Knowledge Discovery & Data Mining

— Data Preprocessing —

Data Transformation

Instructor: Yong Zhuang

yong.zhuang@gvsu.edu

Based on the original version by Professor Jiliang Tang

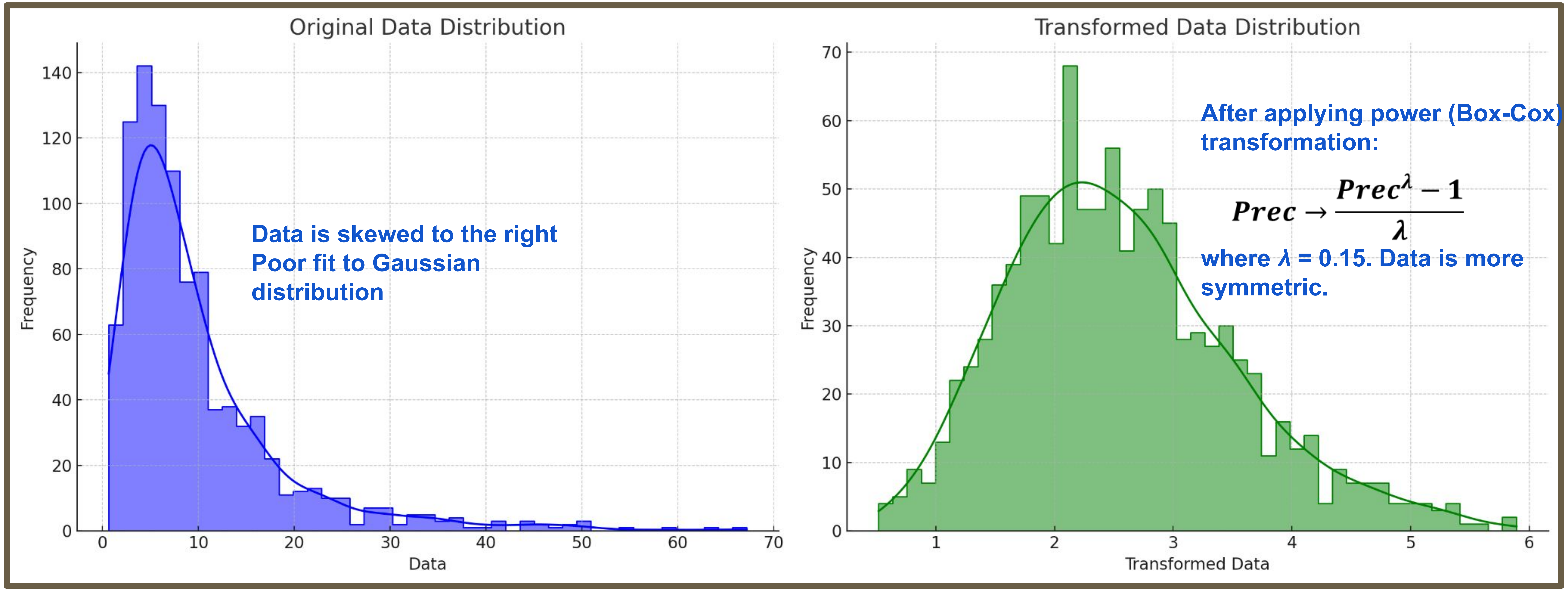
Outline

- Data Transformation
    - Transformation functions
    - Data normalization
        - Min-max
        - Z-score
        - Decimal scaling
    - Data discretization

Data Transformation

- Sometimes, the original attribute values may not be suitable/optimal for the data mining task.
- Attribute transformation maps the entire set of values of a given attribute to **a new set of values** using.
  - Transformation functions
  - Data normalization
  - Data discretization
  - Data compression
  - Sampling

Transformation functions



Data is skewed to the right
Poor fit to Gaussian distribution

After applying power (Box-Cox) transformation:

$$Prec \rightarrow \frac{Prec^{\lambda} - 1}{\lambda}$$

where λ = 0.15. Data is more symmetric.

Data Normalization

In general, expressing an attribute in smaller units will lead to a larger range for that attribute and thus tend to give such an attribute greater effect or “weight.” To help avoid dependence on the choice of measurement units, the data should be **normalized** or **standardized**.

