# MACHINE LEARNING
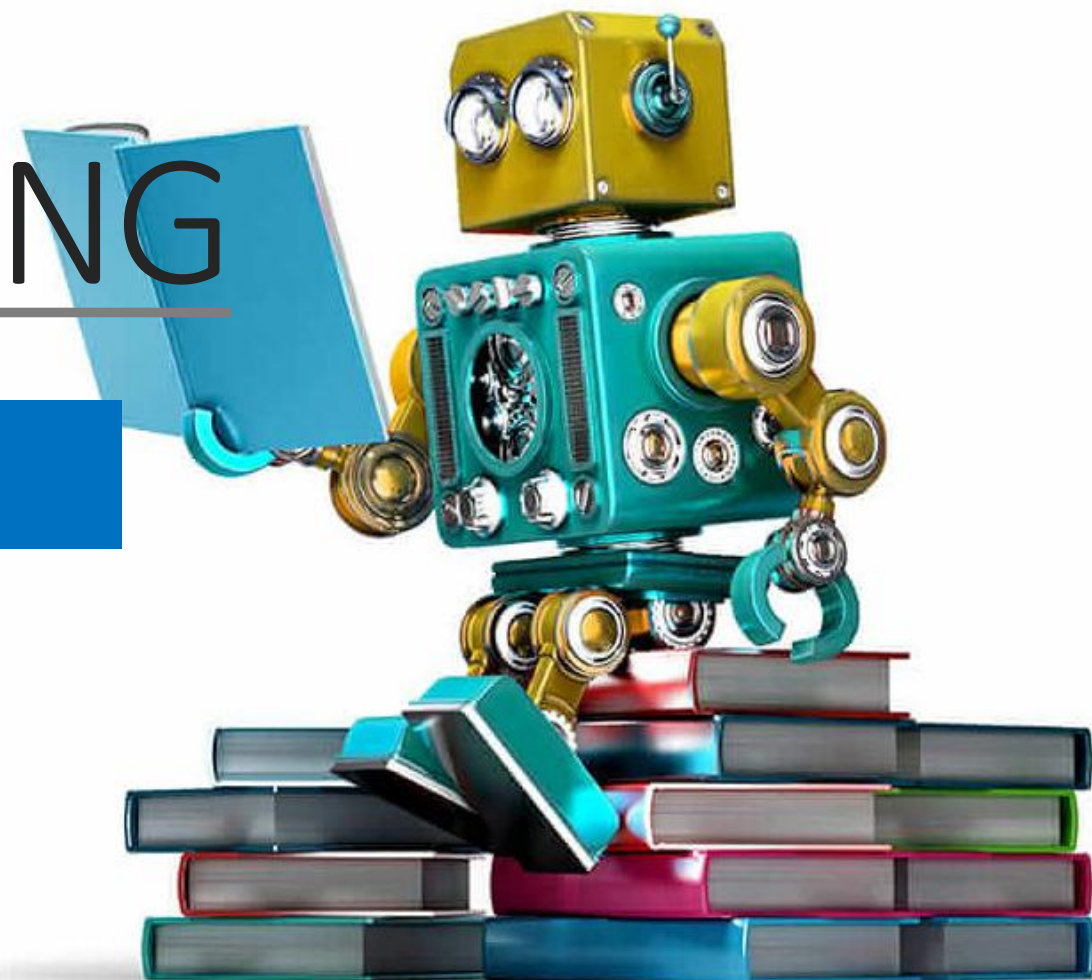
## LAB3 Bayesian Learning

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn

# OBJECTIVES

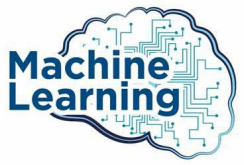**01** **Understand the Naive Bayes algorithm**

**02** **Learn how to implement the Naive Bayes Classifier in python**

# Outline

- ➢ Background and Probability Basics
- ➢ Probabilistic Classification Principle
  - Probabilistic discriminative models
  - Generative models and their application to classification
  - MAP and converting generative into discriminative
- ➢ Naïve Bayes – an generative model
  - Principle and Algorithms (discrete vs. continuous)
  - Example: Play Tennis
- ➢ Zero Conditional Probability and Treatment
- ➢ Summary
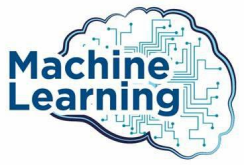
- There are three methodologies:

  ✓ Model a classification rule directly

    Examples: k-NN, linear classifier, SVM, neural nets, ..

  ✓ Model the probability of class memberships given input data

    Examples: logistic regression, probabilistic neural nets (softmax),…

  ✓ Make a probabilistic model of data within each class

    Examples: naive Bayes, model-based ….

