) = 0.464$

e) ((rain, wind))

(rain, weak): C1: 3, C2: 0
(rain, strong): C1: 0, C2: 2

- GINI(rain, weak) = $1 - (3/3)^2 - (0/3)^2 = 0$
- Gain(rain, strong) = $1 - (0/2)^2 - (2/2)^2 = 0$

GINI(rain, wind) = $3/5(0) + 2/5(0) = 0$ ((leaf node of Rain))



data mining

#8. Analisis Asosiasi

- ARM: diberikan himpunan transaksi, cari aturan yg memprediksi kemunculan sebuah item berdasarkan kemunculan item lain dalam transaksi.
  eg: Transaksi Market-Basket

TID	Items

- Definisi:
  1. Itemset: sekumpulan 1/lebih items. eg: {Milk, Bread, Diaper}
     ↳ k-itemset: itemset yg mengandung k items.
  2. Support Count (σ): frekuensi kemunculan sebuah itemset. (support absolute).
     eg: σ({Milk, Bread, Diaper}) = 2
  3. Support (s): fraksi transaksi yg mengandung sebuah itemset. (support relative).
     eg: s({Milk, Bread, ...}) = 2/5
  4. Frequent Itemset: sebuah itemset dikatakan *support* jika ≥ minsup threshold.
  5. Association Rule: ekspresi implikasi dr bentuk $X \rightarrow Y$, dimana X dan Y = itemsets.
     eg: {Milk, Diaper} → {Juice}.
  6. Rule Evaluation Metrics
     - Support (s): fraction of transactions that contain both X and Y.
     - Confidence (c): measures how often items in Y appear in transactions that contain X.

eg: {Milk, Diaper} ⇒ Juice

$$s = \frac{\sigma(\{Milk, Diaper, Juice\})}{|T|} = \frac{2}{5} = 0.4 \qquad c = \frac{\sigma(\{Milk, Diaper, Juice\})}{\sigma(\{Milk, Diaper\})} = \frac{2}{3} = 0.67$$

- Association Rule Mining Task
  ↳ find all rules having: support ≥ minsup, confidence ≥ minconf
  ↳ Brute-force approach:
    1. List all possible association rule.
    2. Compute the support and confidence for each rule.
    3. Prune rules that fail the minsup & minconf thresholds.

- Mining Association Rules
  Two step approach:
  1. Frequent Itemset Generation
     : generate all itemsets whose support ≥ minsup
  2. Rule Generation
     : generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset.

1. Frequent Itemset Generation

Given d unique items:
- Total number of itemsets = 2^d
- Total number of possible association rules:

$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right] = 3^d - 2^{d+1} + 1$$

eg: $d=6 \rightarrow R = 3^6 - 2^{6+1} + 1 = 602$ rules.

- complexity ~ $O(NMw)$ → expensive since $M = 2^d$

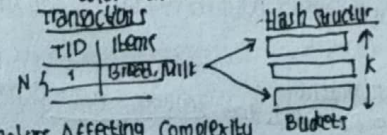
Solusi:
1. Mengurangi jmh kandidat (M)
   - complete search: $M = 2^d$
   - use pruning techniques to reduce M.
2. Mengurangi jmh transaksi (N)
   - Reduce size N as the size of itemset increases.
   - Use vertical-partitioning of the data to apply the mining algorithms.
3. Mengurangi jmh comparisons (NM).
   - Use efficient data structures to store the candidates or transactions.
   - No need to match every candidate against every transaction.

Prinsip Apriori: jika itemset frequent, maka semua subsets harus frequent.

$\forall X, Y: (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$

Method:
- Let k=1
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified.
  ↳ Generate length (k+1) candidate itemsets from length k frequent itemsets
  ↳ Prune candidate itemsets containing subsets of length k that are infrequent.
  ↳ Count the support of each candidate by scanning the DB.
  ↳ Eliminate candidates that are infrequent, leaving only those that are frequent.

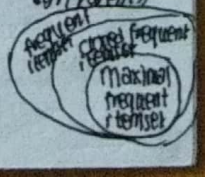
- Reducing Number of Comparisons
  ↳ candidate counting:
  1. scan the database of transactions to determine the support of each candidate itemset.
  2. To reduce the number of comparisons, store the candidates in a hash structure.
     - Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets.



- Factors Affecting Complexity
  1. Choice of minimum support threshold
     - makin kecil support threshold, makin byk frequent itemset.
     - dpt meningkatkan jmh kandidat dan memaksimumkan length of frequent itemset.
  2. Dimensionality (number of items) of the dataset.
     - butuh banyak space utk mengimpan support count setiap item.
     - jika jmh frequent items naik, maka biaya komputasi dan I/O yg meningkat.
  3. Size of database
     - jmh transaksi meningkat, runtime algoritme meningkat.
  4. Average transaction width
     - transaction width increases with denser datasets.
     - menyebabkan peningkatan max length of frequent itemsets dan traversals of hash tree (number of subsets in a transaction increases with its width).

