Knowledge Discovery and Data Mining

Spring 2021

Chap 3. Data Preprocessing

Jiawei Han, Micheline Kamber and Jian Pei, Data Mining: Concepts and Techniques, 3rd ed., The Morgan Kaufmann Series in Data Management Systems Morgan Kaufmann Publishers, July 2011. ISBN 978-0123814791

Tuong Le, PhD

Outline

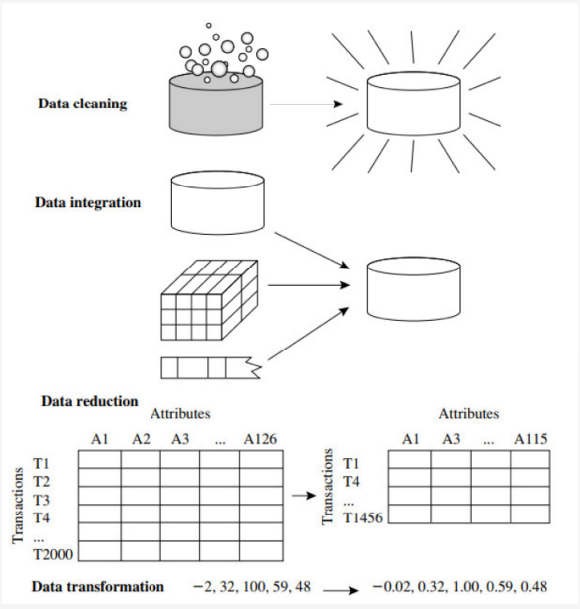
1. Data Preprocessing: An Overview
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation and Data Discretization
6. Summary

Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
  - Accuracy: correct or wrong, accurate or not
  - Completeness: not recorded, unavailable, …
  - Consistency: some modified but some not, dangling, …
  - Timeliness: timely update?
  - Believability: how trustable the data are correct?
  - Interpretability: how easily the data can be understood?

Major Tasks in Data Preprocessing

- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data reduction
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- Data transformation and data discretization
  - Normalization
  - Concept hierarchy generation



Outline

1. Data Preprocessing: An Overview
2. **Data Cleaning**
3. Data Integration
4. Data Reduction
5. Data Transformation and Data Discretization
6. Summary

Data Cleaning

- Data in the Real World is dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., Occupation="" (missing data)
  - Noisy: containing noise, errors, or outliers
    - e.g., Salary="−10" (an error)
  - Inconsistent: containing discrepancies in codes or names, e.g.,
    - Age="42", Birthday="03/07/2010"
    - Was rating "1, 2, 3", now rating "A, B, C"
    - Discrepancy between duplicate records
  - Intentional (e.g., disguised missing data)
    - Jan. 1 as everyone's birthday?

Incomplete (Missing) Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - Equipment malfunction
  - Inconsistent with other recorded data and thus deleted
  - Data not entered due to misunderstanding
  - Certain data may not be considered important at the time of entry
  - Not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification) - not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: time consuming.
- Fill in it automatically with
  - A global constant : e.g., "unknown", a new class?!
  - The attribute mean: mean or median.
  - The attribute mean for all samples belonging to the same class: smarter
  - The most probable value: using regression, inference-based tools using a Bayesian formalism or decision tree induction.

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - Faulty data collection instruments
  - Data entry problems
  - Data transmission problems
  - Technology limitation
  - Inconsistency in naming convention
- Other data problems which require data cleaning
  - Duplicate records
  - Incomplete data
  - Inconsistent data

How to Handle Noisy Data?

