POLYTECHNIC UNIVERSITY
Department of Computer Science / Finance and Risk Engineering

Preparing the Data

K. Ming Leung

Abstract: Techniques for preprocessing data for data mining are discussed. Issues include scaling numerical data, attribute transformation, dealing with missing values, representation of time-dependent data, and outlier detection.

Directory

- **Table of Contents**
- **Begin Article**

Table of Contents

1. Representation of Raw Data

Every sample in our data collection is described with many features. There are many different types of values for every feature. The most common two types are: numeric and categorical.

Numeric values include integer and real-value variables, such as trade volumes and intraday-high stock prices. An attribute with numeric values has two important properties: its values have an order relation ($2 < 5$ and $5 < 7$), and a distance relation (distance between 2.3 and $4.2 = d(2.3, 4.2) = 1.9$).

Categorical variable have neither of these relations. Examples of categorical variables are eye color, sex, or country of citizenship. Two values of a categorical variable can be either equal or not equal (i.e. only the equality relation is supported). For example, Barack Obama and Hillary Clinton have the opposite sex but have the same country of citizenship. A categorical variable with two values can be converted to a numeric binary variable with two values: 0 or 1. A categorical variable with N values can be converted to a numeric binary variable with N values. For example, if the variable eye-color has 4 values:

black, blue, green and brown, they can be coded with 4 binary digits.

Featurevalue	Code
Black	1000
Blue	0100
green	0010
Brown	0001

Another way is to classify variable, based on its values, as continuous or discrete.

Discrete variables are measured using one of two kinds of nonmetric scales – nominal and ordinal.

A nominal scale is an order-less scale, which uses different symbols, characters, or numbers to represent the different states (values) of the variable. For example, in a utility company, customers are identified as residential, commercial, and industrial. These values can be coded alphabetically as A, B, and C, or numerically as 1, 2, and 3 (no ordering or distance measure related to these values). Another example of a numeric nominal scale is our 9-digit social security numbers.

An ordinal scale consists of ordered,discrete gradations or rankings. An ordinal variable is a categorical variable for which an order relation is defined but not a distance measure. An example is the course grade of a student. An 'A' is certainly better than a 'B', but how much better is not clearly defined. Ordinal variable are closely related to the linguistic or fuzzy variables commonly used in spoken English: e.g.AGE (with values young, middle-aged, and old) and INCOME LEVEL (with values low, middle-class, upper-middle-class, and rich).

One can also classify data based on its relation with time. Data that do not change with time are called static data. One the other hand attribute values that change with time are called dynamic or temporal data. Special consideration is need to treat dynamic data.

1.1. The Curse of the Dimensionality

In data mining problems, data samples often have a large number of features or attributes. If each sample can be represented as a vector in a vector space, then the vector is defined in a very high dimensional space.[8] This high dimensional data set causes problems known as the ”curse of dimensionality”. The curse of dimensionality is a term coined by Richard Bellman (a mathematician celebrated for his invention of dynamic programming in 1953) to describe the problem caused by the exponential increase in volume associated with adding extra dimensions to a (mathematical) space.

The curse of dimensionality is a significant obstacle to solving dynamic optimization problems by numerical backwards induction when the dimension of the ’state variable’ is large. It also complicates machine learning problems (used for example in data mining) that involve learning a ’state-of-nature’ (maybe infinite distribution) from a finite (low) number of data samples in a high-dimensional feature space.

The properties of high-dimensional spaces often appear counter-intuitive because our experience with the physical world is in a low-dimensional space (of 2 or 3). Four important properties of high-dimensional data are often the guidelines in the interpretation of input data and data mining results:

- The size of a data set yielding a given density of data points in an n-dimensional space increases exponentially with dimensions. Because if a one-dimensional sample containing m data points has a satisfactory level of density, then to achieve the same density of points in n dimensions, we need m^n data points. For example, if 100 data points gives a sufficiently dense samples in one dimension, then to obtain the same density in a sample space of 10 dimensions, we will need $100^{10} = 10^{20}$ data samples! Because of the curse of the dimensionality, even for the largest real-world data sets, the density is often still relatively low, and may be unsatisfactory for data mining purposes.