- Maximal Frequent Itemset
  : jika none of its immediate supersets is frequent.
  (tidak ada item frequent dgn parent-nya))

- Closed Itemset
  : jika none of its immediates superset has the same support as the itemset.
  (tidak ada item anaknya yg support-nya sama dgn parent)



- Representation of Database
1-Horizontal Data Layout
TID	Items
1	A, B, E
2	B, C, D
3	C, E
4	A, C, D

2-Vertical Data Layout
A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

eg: minsup=2

TID	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan → C1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

→ L1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

2nd scan → C2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

→ L2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

3rd scan → C3

Itemset	sup
{B, C, E}	2

→ L3

Itemset	sup
{B, C, E}	2

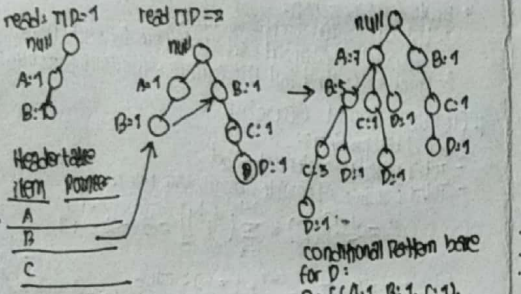
sebenarnya: {A, B, C}, {A, C, E}, {A, B, E}, {B, C, E} → terelimin. si, shg tidak diambil.

+ Apriori: horizontal data format: diurutkan berdasar TID.

#9. FP-Growth
- menggunakan FP-tree sbg representasi terkompresi dr database.
- build tree → pake pendekatan divide and conquer utk dapetin frequent & infrequent itemset.
- Caranya:
1-Baca per transaksi
2-Mulai dgn node null, lalu buat node item baru. Jika item sudah ada, pakai yg lama
3-Node item akan bertambah kemunculannya.
4-Bikin pointer.

eg:

TID	Items
1	A, B
2	B, C, D
3	A, C, D, E
4	A, D, E
5	A, B, C
6	
7	



conditional Pattern base for D:
D = {(A:1, B:1, C:1), (A:1, B:1), (A:1, C:1), (B:1, C:1)}

#1: karena muncul bertama D hanya 1 transaksi.
Frequent itemset (sup ≥ 1):
AD, BD, CD, ACD, BCD
4 3 3 2 2

ECLAT
└ vertical data format: disusun berdasarkan item.

eg:

Itemset	TIDSet
I1	{T100, T400, ...}
I2	{T100, T200, ...}

atau

Item	TID
A	{1, 4, 5, 6, ...}
B	
D	

- Pembangunan Rule
- itemset frequent: L
- find seluruh subset yg tidak kosong f ⊂ L → f → L − f memenuhi conf-min.
eg: {A, B, C, D} adh frequent itemset, kandidat rules:
ABC → D, ABD → C, ACD → B, BCD → A
A → BCD, B → ACD, C → ABD, D → ABC
AB → CD, AC → BD, AD → BC, BC → AD,
BD → AC, CD → AB
- If |L| = k, maka ada $2^k - 2$ kandidat association rules (dgn mengabaikan L → ∅ dan ∅ → L)

- Pruning Berbasis Confidence
if X → Y tidak memenuhi confidence threshold, maka semarang rule X' → Y − X'; dimana X' = subset dr X, yg hrs tidak dpt memenuhi confidence threshold.

$c(R1) = \frac{|AB \cup CD|}{|AB|}$

$c(R2) = \frac{|ABC \cup D|}{|ABC|}$

c(R1) < c(R2): R2 (byk transaksi) tidak memenuhi, analogi R1 (transaksi dikit).

- Confidence: tidak memiliki sifat antimonotone.
eg: c(ABC → D) bisa > atau < dr c(AB → D), tapi confidence ruler dr itemset sama yg sifat antimonotone.
eg: L = {A, B, C, D}

$c(ABC \to D) \geq c(AB \to CD) \geq c(A \to BCD)$

TID	Items
10	ABCD
20	BCE
30	ABCE
40	ABE

$c(ABC \to D) = \frac{\sigma(ABC \cup D)}{\sigma(ABC)} = 1/2 = 0.5$

$c(AB \to CD) = \frac{\sigma(AB \cup CD)}{\sigma(AB)} = 1/3 = 0.33$

$c(A \to BCD) = \frac{\sigma(A \cup BCD)}{\sigma(A)} = 1/3 = 0.33$

- Pembangunan Rule: Apriori
└ kandidat rule dibangun dgn cara menggabungkan 2 rules yg panjangnya prefix yg sama dibagian rule consequent.

(CD → AB) (BD → AC)

↓

(D → ABC): prune rule D → ABC if subset AD → BC tidak punya high confidence.