- Binning
  - First sort data and partition into (equal-frequency) bins
  - Then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - Smooth by fitting the data into regression functions
- Outlier analysis
  - Detect and remove outliers by using clustering

Binning – Example

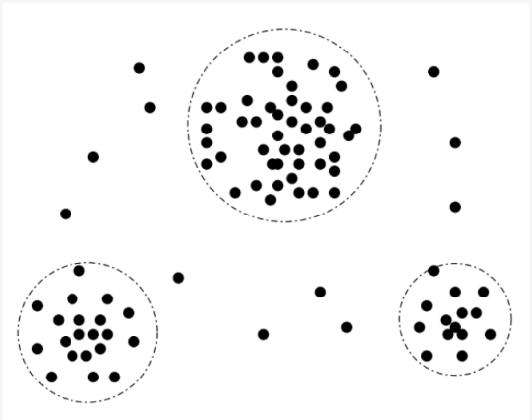
- Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34.

Partition into (equal-frequency) bins:	Smoothing by bin means:	Smoothing by bin boundaries:
Bin 1: 4, 8, 15	Bin 1: 9, 9, 9	Bin 1: 4, 4, 15
Bin 2: 21, 21, 24	Bin 2: 22, 22, 22	Bin 2: 21, 21, 24
Bin 3: 25, 28, 34	Bin 3: 29, 29, 29	Bin 3: 25, 25, 34

- In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.
- Similarly, smoothing by bin medians can be employed, in which each bin value is replaced by the bin median
- In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

Outlier analysis: Example

- A 2-D customer data plot with respect to customer locations in a city, showing three data clusters. Outliers may be detected as values that fall outside of the cluster sets



Outline

1. Data Preprocessing: An Overview
2. Data Cleaning
3. **Data Integration**
4. Data Reduction
5. Data Transformation and Data Discretization
6. Summary

Data Integration

- Data integration:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id ≡ B.cust-#
  - Integrate metadata from different sources
- Entity identification problem:
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - Object identification: The same attribute or object may have different names in different databases
  - Derivable data: One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by **correlation analysis** and **covariance analysis**
  - Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data.
  - Using the χ2 (chi-square) test for nominal data and the correlation coefficient and covariance for numeric attributes.
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Correlation Analysis (Nominal Data)

- For nominal data, a correlation relationship between two attributes, A and B, can be discovered by a χ^2 (chi-square) test. Suppose *A* has *c* distinct values, namely $a_1, a_2 \dots a_c$ and *B* has *r* distinct values, namely $b_1, b_2 \dots b_r$.
- The data tuples described by A and B can be shown as **a contingency table**, with the c values of A making up the columns and the r values of B making up the rows.
- Let (A_i, B_j) denote the joint event that attribute A takes on value a_i and attribute B takes on value b_j, that is, where $(A = a_i, B = b_j)$. Each and every possible (A_i, B_j) joint event has its own cell (or slot) in the table. The χ^2 value is computed as:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

where o_{ij} is the observed frequency (i.e., actual count) of the joint event (A_i, B_j) and e_{ij} is the expected frequency of (A_i, B_j), which can be computed as

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n},$$

where n is the number of data tuples, count$(A = a_i)$ is the number of tuples having value a_i for A, and count$(B = b_j)$ is the number of tuples having value b_j for B.

Correlation Analysis (Nominal Data)

- The χ^2 statistic tests the hypothesis that A and B are independent, that is, there is no correlation between them. The test is based on **a significance level**, with (r-1)×(c-1) degrees of freedom.
- The significance level can be taken from the table of upper percentage points of the χ^2 distribution.
- If the hypothesis can be rejected, then we say that A and B are statistically correlated.
- Otherwise, A and B are independent.