- As the dimensionality, n, increases, so does the radius needed to enclose a given fraction of the data points in an n-dimensional space. Suppose all the data points in an n-dimensional space lies within the n-dimensional hypercube. For a given fraction of data points, p, the edge length, $e_n(p)$, of a hypercube enclosing that fraction of data points is given by

  $$e_n(p) = p^{1/n}.$$

  For example if one wishes to enclose 10% of the sample ($p =$

0.1), the corresponding edge for a two-dimensional space will be $e_2(0.1) = 0.32$, for a three-dimensional space $e_3(0.1) = 0.46$, for a 10-dimensional space $e_{10}(0.1) = 0.80$, and for a 15-dimensional space $e_{15}(0.1) = 0.96$.

This shows that a large neighborhood is required to capture even a small portion of the data in a high-dimensional space.

- Almost every point is closer to an edge than to another sample in a high-dimensional space. For a sample of size m, the expected distance d between data points in an n-dimensional space is

$$d_n(m) = \frac{1}{2}\left(\frac{1}{m}\right)^{1/n}.$$

For example, for a two-dimensional space with 10,000 points the expected distance is $d_2(10000) = 0.005$, for a 10-dimensional space with the same number of sample points $d_{10}(10000) = 0.199$, and for a 100-dimensional space $d_{100}(10000) = 0.456$. Keep in mind that the maximum distance between any point to the edge occurs at the center of the cube, and it is 0.5 irrespective of the dimensionality.

- Almost every point is an outlier.

These rules have serious consequences when dealing with a finite number of samples in a high-dimensional space. From properties (1) and (2) we see the difficulty in making local estimates for high- dimensional samples; we need more sample points to establish the required data density for performing planned mining activities. Properties (3) and (4) indicate the difficulty of predicting a response at a given point, since any new point will on the average be closer to an edge than to the training examples in the central part.

2. Characteristics of Raw Data

Raw data typically have low quality. Many experts in data mining will agree that the most critical steps in a data-mining process is the preparation and transformation of the initial data set. This task often receives little attention in the research literature, mostly because it is considered too application-specific. In the real world, more effort is expended preparing data than applying data-mining methods.

There are two central tasks for the preparation of data:

1. To organize data into standard relational table form.
2. To prepare data sets that lead to the best data-mining performances.

3. Transforming of Raw Data

We will review some general types of transformation of data that are not problem-dependent.

1. Normalizations

   Some data mining methods, typically those that are based on distance computation between points in an n-dimensional space, may need normalized data for best results. If the values are not normalized, the distance measure will overweight those features that have, on an average, larger values. The measured values can be scaled to a specific range, e.g., [-1, 1] or [-1, 1]. Here are three simple and effective normalization techniques:

   a Decimal scaling Decimal scaling moves the decimal point but still preserving most of the original digit value. The typical scale maintains the values in the range of [-1, 1]. Supposed $v(i)$ is the value of the feature v for sample i. First the maximum $|v(i)|$ in the data set, and then the decimal point is moved until the new, scaled, maximum absolute value is just less than 1. The same scaling is then applied to all other $v(i)$. That is, we define scaled variables

$$v'(i) = v(i)/10^k$$

such that k is the smallest integer such that $\max_i |v'(i)| < 1$. Note that v' has exactly the same unit as v, they are scaled differently only.