- attempts to give all attributes an equal weight.
- useful for
  - classification algorithms: neural networks
  - distance measurements: nearest-neighbor classification and clustering

Data Normalization: Min-max

Min-max normalization performs a linear transformation on the original data. Suppose that min$_A$ and max$_A$ are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v_i, of A to v'_i in the range [new_min$_A$, new_max$_A$] by computing

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

if a future input case for normalization falls outside of the original data range for A: "**out-of-bounds**" error

Data Normalization: Min-max

Example. Suppose that the minimum and maximum values for the attribute income are \$12,000 and \$98,000, respectively. We would like to map income to the range [0.0, 1.0]. By min-max normalization, a value of \$73,600 for income is transformed to:

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

Data Normalization: Min-max

Example. Suppose that the minimum and maximum values for the attribute income are \$12,000 and \$98,000, respectively. We would like to map income to the range [0.0, 1.0]. By min-max normalization, a value of \$73,600 for income is transformed to:

$$\frac{73,600-12,000}{98,000-12,000}(1.0-0)+0=0.716$$

Data Normalization: z-score

In **z-score** normalization (or **zero-mean** normalization), the values for an attribute, A, are normalized based on the mean (i.e., average) and standard deviation of A. A value, v_i, of A is normalized to v'_i by computing

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$

where σ_A is the standard deviation of attribute A.

Data Normalization: z-score

Example. Suppose that the mean and standard deviation of the values for the attribute income are \$54,000 and \$16,000, respectively. With z-score normalization, a value of \$73,600 for income is transformed to

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$

Data Normalization: z-score

Example. Suppose that the mean and standard deviation of the values for the attribute income are \$54,000 and \$16,000, respectively. With z-score normalization, a value of \$73,600 for income is transformed to

$$\frac{73,600-54,000}{16,000} = 1.225$$

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$

Data Normalization: z-score

A variation of this z-score normalization replaces the standard deviation of σ_A by the mean absolute deviation of A. The mean absolute deviation of A, denoted S_A, is

$$s_A = \frac{1}{n}(|v_1 - \bar{A}| + |v_2 - \bar{A}| + \cdots + |v_n - \bar{A}|)$$

$$\boxed{\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right) - \bar{x}^2}$$

Thus z-score normalization using the mean absolute deviation is

$$v_i' = \frac{v_i - \bar{A}}{s_A}$$

S_A is more robust to outliers than σ_A

Data Normalization: Decimal Scaling

Normalization by decimal scaling normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value, v_i, of A is normalized to v'_i by computing

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

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

Data Normalization: Decimal Scaling

Example. Suppose that the recorded values of A range from −986 to 917, how to normalize with decimal scaling?

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

j is the smallest integer such that ***Max(|v'$_i$|)*** *< 1*

Data Normalization: Decimal Scaling

Example. Suppose that the recorded values of A range from −986 to 917, how to normalize with decimal scaling?

