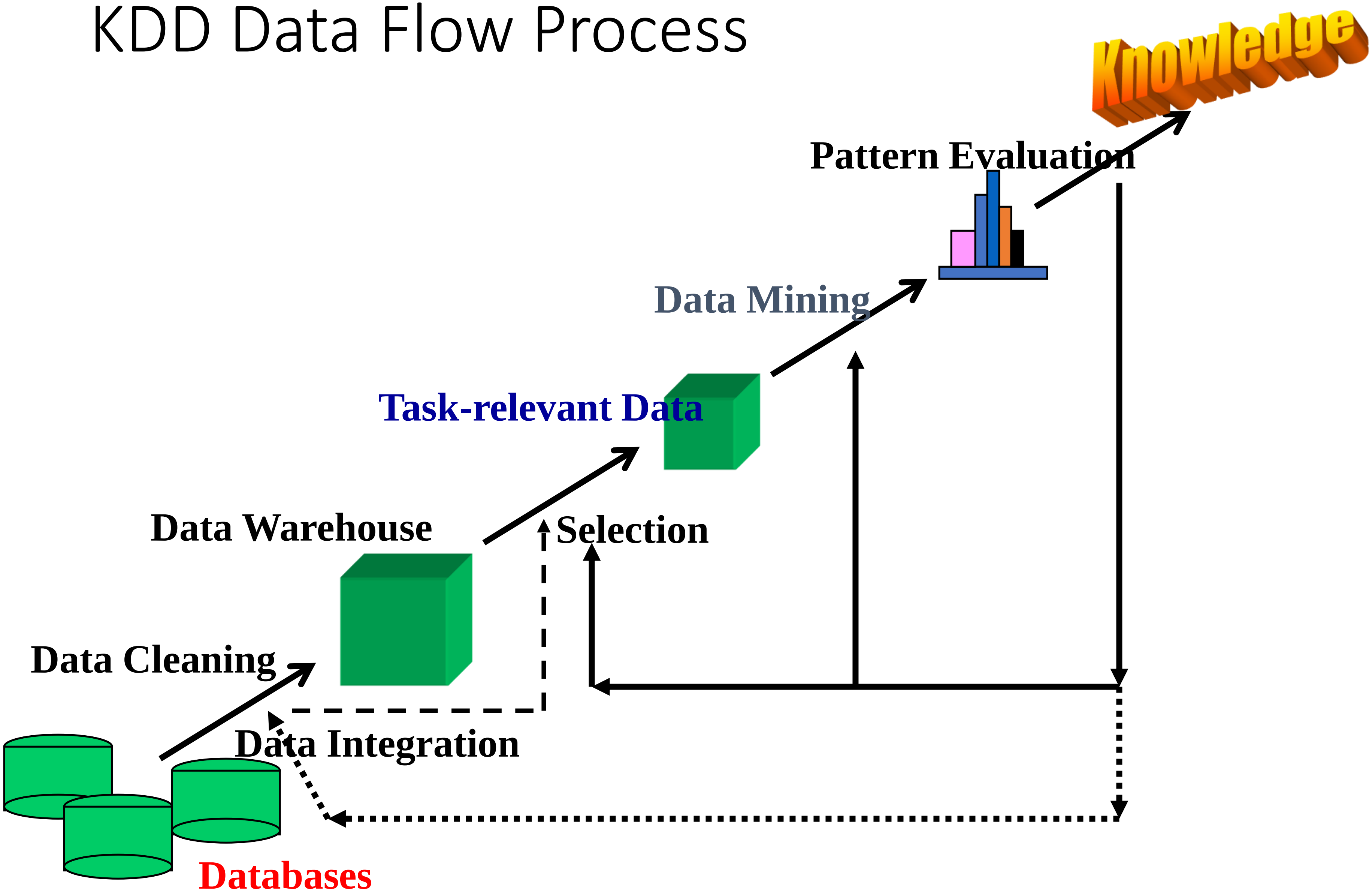
CISC 372 Advanced Data Analytics L5 – Infrastructure & Preprocessing

https://l1nna.com/372

Today

- KDD Process
- Data Attributes
- Data Characteristics
- Data Preprocessing

KDD Data Flow Process



Data Objects

- Data sets are made up of data objects.
- A **data object** represents an entity.
- Examples:
  - sales database: customers, store items, sales
  - medical database: patients, treatments
  - university database: students, professors, courses
- Also called *samples* , *examples, instances, data points, objects, tuples*, and *records*.
- Data objects are described by **attributes**.
- Database rows -> data objects; columns ->attributes.