- Effect of Support Distribution
- If minsup = tinggi: interesting items kadang tidak muncul, eg: produk hanya mahal.
- If minsup: rendah: komputasi lambat, jmh itemset besar
Solusi: multiple minsup.
└ beda2 tiap barang. eg: MS(i): minsup for item i
MS(Milk) = 5%, MS(Coke) = 3%, MS(Broccoli) = 0.1%, MS(Salmon) = 0.5%
MS({Milk, Broccoli}) = min(MS(Milk), MS(Broccoli)) = min(5%, 0.1%) = 0.1%

- Menghitung Interestingness Measure
Tabel kontingensi X → Y

	Y	$\bar{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\bar{X}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	|T|

f_{11} = support of X dan Y
f_{10}: X dan $\bar{Y}$
f_{01}: $\bar{X}$ dan Y
f_{00}: $\bar{X}$ dan $\bar{Y}$

utk menentukan various measures: support, conf, lift, Gini, J-measure, etc.

Statistical Independence
Population of 1000 students.
- 600 students: Soccer (S)
- 700 students: Bike (B)
- 420 students: swim and bike (S, B)

$P(S \wedge B) = \frac{420}{1000} = 0.42$

$P(S) \times P(B) = \frac{600}{1000} \times \frac{700}{1000} = 0.42$

Sehingga: $P(S \wedge B) = P(S) \times P(B)$: statistically independent
$>$: korelasi (+)
$<$: korelasi (−)

- $Lift = \frac{P(Y|X)}{P(Y)}$
- $Interest = \frac{P(X,Y)}{P(X)P(Y)}$
- $PS = P(X,Y) - P(X)P(Y)$
- $\phi\text{-coefficient} = \frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)[1-P(X)]\,P(Y)[1-P(Y)]}}$

eg:

	Coffee	$\bar{Coffee}$	
Tea	15	5	20
$\bar{Tea}$	75	5	80
	90	10	100

Association Rule: Tea → Coffee
- Confidence = P(Coffee | Tea) = P(Coffee ∧ Tea) / P(Tea) = 15/20 = 0.75
P(Coffee) = 90/100 = 0.9,
maka lift = 0.75/0.9 = 0.8333 < 1, jadi jika negatif

- metric: transaction throughput / query throughput, response.

	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date, detailed, flat-relational, isolated	historical, summarized, multidimensional, integrated, consolidated
usage	repetitive	ad-hoc
access	Read/write, index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
# users	thousands	hundreds
DB size	100 MB-GB	100 GB-TB

3. Fact constellations: beberapa tabel fakta yg berbagi tabel dimensi. (kalo ini bintang/galaxy)

#10. Data Warehousing & On-line Analytical Processing
- Data Warehouse
└ dikelola terpisah dari database operasional
└ pemrosesan informasi dgn menyediakan data historis utk dianalisis.
└ mirip koleksi data yg punya karakteristik: subject-oriented, integrated, time-variant, non volatile. yg digunakan utk pengambilan keputusan.
└ Data Warehousing: proses membangun & meng-data warehouse.

1. Subject-Oriented
- diorganisasi menurut subjek penting, eg: customer, product, sales.
- fokus: pemodelan dan analisis data utk pembuatan keputusan, bukan utk operasional harian/pemrosesan transaksi.
- tampilan simpel dan ringkas thd subjek dan menghasilkan data yg tidak berguna utk proses pendukung keputusan.

2. Integrated
- terintegrasi dr beberapa database.
└ relasional db, flat files, on-line transaction records
- menerapkan: data cleaning & data integration.
└ memastikan konsistensi dlm penamaan, struktur encoding, satuan ukuran, dll. thd berbagai sumber data.

3. Time-Variant
- periode waktu data di data warehouse lbh pjg dr sistem operasional.
└ database operasional: hari ini
data di data warehouse: informasi historis. eg: 5-10 thn yg lalu.
- setiap struktur key dlm data warehouse: berisi elemen waktu: eksplisit & implisit.

4. Non-Volatile
- tempat penyimpanan terpisah
- pembaruan data operasional tidak dilakukan (update + recovery data); yg adahnya 2 operasi akses data: initial loading of data, access of data.

- Data Warehouse terpisah:
1. kinerja tinggi utk kedua sistem ← DBMS-dioptimasi utk OLTP; Warehouse-dioptimasi utk OLAP
2. fungsi yg berbeda dr data yg berbeda:
- data yg tidak ada: perlu data historis yg tidak dikelola oleh DB operasional
- konsolidasi data: dr sumber data yg berbeda.
- kualitas data: format yg tidak sama/inkonsisten shg perlu rekonsiliasi.

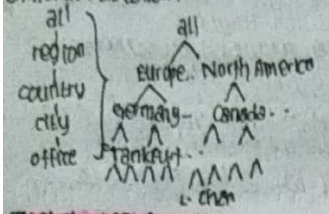
A Multi-dimensional data model
From tables & spreadsheets → data cubes: multidimensional data model
- Dimension tables: item (item_name, brand, type), time (day, week, month, quarter, year).
- Fact table: berisi measure & key ke setiap dimensi table yg berhub.