	Level of Significance α								
df	0.200	0.100	0.075	0.050	0.025	0.010	0.005	0.001	0.0005
1	1.642	2.706	3.170	3.841	5.024	6.635	7.879	10.828	12.116
2	3.219	4.605	5.181	5.991	7.378	9.210	10.597	13.816	15.202
3	4.642	6.251	6.905	7.815	9.348	11.345	12.838	16.266	17.731
4	5.989	7.779	8.496	9.488	11.143	13.277	14.860	18.467	19.998
5	7.289	9.236	10.008	11.070	12.833	15.086	16.750	20.516	22.106
6	8.558	10.645	11.466	12.592	14.449	16.812	18.548	22.458	24.104
7	9.803	12.017	12.883	14.067	16.013	18.475	20.278	24.322	26.019
8	11.030	13.362	14.270	15.507	17.535	20.090	21.955	26.125	27.869
9	12.242	14.684	15.631	16.919	19.023	21.666	23.589	27.878	29.667
10	13.442	15.987	16.971	18.307	20.483	23.209	25.188	29.589	31.421
11	14.631	17.275	18.294	19.675	21.920	24.725	26.757	31.265	33.138
12	15.812	18.549	19.602	21.026	23.337	26.217	28.300	32.910	34.822
13	16.985	19.812	20.897	22.362	24.736	27.688	29.820	34.529	36.479
14	18.151	21.064	22.180	23.685	26.119	29.141	31.319	36.124	38.111
15	19.311	22.307	23.452	24.996	27.488	30.578	32.801	37.698	39.720
16	20.465	23.542	24.716	26.296	28.845	32.000	34.267	39.253	41.309
17	21.615	24.769	25.970	27.587	30.191	33.409	35.719	40.791	42.881
18	22.760	25.989	27.218	28.869	31.526	34.805	37.157	42.314	44.435
19	23.900	27.204	28.458	30.144	32.852	36.191	38.582	43.821	45.974
20	25.038	28.412	29.692	31.410	34.170	37.566	39.997	45.315	47.501
21	26.171	29.615	30.920	32.671	35.479	38.932	41.401	46.798	49.013
22	27.301	30.813	32.142	33.924	36.781	40.289	42.796	48.269	50.512
23	28.429	32.007	33.360	35.172	38.076	41.639	44.182	49.729	52.002
24	29.553	33.196	34.572	36.415	39.364	42.980	45.559	51.180	53.480
25	30.675	34.382	35.780	37.653	40.646	44.314	46.928	52.620	54.950
26	31.795	35.563	36.984	38.885	41.923	45.642	48.290	54.053	56.409
27	32.912	36.741	38.184	40.113	43.195	46.963	49.645	55.477	57.860
28	34.027	37.916	39.380	41.337	44.461	48.278	50.994	56.894	59.302
29	35.139	39.087	40.573	42.557	45.722	49.588	52.336	58.302	60.738
30	36.250	40.256	41.762	43.773	46.979	50.892	53.672	59.704	62.164
40	47.269	51.805	53.501	55.759	59.342	63.691	66.766	73.403	76.097
50	58.164	63.167	65.030	67.505	71.420	76.154	79.490	86.662	89.564
60	68.972	74.397	76.411	79.082	83.298	88.380	91.952	99.609	102.698
70	79.715	85.527	87.680	90.531	95.023	100.425	104.215	112.319	115.582
80	90.405	96.578	98.861	101.880	106.629	112.329	116.321	124.842	128.267
90	101.054	107.565	109.969	113.145	118.136	124.117	128.300	137.211	140.789
100	111.667	118.498	121.017	124.342	129.561	135.807	140.170	149.452	153.174

Correlation Analysis (Nominal Data): Example 1

- Suppose that a group of 1500 people was surveyed. The gender of each person was noted. Each person was polled as to whether his or her preferred type of reading material was fiction or nonfiction.
- Thus, we have two attributes, gender and preferred reading. The observed frequency (or count) of each possible joint event is summarized as the following table.

	Male	Female	Total
Like science fiction	250	200	450
Not like science fiction	50	1000	1050
Total	300	1200	1500

Correlation Analysis (Nominal Data): Example 1

- The expected frequencies are calculated based on the data distribution for both attributes. For example:

$$e_{11} = \frac{count(male) \times count(fiction)}{n} = \frac{300 \times 450}{1500} = 90,$$

	Male	Female	Total
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Total	300	1200	1500

- χ^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$
$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93.$$

- For this 2 × 2 table, the degrees of freedom are (2-1)x(2-1)=1. For 1 degree of freedom, the χ2 value needed to reject the hypothesis at the 0.001 significance level is 10.828.
- It shows that gender and preferred reading are (strongly) correlated for the given group of people.

