Lecture notes for January 31, 2005

Data Transformation

In data transformation, the data are modified into forms more appropriate for mining. Several techniques may be viewed as data transformation techniques:

1. **Smoothing** – these techniques are typically introduced to remove noise from the data. They include binning, clustering, regression.
2. **Generalization** – low level descriptions are typically replaced by higher-level descriptions.
3. **Normalization** – involves rescaling of attributes.
4. **Attribute (feature) construction** – new attributes are constructed and added to the existing ones.

**Normalization:**

Features are typically normalized on by one.

**Min-max normalization:** Performs a linear transformation on the original data values. Suppose that $\min_X$ and $\max_X$ are the minimum and maximum of feature $X$. We would like to map interval $[\min_X, \max_X]$ into a new interval $[\text{new}_\min_X, \text{new}_\max_X]$. Consequently, every value $v$ from the original interval will be mapped into value $\text{new}_v$ using the following formula:

$$\text{new}_v = \frac{v - \min_X}{\max_X - \min_X} \cdot (\text{new}_\max_X - \text{new}_\min_X) + \text{new}_\min_X$$

Min-max normalization preserves the relationships among the original data values. A problem may occur if a value of an unseen data point (that we would like to predict on) is out of $[\min_X, \max_X]$ interval.

**Z-score normalization:** also called zero-mean normalization. The values of attribute $X$ are normalized using the mean and standard deviation of $X$. A new value $\text{new}_v$ is obtained using the following expression:

$$\text{new}_v = \frac{v - \mu_X}{\sigma_X}$$

where $\mu_X$ and $\sigma_X$ are the mean and standard deviation of attribute $X$, respectively. If $\mu_X$ and $\sigma_X$ are not known they can be estimated from the sample (column in $D$ that corresponds to feature $X$). In such a case, we can substitute $\hat{\mu}_X$ for $\mu_X$ and $\hat{\sigma}_X$ for $\sigma_X$ in the expression above.
After zero-mean normalizing each feature will have a mean value of 0. Also, the unit of each value will be the number of (estimated) standard deviations away from the (estimated) mean. Note that z-score normalization may be sensitive to small values of $\sigma_X$.

**Normalization by decimal scaling:** normalizes by moving the decimal point of values of feature $X$. The number of decimal points moved depends on the maximum absolute value of $X$. A modified value $new\_v$ corresponding to $v$ is obtained using

$$new\_v = \frac{v}{10^c}$$

where $c$ is the smallest integer such that $\max(|new\_v|) < 1$.

Example: suppose the range of attribute $X$ is $-500$ to $45$. The maximum absolute value of $X$ is $500$. To normalize by decimal scaling we will divide each value by $1,000$ ($c = 3$). In this case, $-500$ becomes $-0.5$ while $45$ will become $0.045$.

**Attribute Construction:**

New attributes are being constructed from the existing ones. For example, attribute *area* may be constructed from *height* and *width*. Attributes constructed in such a way do not add any new information into the mix. Rather, they use our prior knowledge to help the predictor easily learn the right relationship (this might be very important if the dataset is not large).

**Data Reduction**

After data reduction a new dataset becomes much smaller but preserves the integrity of the original dataset. In other words, a model that learns from the smaller dataset should be of similar or better quality than models learned from the original data. The real advantage is that learning becomes significantly faster.

Some of the techniques that fall into data reduction are:

1. **Dimensionality reduction** - irrelevant, weakly relevant, or redundant features are detected and removed.
2. **Data compression** - encoding is used to reduce the dataset size.
3. **Numerosity reduction** - original dataset is typically sampled.
4. **Concept hierarchy generation** - substitution of a low level concept by a high level concept.

**Dimensionality Reduction**

Datasets may contain thousands of features. Not all of them are equally important for a task at hand and some of the attributes may be quite irrelevant. For example, if we are constructing a dataset for
prediction of a disease based on several features such as body_mass_index, temperature, etc., then feature telephone_number is likely irrelevant.

A straightforward goal of dimensionality reduction is to find the best subset of features with respect to the prediction accuracy of the algorithm. If prediction accuracy is similar to or better than the original accuracy the removed features are said to be irrelevant.

Problem: in a set of \( k \) elements (features) how many subsets are there? There are \( 2^k \) subsets. In the case of 100 features there are \( 2^{100} \) subsets to test. If evaluation of every subset takes 1 second, this would be about \( 4 \times 10^{22} \) years on a single processor machine. Clearly, we have to work with suboptimal algorithms for such problems. Even if one evaluation can be obtained in one clock of a 10GHz machine (subtract 10 from the exponent above) this won’t solve our problem.

One suggested approach: the greedy algorithm

- find a single best feature \( X_i \) and include it in the final set
- find a second feature \( X_j \) such that the accuracy of the predictor using only features \( X_i \) and \( X_j \) is maximized
- proceed until accuracy stops improving (and in each step add one feature)

Data Compression

Compression algorithms are divided into lossless and lossy. In data mining, lossy algorithms are dominant and are frequently used in mining time-series data. Examples of useful transforms are wavelet transform and discrete Fourier transform.

Binning (described earlier) is also one way of data compression. There are many ways of compressing a single feature. Typically, techniques from signal processing are used. Scalar or vector quantization are used. Vector quantization would compress several features at a time.

Numerosity Reduction

Sampling is a typical numerosity reduction technique. There are several ways to construct a sample:

- Simple random sampling without replacement – performed by randomly choosing \( n_1 \) data points such that \( n_1 < n \). Recall, \( n \) is the number of data points in the original dataset \( D \).
- Simple random sampling with replacement – we are selecting \( n_1 < n \) data points, but draw them one at a time (\( n_1 \) times). In such a way, one data point can be drawn multiple times in the same subsample.
- Cluster sample – examples in \( D \) are originally grouped into \( M \) disjoint clusters. Then a simple random sample of \( m < M \) elements can be drawn.
- Stratified sample – \( D \) is originally divided into disjoint parts called strata. Then, a stratified sample of \( D \) is generated by obtaining a simple random sample at each stratum. This helps getting a representative sample especially when the data is skewed (say, many more examples of class 0 then of class 1). Stratified samples can be proportionate and disproportionate.