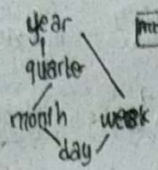
- Modeling data warehouses: dimensions & measures.
1. Star schema: tabel fakta di tengah yg terhubung sejumlah tabel dimensi.
2. Snowflake schema: perluasan star schema, tabel fakta, terhub dgn tabel dimensi yg punya anak.

Measures of Data Cube: 3 categories

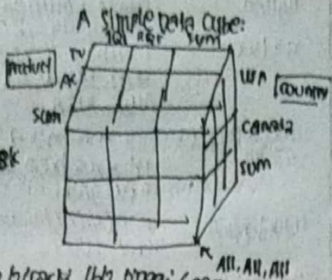
1. Distributive: if hasil turunan dr n aggregate values = turunan dr fungsi tanpa perhitungan.
   eg: count(), sum(), min(), max().
2. Algebraic: if dpt dihitung mengg. algebraic funct dgn M argumen (M = bil. int yg sgt bsr), setiap perhitungan mengg. distributive aggregate funct.
   eg: avg(), min_N(), std-deviation()
3. Holistic: if there is no constant bound on the storage size needed to describe a sub aggregate.
   eg: median(), mode(), rank().

A Concept Hierarchy:
dimension (location)





A Simple Data Cube:

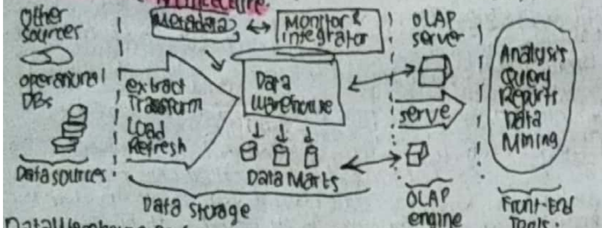


Typical OLAP Operations

1. Roll up (drill-up): summarize data; dgn naik ke hirarki lbh tinggi / pengurangan dimensi.
2. Drill down (roll-down): reverse of roll up, from higher level summary to lower level summary or detailed data, or introducing new dimensions.
3. Slice and dice: project and select. mengambil kolom.
4. Pivot (rotate): reorient the cube, visualization, 3D to series of 2D planes.
5. Other operations: - drill across: involving (across) more than 1 fact table.
   - drill through: through the bottom level of the cube to its back end relational tables (using SQL).

Data Warehouse Architecture

= A Multi-Tiered Architecture



Three Data Warehouse Models:
1. Enterprise warehouse: mencakup seluruh organisasi
2. Data Mart: kelompok pengguna tertentu / terbatas. eg: marketing data mart.
3. Virtual Warehouse: sekumpulan view dr hasil query utk DB operasional.

Data Warehouse Back-End Tools & Utilities:
ETL (Extraction, Transformation, and Learning)
1. Data extraction: dr berbagai sumber
2. Data cleaning: inkonsistensi data dibenerin
3. Data transformation: penyesuaian format agar sesuai.
4. Load: proses² inti: sort, summarize, consolidate, compute views, check integrity, dan build indices and partitions.
5. Refresh: update dat² sources ke warehouse.

Metadata Repository: data yg mendefinisikan warehouse objek. Berisi:
1. deskripsi struktur data warehouse: schema, view, dimensions, hierarchies, data definition, turunan, lokasi, isi data mart.
2. operational meta-data: siapa saja yg pke, kapan aja, pake cara apa.
3. algoritme utk summarization
4. Mapping

OLAP Server Architectures
1. Relational OLAP (ROLAP)
   - mengg. DBMS relational atau extended-relational utk menyimpan & manage data warehouse dan OLAP middle ware.
   - termasuk optimasi backend DBMS, implementasi aggregation navigation logic.
   - greater scalability.
2. Multidimensional OLAP (MOLAP)
   - fast indexing utk pre-computed summarized data.
   - sparse array-based multidimensional storage engine.
3. Hybrid OLAP (HOLAP)
   - flexibility: eg: low level: relational, high level: array
   - eg: Microsoft SQLServer
4. Specialized SQL servers: specialized support for SQL queries over star / snowflake schemas.
   eg: Redbricks.

Data Warehouse Implementation
Indexing OLAP Data:
1. Bitmap Index
   eg: Base Table

Cust	Region	Type
C1	Asia	Retailer
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retailer

Index on Region

RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1

Index on Type

RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0

2. Join Indices

From Data Warehousing to Data Mining
Data Warehouse Usage:
1. Information processing 2. Analytical processing 3. Data Mining

#11. Spatiotemporal Data Mining

- Data spasial: generalisasi titik geografis yg detail mjd area terklaster. ada penggabungan sejumlah area geografis mengg. operasi spasial.
- Spasial Database Systems: utk mengelola data spasial.
  - Image db system: handling digital raster image (eg: satelite sensing, computer tomography), teknik analisis objek & ekstraksi dr image dan some spatial db functionality.
  - Spatial (geometric, geographic) db system: handling objek yg punya identity and well-defined extents, locations, and relationship (id, ukuran, & hub. antar data).