Correlation coefficient (Numeric Data)

- Correlation coefficient:

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B},$$

where n is the number of tuples, a_i and b_i are the respective values of A and B in tuple i, $\bar{A}$ and $\bar{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\sum(a_i b_i)$ is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation. A higher value may indicate that A (or B) may be removed as a redundancy.
- $r_{A,B} = 0$: independent, and there is no correlation between them.
- $r_{A,B} < 0$: negatively correlated. This means that each attribute discourages the other.
- Note that correlation does not imply causality. That is, if A and B are correlated, this does not necessarily imply that A causes B or that B causes A.

Covariance (Numeric Data)

- Consider two numeric attributes *A* and *B*, and a set of *n* observations $\{(a_1, b_1),\ldots,(a_n, b_n)\}$. The mean values of A and B, respectively, are also known as the expected values on A and B, that is

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} \qquad E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}.$$

- The covariance between A and B is defined as

$$Cov(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^{n} (a_i - \bar{A})(b_i - \bar{B})}{n}.$$

- It can be simplified in computation as $Cov(A, B) = E(A \cdot B) - \bar{A}\bar{B}$
- Correlation coefficient: $r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$,

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean or expected values of A and B, σ_A and σ_B are the respective standard deviation of A and B.

Covariance (Numeric Data)

- Positive covariance: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- Negative covariance: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- Independence: $Cov_{A,B} = 0$.
- However, some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

Covariance: Example 1

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- **Question:** If the stocks are affected by the same industry trends, will their prices rise or fall together?
- **Solution:**
  - E(A) = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - E(B) = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6
  - $Cov_{A,B}$ = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4
- Thus, A and B rise together since $Cov_{A,B}$ > 0.

Covariance and Coefficient: Example

- The data set of two TV channels at a time (20:00 - 21:00 every Thursday) for a month is: X = (50772, 73756, 74251, 77601) and Y = (102492, 100406, 97762, 98191). Calculate the Covariance and Coefficient between them.
- **Solution:**
  - E(X) = (50772 + 73756 + 74251 + 77601)/ 4 = 69095
  - E(Y) = (102492 + 100406 + 97762 + 98191) / 4 = 99712.75
  - $\sigma_X = \sqrt{\frac{1}{n}\sum_1^N (x_i - E(X))^2}$ = 10681.69
  - $\sigma_Y = \sqrt{\frac{1}{n}\sum_1^N (y_i - E(Y))^2}$ = 1892.48
  - $Cov_{X,Y}$ = [(50772 * 102492 + 73756 * 100406+ 74251 * 97762 + 77601 * 98191)/4] – 69095 * 99712.75 = -17673758
  - $r_{X,Y}$= $Cov_{X,Y}$ / ($\sigma_X \times \sigma_Y$) = -17673758 / (10681.69 * 1892.48) = -0.87

Outline

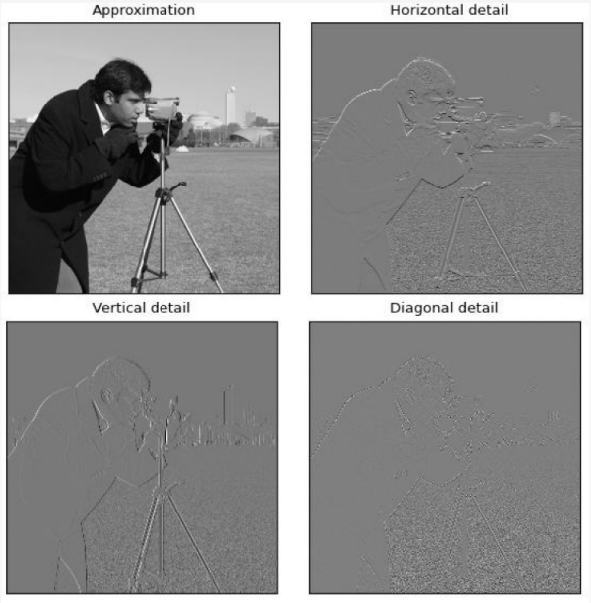
1. Data Preprocessing: An Overview
2. Data Cleaning
3. Data Integration
4. **Data Reduction**
5. Data Transformation and Data Discretization
6. Summary

