Data preprocessing

Why preprocessing ?

1. Real world data are generally
    - Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - Noisy: containing errors or outliers
    - Inconsistent: containing discrepancies in codes or names
2. Tasks in data preprocessing
    - Data cleaning: fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
    - Data integration: using multiple databases, data cubes, or files.
    - Data transformation: normalization and aggregation.
    - Data reduction: reducing the volume but producing the same or similar analytical results.
    - Data discretization: part of data reduction, replacing numerical attributes with nominal ones.

Data cleaning

1. Fill in missing values (attribute or class value):
    - Ignore the tuple: usually done when class label is missing.
    - Use the attribute mean (or majority nominal value) to fill in the missing value.
    - Use the attribute mean (or majority nominal value) for all samples belonging to the same class.
    - Predict the missing value by using a learning algorithm: consider the attribute with the missing value as a dependent (class) variable and run a learning algorithm (usually Bayes or decision tree) to predict the missing value.
2. Identify outliers and smooth out noisy data:
    - Binning
        - Sort the attribute values and partition them into bins (see "Unsupervised discretization" below);
        - Then smooth by bin means, bin median, or bin boundaries.
    - Clustering: group values in clusters and then detect and remove outliers (automatic or manual)
    - Regression: smooth by fitting the data into regression functions.
3. Correct inconsistent data: use domain knowledge or expert decision.

Data transformation

1. Normalization:
   - Scaling attribute values to fall within a specified range.
     - Example: to transform `V in [min, max]` to `V' in [0,1]`, apply `V'=(V-Min)/(Max-Min)`
   - Scaling by using mean and standard deviation (useful when min and max are unknown or when there are outliers): `V'=(V-Mean)/StDev`
2. Aggregation: moving up in the concept hierarchy on numeric attributes.
3. Generalization: moving up in the concept hierarchy on nominal attributes.
4. Attribute construction: replacing or adding new attributes inferred by existing attributes.

Data reduction

1. Reducing the number of attributes
   - Data cube aggregation: applying roll-up, slice or dice operations.
   - Removing irrelevant attributes: attribute selection (filtering and wrapper methods), searching the attribute space (see Lecture 5: Attribute-oriented analysis).
   - Principle component analysis (numeric attributes only): searching for a lower dimensional space that can best represent the data..
2. Reducing the number of attribute values
   - Binning (histograms): reducing the number of attributes by grouping them into intervals (bins).
   - Clustering: grouping values in clusters.
   - Aggregation or generalization
3. Reducing the number of tuples
   - Sampling

Discretization and generating concept hierarchies

1. Unsupervised discretization - class variable is not used.
   - Equal-interval (equiwidth) binning: split the whole range of numbers in intervals with equal size.
   - Equal-frequency (equidepth) binning: use intervals containing equal number of values.
2. Supervised discretization - uses the values of the class variable.
   - Using class boundaries. Three steps:
     - Sort values.
     - Place breakpoints between values belonging to different classes.
     - If too many intervals, merge intervals with equal or similar class distributions.
   - Entropy (information)-based discretization. Example:
     - Information in a class distribution:
       - Denote a set of five values occurring in tuples belonging to two classes (+ and -) as `[+,+,+,-,-]`

        - That is, the first 3 belong to "+" tuples and the last 2 - to "-" tuples
        - Then, `Info([+,+,+,-,-]) = -(3/5)*log(3/5)-(2/5)*log(2/5)` (logs are base 2)
        - 3/5 and 2/5 are relative frequencies (probabilities)
        - Ignoring the order of the values, we can use the following notation: `[3,2]` meaning 3 values from one class and 2 - from the other.
        - `Then, Info([3,2]) = -(3/5)*log(3/5)-(2/5)*log(2/5)`
    - Information in a split `(2/5 and 3/5 are weight coefficients)`:
        - `Info([+,+],[+,-,-]) = (2/5)*Info([+,+]) + (3/5)*Info([+,-,-])`
        - `Or, Info([2,0],[1,2]) = (2/5)*Info([2,0]) + (3/5)*Info([1,2])`
    - Method:
        - Sort the values;
        - Calculate information in all possible splits;
        - Choose the split that minimizes information;
        - Do not include breakpoints between values belonging to the same class (this will increase information);
        - Apply the same to the resulting intervals until some stopping criterion is satisfied.
3. Generating concept hierarchies: recursively applying partitioning or discretization methods.