For example, if the largest value in the data set is 455 and the smallest value is -834, then the value with the largest magnitude is -834. It is scaled to -0.834 by dividing it by $1,000$. All other values of the feature are then divided by the same factor. The scaled values then vary from -0.834 to 0.455.

b Min-max normalization Sometimes there is a problem with the above normalization. Suppose that the data for a feature v are in a range between 150 and 250. The above normalization will yield data in the narrow subinterval [0.15, 0.25] of the entire range. To obtain better distribution of values on a whole, normalized interval, e.g., $[0, 1]$, we can use the min-max formula

$$v'(i) = \frac{v(i) - \min_i v(i)}{\max_i v(i) - \min_i v(i)}$$

where the minimum and the maximum values for the feature v are computed on a set automatically, or they are estimated by an expert in a given domain. Note that v' now is dimensionless.
Similar transformation may be used for the normalized interval $[-1, 1]$. If v" is the new variable then we assume that it is related linearly to v':

$$v\text{"} = av' + b.$$

To find a and b, we require that when $v' = 0$, $v\text{"} = -1$, and when $v' = 1$, $v" = 1$. Thus we have two equations:

$$-1 = b \qquad 1 = a + b$$

from which we have $a = 2$ and $b = -1$. Thus the relation is

$$v" = 2v' - 1.$$

c standard deviation normalization First for a feature v, the mean value $mean(v)$ and the standard deviation $sd(v)$ are computed for the entire data set. Then for each sample i, the feature value is transformed using the equation

$$v'(i) = \frac{v(i) - mean(v)}{sd(v)}.$$

For example if the initial set of values of the attribute is $v = \{1, 2, 3\}$, then $mean(v) = 2$, $sd(v) = 1$, and the set of normalized values is $v' = \{-1, 0, 1\}$.

2. Data smoothing Simple smoothers can be specified that average similar measured values. For example, if the set of values for the given feature F is $\{0.93, 1.01, 1.001, 3.02, 2.99, 5.03, 5.01, 4.98\}$,

then the smoothed values may be $\{1.0, 1.0, 1.0, 3.0, 3.0, 5.0, 5.0, 5.0\}$. Smoothing reduces the number of real values for a feature and thus reduces the dimensionality of the data space and thus improve data mining performance.

3. Differences and ratios Two simple transformations, differences and ratios, could improve data mining performances by better specifying the goal of data mining.

   Instead of optimizing the output $s(t+1)$ of a data mining process, it may be more effective to set the goal of a relative move from current value to a final optimal $s(t+1) - s(t)$. The range of values for the relative moves is generally much smaller.

   Sometimes using $s(t+1)/s(t)$ instead of $s(t+1)$ as the output of a data mining process may improve performance.

   Difference and ratios can also be used to transform not only output values but input values as well. For example, in many medical data sets, there are two features of a patient, height and weight, that are taken as input parameters. Many applications show that better diagnostic results are obtained when an initial transformation is performed using the body-mass index (BMI), which is the weighted ratio between weight and height. This composite feature (which is essentially a ratio) is better than the original parameters to describe some of the characteristics of the patient, such as whether or not the patient is overweight.

4. Missing Values in Data

For many real-world applications, many samples may have feature values missing. If the number of complete samples is sufficiently large, then incomplete samples may be simply deleted from the data before data mining is performed.

It is not uncommon that a large fraction of the samples have missing values, in that case we need to use data mining methods that are insensitive and can deal with missing values, or to find ways to fill-in the missing values.

Sometimes a data miner, together with the domain expert, can manually examine samples that have no values and enter a reasonable, probable, or expected value, based on domain experience. This method works only for small numbers of missing values and relatively small data sets. It is possible to introduce error into the data set.

There are three simple automatic methods to replace missing values:

1. Replace all missing values with a single global constant (its value is highly application-dependent).
2. Replace a missing value with its feature mean.
3. Replace a missing value with its feature mean for the given class (for classification problems where samples are classified in advance).

Problems may result due to incorrect values and bias introduced into the data.

One other way to handle samples with missing values is to treat them as "don't care" values. That is we suppose that these values do not have any influence on the final data-ming results. In that case, a sample with the missing value may be extended to the set of artificial samples, where, for each new sample, the missing value is replaced with one of the possible feature values.