- Important ML taxonomy for learning models

  probabilistic models vs non-probabilistic models

  discriminative models vs generative models

- Based on the taxonomy, we can see different the essence of learning models (classifiers) more clearly.

|  | Probabilistic | Non-Probabilistic |
|---|---|---|
| **Discriminative** | • **Logistic Regression**<br>• **Probabilistic neural nets**<br>• **……..** | • **K-nn**<br>• **Linear classifier**<br>• **SVM**<br>• **Neural networks**<br>• **……** |
| **Generative** | • **Naïve Bayes**<br>• **Model-based (e.g., GMM)**<br>• **……** | **N.A. (?)** |

# Probability Basics

➢ Prior, conditional and joint probability for random variables

     – Prior probability: $P(x)$

     – Conditional probability: $P(x_1|x_2), P(x_2|x_1)$

     – Joint probability: $\mathbf{x}=(x_1, x_2), P(\mathbf{x})=P(x_1, x_2)$

     – Relationship: $P(x_1, x_2)=P(x_2|x_1)P(x_1)=P(x_1|x_2)P(x_2)$

     – Independence: $P(x_2|x_1)=P(x_2), P(x_1|x_2)=P(x_1), P(x_1, x_2)=P(x_1)P(x_2)$

➢ Bayesian Rule

$$P(c|\mathbf{x})=\frac{P(\mathbf{x}|c)P(c)}{P(\mathbf{x})}$$
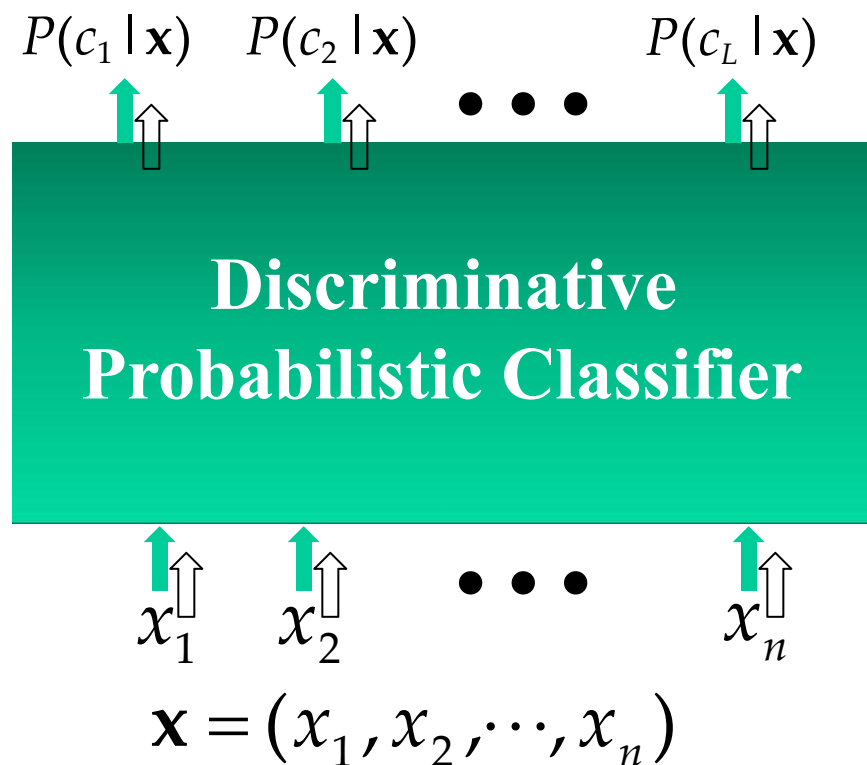
$$Posterior=\frac{Likelihood \times Prior}{Evidence}$$

Discriminative

Generative

➢ Establishing a probabilistic model for classification

– **Discriminative model**

$$P(c \mid \mathbf{x}) \quad c = c_1, \cdots, c_L, \ \mathbf{x} = (x_1, \cdots, x_n)$$

$P(c_1 \mid \mathbf{x}) \quad P(c_2 \mid \mathbf{x}) \qquad\qquad P(c_L \mid \mathbf{x})$

