Wikipendium

TDT4300: Data Warehousing and Data Mining

Tags: data datavarehus datamining database datagruve +

This is a summary of "Introduction to Data Mining" by Tan, Steinback, Kumar.

Data Mining, TAN

Chapter 1 Introduction

Chapter 1.1 What is data mining?

Data mining is the process of automatically discovering useful information in large repositories.

Knowledge discovery in databases (KDD) is the overall process of converting raw data into useful information. Data mining is an integral part of KDD.

Preprosessing in advance of processing do feature selection, dimensionality reduction, normalization data subsetting. Make the data into an appropriate format, reduce noise etc. **Postprocessing** filtering patterns, visualization, pattern interpretation. Only useful and valid results are incorporated into an decision support system (DSS).

Chapter 1.2 Motivating challenges

Scalability Algorithms must support growing data sets.

High dimensionality The algorithms must support the complexity that increases when the high dimensionality (number of features) increases.

Heterogeneous and complex data We need techniques for heterogeneous attributes and complex data models.

Data ownership and distribution sometimes the data needed is not owned or stored in one location/organization. Requires distributed data mining techniques. The chalenge is to find out how to reduce communication in distribution, and how to effectively consolidate the result. There are also of course data security issues.

Non-traditional analysis non traditional methods, types of data and data distribution.

Chapter 1.3 The origins of data mining

Data mining draws upon ideas from sampling, estimation hypothesis testing from Statistics. and search algorithms, modeling techniques and learning theories from AI, pattern recognition, machine learning. Other areas: optimization, evolutionary computing, information theory, signal processing, visualization, information retrieval. Others: database systems, (parallell) computing (for massive data sets), distributed techniques.

Chapter 1.4 Data mining tasks

Two categories: _Predictive tasks _objective is to predict the value of a particular attribute (target/dependent variable) based on the values of other attributes (explanatory/independent variables).

Descriptive tasks objective is to derive patterns (correlations, trends, clusters, trajectories, anomalies) that summarize underlying relationships in data.

Predictive modeling building a model for target variable as a function of the explanatory variables. Two types of predictive modeling tasks: classification for discrete target variables. Regression for continuous target variables. The model should minimize the error between the predicted and the true values of the target variable.

Association analysis used to discover patterns that describe strongly associated features in the data. Patterns represented as implication rules or feature subsets. Extract the most interesting patterns in an efficient manner.

Anomaly detection the task of identifying observations whose characteristics are significantly different from the rest of the data. The observations are called anomalies/outliers. The goal is to find them and not label normal objects as anomalous.

Chapter 2 Data

Chapter 2.1 Types of data

Data set collection of data objects (records).

Data objects described by a number of attributes (variables).

Definition Attribute: is a property or characteristic of an object that may vary, either from one object to another (eye color) or from one time to another (temperature).

Definition Measurement scale: is a rule (function) that associates a numerical or symbolic value with an attribute of an object.

The values used to represent an attribute may have properties that are not properties of the attribute itself, and vice versa.

Attribute types (see table 2.2 p.26): nominal and ordinal (categorical/qualitative), interval and ratio (quantitative, numeric).

Permissible transformations are transformations that do not change the meaning of the attribute (table 2.3 p.27).

Number of values Describing attributes by the number of values - Discrete attribute has a finite/countable infinite set of values. - Special case: binary attributes - Continuous attributes have real number values (temperate, height, weight). represented in floating-point variables.

The number of values can be combined with attribute types, but all do not make sense. Example: Count attributes. Are both discrete(number of values) and ratio(attribute type).

Asymmetric attributes Only presence (non-zero attributes) are regarded as important. As a student who takes a course (1) or not (0) on a university. There are fewer 1´s than 0’s, so it is smarter to focus on the 1’s. Special case: asymmetric binary attributes. important for association analysis. special case: asymmetric discrete, asymmetric continuous attributes.

Data sets general characteristics - Dimensionality - the number of attributes that the object in the data set possess. Curse of dimensionality is issues with high dimensionality. Need reduction in preprocessing phase. - Sparsity - only process and store the non-zero values. - Resolution - capture data at different resolutions to avoid that patterns disappear because of noise etc.

Transaction (market basket) data a special type of record data. each record (transaction) involves a set of items. Like a super market basket.

The data matrix if the data objects in a collection of data all have the same fixed set of numeric attributes. Then the data objects can be thought of as points (vectors in a multidimensional space, where each dimension represents a distinct attribute describing the object. Matrix m,n where m rows for each object and n columns for each attribute. Special case: sparse data matrix, only non-zero attributes are important. Document-term matrix, is a sparse data matrix of a document where each word is an attribute and the order of the words are not important.

Group-based data A graph is a powerful and convenient representation of data.

data with relationships among objects Data often represented as a graph. Data objects are mapped to nodes of the graph, and relationships are paths between objects (with direction and weight). Example, The internet with web pages that links to other pages.

data with objects that are graphs If objects have structure (the object contains sub-objects with relationships) they are frequently represented as graphs. Example molecules. Substructure mining is a branch of data mining.

Ordered data Attributes have relationships that involve time or space.

Sequential data (temporal data) extension of record data where each record has a time associated with it. Example a transaction of groceries with timestamp.

Sequence data consists of a data set that is a sequence of individual entities (sequence of words, letters, etc.). Not timestamps but a position in an ordered sequence. Example DNA sequences (ATGC).

Time series data Special case of sequential data, but each record is a time series (series of measurements over time). Example average monthly tempratures of Minneapolis. temporal autocorrelation is important to consider: if two measurements are close in time, then their values are often similar.

Spatial data Positions, areas, etc. Example: weather data (several attributes for each geographical location). spatial autocorrelation (close points have same values).

Handling non-record data Most data mining algorithms are designed for record data. For non-record data, extract features from data objects and create records for each object. Does not capture all relationships (example time etc.)

Chapter 2.2 Data Quality

Because preventing data quality problems is typically not an option, data mining focuses on the detection and correction of data quality problems the use of algorithms that can tolerate poor data quality. First step of correction: data cleaning.

Measurement and data collection issues Data are not perfect: - Human error - Limitations of measuring devices - Flaws in data collection process - Missing data/data sets - Spurious or duplicate objects

Measurement and data collection error common problem: recorded value differs from true value. Numeric difference = error. Data collection error = omitting data objects/attribute values, inappropriately including a data object.

noise, artifacts: distortion of value, spurious object techniques from signal processing to discover patterns lost in the noise elimination of noise is difficult. Data mining algorithms must be robust, to produce good result even when noise is present. Artifacts are deterministic disortions of data, such as a streak in the same place on a set of photographs.

Precision, bias, accuracy: In statistics, quality of measurement process and resulting data is measured by precision and bias. precision: the closeness of repeated measurements (of the same quantity) to one another. Bias: a systematic variation of measurements from the quantity being measured. Accuracy: the closeness of measurements to the true value of the quantity being measured. depends on precision and bias. Significant digits, use only the number of digits specified in the data.

Outliers: data objects that, in some sense, have characteristics that are different from most of the other data objects in the data set values of an attribute that are unusual with respect to the typical values for that attribute "anomalous" objects noise and outliers are not the same outliers can be legitimate outliers can be of interest (fraud, network intrusion detection)

Missing values: - not collected, refused to be entered (age, weight), conditional parts of forms. - should be taken into account in data analysis. - strategies for dealing with missing data - eliminate data objects or attributes with missing values - (-) lose important information. - (+) simple and effective. - can be done with care (if the data set has only a few objects missing data, it can be expedient). - estimate missing values (reliably) - interpolation with remaining values - ignore the missing values during analysis - if the total number of attributes is high enough - if the number of missing values is low

Inconsistent values: Detect and, if possible, correct. It can be necessary to consult with an external source of information.

Duplicate data: or almost duplicates. To detect and eliminate, two main issues must be addressed: if there are two objects that actually represent a single object, then the values of corresponding attributes may differ, and these inconsistent values must be resolved. care needs to be taken to avoid accidentally combining data objects that are similar, but not duplicates. Such as two distinct people with identical names. Can cause problems for algorithms. Deduplication = dealing with the two issues.

Issues related to applications - Timeliness - some data age as soon as they are collected (snapshots of ongoing processes) - Relevance - Make sure that objects in data set are relevant. common problem: sampling bias - occurs when a samples does not contain different types of objects in proportion to their actual occurrence in the population. - example: survey data only describes only respondents. - Knowledge about the data - data that have documentation. - type of feature (nominal, ordinal, interval, ratio) - scale of measurement (meters, feet, etc.) - origin of data

Chapter 2.3 Data preprocessing

To make data more suitable for data mining. - aggregation - sampling - dimensionality reduction - feature subset selection - feature creation - discretization and binarization - variable transformation

Aggregation "less is more". Combine two or more objects into one object. example: reduce transactions in a store one day to one common transaction. Issue: how are the data combined? Behaviour of groups of object is more stable than individual objects Lose information about interesting details.

Sampling Sampling is a commonly used approach for selecting a subset of the data objects to be analyzed. Statistics use sampling because entire data sets are expensive and time consuming to obtain. Data mining use sampling because the entire data set are expensive and time consuming to process. If sampling is representative, then it is almost as well as using the entire data set. representative if has approximately the same property of interest sampling is a statistically process

Sampling approaches simple random sampling - variation 1: sampling without replacement - variation 2: sampling with replacement - same object can be picked more than once - easier to analyse since the size of the data set is constant - can fail if population consists of different types of objects, with widely different numbers of objects. Might not get representation of all object types.