Attributes

- **Attribute (**or **dimensions, features, variables**): a data field, representing a characteristic or feature of a data object.
  - *E.g., customer _ID, name, address*

- Types:
  - Nominal (aka **categorical**)
  - Binary (aka **binominal**)
  - Ordinal
  - Continuous

Attribute Types

- **Nominal:** categories, states, or “names of things”
  - *Hair_color = {auburn, black, blond, brown, grey, red, white}*
  - marital status, occupation

- **Binary** (aka binominal)
  - Nominal attribute with only 2 states (0 and 1)
  - <u>Symmetric binary</u>: both outcomes equally important
    - e.g., gender
  - <u>Asymmetric binary</u>: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
    - Convention: assign 1 to most important outcome (e.g., HIV positive)

Attribute Types

- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size* = {*small, medium, large*}, grades, army rankings

- **Continuous Attribute**
  - Has real numbers as attribute values
    - E.g., temperature, height, or weight
  - Practically, real values can only be measured and represented using a finite number of digits

Types of Data Sets

- Record
  - Relational records
  - Data matrix, e.g., numerical matrix, crosstabs
- Transaction data
- Document data
- Graph and network
  - World Wide Web
  - Social or information networks
  - Molecular Structures
- Ordered
  - Video data: sequence of images
  - Temporal data: time-series
  - Sequential Data: transaction sequences
  - Genetic sequence data
- Spatial, image and multimedia:
  - Spatial data: maps
  - Image data
  - Video data

Record

Employee ID	First name	Last name	Title	Date of birth	Address ID
1008	Jane	Smith	Sales	2/14/1969	302
1009	Joe	Diner	Clerk	4/16/1968	884
1010	Ed	Smithers	Sales	9/22/1964	992
1011	Tom	Massee	Mgr	4/5/1947	42
1012	Julie	Vahnne	Clerk	6/11/1972	223
1013	John	Smith	Clerk	3/12/1970	302
1014	Donald	Winter	Clerk	5/18/1969	55

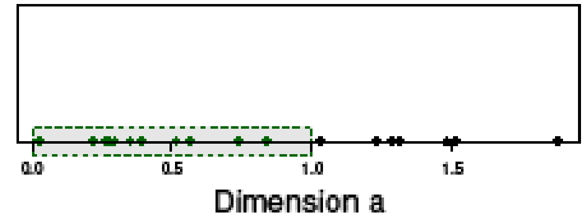
EMPLOYEE table

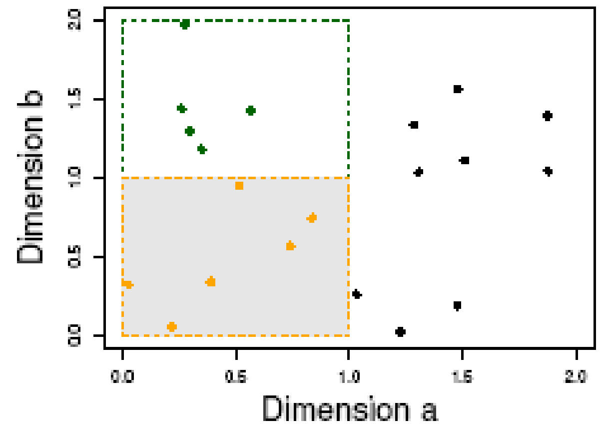
Field

Month	REGION 1	REGION 2	REGION 3	REGION 4	REGION 5	TOTAL
April	13	33	76	2	47	171
May	17	55	209	1	143	425
June	8	63	221	1	127	420
July	13	104	240	6	123	486
August	18	121	274	9	111	533
September	25	160	239	2	88	514
October	9	88	295	2	127	521
November	2	86	292	2	120	502
December	1	128	232	6	155	522
TOTAL	106	838	2078	31	1041	4094

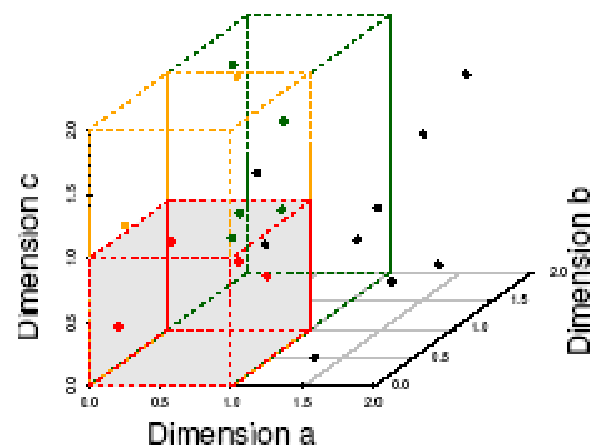
Important Characteristics of Structured Data

- Dimensionality
  - Curse of dimensionality
- Sparsity
  - Only presence counts
- Resolution
  - Patterns depend on the scale
- Distribution
  - Centrality and dispersion