GIS (Geographic Information System): analisis & visualisasi data geografis.
- fungsi analisis: - kemiringan, dll.
  1. pencarian (tematik, per area).
  2. Analisis lokasi (buffer, overlay). aliran air dr sumber sampai apa.
  3. Analisis topografi (slope / aspect, drainage network)
  4. Flow analysis (connectivity, shortest path): jarak terpendek dr sources ke dest.
  5. Distribution (nearest neighbor, proximity, change detection) : jarak terdekat.
  6. Analisis / statistics spasial (pattern, centrality, similarity, topology).
  7. Pengukuran (distance, perimeter, shape, adjacency, direction).

Model dari Objek Spasial:
1. Single object: tiap entity diolah dlm area terpisah. (Point, Line, Region)
   - Model kota, hutan, sungai
2. Kumpulan objek yg pny hub. spasial: deskripsi areanya langsung (setiap point dlm area.)
   - model pengg. lahan, pembagian negara mjd provinsi / distrik / negara ... dlm area.
1. Point: hanya lokasi, tanpa ukuran / area (pusat kota, stasiun cuaca, perpotongan jalan.
2. Line (atau polyline / sekuen dr line segment): jalan, sungai, kabel, instalasi listrik.
3. Region: ruang 2D (negara, danau, taman): terdiri dr bagian yg tidak terhubung.

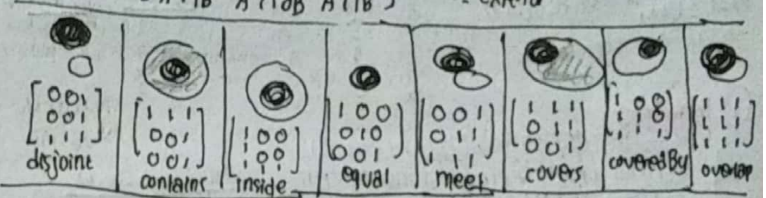
Representasi Fitur Spasial
1. Discrete features: fitur individual yg dapat dibedakan
   - objek 1 dgn objek lain tdk ada objek ditengahnya.
   - termasuk points, lines, polygons.
2. Continuous features: fitur yg muncul secara kontinu diantara observasi. eg: elevasi, curah hujan, disetiap tempat curah hujan nilai beda².

Spatial Relationship
1. Topological - relations: touches, overlap, contains, inside, equal, disjoint, intersect.
2. Metric - relations: distance < d
3. Direction - relation: north, south, west, east

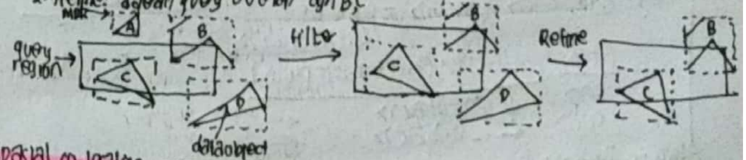
Spatial Relationship dlm matriks

$$\Im_g(A,B) = \begin{bmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{bmatrix}$$

∂ = boundary
$\circ$ = interior
$-$ = exterior



Query Processing
- spatial query language: SQL3, OGIS
- common strategy: filter & refine.
  1. Filter: daerah query overlap dgn MBRs (minimum bounding rectangles) of B, C, D
  2. Refine: daerah query overlap dgn B, C



Spasial co-location: mengenal kemunculan bersama dlm sebuah lokasi / berdekatan.
eg: kebakaran hutan & pohon, burung gereja & perumahan, atau stasiun didekatnya ada rumah sakit → bisa ada peluang jadi ojek online disekitar stasiun karena akan ada byk orang yg mau ke RS.

Spatial Classification
eg: klasifikasi nilai rumah berdasarkan kedekatan dgn danau, jalan raya, gunung, etc.
- Metode spasial classification populer:
  1. Spatial Auto-regression (SAR) 2. Markov Random Field (MRF)
- Algoritme Decision Tree pd spatial data: perlu diproses, merubah algoritme / diimprove.

Spatial Trend Analysis: mendeteksi perubahan & trend sepanjang satu dimensi spasial.
eg: observasi trend perubahan iklim / vegetasi seiring meningkatnya jarak dr laut, tingkat kriminalitas / pengangguran.

Spatial Cluster Analysis
: pengelompokkan daerah² berdasarkan ciri yg sama. dari daerah² yg dikelompokkan mjd cluster, cluster² tsb nantinya dpt teridentifikasi fitur unik yg dpt digunakan utk pemetaan, dibedakan berdasarkan warna.

Spatio-temporal data: data yg memiliki informasi spasial dan berubah seiring waktu. eg: kebakaran hutan, objek bergerak, badai, gempa bumi.