**Discriminative Probabilistic Classifier**

$x_1 \quad x_2 \qquad \cdots \qquad x_n$

$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

- To train a discriminative classifier regardless its probabilistic or non-probabilistic nature, all training examples of different classes must be jointly used to build up a single discriminative classifier.
- Output $L$ probabilities for $L$ class labels in a probabilistic classifier while a single label is achieved by a non-probabilistic classifier .
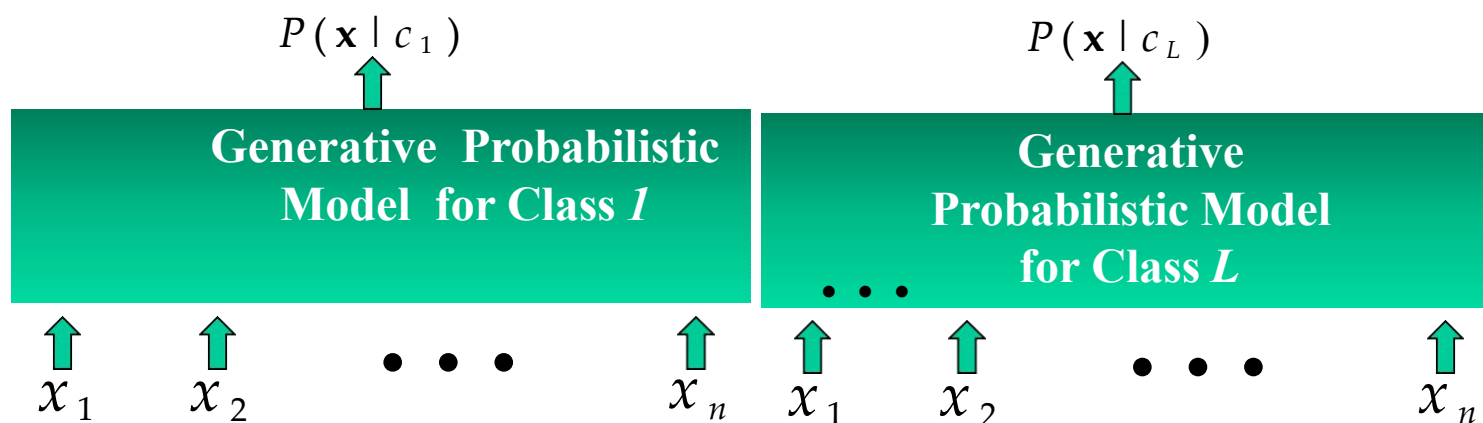
➢ Establishing a probabilistic model for classification (cont.)

    – **Generative model (must be probabilistic)**

$$P(\mathbf{x} \mid c) \quad c = c_1, \cdots, c_L, \quad \mathbf{x} = (x_1, \cdots, x_n)$$

- $L$ probabilistic models have to be trained independently
- Each is trained on only the examples of the same label
- Output $L$ probabilities for a given input with $L$ models
- "Generative" means that such a model produces data subject to the distribution via sampling.

$P(\mathbf{x} \mid c_1)$            $P(\mathbf{x} \mid c_L)$

| Generative Probabilistic Model for Class *1* | Generative Probabilistic Model for Class *L* |

$x_1 \quad x_2 \quad \bullet \bullet \bullet \quad x_n \quad x_1 \quad x_2 \quad \bullet \bullet \bullet \quad x_n$

$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

# Probabilistic Classification Principle

➢ **M**aximum **A P**osterior (**MAP**) classification rule

- For an input $\boldsymbol{x}$, find the largest one from L probabilities output by a discriminative probabilistic classifier $P(c_1 \mid \mathbf{x}), ..., P(c_L \mid \mathbf{x})$.

- Assign $\boldsymbol{x}$ to label $c*$ if $P(c^* \mid \mathbf{x})$ is the largest.

➢ Generative classification with the MAP rule

- Apply Bayesian rule to convert them into posterior probabilities

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{P(\mathbf{x})} \propto P(\mathbf{x} \mid c_i)P(c_i)$$

Common factor for all $L$ probabilities

$$\text{for } i = 1, 2, \cdots, L$$

- Then apply the MAP rule to assign a label

➢ Bayes classification

$$P(c \mid \mathbf{x}) \propto P(\mathbf{x} \mid c)P(c) = P(x_1, \cdots, x_n \mid c)P(c) \text{ for } c = c_1, \dots, c_L.$$

Difficulty: learning the joint probability $P(x_1, \cdots, x_n \mid c)$ is infeasible!