Stratified sampling - starts with prespecified groups of objects. - simple version: extract equal number of objects from each group - complex version: extract the number of objects from a group that is proportional to the size of the group

progressive sampling - size can be difficult to determine. sometimes adaptive or progressive sampling is therefore used. - start with small sample, increase til sufficient size has been obtained.

Dimensionality reduction Benefits: - many data mining algorithms work better if the dimensionality is lower - reduce noise - eliminate irrelevant features - more understandable model - more easily visualized - amount of time and memory required by an algorithm is reduced

Reduction often consists of techniques that replace attributes with a new attribute that is a combination of the old attributes (feature subset selection).

Two important topics: **1)the curse of dimensionality** data is more complex to analyse as the dimensionality increases. - classification: - not enough data objects to allow the creation of a model that reliably assigns a class to all possible objects - clustering: - the definitions of density and the distance between points become less meaningful. Therefore a lot of algorithms have trouble with high-dimensional data. **2)Linear algebra techniques for dimensionality reduction** Principal component analysis (PCA) - technique that for continuous attributes finds new attributes that - are linear combinations of the original attributes - are orthogonal to each other - capture the maximum amount of variation in the data - Singular Value Decomoposition (SVD) - related to PCA

Feature Subset Selection Use only a subset of the features. If redundant and irrelevant features are present, one does not lose information. - redundant features - duplicate much/all information contained in one or more other attributes. - irrelevant features - contain almost no useful information for the task at hand will reflect bias, objective at final algorithm. For n attributes there are 2n subsets. Impractical in most situations. Three standard approaches: 1. embedded 1. occurs naturally as part of the data mining algorithm. the algorithm decides what attributes to use, and what to ignore. 2. filter 1. features are selected before the algorithm runs. 3. wrapper 1. use the target algorithm as a black box to find best subset, but does not enumerate all possible subsets.

architecture for feature subset selection (filter/wrapper) feature selection process consists of: - measure for evaluating a subset - search strategy that controls the generation of a new subset of features - stopping criterion - validation procedure Filter (uses target algorithm) and wrapper (distinct from target algorithm) differs in the way they evaluate a subset.

flowchart of a feature subset selection process (figur i boken)

Stopping criterion is often based on 1+ conditions: - number of iterations - optimal value of the subset, or exceeds a threshold - certain size of subset - evaluation criteria has been achieved evaluation approach - run algorithm with both full data set and featured subset and compare results - run algorithm with different subsets and compare results

feature weighting more important features are assigned a higher weight. Sometimes based on domain knowledge, or can be determined automatically. Features with higher weighting plays a more important role in a model.

Feature extraction Creation of new set of features from the original raw data. Highly domain-specific. example image processing. Features and techniques to extract features are specific to fields.

mapping the data to a new space in example time series of data and if there is a number of periodic patterns and noise present, patterns are hard to detect. Fourier transform applied to the time series patterns can be detected. Underlying frequency information is explicit.

feature construction If the original feature is in a form not suitable for the data mining algorithm. One or more new features are constructed out of the original feature.

Discretization and binarization Some algorithms require data to be categorical attributes (classification) or binary attributes (association patterns). Continuous to categorical = discretization Continuous/discrete to binary = binarization If some categorical attribute has a large number of values, or if some values occur infrequently, it can be beneficial to reduce number of categories by combining values.

Binarization Simple techniques: - if there are m categorical values, then uniquely assign each original value to an integer in the interval [0, m-1]. If the attribute is ordinal, the order must be maintained by the assignment. - convert each of the m integers to a binary number. - If only the presence of an attribute is important, use asymmetric binary attributes. For n attribute there is n*x binary attribute values. - Some attributes needs to be divided into two or more binary attributes. Example gender with one attribute that has value 1 if female, and one attribute that has value 1 if male.

discretization of continuous attributes: Discretization for attributes used for classification or association analysis. Transformation from continuous attribute to categorical attribute: 1. deciding how many categories, and how to map attribute-values to these categories - after the values of the continuous attribute are sorted, divide them into n intervals by specifying n-1 split points. - all the values in one interval are mapped to the same categorical value. 2. problem: - how many split points? - where to place them? 3. result represented as a set of intervals, or as a series of inequalities (xo < x < x1, etc.)

Unsupervised discretization: distinct between discr. methods by whether class information is used (supervised) or not (unsupervised). Unsupervised has relatively simple approaches. As equal width (divide the range of the attribute into a user-specified number of intervals each having same width), can be badly affected by outliers. Equal frequency (puts the same number of objects into each interval) is preferred, handles outliers better. K-means and visually inspecting is also examples. Usually better than no discretization.

Supervised discretization Use the end purpose in mind and use class labels. Produce better result than unsupervised. Things to consider: purity and minimum size of an interval. entropy: (formel i boka) where k is number of class labels, mi is the number of values in the ith interval, mij is the values of class j in the ith interval, pij=mij/mi is the probability of clas j in the ith interval. total entropy: (formel i boka) where w is mi/m (the fraction of values in the ith interval) and m is the number of values.

Entropy is a measure of the purity of an interval. If an interval contains only objects of the same class (pure), the entropy equals 0. If the classes of values in an interval occur equally often, the interval is as impure as possible, and the entropy is maximum.

Simple approach for partitioning is by starting with bisecting the initial values so the resulting two intervals give minimum entropy. assumes ordered set of values. Repeat with the interval with highest entropy, until stopping criterion is reached.

problem: categorical attributes sometimes have too many values can join together based on domain knowledge, or if it does not improve or exist, use a more empirical method such as group together only if grouping improves the classification accuracy or similar.

Variable transformation A transformation that is applied to all the values of a variable. For each object, the transformation is applied to the value of the variable for that object.

Simple functions A simple mathematical function is applied to each value individually. Variable transformaiton should be done with causion since it change the nature of the data. Should be fully appreciated. Example 1/x, increases magnitude for <0,1>, but decreases for <1,n>.

Normalization or standardization Common type of variable transformation. The goal is to make an entire set of values have a particular property. Example: difference of two people based on income and age. Income varies more than age, income can dominate. Mean and standard deviation is often affected by outliers. Mean is therefore replaced by median (middle value), standard deviation is replaced by absolute standard deviation.

Chapter 2.4 Measures of similarity and dissimilarity

Used by a number of data mining techniques; clustering, nearest neighbour classification, anomaly detection. In many cases initial data set is not needed after similarities or dissimilarities have been computed.

Proximity Refers to similarity or dissimilarity.

Similarity between two objects is a numerical measure of the degree to which the two objects are alike. 0 = no similarity 1 = complete similarity.

Dissimilarity between two objects is a numerical measure of the degree to which two objects are different. Distance is a synonym. [0,1] or [0, infinity].

transformations often applied to convert a similarity to a dissimilarity (or vice versa) - d = 1-s and s= 1-d. - s > 0, d<0 and s = -d, and d=-s transform a measure to fall within a specific range. example [0,1] - s′ = (s-min_s)/(max_s-min_s). example: s′ = s-1/9 where min_s = 1 and max_s = 10.

Similarity and dissimilarity between simple attributes objects having a simple attribute. S = 1 if match, s = 0 if dont match. d = 1 if dont match, d = 1 if match. single ordinal attribute. If poor = 0, fair = 1, ok = 2, good = 3, wonderful = 4, then d(good, ok) = 3-2 = 1. downside: assumes equal distance between the values. table 2.7 s.69!!!!

Dissimilarities between data objects

distance Euclidean distance d between x and y is given by following formula: (formel i boka)

n is the number of dimensions. xk and yk is the kth attributes of x and y.

Euclidean distance is generalized by Minkowski distance metric: (formel i boka)

r is a parameter. Three most common examples: 1. r = 1. City block (Manhattan, taxicab, L1 norm) distance. 2. r = 2. Euclidean distance (L2 norm). 3. r = Supremum (Lmax norm) distance. The maximum difference between any attribute of the objects. (formel i boka)

Properties of distances: positivity d(x,y) is positive for all x, y. d(x,y) = 0 only if x = y symmetry d(x,y) = d(y,x) for all x, y triangle inequality d(x,z) d(x,y) + d(y,z) for all points x,y,z. Measures that satisfy all = metrics

Similarities between data objects For similarities the triangle inequality often does not hold.

Examples of proximity measures Simple matching coefficient SMC = number of matching attribute valuesnumber of attributes=f(11)+f(00)f(01)+f(10)+f(00)+f(11) treats presences and absences equally.

Jaccard coefficient handles objects consisting of asymmetric binary attributes. Example if number of true is very little combined with false. J = number of matching presencesnumber of attributes not involved in 00 matches=f(11)/f(01)+f(10)+f(11) Example: Jaccard and SMC x = (1,0,0,0,0,0,0,0,0,0) y = (0,0,0,0,0,0,1,0,0,1) f01 = 2, f10 = 1, f00 = 7, f11 = 1

SMC = (0+7)/(2+1+0+7) = 0,7 J = 0/(2+1+0) = 0

Cosine similarity word documents often does not contain zero entries. Cosine similarities does not depend on (and ignores) 0-0matches, as jaccard measure also does. Cosine similarity also must be able to handle non-binary vectors. If x and y are two documents vectors then cos(x,y) = x * y/||x||||y|| x*y = the vector dot product.