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

*j is the smallest integer such that **Max(|v'$_i$|)** < 1*

Max |v$_i$|* = 986** → ***Min j

*such that **Max(|v'$_i$|)** < 1*

→ ***Max(|v'$_i$|)* =** 0.986 and ***j = 3***

↓

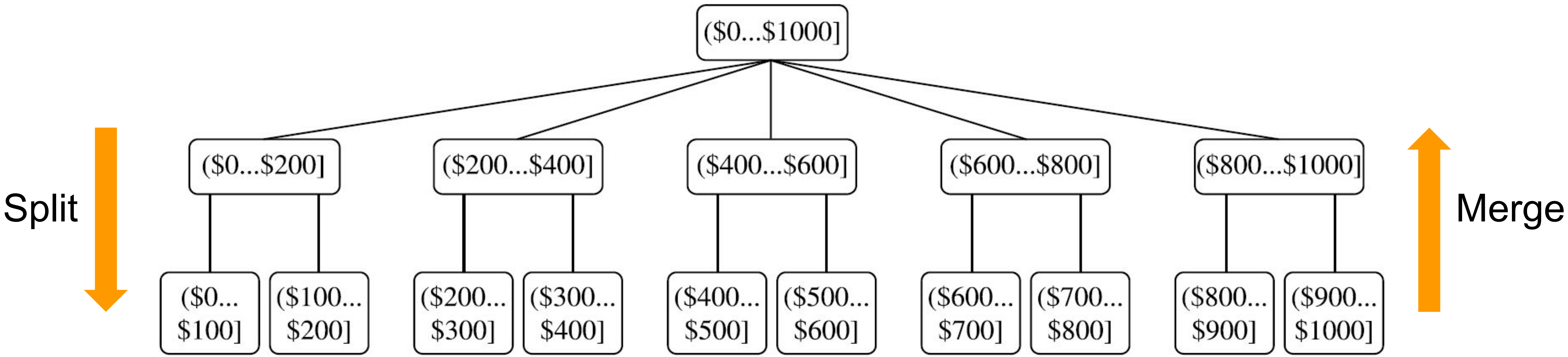
$$v_i' = \frac{v_i}{1000}$$

Data Discretization

Data discretization is a common data transformation technique, where 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). The labels, in turn, can be recursively organized into higher-level concepts, resulting in a concept hierarchy for the numeric attribute.

Data Discretization

- Supervised discretization: use class information, Otherwise, unsupervised
- Split (top-down) vs. merge (bottom-up)
- Discretization can be performed recursively on an attribute
- Prepare for further analysis, e.g., classification



Data Discretization Methods

Typical methods: All the methods can be applied recursively

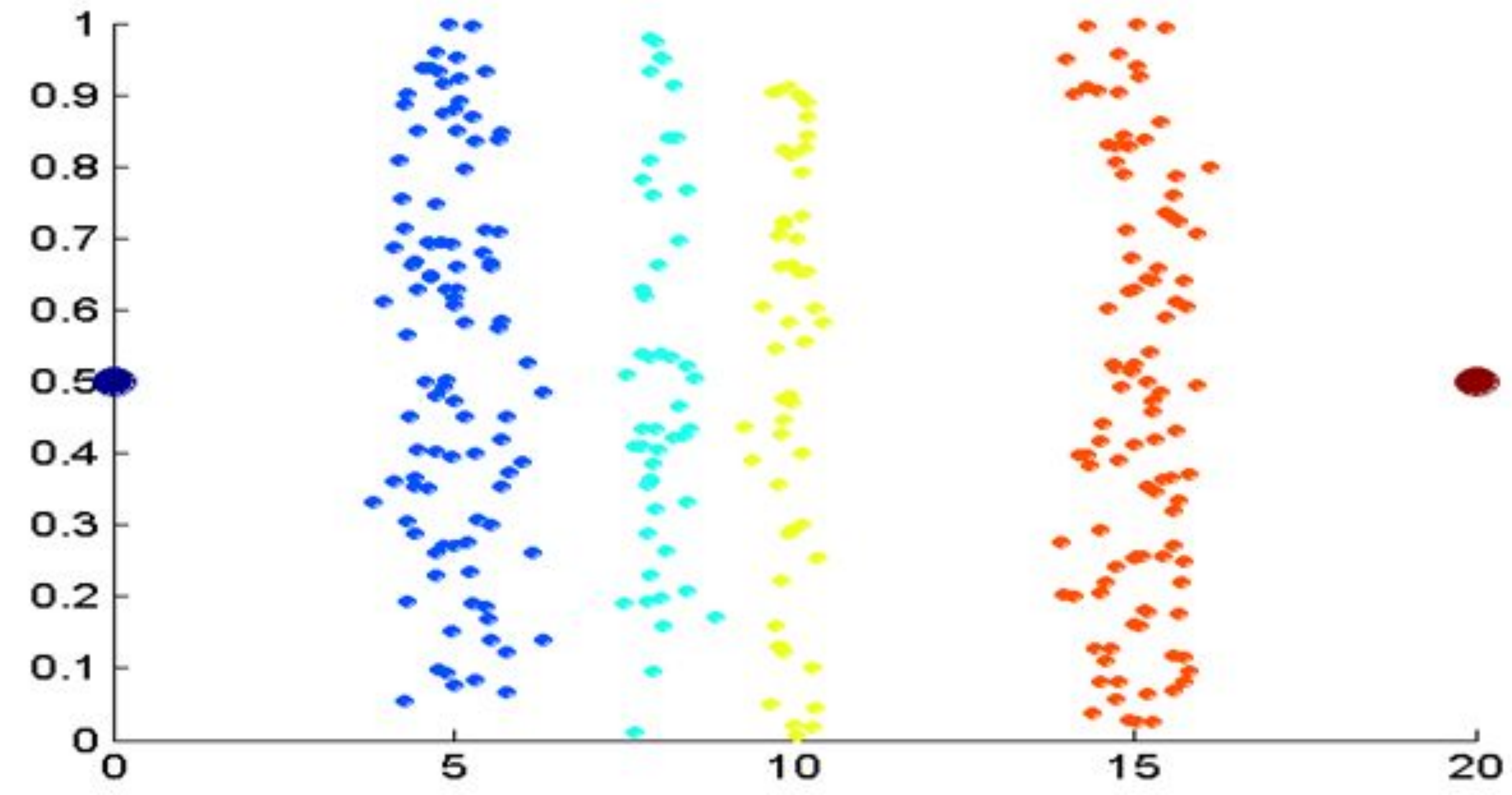
- **Binning**
  - Top-down split, unsupervised
- **Histogram analysis**
  - Top-down split, unsupervised
- **Clustering analysis** (unsupervised, top-down split or bottom-up merge)
- **Decision-tree analysis** (supervised, top-down split)
- **Correlation analysis** (unsupervised, bottom-up merge)