Data Reduction Strategies

- Data reduction: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
  - Dimensionality reduction, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
  - Numerosity reduction techniques: replace the original data volume by alternative, smaller forms of data representation.
    - Parametric methods: Regression and Log-Linear Models
    - Nonparametric methods: Histograms, clustering, sampling, data cube aggregation
  - Data compression: transformations are applied so as to obtain a reduced or "compressed" representation of the original data.

Dimensionality Reduction: Wavelet Transformation

- Decomposes a signal into different frequency subbands
  - Applicable to n-dimensional signals
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allow natural clusters to become more distinguishable
- Used for image compression



Dimensionality Reduction: Wavelet Transformation

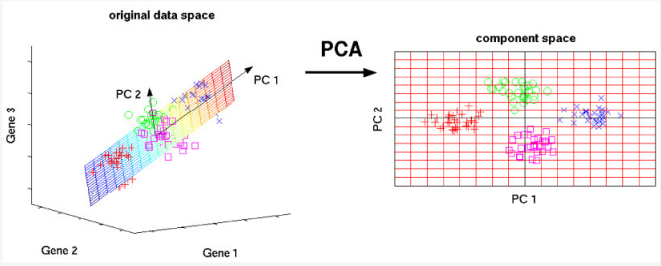
- The Discrete Wavelet Transform (DWT) is a linear signal processing technique that, when applied to a data vector X, transforms it to a numerically different vector, X′, of wavelet coefficients.
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
  - For example, all wavelet coefficients larger than **an user-specified threshold** can be retained. All other coefficients are set to 0. The resulting data representation is therefore very sparse: very fast if performed in wavelet space.
- DWT also works to remove noise without smoothing out the main features of the data, making it effective for data cleaning as well.
- Given a set of coefficients, an approximation of the original data can be constructed by applying the inverse of the DWT used.
- Popular wavelet transforms include the Haar-2, Daubechies-4, and Daubechies-6.

Dimensionality Reduction: Wavelet Transformation

- ❑ Providing good results on sparse or skewed data and on data with ordered attributes
- ❑ Effective removal of outliers
  - Insensitive to noise, insensitive to input order
- ❑ Efficient
  - Complexity O(N)
- ❑ Wavelet transforms have many real-world applications
  - The compression of fingerprint images,
  - Computer vision,
  - Analysis of time-series data, and
  - Data cleaning
- ❑ Recommend package: PyWavelets (https://pywavelets.readthedocs.io/en/latest/)

Dimensionality Reduction: Principal Component Analysis (PCA)

- ❑ Suppose that the data to be reduced consist of tuples or data vectors described by n attributes or dimensions. PCA searches for k n-dimensional orthogonal vectors that can best be used to represent the data, where $k \leq n$.



- ❑ The original data are projected onto a much smaller space, resulting in dimensionality reduction.
- ❑ PCA can be applied to ordered and unordered attributes, and can handle sparse data and skewed data.
- ❑ In comparison with wavelet transforms, PCA tends to be better at handling sparse data, whereas wavelet transforms are more suitable for data of high dimensionality.
- ❑ Recommend package: Sklearn (https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html)

Dimensionality Reduction: Attribute Subset Selection

- Redundant attributes
  - Duplicate much or all of the information contained in one or more other attributes
  - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
  - Contain no information that is useful for the data mining task at hand
  - E.g., students' ID is often irrelevant to the task of predicting students' GPA
- This can result in discovered patterns of poor quality. In addition, the added volume of irrelevant or redundant attributes can slow down the mining process.
- Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes (or dimensions)

Dimensionality Reduction: Attribute Subset Selection with Sklearn

- The classes in the ***sklearn.feature_selection*** module can be used for feature selection/dimensionality reduction on sample sets, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets.
- **Removing features with low variance**: VarianceThreshold is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.
- **Univariate feature selection**: This approach works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator.

