多元统计分析

第13讲 判别分析(II)

Johnson & Wichern Ch11.4-6

统计学研究中心 邓婉璐
wanludeng@tsinghua.edu.cn

2018-2019春季学期

Outline

- Classification for g Populations
- Additional notes on sample spaces and CCA vs LDA
- Summary
- KNN

Fisher’s Approach

Recall ANOVA (to MANOVA)

- ANOVA:
- Model $X_{lj} = \underbrace{\mu}_{\text{overall mean}} + \underbrace{\tau_l}_{\text{treatment effect}} + \underbrace{e_{lj}}_{\text{random error}}$

Constraint for identifiability

$$\sum_{l=1}^{g} n_l \tau_l = 0$$

- Target $H_0 : \tau_1 = \tau_2 = \cdots = \tau_g = 0$
- Sample $\underbrace{x_{lj}}_{\text{observation}} = \underbrace{\bar{x}}_{\text{overall sample mean}} + \underbrace{(\bar{x}_l - \bar{x})}_{\text{estimated treatment effect}} + \underbrace{(x_{lj} - \bar{x}_l)}_{\text{residual}}$

$$\underbrace{\sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj} - \bar{x})^2}_{\text{SS}_{cor}} = \underbrace{\sum_{l=1}^{g} n_l(\bar{x}_l - \bar{x})^2}_{\text{SS}_{tr}} + \underbrace{\sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj} - \bar{x}_l)^2}_{\text{SS}_{res}}$$

SS_{cor}: total (corrected) SS

SS_{tr}: between (samples) SS

SS_{res}: within (samples) SS

Recall ANOVA (to MANOVA)

- ANOVA table for comparing univariate population means

Source of variation	Sum of squares (SS)	Degrees of freedom (d.f.)
Treatments	$\mathrm{SS}_{tr} = \sum_{l=1}^{g} n_l (\bar{x}_l - \bar{x})^2$	$g-1$
Residual (error)	$\mathrm{SS}_{res} = \sum_{l=1}^{g} \sum_{j=1}^{n_l} (x_{lj} - \bar{x}_l)^2$	$\sum_{l=1}^{g} n_l - g$
Total (corrected for the mean)	$\mathrm{SS}_{cor} = \sum_{l=1}^{g} \sum_{j=1}^{n_l} (x_{lj} - \bar{x})^2$	$\sum_{l=1}^{g} n_l - 1$

Recall ANOVA (to MANOVA)

- MANOVA:
- Model $X_{lj} = \mu + \tau_l + e_{lj}, j = 1, \cdots, n_l, l = 1, \cdots, g$
- Target $H_0 : \tau_1 = \tau_2 = \cdots = \tau_g = 0$
- Sample $x_{lj} = \bar{x} + (\bar{x}_l - \bar{x}) + (x_{lj} - \bar{x}_l)$

Constraint for identifiability

$$\sum_{l=1}^{g} n_l \tau_l = 0$$

observation = overall sample mean + estimated treatment effect + residual

$$\sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj}-\bar{x})(x_{lj}-\bar{x})' = \sum_{l=1}^{g} n_l(\bar{x}_l-\bar{x})(\bar{x}_l-\bar{x})' + \sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj}-\bar{x}_l)(x_{lj}-\bar{x}_l)'$$

total (corrected) SS = between (samples) SS + within (samples) SS

Recall ANOVA (to MANOVA)

- MANOVA table for comparing population mean vectors

Source of variation	Matrix of sum of squares and cross products (SSP)	Degrees of freedom (d.f.)
Treatments	$B = \sum_{l=1}^{g} n_l(\bar{x}_l - \bar{x})(\bar{x}_l - \bar{x})'$	$g-1$
Residual (error)	$W = \sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj} - \bar{x}_l)(x_{lj} - \bar{x}_l)'$	$\sum_{l=1}^{g} n_l - g$
Total (corrected for the mean)	$B + W = \sum_{l=1}^{g}\sum_{j=1}^{n_l}(x_{lj} - \bar{x})(x_{lj} - \bar{x})'$	$\sum_{l=1}^{g} n_l - 1$

More than 2 Populations (g Populations)

- Assumption: $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_g = \Sigma$
- Denote $B_\mu = \sum_{i=1}^{g} (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})'$, $\bar{\mu} = \frac{1}{g}\sum_{i=1}^{g} \mu_i$
- Consider linear combination $Y = a'X$
- Then $E(Y) = a'E(X \mid \pi_i) = a'\mu_i$ for population π_i