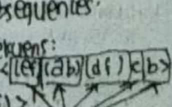
#12. Sequential Pattern Mining

- diberikan himpunan sekuens dan support threshold, temukan himpunan lengkap dari frequent subsequences.

Basis data sekuens

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

kejadian terurut: a, (abc), (ac) ...



sebuah elemen dpt mengandung sekumpulan item. item dlm sebuah elemen adalah tdk terurut dan dituliskan secara alfabetik.

α: subsekuens dr β ($\alpha \subseteq \beta$)
β: supersekuens dr α
eg: α = <(ab), d>
β = <(abc), (de)>

<a(bc)dc> adh subsequence dr <a(abc)(ac)d(cf)>
<(ab)c(de)> adh supersequence dr <(ab)d>
diberikan support threshold min-sup = 2, <(ab)c> adh sequential pattern

- Aplikasi:
  - Urutan pembelajaan dr konsumen: pertama membeli komputer, kemudian CD-ROM, dan kemudian digitalcamera, dlm 3 bulan.
  - perawatan medis, bencana alam (eg: gempa bumi), proses sains dan rekayasa, stock dan pemasaran, dll.
  - pola panggilan telepon, weblog click streams
  - sekuens DNA dan struktur gen.

- Metode²:
  1. Pendekatan berbasis Apriori < GSP / SPADE
     - sifat dasar: jika sebuah sekuens S adh tidak frequent, maka tidak ada dari super-sekuens dr S yg frequent.
       eg: <hb>: infrequent → <hab> dan <(ah)b> ~~that~~ infrequent.
  2. Pendekatan berbasis Pattern-Growth:
     - FreeSpan
     - PrefixSpan

① GSP (Generalized Sequential Pattern) Mining ((sama proses dgn Apriori di ARM))

min-sup = 2

seq. ID	sequence
10	<(bd)cb(ac)>
20	...

→ 1st scan →

kandidat	sup
<a>	..
<b>	..
<c>	..

→ kemunculan tiap transaksi

- keuntungan dr Apriori pruning: mengurangi ruang pencarian
- Bottlenecks:
  - scan basis data berkali²
  - membangkitkan himpunan besar dr sekuens kandidat.
  ((perlu metode mining yg lbh efisien))

② SPADE (Sequential Pattern Discovery Using Equivalent Class)
- format vertikal
- basis data sekuens dipetakan ke himpunan berukuran besar dr item: <SID, EID>

eg:

SID	sequence
1	<a(abc)(ac)>
2	<(ad)c(bc)>

bikin 2 tabel (2 kejadian + 1 kejadian)

→

SID	EID	items
1	1	a
1	2	abc
1	3	ac
2	1	ad
2	2	c
2	3	bc

→

a

SID	EID
1	1
1	2
1	3
2	1

b

SID	EID
1	2
2	3

...

↓

ab

SID	EID(a)	EID(b)
1	1	2
2	1	3

ba

SID	EID(b)	EID(a)
1	2	3
2	3	4

③ PrefixSpan (Prefix-Projected Sequential Pattern Growth)
- berbasis proyeksi, tetapi hnya proyeksi berbasis prefiks: proyeksi yg sedikit & jmh sekuens menyusut dgn cepat.
- cara kerja: cari suffix dgn panjang prefix bertambah

eg: <a>, <aa>, <a(ab)> = prefix dr <a(abc)(ac)>

prefix	suffix (proyeksi berbasis prefix)
<a>	<(abc)(ac)>
<aa>	<(_bc)(ac)>

efisiensi PrefixSpan:
- tidak ada kandidat sekuen yg perlu dibangkitkan
- projected databases, tetapi berukuran kecil
- biaya dlm PrefixSpan: membuat projected databases, dpt diperbaiki dgn bi-level projection.

Kendala pd SPM
1. kendala item: carilah pola weblog yg hny terkait online-bookstore.
2. kendala panjang (length): cari pola yg memiliki sedikitnya 20 barang.
3. kendala super pattern: cari super patterns dr "PC → digitalcamera"
4. kendala agregat: cari pola yg rata² harga barangnya melebihi $100
5. kendala ekspresi reguler:
   - carilah pola "starting from Yahoo homepage, search for hotels in Washington DC area".
   - Yahootravel (WashingtonDC | DC)(hotel | motel | lodging)
6. kendala durasi: carilah pola sekitar ± 24 jam dr pengambilan
7. kendala gap: carilah pola pembelian dgn gap antara setiap pembelian yg berurutan kurang dr 1 bulan".

#13 Text & Web Mining

- data mining yg mengg. dokumen teks sbg data.
- mengg. metode Information Retrieval (IR) utk praproses dokumen teks.
- proses yg secara otomatis mengekstrak informasi komprehensif yg bermakna, berguna dan blm diketahui sebelumnya dr repository dokumen tekstual.