Dimensionality Reduction: Removing features with low variance

As an example, suppose that we have a dataset with boolean features, and we want to remove all features that are either one or zero (on or off) in more than 80% of the samples. Boolean features are Bernoulli random variables, and the variance of such variables is given by $Var[X] = p(1-p)$. Therefore, we can select using the threshold 0.8 * (1 - 0.8).

```
>>> from sklearn.feature_selection import VarianceThreshold
>>> X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
>>> sel = VarianceThreshold(threshold=(.8 * (1 - .8)))
>>> sel.fit_transform(X)
array([[0, 1],
       [1, 0],
       [0, 0],
       [1, 1],
       [1, 0],
       [1, 1]])
```


Dimensionality Reduction: Univariate feature selection

- Scikit-learn exposes feature selection routines as objects that implement the transform methods:
  - SelectKBest removes all but the highest scoring features
  - SelectPercentile removes all but a user-specified highest scoring percentage of features
  - GenericUnivariateSelect allows to perform univariate feature selection with a configurable strategy. This allows to select the best univariate selection strategy with hyper-parameter search estimator.
- For instance, we can perform a χ2 test to the samples to retrieve only the two best features as follows:

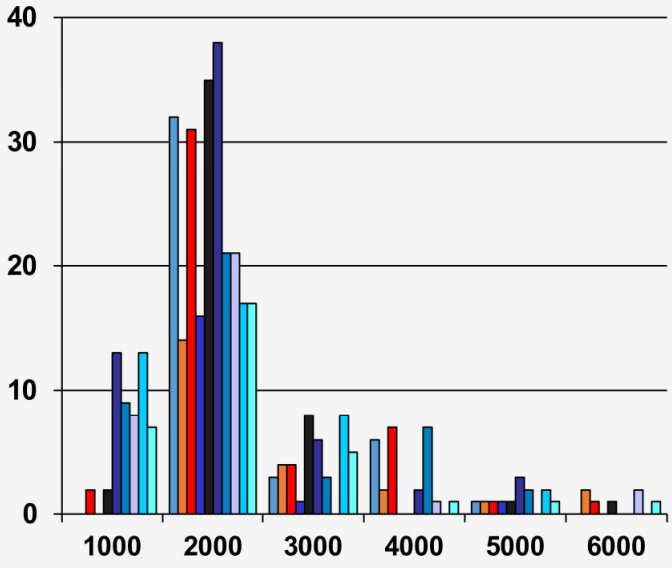
```
>>> from sklearn.datasets import load_iris
>>> from sklearn.feature_selection import SelectKBest
>>> from sklearn.feature_selection import chi2
>>> X, y = load_iris(return_X_y=True)
>>> X.shape
(150, 4)
>>> X_new = SelectKBest(chi2, k=2).fit_transform(X, y)
>>> X_new.shape
(150, 2)
```

Numerosity Reduction

- Reduce data volume by choosing alternative, smaller forms of data representation: **Parametric methods and Non-parametric methods**.
- Parametric methods (e.g., regression)
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the actual data (Outliers may also be stored).
  - Ex.: Regression and Log-linear models
- Non-parametric methods
  - Nonparametric methods for storing reduced representations of the data.
  - Major families: **histograms**, **clustering**, **sampling**.