  $Var(Y) = a'Cov(X)a = a'\Sigma a$ for all populations
- Thus the overall mean

$$\bar{\mu}_Y = \frac{1}{g}\sum_{i=1}^{g} \mu_{iY} = \frac{1}{g}\sum_{i=1}^{g} a'\mu_i = a'(\frac{1}{g}\sum_{i=1}^{g} \mu_i) = a'\bar{\mu}$$

with $\mu_{iY} = a'\mu_i$

More than 2 Populations (g Populations)

- The separation

$$\frac{\text{sum of squared distances from populations to overall mean of Y}}{\text{variance of Y}} = \frac{\sum_{i=1}^{g}(\mu_{iY} - \bar{\mu}_Y)^2}{\sigma_Y^2} = \frac{\sum_{i=1}^{g}(a'\mu_i - a'\bar{\mu})^2}{a'\Sigma a}$$

$$= \frac{a'\left(\sum_{i=1}^{g}(\mu_i - \bar{\mu})(\mu_i - \bar{\mu})'\right)a}{a'\Sigma a}$$

$$= \frac{a'B_\mu a}{a'\Sigma a}$$

- How to modelling?

More than 2 Populations (g Populations)

- Sample counterparts: $B = \sum_{i=1}^{g} n_i(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})'$

$$W = \sum_{i=1}^{g}(n_i - 1)S_i = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)'$$

More than 2 Populations (g Populations)

Result

Let $\hat{\lambda}_1, \cdots, \hat{\lambda}_s > 0$ denote the $s \leq \min(g-1, p)$ nonzero eigenvalues of $W^{-1}B$ and $\hat{e}_1, \cdots, \hat{e}_s$ be the corresponding eigenvectors (scaled so that $\hat{e}' S_{pooled} \hat{e} = 1$). Then the vector of coefficients â that maximizes the ratio

$$\frac{\hat{a}' B \hat{a}}{\hat{a}' W \hat{a}} = \frac{\hat{a}' \left(\sum_{i=1}^{g} n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})' \right) \hat{a}}{\hat{a}' \left(\sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)' \right) \hat{a}}$$

is given by $\hat{a}_1 = \hat{e}_1$. The linear combination $\hat{a}_k' x = \hat{e}_k' x$ is, called the sample kth discriminant, $k \leq s$.

More than 2 Populations (g Populations)

- Properties

$$\hat{a}_i' S_{pooled} \hat{a}_k = \begin{cases} 1 & \text{if } i = k \le s \\ 0 & \text{otherwise} \end{cases}, \quad \frac{W}{n_1 + \cdots + n_g - g} = S_{pooled} \text{ is the estimate of } \Sigma.$$

- How to classify?

- Denote

$$Y_k = a_k' X = k\text{th distriminant}, k \le s \qquad Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix} \text{ has mean vector } \mu_{iY} = \begin{bmatrix} \mu_{iY_1} \\ \vdots \\ \mu_{iY_S} \end{bmatrix} = \begin{bmatrix} a_1' \mu_i \\ \vdots \\ a_s' \mu_i \end{bmatrix}$$

- Assign y to the k[th] group if y is closest to the mean of the k[th] group than with the means of other groups. 'closeness' in the following sense:

$$(y - \mu_{iY})'(y - \mu_{iY}) = \sum_{j=1}^{s} (y_j - \mu_{iY_j})^2$$

Linear discriminant analysis (LDA)

Classification

More than 2 Populations (g Populations)

- ECM: $P(k \mid i) = P(X \in R_k \mid X \in \pi_i) = \int_{R_k} f_i(x)\,dx$

$$ECM(1) = c(2 \mid 1)P(2 \mid 1) + \cdots + c(g \mid 1)P(g \mid 1)$$

$$ECM = p_1 ECM(1) + \cdots + p_g ECM(g) = \sum_{i=1}^{g} p_i \left(\sum_{\substack{k=1 \\ k \neq i}}^{g} P(k \mid i)c(k \mid i) \right)$$

$$\min_{R_1,\ldots,R_g} ECM$$

- To achieve minimum of ECM, assign x to k^{th} group if the following is minimum among all g groups: $\sum_{\substack{k=1 \\ k \neq i}}^{g} p_i f_i(x)c(k \mid i)$

More than 2 Populations (g Populations)

- ECM when c's are all equal:

The regions that minimize the ECM are defined by the values x for which the following inequalities hold:

$$R_k : p_k f_k(x) > p_i f_i(x), \forall i \neq k.$$

- Posterior: equivalent results as above.

$$P(X \in \pi_k \mid X = x_0) = \frac{P(X = x_0 \mid X \in \pi_k) P(X \in \pi_k)}{P(X = x_0 \mid X \in \pi_1) P(X \in \pi_1) + \cdots + P(X = x_0 \mid X \in \pi_g) P(X \in \pi_g)}$$

$$= \frac{p_k f_k(x_0)}{p_1 f_1(x_0) + \cdots + p_g f_g(x_0)}$$

Summary

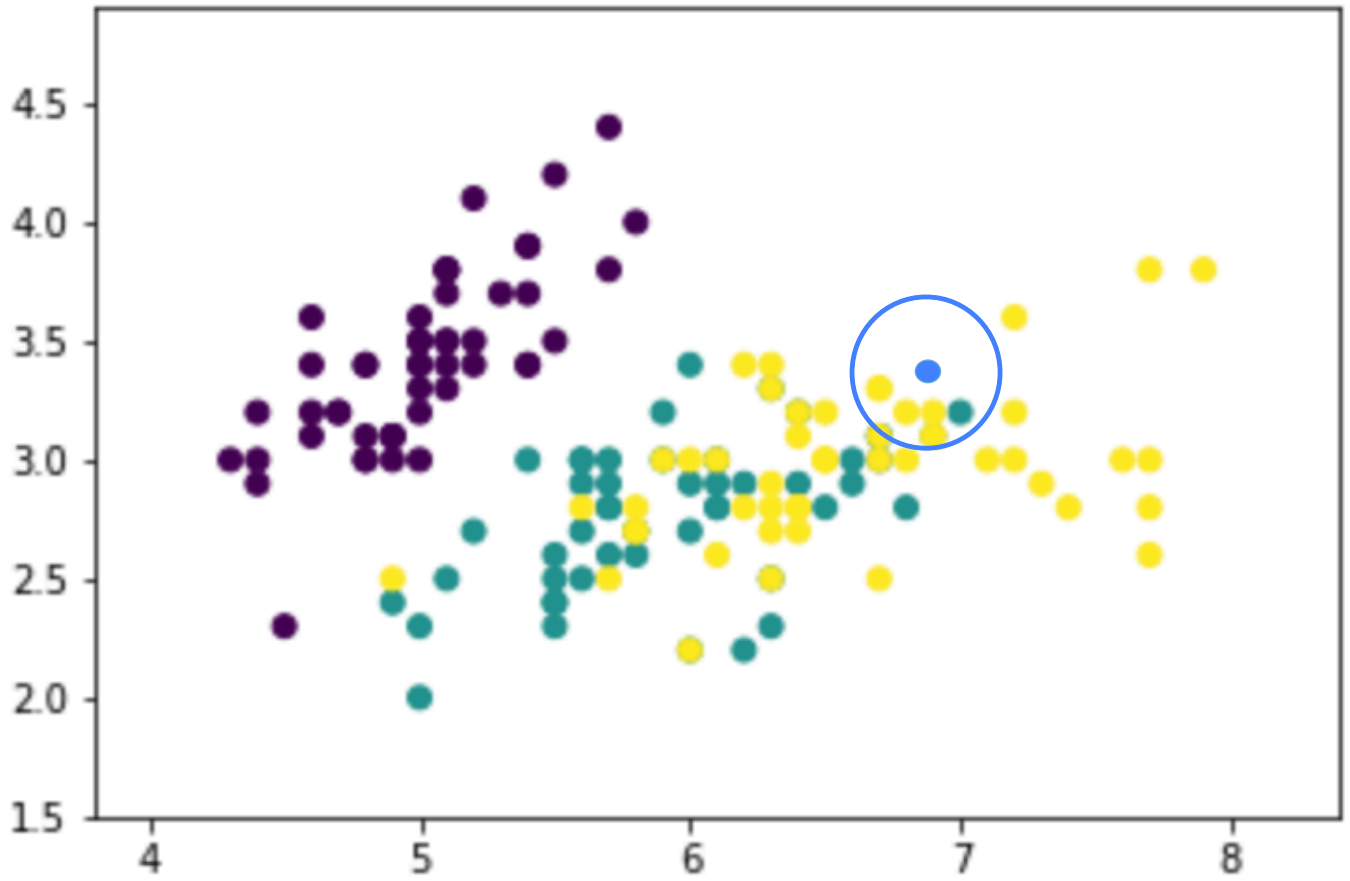
Summary for the Semester

- Level 1: know what kind of methods to search for, in application; Implementation
- Level 2: know the theoretical support (assumptions, etc)
- Level 3: be aware of connections to similar / previous methods
- Level 4: summarize the general methodology for new problem

Example for Level 4: Another Classification Method KNN

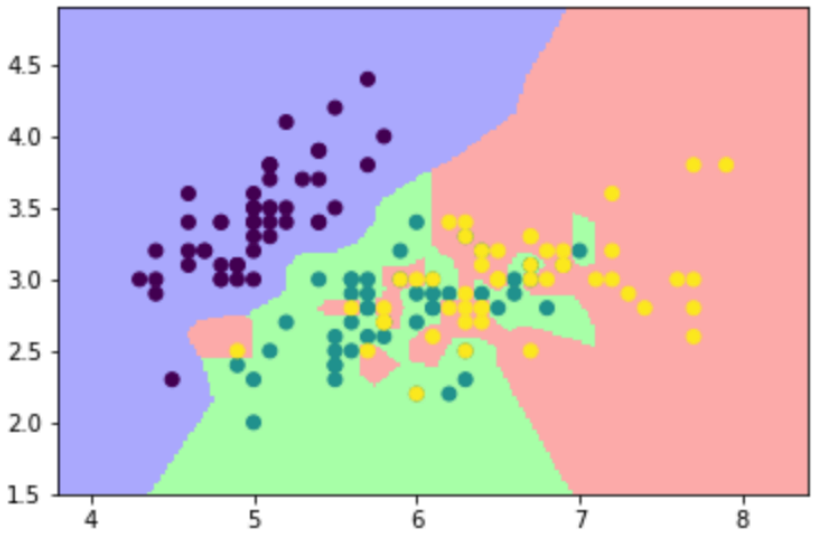
k-近邻法(K Nearest Neighbor KNN)

- 自然地想法：“物以类聚，人以群分”
  - 定义距离
  - - 例如，欧氏距离
  - 确定要利用的邻居个数K
  - - 例如，k = 5
  - 对每个点，用距离最近的K个邻居的
  - 类别信息预测其类别
  - - 例如，k个邻居中第2类最多，则预测
  - 该点类别为2.
- 预处理：可以对数据标准化

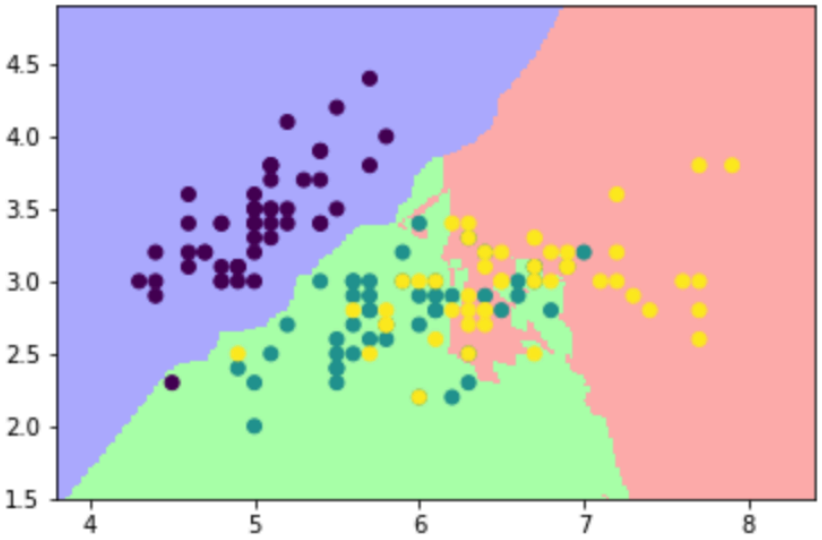


k-近邻法(KNN)