(s.75 for regneeksempel)

if cos = 0, the angle is 90 degrees, and x and y do not share any terms.

Extended jaccard coefficient (tanimoto coefficient) useful for document data. reduces to jaccard in case of binary attributes. EJ(x,y) , formel i boka..

Chapter 3 Exploring Data

Data exploration The preliminary investigation of the data in order to better understand its specific characteristics. It can aid in preprocessing and data analysis techniques. Help in data mining when patterns can be found visually. Understand and interpret results.

This chapter overlap with Exploratory Data Analysis (EDA). Created in the 1970s.

Chapter 3.2 Summary statistics

Summary statistics are quantities that capture various characteristics of a potentially large set of values with a single number or a small set of numbers. Mean, standard deviation etc.

Frequency For unordered categorical values, frequency is almost the only useful thing to compute. frequency (vi) = number of objects with attribute value vim m is the number of objects.

Mode the mode of a categorical attribute is the value that has the highest frequency. May contain information about the presence of missing values (heights in cm and not in mm, or if a value is used to indicate a missing value).

Percentiles Ordered data. Given an ordinal/continuous attribute x and a number p (0,100), the pth percentile xp is a value of x such that p% of the observed values of x are less than xp. Eksempel s.101.

Measures of location Mean x= 1mi = 1mxi The middle of a set of values, if the values are distributed in a symmetric manner. If not, use median. Sensitive to outliers. Trimmed mean, when a percentage p of the top and bottom are trimmed, and mean is calculated as usual. (median is trimmed mean where p=100%).

Median median (x) = x(r + 1) if m is odd (the middle value) = ½ (x(r) + x(r + 1)) if m is even (the average of the two middle values) Not so sensitive to outliers.

Measures of spread Measures if widely spread or relatively concentrated around a single point.

range Simplest measure of spread. range(x) = max(x)-min(x)=x(m)-x(1) in the set {x1, x2, …, xm}. Not good if most of the values are concentrated, but there are some extreme min and max outliers.

variance variance = sx2 = 1m-11 = 1m(xi-x)2 Standard deviation (sx) is the square root of the variance. Variance is computed with mean, and is therefore also sensitive to outliers. More robust measures: ADD: absoulte average deviation ADD = 1m1 = 1m|xi-x | MAD: median absolute deviation MAD(x) = median({|x1 - x|, … ,|xm - x|}) IQR: interquaratile range IQR = x75% - x25%

Multivariate summary statistics measures of location can be obtained by computing mean and median separately for each attribute. For multivariate data, it can be computed independent of each other. For continuous variables, the spread is most commonly captured with covariance matrix S.

si j = covariance(xi,yi) covariance(xi , yi) = 1m-11 = 1m(xki-xi)(xkj-xj)

The variances lies along the diagonal. Covariance is the measure of how two attributes vary together and depends on the magnitudes of the variables. Near 0, not have a relationship.

Correlation measure how strong two attributes are linearly related. Preferred for data exploration. correlation (xi,yi) = covariance(xi, yi)s(i)s(j)

Diagonal = 1, other varies between (-1, 1).

Other ways to summarize data Skewness The degree to which the values are symmetrically distributed around a mean.

Chapter 3.3 Visualization

the display of information in a graphic or tabular format. Visual data mining.

Motivation for visualization use information that is “locked up in people’s heads”. Maps of world and temperature. eliminate patterns and focus on the important ones.

General concepts Representation: Mapping data to graphical elements First step. Objects usually represented as: if only a single categorical attribute is being considered categories based on values of the attribute are displayed as an entry in a table or an area on a screen. if an object has multiple attributes objects can be represented as a row (/column) in a table or as a line in a graph if an object is a point in a 2- or 3-dimensional space points can be represented as a geometric figure circle, cross, box…

Techniques based on number or type of attributes involved, or if the data has some special characteristics.

visualizing small numbers of attributes Stem and leaf plots one-dimensional integer or continuous data. Split the values into groups, where each group contains those values that are the same except for the last digit each group becomes a stem last value in a group is a leaf example: 35, 36, 42, 51. stems: 3, 4, 5 leaves: 5, 6, 2, 1.

Histogram Stem and leaf plots are a type of histogram where values are divided into bins.
For categorical data, each value is a bin. If too many bins, combine. for continuous attributes, the range of values is divided into bins of (often) equally width. Representation: Each bin is represented by a bar. Bar plot. Variations: relative (frequency) histogram - change in scale in y axis. pareto histogram - categories are sorted by count. Decreases from left to right. two-dimensional histograms - each attribute is divided into intervals ,and the two sets of intervals define two-dimensional rectangles of values. can show how to values of an attribute co-occur. (figur på s.114)

Box plots figur s. 115. outliers = “+”. Compact and many can be shown in same plot.

Pie chart typically used with categorical attributes with relatively small number of values. not popular in technical publications because the size of relative areas can be hard to judge.

Cumative distribution function (CDF) - Percentile plots For each value in a statistical distribution a CDF shows the probability that a point is less than that value. For each observed value an Empirical CDF shows a fraction of points that are less than this value. (figur s.117)

Scatter plots each data object is plotted as a point in the plane using the values of the two attribute as x and y coordinates. main uses for scatter plots: graphically show the relationship between two attributes. scatter plots can also be used to detect non-linear relationships, either directly or using a scatter plot of the transformed attributes. when class labels are available, they can be used to investigate the degree to which two attributes separate the classes.

visualizing spatio-temporal data when data has spatial or temporal attributes. The earth is a spatial grid. various points in time. Data may have only a temporal component, such as time series data that give the daily prices of stocks.

contour plots if a three-dimensional data where the third dimension has a continuous value (temperature, elevation). Visualized as contour plot, which breaks the plane (the first two dimensions) into separate regions where the third dimension is roughly the same. surface plot the third attribute is the height above the plane (x, y). Require the z to be defined for all combinations of x,y.

Vector field plot characteristics may have both a magnitude and a direction. Display both. (figur s. 123).

Lower-dimensional slices When a spatio-temporal data set is recorded over time = 4 dimensions. Not easily displayed. Separate slices can be displayed (example one for each month).

Animation To deal with slices of data. Display successive two-dimensional slices of data.

visualizing higher-dimensional data Can show higher dimensions, but show only some aspects of the data.

Matrices array of values. If class labels are known, data should be reordered so similar class labels are grouped together. If different attributes have different ranges, it is standardized to have a mean of 0, and a standard deviation of 1.

If class labels are not known, matrix recording or seriation can be used to rearrange the rows and columns of the similarity matrix, so that groups of highly similar objects and attributes are together and can be visually identified.

parallel coordinates one coordinate axis for each attribute. axes are parallel to one another instead of pendicular (x,y liksom). An object is represented as a line (not a point). - each value represent a point, a line is drawn between points to form the line - figur s.127

star coordinates and chernoff faces encode objects as glyphs or icons. star coordinates - uses one axis for each attribute, and radiate from a center point. - typically, all attribute values are range (0,1). chernoff face - program to map values to features of a face. - figur s.129 they do not scale well, and are of limited use in data mining.

Do's and Dont's Guidelines - ACCENT principle for effective pragpical display (d.A. Burn).

Apprehension - ability to correctly perceive relations among variables. Does the graph maximize apprehension of the relations among variables? **C**larity - ability to visually distinguish all the elements of a graph. are the most important elements or relations visually most prominent? **C**onsistency - ability to interpret a graph based on similarity to previous graphs. are the elements, symbol shapes, and colors consistent with their use in previous graphs? **E**fficiency - ability to portray a possibly complex relation in as simple a way as possible. are the elements of the graph economically used? is the graph easy to interpret? **N**ecessity - the need for the graph, and the graphical elements. Is the graph a more useful way to represent the data than alternatives? are all the graph elements necessary to convey the relationships? **T**ruthfulness - ability to determined the true value represented by any graphical element by its magnitude relative to the implicit or explicit scale. are the graph elements accurately positioned and scaled?

Tufte's guidelines for graphical excellence: - is the well-designed presentation of interesting data - a matter of substance, of statistics and of design. - consists of complex ideas communicated with clarity, precision, and efficiency. is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space. - is nearly always multivariate. - requires telling the truth about the data.

Chapter 3.4 OLAP and multidimensional data analysis

viewing data sets as multidimensional arrays. OLAP = On-Line Analytical Processing. OLAP systems has a strong focus on the interactive analysis of data and typically provide extensive capabilities for visualizing the data and generating summary statistics.

Representing data as a table each row is an object, each column is an attribute. Multidimensional table, s.132.

General case for multidimensional data: starting point: - tabular representation of the data = fact table. step 1: - identification of the dimensions and identification of an attribute that is the focus of the analysis. - The dimensions is categorical attributes, or converted continuous attributes. - The size of the dimension is the number of attribute values. - Content of each cell = target quantity. step 2: - each row is mapped to a cell. The indices of the cell are specified by the values of the attributes (dimensions). The value of the cell is the value of the target attribute. Cells not defined assumes to be 0.

Analyzing multidimensional data - data cubes: computing aggregate quantities - aggregation is important to find a total/summary of data. Also to avoid too many entries when representing/computing. - There are a number of different aggregations (totals) to be computed for a multidim.array, depending on how many attributes we sum over. - For n dimensions, where we sum over k dimensions = nkarrays of totals, each with dimensions n - k (ikke den derre streken midt på. damn g.docs.equations).

data cube a multidimensional representation of the data, together with all possible totals (aggregates). The size of each dimension does not need to be equal. Can have more or less than 3 dimensions. Generalization of cross-tabulation (statistical terminology).