Histogram Analysis

- Histograms use binning to approximate data distributions and are a popular form of data reduction.
- Divide data into buckets and store average (sum) for each bucket
- Histograms are highly effective at approximating both sparse and dense data, as well as highly skewed and uniform data
- Singleton buckets are useful for storing high-frequency outliers

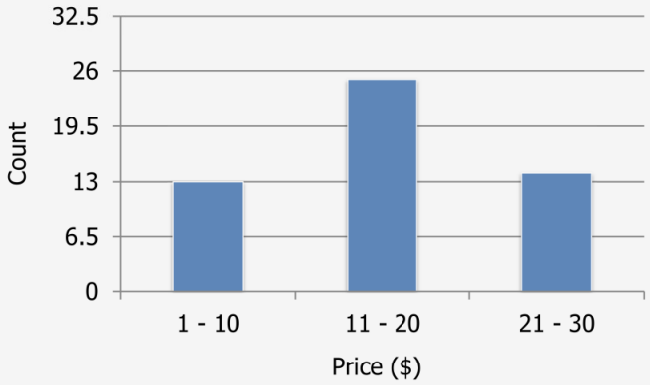


Histogram Analysis – Example 1

- The following data are a list of AllElectronics prices for commonly sold items (rounded to the nearest dollar): 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30. Draw a histogram for the data for each bucket represents a different $10 range for price.

- **Solution**

Price ($)	Count
1 - 10	13
11 - 20	25
21 - 30	14



Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only

- In data reduction, the cluster representations of the data are used to replace the actual data

- It is much more effective for data that can be organized into distinct clusters than for smeared data

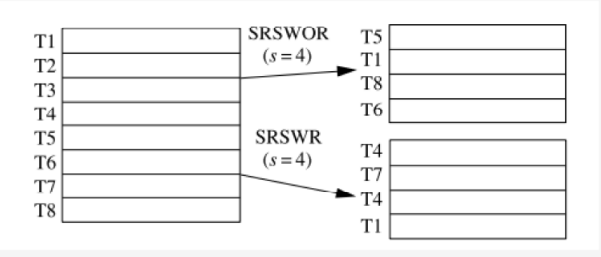
- There are many choices of clustering definitions and clustering algorithms which are further described in Chapters 10.

Sampling

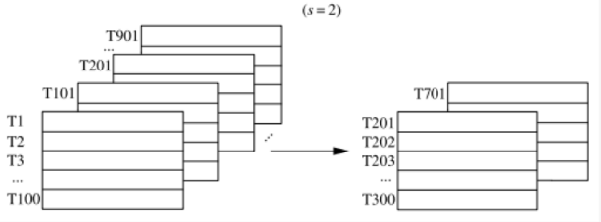
- Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random data sample (or subset).
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Sampling is a natural choice for the progressive refinement of a reduced data set.
- Recommend package: Imblearn.Under_sampling (https://imbalanced-learn.readthedocs.io/en/stable/index.html)

Types of Sampling

Simple random sample without replacement and Simple random sample with replacement



Cluster sample



Stratified sample

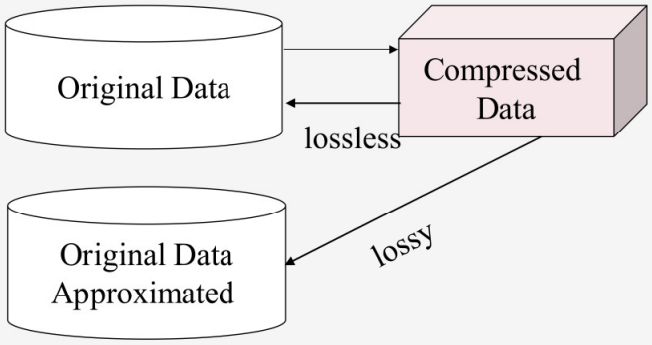
(according to *age*)

T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Data Compression

- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless, but only limited manipulation is possible without expansion
- Audio/video compression
  - Typically lossy compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
  - Typically short and vary slowly with time
- Dimensionality and numerosity reduction may also be considered as forms of data compression



Outline

1. Data Preprocessing: An Overview
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation and Data Discretization
6. Summary