Discretization by binning

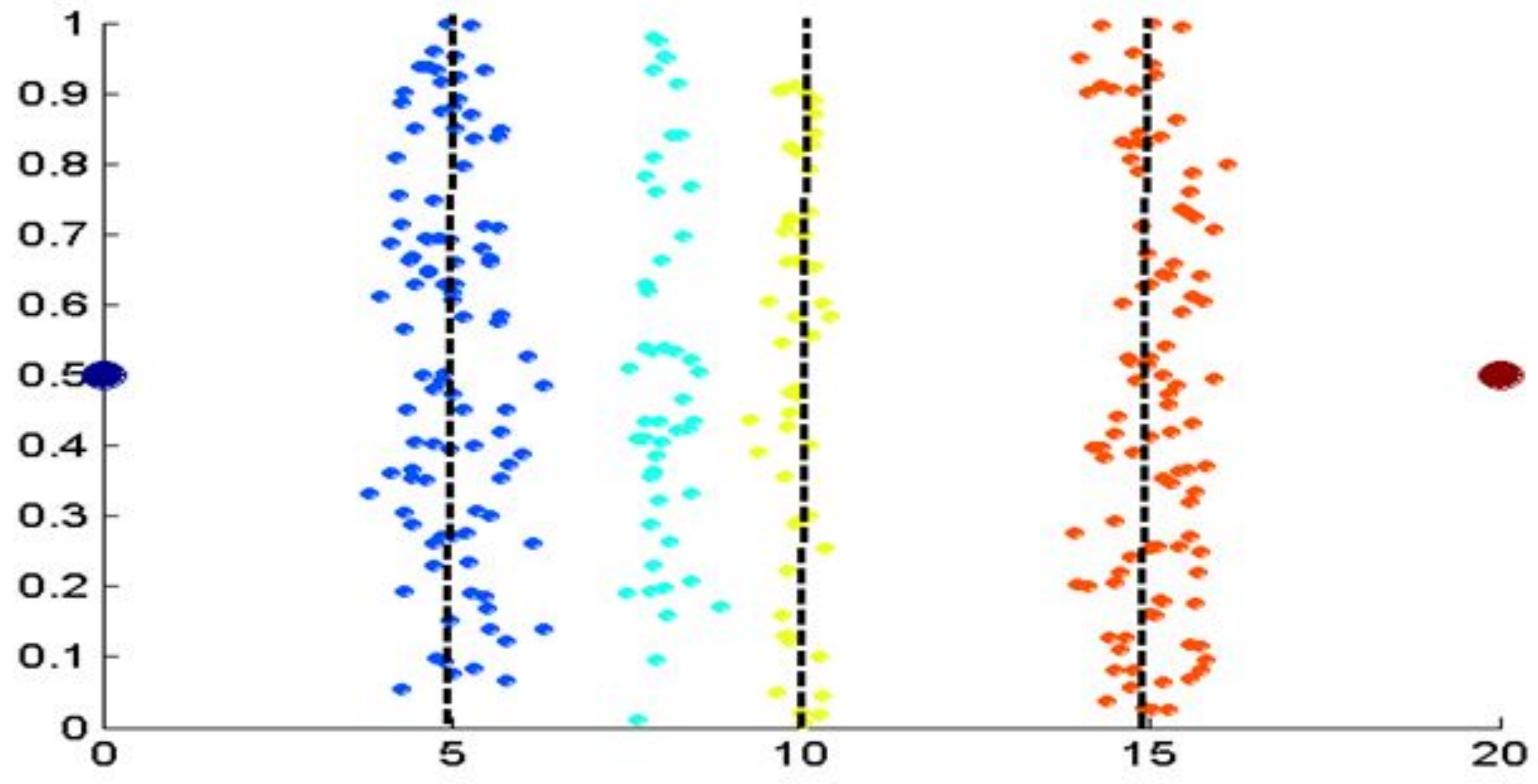
Binning is a **top-down** splitting technique based on a specified number of bins.

- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

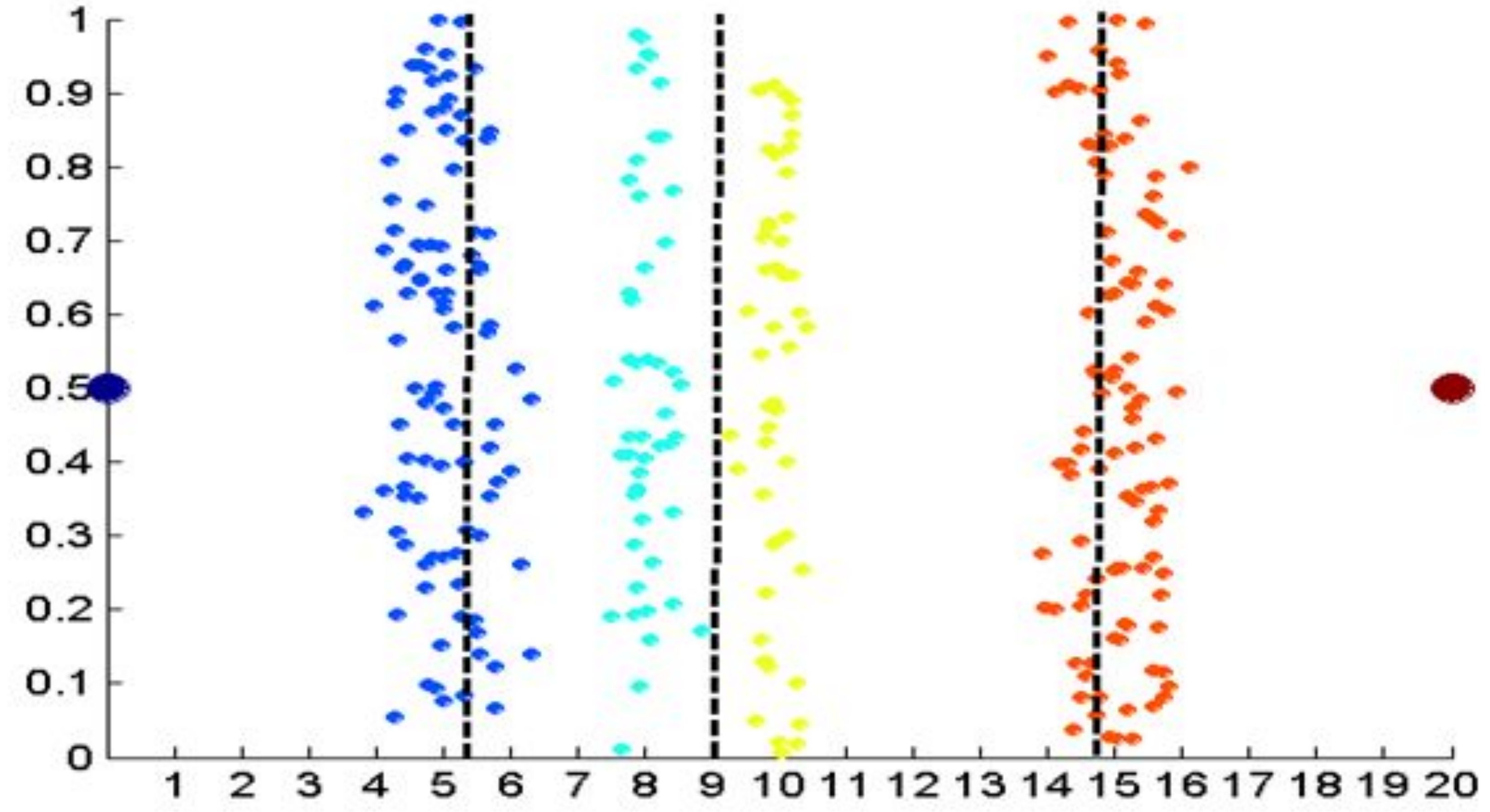