Dimensionality reduction collapses a column into a single cell. If sj is the number of possible values of the jth dimension, the number of cells is reduced by a factor of sj. pivoting is aggregating over all dimensions except 2. result: cross-tabulation with dim= 2.

Slicing and dicing slicing is selecting a group of cells from the entire multidim. array by specifying a specific value for one or more dimensions. dicing is selecting a subset of cells by specifying a range of attribute values. (selecting a subset from the complete array).

Roll-up, drill-down data categories can be viewed as hierarchies. A chair is a furniture, and a chair can be divided into lounge chair, sitting chair, armchair etc. roll-up to roll data up in a single category (from days to one month). drill-down to drill down data from a single category to several categories (from month to days).

Chapter 4 Classification: Basic Cocnepts, Decision Trees, and Model Evaluation

Classification The task of assigning objects to one of several predefined categories (mapping input attribute set x into its class y). A problem that encompasses many diverse applications. Detecting spam messages medical results galaxy-classifications based on shape

Chapter 4.1 Preliminaries

Record The input for a classification task is a collection of records. Each record (instance, example) is characterized by a tuple (x, y). x = attribute set y = special attribute, class label

Definition classification: The task of learning a target function f that maps the attribute set x to one of the predefined class labels y.

f is known as a classification model. Useful for Descriptive modeling serve as tool for distinguish objects of different classes Predictive model predict the class label of unknown records.

Chapter 4.2 General approach to solving a classification problem

Classification technique / classifier To build a classification model from an input data set. Each technique employs a learning algorithm to find the best model for x and y.

Training set a set where labels are known. Is used to build a classification model which is applied to the test set. A test set has unknown labels.

Confusion matrix counts of correct and incorrect results from the test records.

Performing metric One number telling the accuracy shown in the confusion matrix.

Accuracy = number of correct predictions / total number of predictions Error rate = number of wrong predictions / total number of predictions

Chapter 4.3 Decision tree induction

decision tree is a classifier (classification technique).

Root node 0 incoming edges, 0 or more outgoing edges

Internal nodes 1 incoming edges, two or more outgoing edges

Leaf (terminal) nodes 1 incoming edge, 0 outgoing edges

Root and internal nodes contains attribute test conditions.

Algorithms for making decision trees: *Hunt's algorithm* basis for: ID3, C4.5, CART grow a tree recursively. - Step 1: - if all records in Dt belongs to the same class yt, then t is a leaf node labeled as yt. - Step 2: - If Dt contains records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition, and Dt is distributed to the children based on the outcomes. - Algorithm is then applied to each child node.

Will work if every combination of attribute values is present in the training data, and each combination has a unique class label.
Additional conditions: - if a child in step 2 can be empty, there are no records associated with the node. In step 2, if all records associated with Dt have identical attribute values, it is not possible to split those records any further.

Design issues of Decision tree induction Learning algorithm should address these issues: How should the training records be split? How should the splitting procedure stop?

Methods for expressing attribute test conditions Attribute types: - binary - must generate two potential outcomes - nominal - multiway split (1 to several) - binary split (OR - conditions) - Ordinal - ordered binary, and must keep the order of the values (not small - large and medium - extra large) - continuous - can be expressed as comparison test (A < v) (A > v).

Measures for selecting the best split the measures are often based on the degree of impurity of the child nodes. Smaller degree, more skewed the class distribution. (0,1) = 0 impurity (0.5, 0.5) = highest impurity.

s.158 entropy, gini og classification error som eksempler på impurity measures for binary classification. og graf for å sammenligne.

Gain The difference between impurity of parent node (before splitting) and child node (after splitting). The larger the difference, the better the test condition. Used to decide how good a test condition performs.

Decision trees algorithms often use a test condition that maximize the gain. I(parent) is same for all test conditions, so max the gain means minimize the weighted average impurity measures of the child nodes.

When entropy is used, the difference in entropy is known as information gain.

s. 160 Splitting binary attributes splitting of nominal attributes splitting of continuous atrributes Gain ratio Impurity measures (entropy, Gini index) tend to favor attributes with large number of distinct values.

Strategies to avoid too pure outcome (gender vs car brand vs customer ID): 1. restrict test conditions to binary splits only. - employed by several decision tree alg. 2. modify the splitting criterion to take into account the number of outcomes produced by the attribute test condition. - gain ratio (information gain /split info)

(mangler, s.164.. TreeGrowth algorithm (decision tree induction) og Example: Web robot detection)

Characteristics of decision tree induction 1. nonparametric approach for building classification models. Does not require any prior assumptions. 2. Finding an optimal decision tree is an NP-complete problem. Many alg are heuristic. 3. Inexpensive techniques make it possible to quickly construct models even when training set is large. 4. Easy to interpret. 5. provide an expressive representation for learning discrete-valued functions. 6. D.T.alg are robust to presence of noise (especially overfitting). 7. attribute-redundancy (langt punkt, se s. 169) 8. number of records become smaller when traversing down the tree. Data fragmentation problem is when number at leaf node is too small to make a statistically significant decision. 9. subtree can be replicated multiple times in a D.T. Makes the D.T more complex and more difficult to interpret. 10. border between two regions of different classes is known as decision boundary. 1. Oblique decision tree allow more than one attribute in a test condition. 2. Constructive induction Partition the data into homogeneous non-rectangular regions. crates composite attributes representing existing attributes arithmetic or logical. 11. choice of impurity method has little effect on the performance of the decision tree induction algorithms because the measures are so consistent with each other.

Chapter 4.4 Model overfitting

errors by a classification model divided into 2 types: 1. training errors (resubstitution / apparent error) - number of misclassification errors committed on training records. 2. generalization errors. - the expected error of the model on previously unseen records.

Model overfitting When a model has low training error but higher generalization error than another model with higher training error.

Also when the node number becomes too high, the test error begins to increase even though its training error rate continue to decrease (see p.174).

Model underfitting High error when the tree size is small. The model has yet to learn the true structure of the data.

Handling the overfitting in DT induction strategies: pre-pruning (early stopping rule) alg is halted before generating a fully grown tree that perfectly fits the entire data set. a restrictive stopping condition required. too high threshold = underfitting, too low might cause overfitting. Post-pruning initially grown to full size. then tree-pruning step. trim tree in bottum-up-fashion. replacing subtree with new leaf node or the most frequently used branch of the substree. Terminate when no further improvements can be seen.

chapter 4.5 Evaluating the performance of a classifier

Holdout method two sets: training set, test set. Classification model is induced from the training set. Performance evaluated on test set. Limitations: fewer labels available for training than for testing. the model may be highly dependent on the composition of the training and test sets. smaller training set, larger variance of the model. and the other way around.

Random subsampling Repeating the holdout method. Overall accuarcy: acc(sub) = i = 1kacc(i)/k Problems: not utilize as much data as possible for training. No control for how many times a record is used for training and testing.

Cross-validation each record is used the same number of times for training and 1(once) for testing. divide into sets. K-fold. each run a partition is used for testing, the rest of them are used for training. Runs k times, each partition is used once for testing.

special set. k = N, leave-one-out approach.

bootstrap Does not (as the others) assume that the training records are sampled without replacement → no duplicate records in the training and test sets. In bootstrap, the records are placed back in the pool, and are as likely as the others to be drawn.

Probability of a record to be chosen: 1 - (1- 1/N)N. If N is large, it approaches 1-e-1 = 0.632.

Accuracy = accboot= 1bi = 1b(0.532 x ei+0.368 + accs)

CHAPTER 5 CLASSIFICATION: ALTERNATIVE TECHNIQUES

Nearest-neighbor classifiers

From chapter 4, a two-step process for classification: an inductive step for constructing a classification model from data, a deductive step for applying the model to test examples.

eager learners: Decision tree and rule-based classifiers. Designed to learn a model that maps the input attributes to the class label as soon as the training data becomes available.

Lazy learners: delay the process of modelling the training data until it is needed to classify the test examples. Example: Rote classifier (memorizes the entire training data and performs classification only if the attributes of a test instance match one of the training examples exactly. Drawback: some records may not be classified.

Approach: Nearest neighbor find all the training examples that are relatively similar to the attributes of the test example; nearest neighbors. Can use them to determine class labels of the test example. “if it walks like a duck, quacks like a duck, and looks like a duck, then it’s probably a duck”.

Algorithm

Computation can be costly if training example is to large. However, efficient indexing techniques are available to reduce the amount of computations needed to find the nearest neighbors of a test example.

