**Question: what is Concept Hierarchy? How Concept Hierarchy is generated for Numerical and categorical data?**

Concept Hierarchy reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior).

**Concept hierarchy generation for numeric data is as follows:**

* **Binning (see sections before)**
* **Histogram analysis (see sections before)**
* **Clustering analysis (see sections before)**
* **Entropy-based discretization**
* **Segmentation by natural partitioning**
* **Binning**
  + In binning, first sort data and partition into (equi-depth) bins then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
* **Histogram analysis**
  + Histogram is a popular data reduction technique
  + Divide data into buckets and store average (sum) for each bucket
  + Can be constructed optimally in one dimension using dynamic programming
  + Related to quantization problems.
* **Clustering analysis**
  + Partition data set into clusters, and one can store cluster representation only
  + Can be very effective if data is clustered but not if data is “smeared”
  + Can have hierarchical clustering and be stored in multi-dimensional index tree structures
* **Entropy-based discretization**
  + Given a set of samples S, if S is partitioned into two intervals S1 and S2 using boundary T, the entropy after partitioning is enter image description here

– S1 & S2 correspond to samples in S satisfying conditions A<v &amp;="" a="">=v

* + The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization.
  + The process is recursively applied to partitions obtained until some stopping criterion is met, e.g., Ent (S)- E(T,S)>δ
  + Experiments show that it may reduce data size and improve classification accuracy
* **Segmentation by natural partitioning**
  + 3-4-5 rule can be used to segment numeric data into relatively uniform, “natural” intervals.
  + If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width intervals
  + If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 intervals
  + If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 intervals

**Concept hierarchy generation for categorical data is as follows:**

* **Specification of a set of attributes, but not of their partial ordering**
  + Auto generate the attribute ordering based upon observation that attribute defining a high level concept has a smaller # of distinct values than an attribute defining a lower level concept
  + Example : country (15), state\_or\_province (365), city (3567), street (674,339)
* **Specification of only a partial set of attributes**
  + Try and parse database schema to determine complete hierarchy

**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.