This approach has the problem that the number of samples is made much larger. For example, if one three-dimensional sample X is given as $X = \{1, ?, 3\}$, where the second value is missing, the process will generate 5 artificial samples if the feature domain is $[0, 1, 2, 3, 4]$

$$\{1, 0, 3\}, \{1, 1, 3\}, \{1, 2, 3\}, \{1, 3, 3\}, \{1, 4, 3\}$$

Another approach is generate a predictive model, based on techniques such as regression, Bayesian formalism, clustering, or decision-tree induction, to predict each of the missing values.

5. Time-Dependent Data

Real-world problems with time dependencies required special preparation and transformation of data. We will start with the simplest case – a single feature measured over time. This feature has a series of values measured over fixed time units. For example, a temperature reading measured every hour, or the price of a stock recorded at the end of every day. This classical univariate time-series for the variable

X can be expressed as

$$\mathbf{X} = \{t(1), t(2), t(3), \ldots, t(n)\}$$

where $t(n)$ is the most recent value.

For many time-series problems, the goal is to predict $t(n+1)$ from previous values of the feature. One of the most important steps in preprocessing such data is the specification of a window or a time lag.

For example, if the time series consists of eleven measurements

$$\mathbf{X} = \{t(0), t(1), t(2), t(3), t(4), t(5), t(6), t(7), t(8), t(9), t(10)\}$$

and if the window is chosen to be 5, then the input data is reorganized into a tabular form with 6 samples, which is more convenient for data mining.

Sample	M1	M2	M3	M4	M5	Next Value
1	t(0)	t(1)	t(2)	t(3)	t(4)	t(5)
2	t(1)	t(2)	t(3)	t(4)	t(5)	t(6)
3	t(2)	t(3)	t(4)	t(5)	t(6)	t(7)
4	t(3)	t(4)	t(5)	t(6)	t(7)	t(8)
5	t(4)	t(5)	t(6)	t(7)	t(8)	t(9)
6	t(5)	t(6)	t(7)	t(8)	t(9)	t(10)

Instead of predicting the next value, the problem can be modified to predict values in the future several time units in advance. That is, given the values $t(n-i),\ldots,t(n)$, we want to predict the value of $t(n+j)$ for fixed given positive integers i and j. Note that the window size gives the number of artificial feature in a tabular representation of the time series, and it is given by $i+1$. The total number of samples in the standard tabular form is given by the total of original data points minus $i+j$.

In the previous example, taking $i=4$ (so that the window size is 5) and $j=3$, the preprocessed data can be put in standard tabular form as shown.

Sample	M1	M2	M3	M4	M5	Next Value
1	t(0)	t(1)	t(2)	t(3)	t(4)	t(7)
2	t(1)	t(2)	t(3)	t(4)	t(5)	t(8)
3	t(2)	t(3)	t(4)	t(5)	t(6)	t(9)
4	t(3)	t(4)	t(5)	t(6)	t(7)	t(10)

Of course, the more further out in the future, the more difficult and less reliable is the forecast.

The goal for a time series can easily be changed from predicting the next value in the series to classification into one of predefined categories. For example, instead of predicting the value of $t(n+1)$, the new classified output will be binary: T for $t(n+1) \geq thresholdvalue$ and F if $t(n+1) < thresholdvalue$.

The best choice of window size is obviously problem-specific. The answer depends on the knowledge of the application and past experiences.

Two small a window size means that we are not using enough of the most relevant recent data to make a prediction. Using too large a window means that (1) we are using too many older values of a feature which may be historical relics that are no longer relevant, (2) the dimensionality of the problem is large.

Besides standard tabular representation of time series, sometimes it is better to perform further preprocessing on the data. It may be better to predict the difference $t(n+1)-t(n)$ or the ratio $t(n+1)/t(n)$ than on the value $t(n+1)$. When differences or ratios are used to specify the goal, features measuring the differences or ratios for input features may also be advantageous.

Time-dependent samples are specified in terms of a goal and a time lag or window of size m. One way of summarizing features in the data set is to average them, producing "moving averages". A single average summarizes the most recent m features values for each sample, and for each increment in time moment i, its value is

$$MA(i,m) = \frac{1}{m} \sum_{j=i-m+1}^{i} t(j).$$

Knowledge of the application can aide in specifying reasonable sizes for m. Error estimation should validate these choices.