Once the list is obtained, the test example classified based on the majority class of its nearest neighbors: Majority Voting: y'=argmaxv(xi,yi)DzI(v = yi) v is class label, yi is the class label for one of the nearest neighbors, and I(*) is an indicator function that returns the value ‘ if its argument is true (0 false). Every neighbor has same impact = sensitive for choice of k. Reduce inpact of k: can weight the influence of each neighbor xi according to its distance: wi = 1/d(x’, xi)2. Then farther away x’s has less impact. distance-weighted Voting: y'=argmaxv(xi,yi)DzwiI(v = yi) Characteristics of nearest - neighbor classifiers part of a more general technique known as instance-based learning (make predictions without making an abstraction(model)). require a proximity-measure to determine the similarity or distance between instances require a classification function that returns the predicted class of a test instance based on its proximity to other instances. Lazy learners do not require model building. but require to compute proximity indicidually between the test and the training examples. eager use a lot of resources on model, and is fast on the classifying. Make their predictions based on local information, trees and rule-based classifiers try to find a global model that fits the entire input space. Nearest neighbor classifiers (with small k) are quite susceptible to noise. can produce arbitrarily shaped decision boundaries. They provide a more flexible model representation compared to eager methods (often constrained to rectilinear decision boundaries). boundaries have high variability because they depend on the composition of training examples. Increasing the number of nearest neighbors may reduce such variability. can produce wrong predictions unless appropriate proximity measure and data preprocessing steps are taken. such as weight (xx kg) and height (x,xx m). Weight will dominate unless not taken into consideration.

CHAPTER 6 ASSOCIATION ANALYSIS: BASIC CONCEPTS AND ALGORITHMS

TID = transaction ID

Association analysis useful for discovering interesting relationships hidden in large datasets. Relationships can be represented as association rules.

section 6.1 Problem definition

Binary representation market basket represented in a binary format. each transaction is a row, and a present item = 1, non-present = 0. Ignores the quantity, price etc.

Asymmetric binary value an item that is more important when present than absent.

itemset I = {i1,i2,...id} is the set of all itemset. If an itemset contains k items, it is called a k-itemset. {beer, diapers, milk} is a 3-itemset.

Transaction set T = {t1,t2,...,tn} is the set of all transactions. ti contains a subset of items chosen from I.

Support count The number of transactions that contain a particular itemset. (X) = |{ti | X ti, tiT}

association rule implication expression of form X → Y, where X and Y are disjoint itemsets. Strength of rule can be measure with its support and confidence.

support how often a rule is applicable to a given data set. support, s(XY)=(XY)N

Used to eliminate rules with low support (most likely to be random and uninteresting).

Confidence how frequently items in Y appears in transactions that contain X. confidence, c(XY)=(XY)(X) Measures the reliability of the inference made by the rule. Gives the conditional property of Y given X.

Association rule mining problem (definition): given a set of transactions T, find all the rules having support minsup and confidence minconf, where minsup and minconf are the corresponding support and confidence thresholds.

Brute-force (1) compute support and confidence for all possible rules. Expensive, exponentially many rules that can be extracted from a data set: R = 3d - 2d+1 + 1, where d is the number of items in the data set.

section 6.2 Frequent itemset generation

itemset of k items can generate 2k -1 frequent itemsets. Ways to reduce computational complexity: reduce the number of candidate itemsets (M) apriori principle, does not count support values reduce the number of comparisons instead of matching each candidate itemset against every transaction. Uses advanced data structures to store the candidate itemsets or to compress the data sets.

Apriori principle if an itemset is frequent, then all of its subsets must also be frequent. and if an itemset is infrequent, then all its supersets is infrequent too. Can prune based on this = support-based pruning.

Anti-monotone property: the support of an itemset never exceeds the support for its subsets. X is subset of Y, f(Y) must not exceed f(X). Anti-monotone (Downward closed).

Monotone property: Monotone (upward closed) = if X is a subset of Y, f(X) must not exceed f(Y).

Apriori algorithm (s.337) initially makes a single pass over the data set to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemset F1 will be known. next, the algorithm iteratively generate new candidate k-itemsets using the frequent (k-1)-itemsets found in the previous iteration. The algorithm passes once more to count the support. Then the algorithm eliminates all candidate itemsets whose support counts are less than minsup. The algorithm terminates when there are no new frequent itemsets generated.

The frequent-itemset-generation part of the Apriori algorithm is a level-wise algorithm it traverses the itemset lattice one level at a time (k = 1 til k = n) generate-and-test first generate candidate itemsets from frequent sets, and then test each candidate. total number of iterations is kmax + 1 (kmax is the max size of the frequent itemsets).

List of requirements for an effective candidate generation procedure: avoid generating too many unnecessary candidates. unnecessary: at least one of its subsets is infrequent. Anti-monotone property! the candidate set is complete (no frequent itemsets are left out by the procedure). a candidate itemset should not be generated more than once.

Brute-force method (2) computations needed for each candidate is O(k). The overall complexity is O(k =1dkd over k= O(d*2d-1)

Fk-1 x F1 method (figur s.340) A frequent itemset is extended with another frequent items. If k=3, a k-1-itemset is extended with a k=1-itemset. Produces O(|Fk-1| x |F1|) candidate k-itemsets, where |Fj| is the number of frequent j-itemsets. Overall complexity: O(kk|Fk-1| |F1|) The method is complete. The method does not prevent that an itemset is generated more than once. can be prevented with sorted items in lexical order, and each itemset(k-1) is extended with an item that is lexically larger. {bread, diapers} with {milk} not {diapers, milk} with {bread} Better than brute-force, but still produce a lot of unnecessary candidates. heuristics available to improve every k-itemset survived pruning step, should be present in k-1 candidates. Otherwise, it is infrequent.

Fk-1 x Fk-1 method (figur s.342) The procedure merges a pair of frequent (k-1)itemsets only if their first k-2 items are identical. Lexical ordering to prevent duplicate candidates. Additional candidate pruning is needed to ensure that the remaining k-2 subsets of the candidate is frequent.

Support counting one approach: compare each transaction against every candidate itemset, and to update the support counts of candidates contained in the transaction. expensive (computationally) another approach: enumerate the itemsets contained in each transaction and use them to update the support counts of their respective candidate itemsets. figur s.343 som forklarer fremgangsmåten. first enumerate, then increment support if it is a match between a k-2(?) itemset and a candidate. Hash tree function (figur s.344) sort both candidate itemsets into buckets, and transaction itemsets (of similar size) into same buckets. Match needs only to be done inside each bucket. les på dette… (s.344)

Computational complexity can be affected by following factors: support threshold lowering threshold → more itemsets is frequent. More complex. number of items (dimensionality) more space needed to store the support counts of items. number of transactions run time increases since it repeatedly passes over the data set average transaction width the maximum size of frequent itemsets tend to increase. Complexity increases. more itemsets are contained in the transaction, number of hash tree traversals increases during counting.

Time complexity: generation of frequent 1-itemsets for each transaction, need to update the support count for every item present in the transaction. w is average transaction width N is total number of transactions O(Nw) is the time the operation requires. candidate generation blæ (se s 349). support counting cost for support counting: O(Nk(W over k)k) W is the maximum transaction width, k is the cost for updating the support count of a candidate k-itemset in the hash tree.

section 6.3 Rule generation

Each itemset of size k can produce max 2k - 2 association rules. Partition an itemset Y into non-empty subset X and Y-X, such that X→ Y-X satisfies the confidence threshold. (all subsets are infrequent due to anti-monotone property).

confidence for rule X → Y-X = (Y)(X)

Confidence-based pruning No anti-monotone property for confidence.

Theorem: If a rule X → Y-X does not satisfy the confidence threshold, then any rule X′ → Y - X′ where X′ is a subset of X, must not satisfy the confidence threshold as well.

Rule generation in apriori algorithm (tegn inn fra s. 350)

algoritme:

les også eksempel s. 353.

section 6.4 Compact representation of frequent itemsets in practice, the number of frequent itemsets produced from a transaction data set can be very large. identify a small representative set of itemsets from which all other frequent itemsets can be derived.

Maximal frequent itemsets Definition: A maximal frequent itemset is defined as a frequent itemset for which none of its immediate supersets are frequent.

figur 6.16: Itemsets are divided into frequent/infrequent itemsets. {a,d} is maximal frequent because all of its immediate supersets ({a,b,d},{a,c,d},{a,d,e}) are infrequent. Non-maximal = non of its immediate subsets are frequent.

Maximal frequent itemsets provide a compact representation because they form the smallest set of itemsets from which all frequent itemsets can be derived.

In figure 6.16 the maximal frequent itemsets are {a,c,e}, {a,d} and {b,c,d,e}.

Maximal frequent itemsets do not contain info about support of their subsets. Additional pass over data set is needed to determine support counts of the non-maximal frequent itemsets.

Closed frequent itemsets minimal representation without losing support information.

definition closed itemset: an itemset X is closed if none of its immediate supersets has exactly the same support count as X.

or: X is not closed if at least one of its immediate supersets has the same support count as X.

figur 6.17

definition closed frequent itemset: an itemset is a closed frequent itemset if it is closed and its support is greater than or equal to minsup.

The closed frequent itemsets to determine the support counts for the non-closed frequent itemsets. {a,d} must have the same support as {a,c,d} since it has larger support count than {a,b,d} and {a,d,e}.

figur 6.18

closed frequent itemset remove some of the redundant association rules. All maximum frequent itemsets are closed because none of the maximal frequent itemsets can have the same support count as their immediate supersets.

section 6.5 Alternative methods for generating frequent itemsets

Apriori - earliest, significant performance improvement, but still incurs considerable I/O overhead since it requires several passes. If transaction width is too high, it can also decrease.

Traversal of itemset lattice (tre fra {} på topp til {a,b,c,d,e} på bunn).

General-to-specific, specific-to-general, bidirectional: apriori uses this search strategy. Merging of k-1-itemsets to obtain candidate k-itemsets. Alternatively can look at most specific first and then more general.