- Text Mining Task
  1. Diberikan:
     - sumber dokumen tekstual.
     - kueri terbatas (berbasis teks) yg didefinisikan dgn baik.
  2. Temukan:
     - kalimat dgn informasi relevan.
     - ekstrak informasi relevan & abaikan informasi yg tdk relevan.
     - hubungkan informasi & keluaran yg slg berhub. dlm format yg sdh ditetapkan sebelumnya.

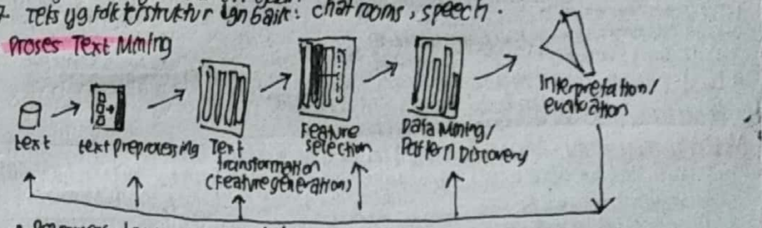
- Task:
  - search & retrieval
  - analisis semantic
  - clustering
  - kategorisasi
  - feature extraction
  - pembuatan ontologi
  - dynamic focusing

	Data Mining	Text Mining
objek yg diteliti	data numerik & kategorikal	text
struktur objek	basdat relasional	free form text
tujuan	prediksi outcome dr situasi mendatang	temu kembali informasi yg relevan, distill the meaning, mengkategorikan & penyampaian target.
metode	machine learning: SKAT, DT, NN, GA, MBR, MBA	indexing, pemrosesan neural network khusus, linguistik, ontologies
ukuran market saat ini	100.000 analist pd perusahaan besar & menengah	100.000.000 pekerja korporat dan pengguna individual
kematangan	implementasi yg luas sejak 1994	implementasi yg luas mulai 2000.

- Aplikasi,
  1. Pemasaran: menemukan kelompok pembeli yg potensial berdasarkan profil teks pengguna. eg: amazon
  2. Industri: mengidentifikasi situs web kelompok pesaing. eg: produk pesaing & harganya.
  3. Pencarian kerja: mengidentifikasi parameter dlm pencarian pekerjaan. eg: www.flipdog.com
  4. Search engines
  5. Enterprise portals
  6. knowledge management systems
  7. e-Business systems
  8. vertical apps:
     - kategorisasi & routing email
     - kategorisasi catatan call center
     - sistem CRM.

- Karakteristik Teks:
  1. Basis data tekstual berukuran besar
     - pertimbangan efisiensi < 2.000.000.000 hal. web / hmpir semua publikasi yg tersedia dlm bentuk elektronik.
  2. Dimensi tinggi (sparse input): pertimbangan setiap word/phrase sbg sebuah dimensi.
  3. Beberapa mode input.
     - eg: web mining: informasi mengenai pengguna dibangkitkan oleh semantics, mencari pola & outside knowledgebase.
  4. Ketergantungan (dependensi): informasi yg relevan adh hub. conjunction yg kompleks dr word/phrase. eg: kategorisasi dokumen, disambigu pronoun.
  5. Ambiguitas < ambiguitas kata: pronouns (he, she, ...), "buy" "purchase" / ambiguitas semantik: the king saw the rabbit with his glasses (8 makna).
  6. Noisy data: eg: kesalahan ejaan
  7. Teks yg tdk terstruktur dgn baik: chat rooms, speech.

- Proses Text Mining



- Praproses teks < analisis teks semantic / syntactic
- Pembangkitan fitur: bag of words
- seleksi fitur < simple counting / statistics
- Text/Data Mining
  - Classification: supervised learning
  - Clustering: unsupervised
- analisis hasil.

// Analisis teks synthetic/semantic
1. Part of speech (pos) tagging
   - tentukan pos yg sesuai utk setiap word. eg: John (noun) gave (verb) the (det) ball (noun).
2. Word sense disambiguation
   - berbasis konteks / proksimity
   - sgt akurat ~98% akurat.
3. Parsing < membangkitkan parse tree (graf) utk setiap kalimat / setiap kalimat adh graf yg berdiri sendiri.

// Pembangkitan fitur: Bag of words
- dokumen teks direpresentasikan oleh word yg terdapat didalamnya (& kemunculannya)
  eg: "Lord of the rings" → {"the", "Lord", "rings", "of"}
  sgt efisien, membuat proses learning mjd lbh sederhana & mudah, urutan kata tdk begitu penting utk aplikasi tertentu.
- stemming: mengidentifikasi kata dgn rootnya. eg: flying, flew → fly.
  - mengurangi dimensionalitas.
- stop words: kata² umum yg tdk membantu proses text mining. eg: "the", "a", "an", "you", ...

// Seleksi fitur
- mengurangi dimensi: proses learning mrpkn tugas yg sulit pd data dimensi yg tinggi
- fitur tdk relevan: tidak semua fitur membantu.