Moving averages weight all time points equally in the average. Typical examples are moving averages in the stock market such as 200-day moving average for DOW and NASDAQ. The objective is to smooth neighboring points by a moving average to reduce the random variation and noise components.

Another type of average is an exponential moving average (EMA) that gives more weight to the most recent time points. It is defined recursively as

$$EMA(i,m) = p\,t(i) + (1-p)\,EMA(i-1,m-1), \qquad EMA(i,1) = t(i)$$

where p is a fixed parameter value between 0 and 1. Recursion starts with $m = 1$, then $m = 2$, etc. until the final value of m.

$$EMA(i,1) = t(i)$$

$$\begin{aligned} EMA(i,2) &= p\ t(i) + (1-p)\ EMA(i-1,1) \\ &= p\ t(i) + (1-p)\ t(i-1) \\ EMA(i,3) &= p\ t(i) + (1-p)\ EMA(i-1,2) \\ &= p\ t(i) + (1-p)\ \left(p\ t(i-1) + (1-p)\ t(i-2)\right) \end{aligned}$$

The larger p is, the more the the most recent time points are counted in the EMA. In the limit $p = 1$, we have $EMA(i, m) = t(i)$. In the opposite limit of $p = 0$, we see that only the most distant point within the window counts

$$\begin{aligned} EMA(i, m) &= EMA(i-1, m-1) = EMA(i-2, m-2) = \ldots \\ &= EMA(i-m+1, 1) = t(i-m+1). \end{aligned}$$

As usual, application knowledge or empirical validation determines the value of p. The exponential moving average has performed very well for many finance and business-related applications, producing results superior to the moving average.

A moving average summarizes the recent past, but spotting a change in the trend of the data may additionally improve forecasting performances. Characteristics of a trend can be measured by composing features that compare recent measurements to those of the more distant past. Three simple comparative features are

1. $t(i) - MA(i, m)$, the difference between the current value and a moving average;

2. $MA(i, m) - MA(i - k, m)$, the difference between two moving averages, usually of the same window size, and
3. $t(i)/MA(i, m)$, the ratio between the current value and a moving average, which may be preferred for some applications.

In general, the main components of the summarizing features for a time series are

1. current values,
2. smoothed values using moving averages, and
3. derived trends, differences, and ratios.

The immediate extension of a univariate time series is to a multivariate one. Instead of having a single measured value at time i, $t(i)$, multiple measurements, $a(i)$ and $b(i)$, etc. are taken at the same time. Typically each series is transformed into features, and the values of the features at each distinct time i are merged into a single sample. The resulting data in standard tabular form are shown here in the case of 2 features assuming a window size of 3.

Time	a	b
1	5	117
2	8	113
3	4	116
4	9	118
5	10	119
6	12	120

Sample	$a(n-2)$	$a(n-1)$	$a(n)$	$b(n-2)$	$b(n-1)$	$b(n)$
1	5	8	4	117	113	116
2	8	4	9	113	116	118
3	4	9	10	116	118	119
4	9	10	12	118	119	120

In real-world applications, one can often find hybrid problems having both time series and features that are not dependent on time. In these cases, standard procedures for time-series transformation and summarization of attributes are performed. High dimensions of the resulting data can be reduced at the next phase of a data-ming process: data reduction.

Some data sets have features that depends on several times that are attributes of described entities. One important class of data belonging to this type is survival data. Survival data are data concerning how long it takes for a particular event to happen, e.g. how long does a patient lives, how long before a machine breaks done, how long before a customer arrives at a bank to seek service from a bank teller.

There are two main characteristic of survival data. The first one ic called censoring. It can happen that a certain event has not yet happen by the end of the study period. So, for some patients in a medical trial we might know that the patient was still alive after 5 years, but we do not know when the patient died. This sort of observation is called a censored observation.