Bidirectional combine. Can find frequent itemset border.

figur 6.19

Equivalance classes: Partition the lattice into disjoint groups of nodes (equivalence classes). A frequent itemset generation algorithm is used within each class. Equivalence classes can also be defined according to its prefix or suffix label of an itemset. First search for frequent itemset with prefix a, then b,c, d, etc. Suffix starts from the back.

figur 6.20

Breadth-first, depth-first: Apriori is a breadth-first. Starts with 1-itemsets, then 2-etc.

Depth first is often used by algorithms designed to find maximal frequent itemsets. Finds the border more quickly.

figur 6.21

figur 6.22

Representation of transaction data set choice of representation can affect the I/O cost.

Horizontal: adopted by apriori etc.

vertical: TID list may be too large to fit into memory, and need more sophisticated methods to compress TID-list.

section 6.6 FP-growth algorithm

Not generate-and-test. Instead it encodes the data set using a compact data structure called an FP-tree and extracts frequent itemsets directly from this structure.

FP-tree representation Compressed representation of the input data. Read the transactions one at a time, and map each transaction onto a path in the tree. Paths may overlap due to similar items. More overlap is more compression. If tree can fit in main memory, frequent data sets can be extracted directly without passes over data stored on disk.

figur 6.24 -- les beskrivelse 1-5 under Best-case scenario. All transactions contains the same items and there is only one branch in the tree. Worst case: none of the branches overlap. The pointers requires space, so the tree requires more space in memory than original data set.

Pointers (between nodes that have the same items) help to facilitate the rapid access of individual items in the tree.

If ordering by decreasing support, it will result in this tree: figur 6.25

section 6.7 Evaluation of association patterns

Need to have evaluation criterias to decide the quality of the association patterns made by the algorithms.

First set of criteria derived from statistical arguments. Patterns that involve a set of mutually independent items or cover very few transactions are considered uninteresting because they may capture spurious relationships in the data. Patterns can be eliminated by applying an: objective interestingness measure uses statistics derived from data to determine if a pattern is interesting. support, confidence, correlation.

Second set: subjective arguments. A pattern is considered subjectively uninteresting unless it reveals unexpected information about the data or provides useful knowledge that can lead to profitable actions. {Butter} → {Bread} vs {Diapers} → {Beer}. Requires a lot of domain knowledge. Approaches for incorporating subjective knowledge into pattern discovery: visualization - interpreting and verifying the discovered patterns. template-based approach - only rules that satisfy a user-specified template are returned to the user. subjective interestingness measure - ?? a subjective measure can be defined based on domain info such as concept hierarchy or profit margin of items. the measure can then be used to filter patterns that are obvious and non-actionable.

Objective measures of interestingness A data driven approach for evaluating the quality of association patterns. Domain-independent, requires minimal input from the users, other than a threshold.

An objectibe measure is usually computed based on the frequency counts tabulated in a contingency table: (figur 6.7)

Limitations of the support-confidence framework Support eliminates low support items, even though they might be of interest. Limitation of confidence: A tea-drinker that drinks coffee is 75% chance, and seems like a good rule, but when 80% drinks coffee regardless of tea or not. Therefore the rule is not good.

(skjønte ikke helt det som står på s. 373)

Interest factor tea-coffee example show that high-confidence rules can be misleading because they ignores the support. Can solve by applying a metric; lift: Lift =c(AB) s(B) Which computes the ratio between the rule's confidence and the support of the imteset in the rule consequent.

Binary values, it is called interest factor I(A,B)=s(A,B)s(A) x s(B)=Nf11f1+ f+1 ….????....

CHAPTER 8 CLUSTER ANALYSIS: BASIC CONCEPTS AND ALGORITHMS

Cluster analysis divide the data into groups (clusters) that are meaningful (should capture real structure of data) or/and useful (ex. for summarization).

Clustering Dividing into groups (clustering).

Classification Assigning particular objects to these groups (classification).

Taxonomy hierarchical classification.

Fields for clustering use biology: find groups of genes that have similar functions. information retrieval: group search results into a small number of clusters. Climate: find patterns in the atmospheric pressure and areas of ocean that have a significant impact on climate. psychology and medicine: identify different types of depression. detect patterns in the spatial or temporal distribution of a disease. business: segment customers into a small number of groups for additional analysis and marketing activities.

Cluster prototype A data object that is representative for the other objects in the cluster. Used by several techniques.

Cluster analysis The study of techniques for finding the most representative cluster prototypes: summarization Apply the analysis technique (algorithm) to a small set of only cluster prototypes. Should result in almost as good result as for whole set, but depends on accuracy and number of prototypes. compression table created with the prototypes for each cluster (prototype has an integer value that is its index in the table). Each object is represented by the index of the prototype associated with its cluster = vector quantization. Used with image, sound, video data where many of the data is highly similar to another some loss of information is acceptable substantial reduction in the data size is desired efficiently finding nearest neighbours can require computing the pairwise distance between all points. Reduce number of computations by using the prototypes if objects are relatively close to their prototype. if two prototypes are far apart, their objects cannot be nearest neighbours. The nearness is measured by the distance between their prototypes.

section 8.1 Overview

Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The goal is similar objects in a group, that is different from objects outside the group. The greater the similarity, the greater the difference between groups = better clustering.

The definition of a cluster is imprecise, and it should depend on the nature of the data and the desired result.

Clustering could be regarded as a form of classification in that it creates a labeling of objects with class (cluster) labels. Cluster analysis is sometimes referred to as unsupervised classification (supervised is as in chapter 4).

Different types of clustering

Hierarchical versus partitional Nested (hierarchical) clustering allowed with subclusters, one get a set of nested clusters organized as a tree. often (but not always) the leaf nodes are singleton clusters of individual data objects. Can be viewed as a sequence of partitional clusterings. unnested (partitional) clustering each object is in exactly one cluster(subset).

Exclusive versus overlapping versus fuzzy exclusive clustering assign each object to a single cluster. If it is reasonable to place an object in several clusters, it is addressed as non-exclusive clustering. overlapping (non-exclusive) clustering can simultaneously belong to more than one group (class). a student can both be enrolled and an employee at an university. rather than making an arbitrary assignment with placing an object in only one cluster, one can place the object in all of the “equally good” clusters. fuzzy clustering every object belongs to every cluster with a membership weight that is between 0 and 1 (absolutely belongs). often impose that the weights should sum up to 1 for each object. good to avoid assigning an object to only one when it may (but not certain as the student/employee) be close to several.

Complete versus partial complete clustering assigns every object to a cluster. May be desired if clustering is to organize documents, then all documents must be included to guarantee browsing. partial clustering does not. Some data objects may not belong to well-defined groups. Some data objects may be noise, outliers or uninteresting background.

Different types of clusters

Well separated data contains natural clusters that are quite far from each other. The distance between two points within a group is always smaller than the distance between two points from different groups. Can have any shape (globular, etc.) Prototype-based cluster where each object is closer to the prototype in the cluster than to a prototype in a different cluster. For continuous attributes, the prototype is often a centroid. For categorical attributes, the prototype is often medoid (the most representative point in the cluster). Center-based clusters: when the prototype is the most central point. tend to be globular clusters. graph-based when data is represented as a graph and nodes with connections. A cluster is a group of nodes with connections, but no connections with nodes outside the cluster. Important example: contiguity-based clusters two objects are connected only if they are within a specified distance of each other. can have trouble when noise is present. other example: clique set of nodes in a graph that are completely connected to each other. If we start by addings connections in the order of their distance, a cluster is made when there is a clique. Tend to be globular. Density-based cluster is a dense region of objects that is surrounded by a region of low density. Often employed when the clusters are irregular of intertwined, and when noise and outliers are present. shared-property (conceptual clusters) clusters as a set of objects that share some property. Emcompasses all previous definitions. Includes new types of clusters. Process of finding new (rare) tpes is called coceptual clustering.

Road map K-means property-based, partitional clustering technique that attempts to find a user-specified number of clusters (K), which are represented by their centroids. agglomerative hierarchical clustering refers to a collection of closely related clustering techniques that produce a hierarchical clustering by starting with each point as a singleton cluster and merge. DBSCAN density-based clustering algorithm that produces a partitional clustering, in which the number of clusters is automatically determined by the algorithm. low-density region points is considered noise and omitted; does not produce a complete clustering.

figur 8.2

section 8.2 K-means

K-means define a prototype in terms of a centroid (mean of a group of points), typically applied to objects in a continuous n-dimensional space.

K-medoid defines a prototype in terms of a medoid (most representative point for a group of points), can be applied to a wide range of data since it requires only a proximity measure for a pair of objects. Must be an actual data point.

Basic K-means algorithm choose K initial centroids (user specified). K is the number of wanted clusters. each point is then assigned to the closest centroid. Clusters are formed. For each cluster the centroid is updated based on the points assigned to the cluster. Repeat and update until no point changes clusters (the centroids remain the same).

Assigning points to the closest centroid Proximity measures: Eucledian (L2), data points in euclidean space (more general) cosine similarity, more appropriate for documents (more general) Manhatten (L1), Euclidean data (more special) Jaccard measure, documents (more special) Low-dimensional euclidean space it can be possible to avoid many of the computations. Also Bisecting K-means is a method for reducing the number of similarities computed.

Centroids and objective functions need to specify an objective function and the proximity measure, then the centroids that we should choose can often be determined mathematically.