Data Transformation

- Data transformation is the process of changing the format, structure, or values of data.
- Methods
  - Smoothing: Remove noise from data. Techniques include binning, regression, and clustering.
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization, data cube construction
  - **Normalization**: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - **Discretization**
  - **Concept hierarchy generation for nominal data**

Normalization

- Min-max normalization: to $[new\_\min_A, new\_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- Z-score normalization (μ: mean, σ: standard deviation):

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

- Normalization by decimal scaling

$$v' = \frac{v}{10^j}$$

Where j is the smallest integer such that Max(|v'|) < 1

Normalization – Example

- Min - Max Normalization: Suppose that the minimum and maximum values for the attribute income are $12,000 and $98,000. By min-max normalization, what is a value of $73,600 for income in range [0.0, 1.0]?

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A = \frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1 - 0) + 0 = 0.716$$

- Z-scored Normalization: Suppose that the mean and standard deviation of the values for the attribute income are $54,000 and $16,000. With z-score normalization, what is a value of $73,600 for income?

$$v' = \frac{v - \mu_A}{\sigma_A} = \frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

- Normalization by decimal scaling: Suppose that the recorded values of A range from −986 to 917. The maximum absolute value of A is 986. Computing the normalization by decimal scaling with j = 3.

$$v' = \frac{v}{10^j} = \frac{-986}{10^3} = -0.986 \qquad v' = \frac{v}{10^j} = \frac{917}{10^3} = 0.917$$

Data Discretization

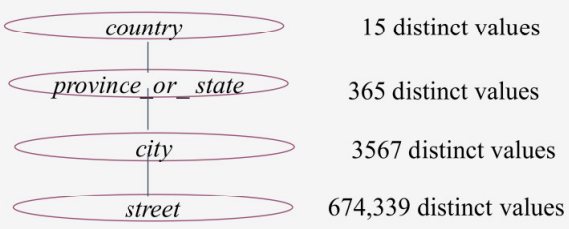
- Data discretization is forms of data reduction. The raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior).
- Typical methods: All the methods can be applied recursively
  - **Binning:** Top-down split, unsupervised
  - **Histogram analysis:** Top-down split, unsupervised
  - **Clustering analysis** (unsupervised, top-down split or bottom-up merge): partitioning the values of a numeric attribute, A, into clusters or groups. Clustering takes the distribution of A into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results.
  - **Decision-tree analysis** (supervised, top-down split)
  - **Correlation (e.g., χ^2) analysis** (unsupervised, bottom-up merge)

Concept Hierarchy Generation for Nominal Data

- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
  - street < city < state < country
- Specification of a hierarchy for a set of values by explicit data grouping
  - {Urbana, Champaign, Chicago} < Illinois
- Specification of only a partial set of attributes
  - E.g., only street < city, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
  - E.g., for a set of attributes: {street, city, state, country}

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
- Suppose a user selects a set of location-oriented attributes—street, country, province or state, and city—from the AllElectronics database, but does not specify the hierarchical ordering among the attributes. A concept hierarchy for location can be generated automatically.
  - First, sort the attributes in ascending order based on the number of distinct values in each attribute. This results in the following : country (15), province or state (365), city (3567), and street (674,339).
  - Second, generate the hierarchy from the top down according to the sorted order, with the first attribute at the top level and the last attribute at the bottom level.
  - Finally, the user can examine the generated hierarchy, and when necessary, modify it to reflect desired semantic relationships among the attributes.



Outline

1. Data Preprocessing: An Overview
2. Data Cleaning
3. Data Integration
4. Data Reduction
5. Data Transformation and Data Discretization
6. **Summary**

Summary

- **Data quality** is defined in terms of accuracy, completeness, consistency, timeliness, believability, and interpretabilty.
- **Data cleaning** routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.
- **Data integration** combines data from multiple sources to form a coherent data store.
- **Data reduction** techniques obtain a reduced representation of the data while minimizing the loss of information content.
- **Data transformation** routines convert the data into appropriate forms for mining.
- **Data discretization** transforms numeric data by mapping values to interval or concept labels.