➢ Naïve Bayes classification

– Assume all input features are class conditionally independent!

$$P(x_1, x_2, \cdots, x_n \mid c) = P(x_1 \mid x_2, \cdots, x_n, c)P(x_2, \cdots, x_n \mid c)$$

Applying the independence assumption

$$= P(x_1 \mid c)P(x_2, \cdots, x_n \mid c)$$

$$= P(x_1 \mid c)P(x_2 \mid c) \cdots P(x_n \mid c)$$

– Apply the MAP classification rule: assign $\mathbf{x}' = (a_1, a_2, \cdots, a_n)$ to $c^*$ if

$$[P(a_1 \mid c^*) \cdots P(a_n \mid c^*)]P(c^*) > [P(a_1 \mid c) \cdots P(a_n \mid c)]P(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

estimate of $P(a_1, \cdots, a_n \mid c^*)$

esitmate of $P(a_1, \cdots, a_n \mid c)$

- Algorithm: Discrete-Valued Features
  - Learning Phase: Given a training set $\mathbf{S}$ of $F$ features and $L$ classes,

    For each target value of $c_i$ $(c_i = c_1, \cdots, c_L)$

    $\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in S;

    For every feature value $x_{jk}$ of each feature $x_j$ $(j = 1, \cdots, F; k = 1, \cdots, N_j)$

    $\hat{P}(x_j = x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ with examples in S;

    Output: $F * L$ conditional probabilistic (generative) models

  - Test Phase: Given an unknown instance $\mathbf{x'} = (a_1', \cdots, a_n')$

    "Look up tables" to assign the label $c^*$ to $\mathbf{X'}$ if

    $$[\hat{P}(a_1' \mid c^*) \cdots \hat{P}(a_n' \mid c^*)]\hat{P}(c^*) > [\hat{P}(a_1' \mid c) \cdots \hat{P}(a_n' \mid c)]\hat{P}(c), \quad c_i \neq c^*, c = c_1, \cdots, c_L$$

# Exampe-Play Tennis

## PlayTennis: training examples

| Day | Outlook | Temperature | Humidity | Wind | PlayTennis |
|-----|---------|-------------|----------|------|------------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

- Learning Phase

| Outlook | Play=*Yes* | Play=*No* |
|---------|------------|-----------|
| *Sunny* | 2/9 | 3/5 |
| *Overcast* | 4/9 | 0/5 |
| *Rain* | 3/9 | 2/5 |

| Temperature | Play=*Yes* | Play=*No* |
|-------------|------------|-----------|
| *Hot* | 2/9 | 2/5 |
| *Mild* | 4/9 | 2/5 |
| *Cool* | 3/9 | 1/5 |

| Humidity | Play=*Yes* | Play=*No* |
|----------|------------|-----------|
| *High* | 3/9 | 4/5 |
| *Normal* | 6/9 | 1/5 |

| Wind | Play=*Yes* | Play=*No* |
|------|------------|-----------|
| *Strong* | 3/9 | 3/5 |
| *Weak* | 6/9 | 2/5 |

$P$(Play=*Yes*) = 9/14     $P$(Play=*No*) = 5/14

- Test Phase

  – Given a new instance, predict its label

    **x'**=(Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)

  – Look up tables achieved in the learning phrase

    P(Outlook=*Sunny*|Play=*Yes*) = 2/9      P(Outlook=S*unny*|Play=*No*) = 3/5

    P(Temperature=*Cool*|Play=*Yes*) = 3/9      P(Temperature=*Cool*|Play==*No*) = 1/5

    P(Huminity=*High*|Play=*Yes*) = 3/9      P(Huminity=*High*|Play=*No*) = 4/5

    P(Wind=*Strong*|Play=*Yes*) = 3/9      P(Wind=*Strong*|Play=*No*) = 3/5

    P(Play=*Yes*) = 9/14      P(Play=*No*) = 5/14

  – Decision making with the MAP rule

    P(*Yes*|**x'**) ≈ [P(*Sunny*|Y*es*)P(*Cool*|*Yes*)P(*High*|Y*es*)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053

    P(*No*|**x'**) ≈ [P(*Sunny*|N*o*) P(*Cool*|N*o*)P(*High*|N*o*)P(*Strong*|N*o*)]P(Play=*No*) = 0.0206

    Given the fact P(*Yes*|**x'**) < P(*No*|**x'**), we label **x'** to be "*No*".

# Naive Bayes

- Algorithm: Continuous-valued Features

  – Numberless values taken by a continuous-valued feature

  – Conditional probability often modeled with the normal distribution

  $$\hat{P}(x_j \mid c_i) = \frac{1}{\sqrt{2\pi}\,\sigma_{ji}} \exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

  $\mu_{ji}$ : mean (avearage) of feature values $x_j$ of examples for which c $= c_i$

  $\sigma_{ji}$ : standard deviation of feature values x $_j$ of examples for which $c = c_i$

  – Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_F)$, $C = c_1, \cdots, c_L$

  Output: FxL normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$

  – Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$

  - Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase

  - Apply the MAP rule to assign a label (the same as done for the discrete case)

# Naive Bayes

- Example: Continuous-valued Features

  – Temperature is naturally of continuous value.