Discretization by binning



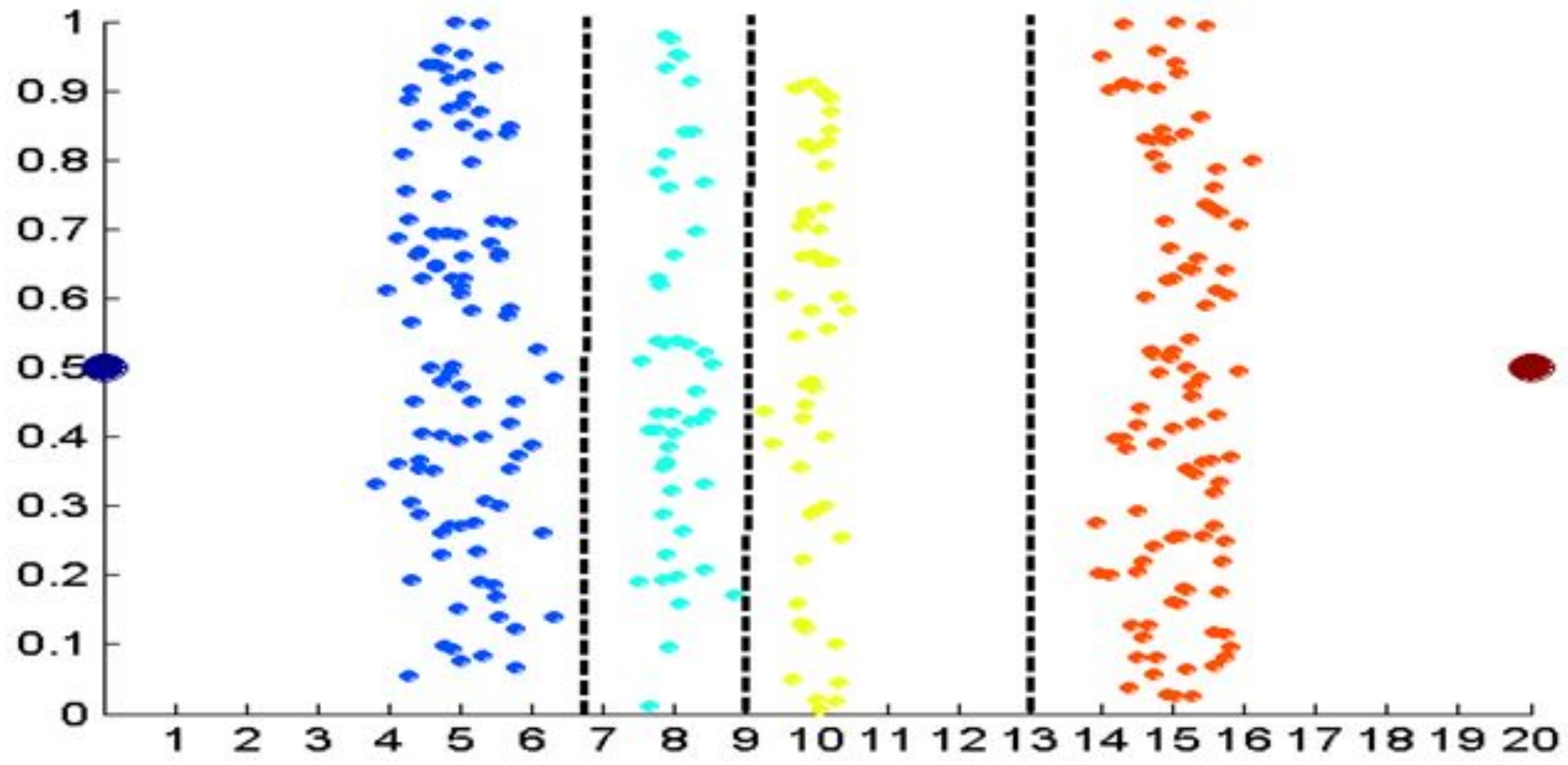
Data



Equal-width



Equal-depth