// Text/Data Mining
1. Klasifikasi
   - diberikan: koleksi dr record dan label (trainingset). setiap record mengandung sekumpulan fitur (atribut) & kelas true (label).
   - tentukan: model utk kelas sbg fungsi dr nilai² fitur.
   - tujuan: record yg blm ada sebelumnya diberi label seakurat mungkin.
     - test set digunakan utk menghitung akurasi model.
2. Clustering
   - diberikan: sekumpulan dokumen & similarity measure diantara dokumen.
   - tentukan: clusters ∋ dokumen didlm 1 cluster mirip dgn dokumen lainnya, dokumen dlm cluster terpisah tidak mirip dgn dok lainnya.
   - tujuan: temukan kumpulan dokumen yg tepat.
   - similarity measures: euclidean distance jika atribut adh kontinu

Web Mining

- mencari informasi menarik & berguna dr konten dan penggunaan web.
  1. Web search: Google, Yahoo, MSN, Ask, ...
  2. specialized search: eg: google.com /shopping (comparison shopping), job ads (Flikr)
  3. eCommerce: rekomendasi eg: Netflix, Amazon. memperbaiki tingkat konversi: produk terbaik berikutnya yg ditawarkan.
  4. Iklan, eg: Google Adsense
  5. Fraud detection: click fraud detection
  6. meningkatkan rancangan & kinerja website.
- teknik² daming utk mencari & mengekstrak secara otomatis informasi dr dokumen/layanan web. (Etzioni, 1996)

- Why?

1. Web kaya akan informasi tekstual
   - Book/CD/Video stores eg. Amazon
   - Information restoran eg. Zagat
   - Harga mobil eg. Carpoint
2. Banyaknya data pd pola akses pengguna
3. Possible to retrieve "previously unknown" information si akses & penggunaan.

- Fitur unik pd web

- koleksi dokumen berukuran besar yg mengandung: informasi hyperlink & informasi akses & penggunaan.
- web sgt dinamis: hal web scr konten dibangkitkan (dibuang)

	Web Mining	Data Mining
Struktur	web bkn mirip relasi	Informasi tekstual dan struktur linkage
skala	data penggunaan berukuran besar & berkembang scr cpt.	data yg dibangkitkan perhari dpt dibandingkan dgn data warehouse konvensional terbesar.
kecepatan	seringkali perlu bereaksi utk mengembangkan pola penggunaan scr real-time eg: merchandising	manusia tdk terlibat dlm proses.

Taksonomi Web Mining

- Web Mining
  - Web Content Mining
    - Web Page Content Mining
      - identifikasi informasi dlm halaman web
      - membedakan halaman web personal dr hal lainnya
    - Search Result Mining
      - kategorisasi dokumen meng. frase dlm judul & potongan informasi.
  - Web Structure Mining ← mengg. interkoneksi antara hal. web utk memberikan bobot pd halaman.
  - Web Usage Mining
    - General Access Pattern Tracking
      - Memahami pola akses & trend utk memperbaiki struktur.
    - Customized Usage Tracking
      - Menganalisis pola akses dr pengguna utk memperbaiki respon.

// Pendekatan² Web Content Mining

1. Information Retrieval: membantu & meningkatkan pencarian & filtering informasi utk pengguna biasanya berdasarkan pd profil pengguna yg disimpulkan/diminta.
2. Basis data: utk memodelkan data pd web & mengintegrasikannya → kueri yg kompleks selain berbasis keywords dapat dilakukan.

	IR View	DB View
view of data	unstructured, semi-structure	semi-structured, website as DB
main data	text docs hypertext docs	hypertext docs
representation	Bag of words, n-grams (terms, phrases) concepts or ontology/ relational	edge-labeled graph relational
methods	machine learning, statistics.	ILP, Association Rule
apps	categorization, clustering, finding extraction rules, finding pattern in text, user modeling	finding frequent substructures, web site schema discovery.

// Web Structur Mining

— utk menemukan struktur link dr hyperlinks pd level antar dokumen utk membangun ringkasan struktur tentang situs web, menemukan hal. web yg authoritative, hyperlinks dpt merujuk authority.

- view of data, main data } Links structure
- representation: graph
- methods: proprietary algorithms
- apps: categorization, clustering

// Web Usage Mining = Web log Mining

→ teknik mining utk menemukan pola penggunaan yg menarik dr data sekunder yg diturunkan dr interaksi pengguna ketika menjelajahi web.

- view of data: interactivity
- main data: server logs, browser logs
- representation: relational table, graph
- methods: machine learning, statistics, association rules.
- apps: site construction, adaptation & management marketing, user modelling.
  1. menargetkan kustomer yg potensial utk produk elektronik
  2. memperluas kualitas dan pengantaran internet information services kepada pengguna akhir.
  3. memperbaiki performa sistem web server
  4. mengidentifikasi lokasi iklan yg potensial
  5. memfasilitasi personalisasi/ situs adaptif
  6. memperbaiki desain situs 7. deteksi fraud/intrusion. 8. memprediksi aksi pengguna