Data in Euclidean space As objective function, can choose SSE (sum og squared error) aka. scatter: calculate the error of each data point (its Euclidean distance to the nearest centroid). If two different cluster sets from two different runs of K-means have different SSE, we choose the one with the lowest SSE (the centroids are better). SSE = i = 1KxC1dist(ci,x)2 dist is the standard Euclidean(L2) distance between two objects in Euclidean space.

(Equation 8.2 s.500) The mean of the ith cluster:

The mean is the centroid that minimizes the SSE. Document data K-means can also be used on document data and with cosine similarity measure. Cohesion: to maximize the similarity of the documents in a cluster to the cluster centroid. The mean is the cluster centroid. Total cohesion (eq.8.3), the same as SSE:

The general case

proximity function centroid objective function Manhattan (L1) median Minimize sum of the L1 distance of an object to its cluster centroid squared Euclidean (L22) mean minimize sum of the squared L2 distance of an object to its cluster centroid cosine mean maximize sum of the cosine similarity of an object to its cluster centroid Bregman divergence mean minimize sum of the Bregman divergence of an object to its cluster centroid.

Bregman divergence is actually a class of proximity measures that includes the squared Eucledian distance, the Mahalanobis distance and cosine similarity.

Choosing initial centroid random, different runs produce different total SSEs. Common approach, with poor resulting clusters. approach: do different runs, select clusters with minimum SSE. take a sample of points and cluster them using a hierarchical clustering technique. K clusters are extracted from the hierarchical clustering, and the centroids of those clusters are used as the initial centroids. practical only if: sample is relatively small K is relatively small compared to sample size. Select first point at random, or take the centroid of all points. Then, for each successive initial centroid, select the point that is farthest from any of the initial centroids already selected. Randomly and well-separated centroids. outliers! :(expensive to compute the farthest point from the current set of initial centroids. often therefore applied to the sample of the points. outliers often don’t show up in samples since they’re rare. Time and space complexity little space required (only data points and centroids are stored). O((m+K)n), where m is the number of points and n is the number of attributes.

The time required is also little. O(IKm*n) where I is the number of iterations required for convergence.

Additional issues Handling empty clusters if no points are allocated to a cluster during the assignment step. Need a replacement strategy: choose the point that is farthest away from any current centroid eliminates the point that contributes most to the SSE. choose the replacement centroid from the cluster that has the highest SSE. split cluster and reduce the overall SSE. For several empty clusters, the procedure is repeated.

Outliers Reduce the representativeness of the prototypes and increase the SSE. Useful to discover and eliminate outliers beforehand. But not always (data compression, financial analysis etc.).

Can remove them afterwards by tracking every point and remove the largest contributor over several runs to a high SSE.

Remove small clusters, since they frequently represent groups of outliers.

Reducing the SSE with Postprocessing Find more clusters (higher K). But in most cases we want to improve SSE without increasing K. Often possible because K-means converge to a local minimum.

Techniques to fix up resulting clusters: splitting or merging clusters Escape local SSE minima and still produce a clustering solution with the desired number of clusters. splitting phase: Strategy: slit a cluster. The one with the largest SSE or standard deviation for an attribute etc. strategy: introduce a new cluster centroid. The furthest point or choose randomly from the points with the highest SSE. merge phase: strategy: disperse a cluster.. Accomplished by removing the centroid that correspond to the cluster and reassigning the points to other clusters. Often the cluster that increases the SSE the least. strategy: merge to clusters. the clusters with the closest centroids, or merge the ones that result in the smallest increase in total SSE.

Updating centroids incrementally Centroids can be updated for each assigned point instead of when all points are assigned. It required 0 or 2 updates to cluster centroids at each point (the point stays in the cluster (0) or moves to a new cluster (2)). No empty sets are produced. Better accuracy and faster convergence if weight of points decrease. Difficult to make a good choice for the relative weight. Negative: introduces order dependency (the order the points are introduced). Can be addressed by randomizing points. more expensive.

Bisecting K-means Extension of basic K-means. Simple idea: obtain K clusters, split the set of all points into two clusters, select one of these clusters to split, and so on, until K clusters have been produced.

algoritme 8.2 s. 509:

Ways to split: can choose the largest cluster can choose the one with largest SSE can use a criterion based on both size and SSE. Often refine the resulting clusters by using their centroids as the initial centroids for the basic K-means algorithm. ?? (skjønte ikke helt hva som står videre på s. 509).

Recording the sequence of bisection, we can produce a hierarchical clustering.

K-means and different types of clusters Difficulties with finding the natural clusters (when clusters have non-spherical shapes or widely different sizes or densities). Can be solved if the number of clusters are increased. Find for instance 6-7 clusters instead of 2-3. Each small cluster is pure (contains only points from one of the natural clusters).

Strengths and weaknesses strengths simple can be used for a wide variety of data types efficient, even though multiple runs are often performed bisecting K-means are even more efficient weaknesses can’t handle all types of data non-globular clusters, different size and density etc. difficulties with outliers can help with detection and removal restricted to data for which there is a notion of a center (centroid) can use K-medoid, but is more expensive.

section 8.3 Agglomerative Hierarchical Clustering

Old, but in widespread use. Two basic approaches for hierarchical clustering: Agglomerative starts with points as individual clusters, and at each step merge the closest pair of clusters. Requires definition of cluster proximity. (most common) Divisive starts with one, all-inclusive cluster, and at each step split a cluster until only singleton clusters of individual points remain. Need to decide which cluster to split at each step and how to do the splitting.

Dendrogram tree-like diagram used to display the hierarchical clustering. Shows the relationships and the order of the merging (or splitting). (figur s. 516)

Nested cluster diagram Another method of graphical representation. (figur s. 516)

Defining proximity between clusters MIN defines cluster proximity as the proximity between the closest two points that are in different clusters, (or the shortest edge between two nodes in different subsets of nodes). Single link (similarities) MAX the proximity between the farthest two points in different clusters. complete link (similarities) Group average defines cluster proximity to be the average pairwise proximities (average length of edges) of all pairs of points from different clusters. figur 8.14 s. 517

Centroids If we use centroids, the cluster proximity is defined as the proximity between cluster centroids. Ward's method assumes that a cluster is represented by its centroid, but it measures the proximity between two clusters in terms of the increase in the SSE that result from merging two clusters.

Time and space complexity Space uses a proximity matrix. Storage of 1/2m2 proximities (assuming the matrix is symmetric), m is the number of data points. Space needed to keep track of the clusters is proportional to the number of clusters(m-1). Total space: O(m2). Time O(m2) required to compute the proximity matrix. m-1 iterations to merge clusters (m clusters at the start, and two clusters merged for each iteration). Overall time: O(m2 log m).

Specific techniques Single link/MIN: Add shortest links one at a time. Handles non-elliptical shapes well, but is sensitive to noise and outliers.

Complete link/MAX/CLIQUE Farthest points away, but the shortest link? (gjør eksemplet i boka). A group is not a cluster until all the points in it are completely linked (form a clique).

Group average An intermediate approach between the single and complete link approaches. cluster proximity:

where mi and mj is the size of clusters Ci and Cj

Ward's method and centroid methods Proximity between two clusters is the increase in the squared error that results when two clusters are merged. Uses same objective function as K-means clustering. Similar to the group average method when the proximity between two points is taken to be the square of the distance between them.

Possibilities of inversions. Two clusters that are merged may be more similar (less distant) than the pair of clusters that were merged in a previous step.

Key issues in hierarchical clustering Lack of global objective function the techniques decide locally with different criterias, at each step, which clusters should be merged. Ability to handle different cluster sizes applies only where it involves sums: ward's and group average. how to treat the relative sizes of the pairs of clusters that are merged. two approaches (applied to the points) weighted: treats all clusters equally gives the points in a cluster different weights unweighted: takes the number of points in each cluster into account. gives points in a cluster same weight unweighted is preferred unless other is stated (perhaps classes of objects have been unevenly sampled). Merging decisions are final once a decision to merge is done, it cannot be undone later. Prevents a local optimization criterion from becoming a global optimization criterion. The clusters are not even stable (a points in a cluster may be closest to a centroid in a different cluster). Some functions try to improve by moving branches of the tree around to improve a global objective function. Or uses a partitional clustering technique (K-means etc.) to create many small clusters, and then perform hierarchical clustering with the little clusters as starting point.

Strenght and weaknesses strengths used because underlying application (creation of a taxonomy) requires a hierarchy. Studies that says that these algorithms can produce better-quality clusters. Weaknesses Expensive (high storage and computational requirements). All merges are final (especially bad for noisy, high-dimensional data, for example a document). can be solved by first using K-means or another method (som beskrevet over).

section 8.4 DBSCAN

density-based clustering locates regions of high density that are separated from one another by regions of low density. DBSCAN is simple and effective density-based clustering algorithm.

Traditional density: center-based approach (other definitions of density is in chapter 9 (ikke pensum)). Density is estimated for a particular point in the data set by counting the number of points within a specified radius, Eps, of that point. It includes the point itself. Simple to implement. Need a proper Eps; Density of any point will depend on the radius (1(itself) to m (all points). Classification of points interior of a dense region (core point) of the number of points within a given a neighborhood around the point as determined by the distance function and a user-specified distance parameter, Eps, exceeds a certain threshold, MinPts (user-specified). on the edge of a dense region (border point) falls within a sparsely occupied region (noise point) All points that's not a core or a border point.

DBSCAN algorithm Any two core points that are close enough (within a distance Eps of one another) are put in the same cluster. Likewise, any border point that is close enough to a core point is put in the same cluster as the core point. Noise points are discarded. Label all points as core, border or noise points eliminate noise points put an edge between all core points that are within Eps of each other. Make each group of connected core points into a separate cluster. Assign each border point to one of the clusters of its associated core points.

Time and space complexity Time O(m x time to find points in the Eps-neighborhood), m is the number of points. worst case O(m2). Low-density, can use kd-trees to allow efficient retrieval of points within a given distance of a specified point, best-case: O(m log m). Space all dimensions: O(m). Need only to keep small amount of data for each point (cluster label, identification of core/border/noise).

Selection of DBSCAN parameters Basic approach: look at the behaviour of the distance from a point to its kth nearest neighbor (k-dist). Will be small for points in a cluster and large for those who's not, but will vary due to density. Compute k-dist for all points, arrange in increasing order (plot values), an sharp change of k-dist will be a suitable value for Esp. The value of k for the selected Eps is MinPts (points for which k-dist is less than Eps = core points).

Clusters of varying density DBSCAN can have trouble with density if the density of clusters varies widely. High Eps will detect high density, but lower density-clusters are marked as noise. Low Eps will find low-density clusters, but higher density clusters will be looked at as one.

Strength and weaknesses strenght density-based definition of cluster relatively restistant to noise can handle arbitrary shapes and sizes weaknesses trouble with varying density high-dimensional data (hard to define density) can be expensive when the computation of nearest neighbors requires computing all pairwise proximities (high-dim data).

section 8.5 CLUSTER EVALUATION

Not well-developer of commonly used part of cluster evaluation. Important. Almost every clustering algorithm will find clusters in a data set, even if that data set has no natural cluster structure.

Important issues of cluster validation determining the cluster tendency of a set of data; distinguishing whether non-random structure exists in the data. Determining the correct number of clusters. Evaluating how well the results of a cluster analysis fit the data without reference to external information. comparing the results of a cluster analysis to externally known results, such as externally provided class labels. Comparing two sets of clusters to determine which is better.

1, 2 and 3 use no external information, they are unsupervised techniques. 5 is both. 3,4,5 look at individual clusters (maybe no need to analyse all).

Evaluation measures unsupervised (internal indices) measures the goodness of a clustering structure without respect to external information. SSE. Further divided into two classes measures of cluster cohesion (tightness, compactness) how closely related the objects in a cluster are measures of cluster separation (isolation) how distinct or well-separated a cluster is from other clusters. Supervised (external indices) measure the extent to which the clustering structure discovered by a clustering algorithm matches some external structure. Entropy (cluster label match) Relative compare different clusterings of clusters. Supervised/unsupervised evaluation measure with the purpose of comparison. Use both SSE, Entropy etc.

Unsupervised cluster evaluation using cohesion and separation many internal measre of cluster validy for partitional clustering schemas are based on the nitions of cohesion or separation. Overall cluser validity for a set of K clusters as a weighted sum of the validity of indiciduals clusters; overall validity = i = 1kwivalidity(Ci) Validity function can be cohesion, separation or some combinations of those quantities. Cohesion: higher values are best separation: lower values are best.

Graph-based view of cohesion and separation cohesion of a cluster can be looked at as the sum of the weights of the links in the proximity graph that connect points within the cluster. cohesion (Ci) = x og y Ciproximity (x,y)

The separation between two clusters can be measured by the sum of the weights of the links from points in one cluster to points in the other cluster. separation(Ci,Cj) = x Ci, yCjproximity (x,y) Prototype-based view of cohesion and separation Cohesion can be defined as the sum of the proximities with respect to the prototype (centroid/medoid) of the cluster. cohesion (Ci) = x Ciproximity (x,ci) (squared Euclidean distance) Separation between two clusters is the proximity of the two cluster prototypes. two measures of separation. ci is the centroid for Ci and c is the overall prototype. separation (Ci) = proximity(ci , cj) separation(Ci) = proximity (ci,c) figur 8.28 s.538

Overall measures of cohesion and separation Any unsupervised measure of cluster validity potentially can be used as an objective function for a clustering algorithm, and vice versa.

(her er endel på eget ark)

Unsupervised cluster evaluation using the proximity matrix measuring cluster validity via correlation evaluate the goodness by looking at the correlation between the actual similarity matrix and an ideal version of the similarity matrix based on the cluster labels. Ideal cluster: similarity 1 to points in the cluster, 0 to points outside the cluster.

Matrix: one row and one column for each data point. Assign 1 to an entry if it belongs to the same cluster. 0 if not.

High correlation indicates that the points that belong to the same cluster are close to each other.

technique: order the similarity matrix with respect to cluster labels and then plot it.

takes 0(m2) to compute the proximity matrix (m is the number of objects). Large, but can use sampling. May need to oversample small clusters and undersample large ones to obtain adequate representation of all clusters.

unsupervised evaluation of hierarchical clustering previous approaches: partitional clusterings.

cophenetic distance the distance between two objects is the proximity at which an agglomerative hierarchical clustering technique puts the objects in the same cluster for the first time. Se eksempel s.545.

CoPhenetic correlation coefficient (CPCC) is the correlation between the entries of this matrix and the original dissimilarity matrix and is a standard measure of how well a hierarchical clustering fits the data. To evaluate which type of hierarchical clustering is best for a particular type of data. Better with higher values.

Determining the correct number of clusters can try to find natural number in data set by looking for the number of clusters at which there is a knee, peak or dip in the plot of the evaluation measure. clusters may be intertwined or overlapping, or can be nested. can look at the average silhoutte coefficient and the SSE together (s.847) when caution is needed.

cluster tendency to determine if a data set has clusters, one can try to cluster it. but algorithms will find some even if there are not any natural. use several algorithms, can approve if some of the clusters are of good quality. try to evaluate whether a data set has clusters without clustering. use statistical tests for spatial randomness. many approaches, most in low-dim Euclidean space. Supervised measures of cluster validity have external information, usual to measure the degree of correspondence between the cluster labels and the class labels. The point of comparison is to see if a manually labeling can be done just as good by a cluster analysis. Two approaches: Classification-oriented use measures of classification (entropy, purity, F-measure, precision, recall, etc) to evaluate the extent to which a cluster contains objects of a single class. entropy: the degree to which each cluster consists of objects of a single class. (kapittel 4) purity: the extent to which a cluster contains objects of a single class.
purity= sum (from i=1 to K) (mi / m) * ei pi = max(j) pij Se på entropy, den er nesten lik. precision: the fraction of a cluster that consists of objects of a specified class. precision of cluster i with respect to class j is precision(i,j)=pi j Recall: the extent to which a cluster contains all objects of a specified class. recall(i,j) = mij / mj mj is the number of objects in class j. F-measure: a combination of precision and recall that measures the extent to which a cluster contains only objects of a particlular class and all objects of that class. F(i,j) = (2 x precision(i,j) x recall(i,j))/(precision(i,j)+(recall(i,j)). Se eksempel og tabell s.550 similarity-oriented similarity measures for binary data (Jaccard measure etc.) to measure the extent to which two objects that are in the same class are in the same cluster and vice versa. use matrices: ideal cluster similarity matrix: has 1 in the ijth entry if two objects i and j, are in the same cluster. 0 otherwise. ideal class similarity matrix: 1 in the ijth entry if two objects i and j, are in the same class. 0 otherwise. regner ut correlation mellom de to, hvordan? Rand statistic and jaccard coefficient to measure cluster validity: Rand statistic = f00 + f11 / alle jaccard coeffisient = f11 / (f01 + f10 + f11)

same cluster different cluster same class f11 f10 different class f01 f00 Cluster validity for hierarchical clusterings supervised evaluation for hierarchical clustering is more difficult: pre existing hierarchical structure often does not exist. Key idea evaluate whether a hierarchical clustering contains, for each class, at least one cluster that is relatively pure and includes most of the objects of that class. Compute F-measure for each class. For each class, take the max F-measure for any cluster. Calculate the overall F-measure for the hierarchical clustering by computing the weighted average of all per-class F-measures weights based on class size F = mjm maxi F(i,j) max is taken over all clusters i, at all levels. mj is the number of objects in class j. m is the total number of objects.

Assessing the significance of cluster validity measures Difficult to interpret the significance of the number/value obtained by a cluster validity. purity: 1 good, 0 not good Entropy: 0 good SSE: 0 good. May not be a minimum or maximum value. And intermediate values need to be interpret. Can use a limit (absolute standard, tolerate a certain level of error etc.). Can interpret the value of our validity measure in statistical terms. the value is good if it is unusual (if it is unlikely to be the result of random chance). When comparing two clusterings (relative measure) the difference may not need to be significant. 0,1% different would many not regard as significant. two aspects to this significance: whether the difference is statistically significant (repeatable) whether the magnitude of the difference is meaningful with respect ot the application.

Chapter 5 Classification: Alternative techniques

Chapter 6 Association Analysis: Basic Concepts and Algorithms

Chapter 8 Cluster Amalysis: Basic Concepts and Algorithms

Written by

sigveseb, cristea, kjersva, sunayana, iverjo, sindreij, hanskhe, Esso

Last updated: 7 years ago.