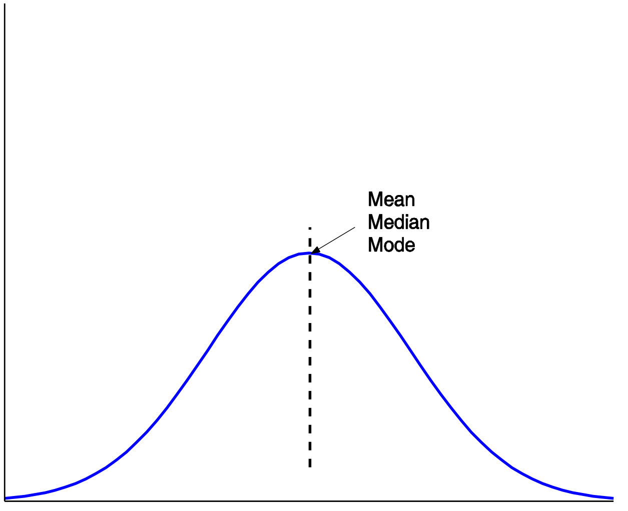
(b) 6 Objects in One Unit Bin

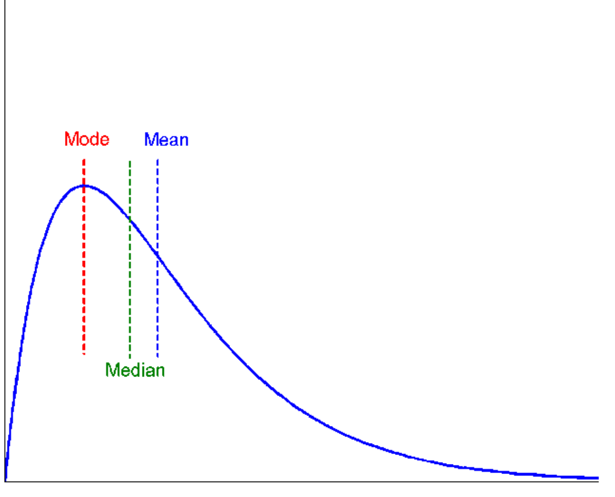


(c) 4 Objects in One Unit Bin

Important Characteristics of Structured Data

- Dimensionality
  - Curse of dimensionality
- Sparsity
  - Only presence counts
- Resolution
  - Patterns depend on the scale
- Distribution
  - Centrality and dispersion





Data Does not Come Prepared (mostly)

- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
    - e.g., do not need both age and birth year
- Data transformation
  - Generalize and/or normalize data
    - e.g. do not need to know specific address of a customer, city or postal code is good enough.

Data Does not Come Prepared (mostly)

- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
    - e.g., do not need both age and birth year
- Data transformation
  - Generalize and/or normalize data
    - e.g. do not need to know specific address of a customer, city or postal code is good enough.

Normalization

- Scaling individual samples to have unit norm.
- Could be L1 norm or L2 norm
  - Very important for vector space model (think about dot-product)

```python
from sklearn import preprocessing
import numpy as np
from numpy.linalg import norm

X_train = np.array([[ 1., -1.,  2.],
                    [ 2.,  0.,  0.],
                    [ 0.,  1., -1.]])
X_scaled = preprocessing.normalize(X_train, norm='l2')

print(X_scaled)
print('norm', norm(X_scaled, axis=1))
print('std ', X_scaled.std(axis=0))
print('mean', X_scaled.mean(axis=0))
```

```
[[ 0.40824829 -0.40824829  0.81649658]
 [ 1.          0.          0.        ]
 [ 0.          0.70710678 -0.70710678]]
norm [1. 1. 1.]
std  [0.41053309 0.46075826 0.62254262]
mean [0.4694161  0.0996195  0.03646327]
```

Standardization

- A common requirement
  - Making the feature a Gaussian **with zero mean** and **unit variance**

```python
from sklearn import preprocessing
import numpy as np
X_train = np.array([[ 1., -1.,  2.],
                    [ 2.,  0.,  0.],
                    [ 0.,  1., -1.]])
X_scaled = preprocessing.scale(X_train)

print(X_scaled)
print('mean', X_scaled.mean(axis=0))
print('std ', X_scaled.std(axis=0))
```

```
[[ 0.         -1.22474487  1.33630621]
 [ 1.22474487  0.         -0.26726124]
 [-1.22474487  1.22474487 -1.06904497]]
mean [0. 0. 0.]
std  [1. 1. 1.]
```

Categorical Encoding

- Converting categorical attribute (aka nominal) into numeric features (or a one-hot encoding vector)

```python
from sklearn import preprocessing
import numpy as np
from numpy.linalg import norm

enc = preprocessing.OrdinalEncoder()
X = [
     ['male', 'from US', 'uses Safari'],
     ['female', 'from Europe', 'uses Firefox']]
enc.fit(X)

enc.transform([
               ['female', 'from US', 'uses Safari']
               ])
```

```
array([[0., 1., 1.]])
```

Continuous Values? Discretization

- Bin-based Discretization

```
from sklearn import preprocessing
import numpy as np
from numpy.linalg import norm

X = np.array([[ -3., 5., 15 ],
              [  0., 6., 14 ],
              [  6., 3., 11 ]])

est = preprocessing.KBinsDiscretizer(
    n_bins=[3, 2, 2], encode='ordinal'
    ).fit(X)

est.transform(X)
```

```
array([[0., 1., 1.],
       [1., 1., 1.],
       [2., 0., 0.]])
```

Continuous Values? Discretization

- Thresholding numerical features => binominal

```
X = [[ 1., -1.,  2.],
     [ 2.,  0.,  0.],
     [ 0.,  1., -1.]]

binarizer = preprocessing.Binarizer(threshold=1.5).fit(X)

binarizer.transform(X)
```

```
array([[0., 0., 1.],
       [1., 0., 0.],
       [0., 0., 0.]])
```