The second characteristic is that the input values are time-dependent. If a smoker quits smoking or starts with a new drug during the study, it is important to know what data to include and how to represent these changes in time. Data-mining analysis for these types of problems concentrates on the survivor function or the hazard function. The survivor function is the probability of the survival time being greater than the time t. The hazard function indicates how likely a failure of an industrial component is at time t, given that a failure has not occurred before time t.

6. Outlier Analysis

Often in large data sets, there exist samples that do not comply with the general behavior of the data model. Such samples, which are significantly different or inconsistent with the remaining set of data, are called outliers. Outliers can be caused by measurement errors or they may be the result of inherent data variability. For example, if, in the database, the number of children for one person is 25, this may be due to an entry mistake or it could be correct and represent real variability of the given attribute. The problem of defining outliers is nontrivial, especially for high dimensional samples.

Many data-mining algorithms try to minimize the influence of outliers on the final model, or to eliminate them in the preprocessing phase. This approach may work well if one is interested in looking for general trends hidden in the data. Outliers are non-representative enough to influence the outcome.

On the other hand, some data-mining application are focused on outlier detection. It is then essential to retain the outliers. Typical applications include detecting fraudulent credit transactions, epidemic spreading of diseases, and the prediction of collapse of the stock market.

6.1. Statistical Outlier Detection

First let us consider one-dimensional samples. The simplest approach to outlier detection is to use statistics. Assuming that the distribution of values is known, it is necessary to find basic statistical parameters such as mean value, μ and variance, σ. Based on these values and the expected number of outliers, it is possible to establish the threshold value, θ, as a function of the variance. Samples out of the threshold value are candidates for outliers. The main problem with this approach is that the data distribution is often unknown in real-world applications.

To illustrate this method, suppose we have the following data set representing the feature Age with 20 different sample:

$\{3, 56, 23, 39, 156, 52, 41, 22, 9, 28, 139, 31, 55, 20, -67, 37, 11, 55, 45, 37\}$

We find that $\mu = 39.6$ and $\sigma = 45.89$ If we select the threshold value for normal distribution of data as

$$\theta = \mu \pm 2\sigma$$

then, all data that are out of the range $[-52.2, 131.4]$ will be potential outliers. Additional knowledge of the characteristics of the feature (Age cannot be negative) eliminates one sample point, and reduces the range to $[0, 124.2]$. [2] So there are three values that are outliers based on our criteria: 156, 139 and -67. With absolute certainty we know that the negative sign is due to typographical error. Such error very likely also introduce extra digits in the first two values.

6.2. Distance-based Outlier Detection

Distance-based outlier detection is a second method that eliminates some of the shortcomings of the statistics method. This method is applicable to multidimensional samples while statistical method analyze only a single dimension, or several dimensions, but separately.

Distance-based outliers are those samples which do not have enough sample points as its neighbors. More precisely, given a data set $\mathbf{S}$, a sample $\mathbf{s}^{(i)}$ is an outlier if at least a fraction p of the samples lies at a distance greater than d away from it. Parameters p and d are either given in advance using knowledge about the data, or their values may be adjusted during preprocessing by trial-and-error.

To illustrate this approach, we consider a set $\mathbf{S}$ of $m = 7$ two-dimensional sample points:

$$\mathbf{S} = \{(2,4),(3,2),(1,1),(4,3),(1,6),(5,3),(4,2)\}$$

and choose $p = 4/(m-1) = 4/6$ and $d = 3$.

We first compute the Euclidean distances among these sample points, $d(\mathbf{s}^{(i)}, \mathbf{s}^{(j)}) = \|\mathbf{s}^{(i)} - \mathbf{s}^{(j)}\|_2$, as shown in the table.