  **Yes**: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

  **No**: 27.3, 30.1, 17.4, 29.5, 15.1

  – Estimate mean and variance for each class

  $$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

  $\mu_{Yes} = 21.64, \ \sigma_{Yes} = 2.35$

  $\mu_{No} = 23.88, \ \sigma_{No} = 7.09$

  – **Learning Phase**: output two Gaussian models for P(temp|C)

$$\hat{P}(x \mid Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left( -\frac{(x-21.64)^2}{2 \times 2.35^2} \right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left( -\frac{(x-21.64)^2}{11.09} \right)$$

$$\hat{P}(x \mid No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left( \frac{(x-23.88)^2}{2 \times 7.09^2} \right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left( -\frac{(x-23.88)^2}{50.25} \right)$$

# Zero conditional probability

- If no example contains the feature value

    – In this circumstance, we face a zero conditional probability problem during test

$$\hat{P}(x_1 \mid c_i) \cdots \hat{P}(a_{jk} \mid c_i) \cdots \hat{P}(x_n \mid c_i) = 0 \ \text{ for } \ x_j = a_{jk}, \ \hat{P}(a_{jk} \mid c_i) = 0$$

    – For a remedy, class conditional probabilities re-estimated with

$$\hat{P}(a_{jk} \mid c_i) = \frac{n_c + mp}{n + m} \qquad \textbf{(m-estimate)}$$

$n_c$ : number of training examples for which $x_j = a_{jk}$ and $c = c_i$

$n$ : number of training examples for which $c = c_i$

$p$ : prior estimate (usually, $p = 1/t$ for $t$ possible values of $x_j$)

$m$ : weight to prior (number of "virtual" examples, $m \geq 1$)

# Zero conditional probability

- Example: P(outlook=overcast|no)=0 in the play-tennis dataset

  - Adding **m** "virtual" examples

    - In this dataset, training examples for the "no" class is 5.

    - We can only add **m=1** "virtual" example in our m-esitmate remedy.

  - The "outlook" feature can takes only 3 values. So **p=1/3**.

  - Re-estimate P(outlook|no) with the m-estimate

$$P(overcast|no) = \frac{0 + 1 * \left(\frac{1}{3}\right)}{5 + 1} = \frac{1}{18}$$

$$P(sunny|no) = \frac{3 + 1 * \left(\frac{1}{3}\right)}{5 + 1} = \frac{5}{9} \qquad P(rain|no) = \frac{2 + 1 * \left(\frac{1}{3}\right)}{5 + 1} = \frac{7}{18}$$

# Summary

- Probabilistic Classification Principle

  - Discriminative vs. Generative models: learning $P(c|x)$ vs. $P(x|c)$

  - Generative models for classification: MAP and Bayesian rule

- Naïve Bayes: the conditional independence assumption

  - Training and test are very efficient.

  - Two different data types lead to two different learning algorithms.

  - Working well sometimes for data violating the assumption!

- Naïve Bayes: a popular generative model for classification

  - Performance competitive to most of state-of-the-art classifiers even in presence of violating independence assumption

  - Many successful applications, e.g., spam mail filtering

  - A good candidate of a base learner in ensemble learning

  - Apart from classification, naïve Bayes can do more…
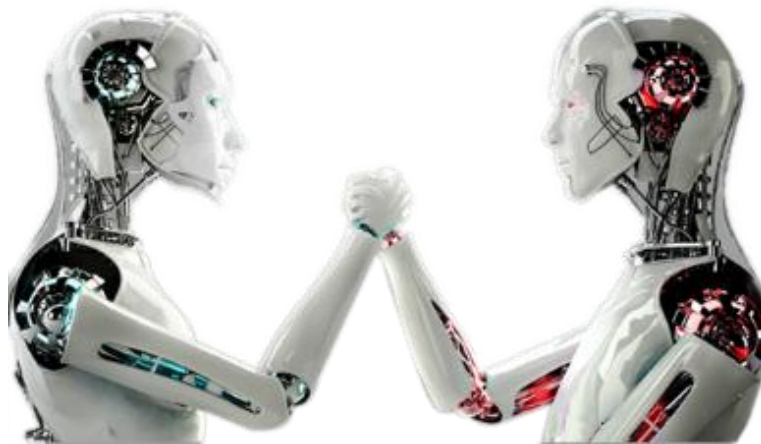
# Lab Task

1. Complete the exercises and questions in the lab03_Bayes.pdf
2. Submit your result file with an extension ".ipynb" to BB.

# Thanks

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn