# DATA PREPROCESSING

Chen Chen

January 28, 2021

# OUTLINE

- Data Analysis
- Data Preprocessing

# DATA ANALYSIS

- Continuous Feature
  - Count
  - Mean
  - Variance
  - Standard Variance
  - Max
  - Min

# CODE EXAMPLE

```
import numpy as np

data = np.loadtxt("iris.csv", delimiter=',')

data.shape

c1 = data[:, 0]

np.sum(c1),np.mean(c1), np.std(c1),
np.var(c1), np.max(c1),np.min(c1),
np.argmax(c1),np.argmin(c1),
```

# CODE EXAMPLE CNTD

```
x= data[:,:4]
x=data[:,:-1]
np.mean(x)
np.mean(x, axis=0)
```

# DATA ANALYSIS

- Categorial Feature
  - Count
  - Unique
  - Top, Freq
- Target(Label, Prediction)
  - Classification
  - Regression
  - Ranking

# CODE EXAMPLE

```
data_dia = np.loadtxt("diabetes.csv", delimiter=",", skiprows=1)

data_dia.shape

c1 = data_dia[:,0]

np.median(c1)

np.unique(c1)
```

# OUTLINE

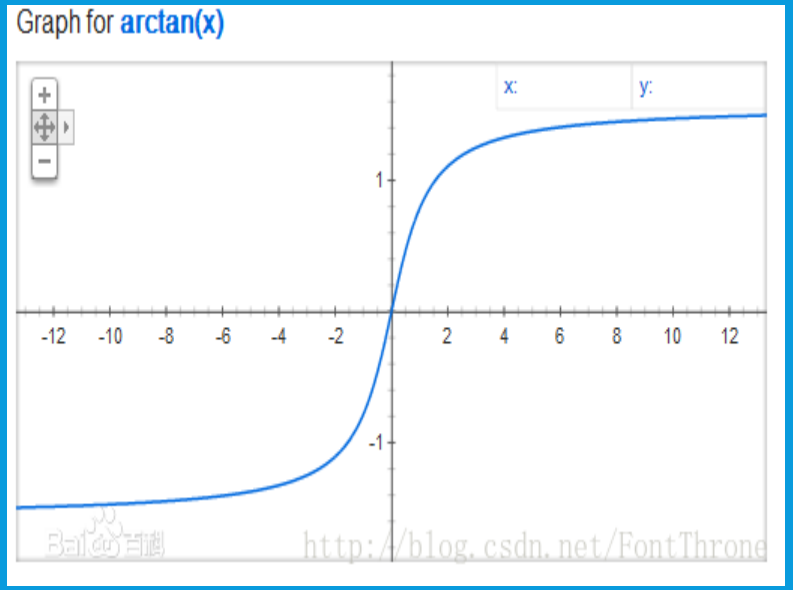
- Data Analysis
- Data Preprocessing

# DATA PREPROCESSING

- Continuous Feature
  - Normalization(归一化)
  - Standardization(标准化)
  - Zero-Centered(零中心化)
- Categorical Feature
  - OneHotEncoding

# DATA PREPROCESSING NORMALIZATION

- Role: map data into [0,1] or [-1,+1]
- Normalization
  - min-max normalization [0,1]
  - mean normalization [-1,+1]
  - logarithmic normalization
    - $Log_{10}(x)/log_{10}(max)$, $log_{10}(x)$
  - arc tangent normalization[-1,+1]
    - 2*atan(x)/pi



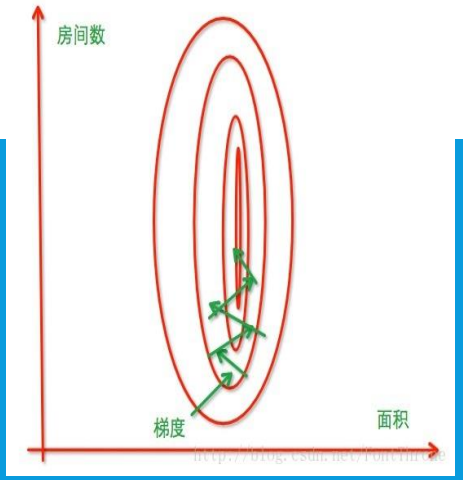
# MAX-MIN NORM

```
data = np.loadtxt("iris.csv", delimiter=',')
x = data[:, :-1]
max_per_column = np.max(x,axis=0)
min_per_column = np.min(x,axis=0)
x_norm = (x-min_per_column)
/(max_per_column - min_per_column)
```

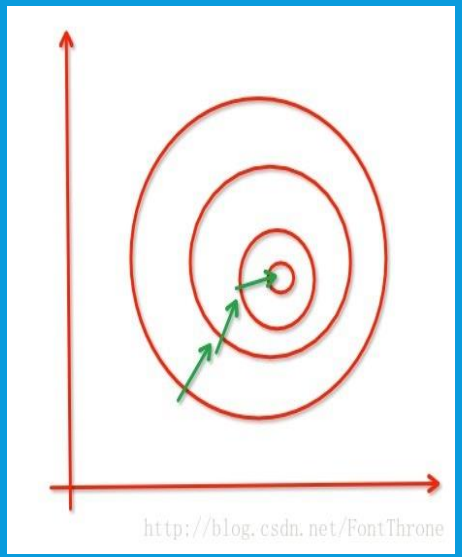
# MEAN NORM

- Do it by yourself

# DATA PREPROCESSING NORMALIZATION




- Advantage
  - Speed up the convergence for SGD
  - Improve the precision
  - Avoid exploding gradient for Deep Learning
- Disadvantage
  - Impact of outliers(max and min value)
  - Bad robustness(small data)
  - Influence on geometrical shape of data



# DATA PREPROCESSING

- Continuous Feature
  - Normalization(归一化)
  - Standardization(标准化)
  - Zero-Centered(零中心化)
- Categorical Feature
  - OneHotEncoding

# DATA PREPROCESSING STANDARDIZATION

- Role: proportional scaling
- When
  - Features have different units, ignore the measurement and make features comparable
  - Do not change original distribution, do not influence the geometrical shape

# STANDARDIZATION

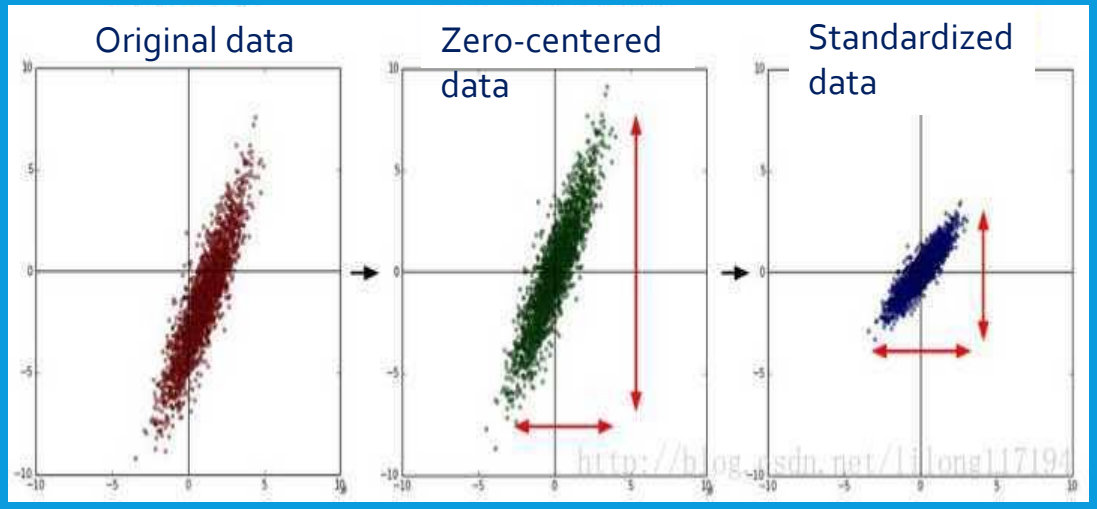
```
x = data[:, :-1]
mean = np.mean(x, axis = 0)
std = np.std(x, axis = 0)
x_std = (x - mean)/std
```

# DATA PREPROCESSING

- Continuous Feature
  - Normalization(归一化)
  - Standardization(标准化)
  - Zero-Centered(零中心化)
- Categorical Feature
  - OneHotEncoding

# DATA PREPROCESSING STANDARDIZATION

- Z-score standardization



# DATA PREPROCESSING

- Continuous Feature
  - Normalization(归一化)
  - Standardization(标准化)
  - Zero-Centered(零中心化)
- Categorical Feature
  - OneHotEncoding

# ONEHOT ENCODING

```
c1 = data_dia[:,0]
m=len(c1)
n=len(np.unique(c1)))
pos = np.array(c1, dtype=np.int)
ohe = np.eye(m,n)[pos]
c1[:10]#[0,1]=male     [1,0,0,0,0,0]
ohe[:10]#[1,0]=female  [0,1,0,0,0,0]
```