Sample	$\mathbf{s}^{(1)}$	$\mathbf{s}^{(2)}$	$\mathbf{s}^{(3)}$	$\mathbf{s}^{(4)}$	$\mathbf{s}^{(5)}$	$\mathbf{s}^{(6)}$	$\mathbf{s}^{(7)}$
$\mathbf{s}^{(1)}$	0.000	2.236	3.162	2.236	2.236	3.162	2.828
$\mathbf{s}^{(2)}$	2.236	0.000	2.236	1.414	4.472	2.236	1.000
$\mathbf{s}^{(3)}$	3.162	2.236	0.000	3.606	5.000	4.472	3.162
$\mathbf{s}^{(4)}$	2.236	1.414	3.606	0.000	4.242	1.000	1.000
$\mathbf{s}^{(5)}$	2.236	4.472	5.000	4.242	0.000	5.000	5.000
$\mathbf{s}^{(6)}$	3.162	2.236	4.472	1.000	5.000	0.000	1.414
$\mathbf{s}^{(7)}$	2.828	1.000	3.162	1.000	5.000	1.414	0.000

Using the table, we can compute for each sample point the fraction of points lying at a distance greater than $d = 3$ away from it. The results are shown in the following table.

Sample	p
$\mathbf{s}^{(1)}$	$2/6$
$\mathbf{s}^{(2)}$	$1/6$
$\mathbf{s}^{(3)}$	$5/6$
$\mathbf{s}^{(4)}$	$2/6$
$\mathbf{s}^{(5)}$	$5/6$
$\mathbf{s}^{(6)}$	$3/6$
$\mathbf{s}^{(7)}$	$2/6$

For the chosen threshold value of $p = 4/6$, we see that $\mathbf{s}^{(3)}$ and $\mathbf{s}^{(5)}$ are outliers.

In two-dimension, one can easily obtain the same result visually by graphing the points. However in higher dimensions, graphical methods are practically impossible and analytical approaches are necessary.

6.3. Deviation-based Techniques

Deviation-based techniques are the third class of outlier-detection methods. These methods define the basic characteristics of the sample set, and all samples that deviate from these characteristics are outliers.

An example is the sequential-exception technique which is based on the use of a dissimilarity function. For a given set of m samples, a possible dissimilarity function is the total variance of the sample set. The task is to find the smallest subset of samples whose removal results in the greatest reduction of the dissimilarity function for the residual set.

Due to the combinational explosion of different selection of the set of potential outliers (the so called exception set), the problem is NP-hard. If we settle for a less-than-optimal answer, the algorithm's complexity can be reduced to linear using a sequential approach. Using the greedy method, and the algorithm reduces the size sequentially, sample by sample (or subset by subset) by selecting at each step the one that causes the greatest decrease in the total variance.

References

[1] M. Kantardzic, *Data Mining - Concepts, Models, Methods, and Algorithms*, IEEE Press, Wiley-Interscience, 2003, ISBN 0-471-22852-4. This is our adopted textbook, from which this set of lecture notes are derived primarily.

[2] First notice that the numbers for the mean, variance, and threshold values are slightly off. Second, using the full data set including the one with negative age, the computed range was $[-52.17, 131.4]$. However when we eliminate the sample point with negative age, we must recompute all statistical parameters. The new values are $\mu' = 45.01$, $\sigma' = 39.47$, and the upper threshold value $\theta' = 124.2$. (Notice the big changes in these values compared with the old ones. The new range is now given by $[0, 124.2]$. The conclusion is still the same, namely 156, 139 and -67 are outliers. 29

[3] Michael W Berry and Murray Browne, *Lecture Notes in Data Mining*, 2006. ISBN-13 978-981-256-802-1, ISBN-10 981-256-802-6.

[4] Daniel T. Larose, *Data Mining Methods and Models*, Wiley-IEEE, 2006, ISBN 0471756474.

[5] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, Elsevier 2006, ISBN 1558609016.

[6] D. Hand, H, Mannila, and P. Smith, *Principles of Data Mining*, MIT Press, Cambridge, MA, 2001.

[7] C.Westphal, and T. Baxton, *Data Mining SOlutions: Methods and Tools for Solving Real-World Problems*, John-Wiley, New York, 1998.

[8] When one or more of its features have nominal values, then a vector space cannot strictly be defined.

6