Clustering

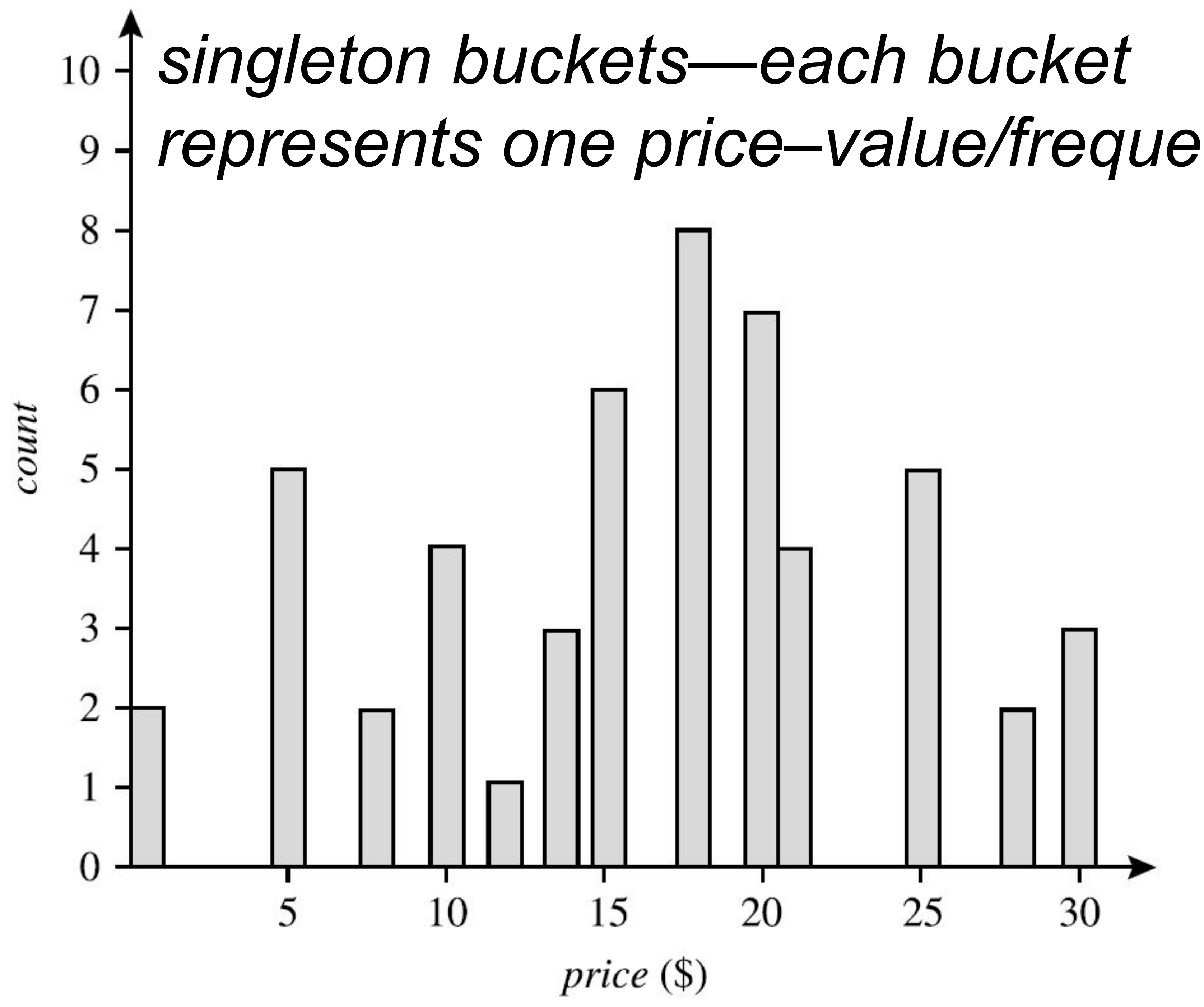
Source: CSE 881: Lecture 4, Professor Jiliang Tang

Discretization by histogram analysis

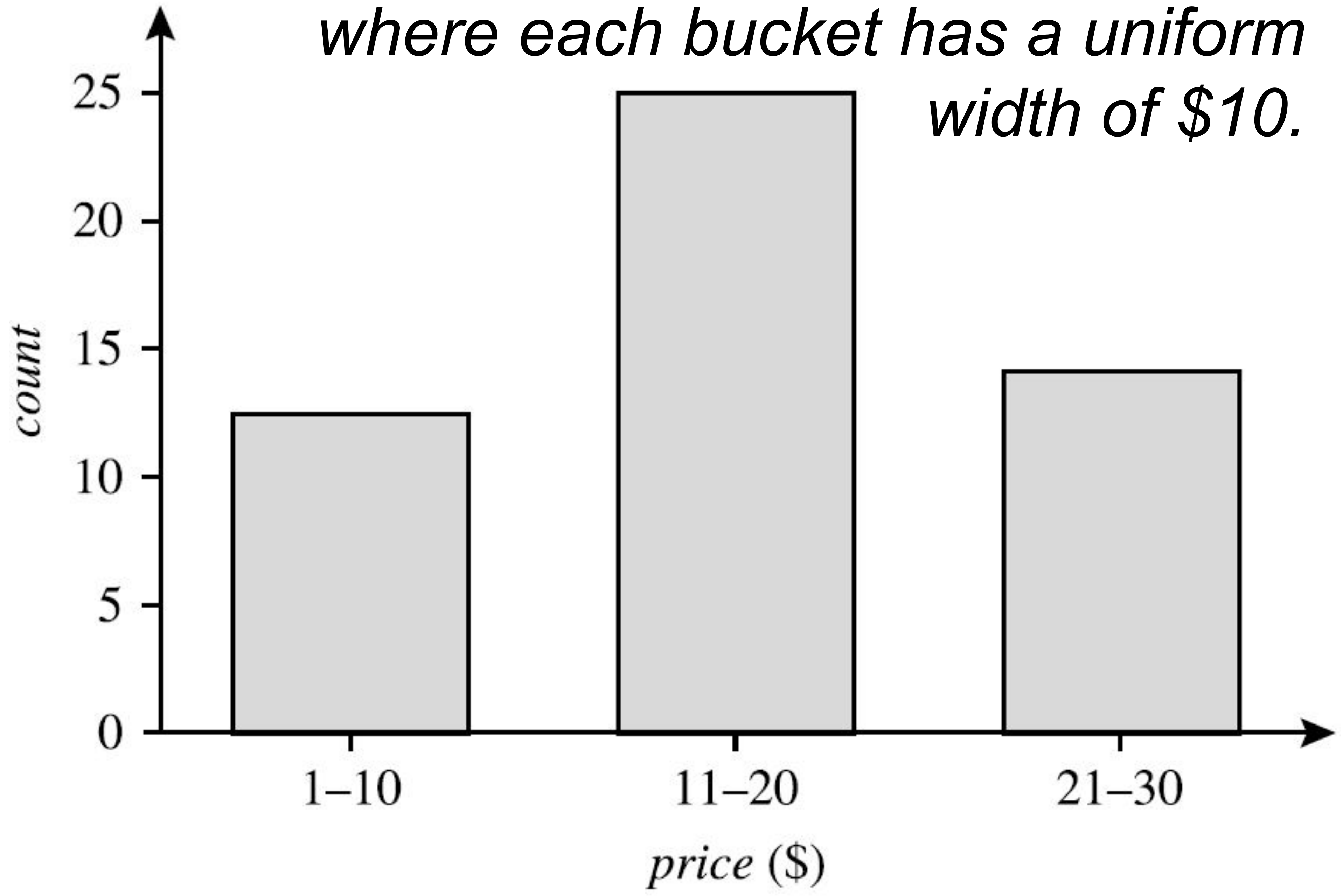
Histogram analysis is an unsupervised discretization technique because it does not use class information. A histogram partitions the values of an attribute, A, into disjoint ranges called buckets or bins. If each bucket represents only a **single attribute-value/frequency** pair, the buckets are called **singleton** buckets. Singleton buckets are useful for storing high-frequency outliers.

Discretization by histogram analysis

Example. The following data are a list of prices for commonly sold items in the company (rounded to the nearest dollar). The numbers have been sorted: 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.



singleton buckets—each bucket represents one price–value/frequency pair



equal-width histogram for price, where each bucket has a uniform width of $10.

Summary

- Data Transformation
    - Transformation functions
    - Data normalization
        - Min-max
        - Z-score
        - decimal scaling
    - Data discretization: Binning, Clustering analysis, Histogram analysis