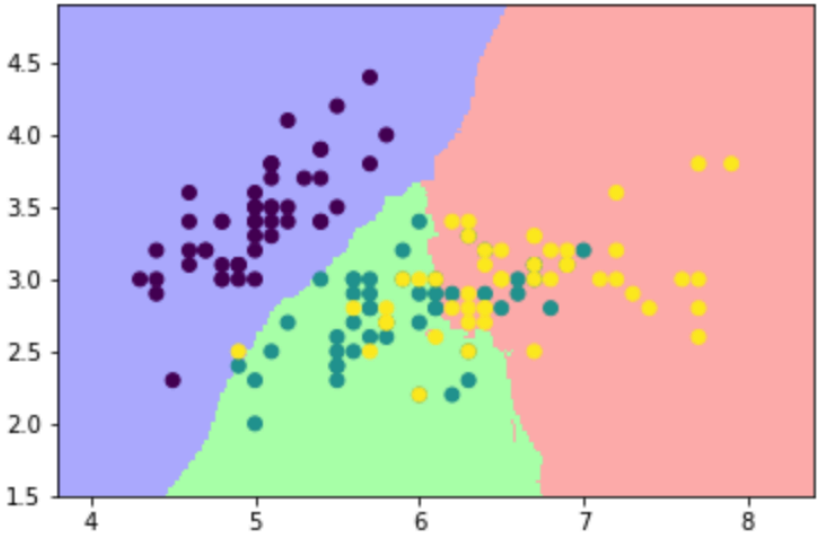
- 如何选择K？K越大，边界越光滑
- 选择准则：预测准确度的度量



K = 1



K = 8



K = 30

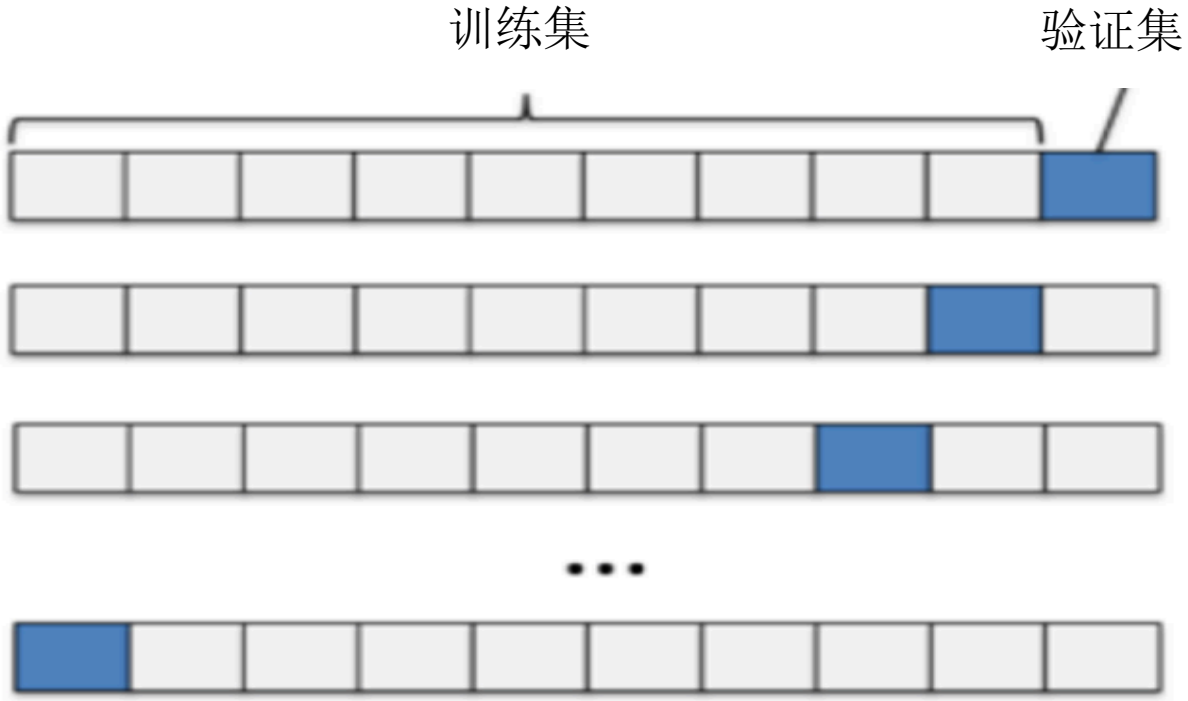
测试与验证

- 可以看到每次改变训练集的样本组成，得到的模型都不同
- 为了更具普适性，利用多组不同的训练集训练、调参

- 将数据集分为3份：训练(train)、验证(validation)、测试(test)
- 利用训练集和验证集调整参数，确定最终模型（可以多次使用）
  - 利用训练集训练模型
  - 根据验证集上的表现调整参数
- 利用测试集评估最终模型的拟合优度（只能用一次），通常样本量占总的20%

测试与验证

- 交叉验证法(cross validation)：
  - 保留测试集不动，下面对训练集进一步分割为训练和验证集：
  - LOOCV(leave-one-out cross validation): 对每个样本点x，用排除x的全部数据作训练集，预测x的分类
  - n-fold CV: 将样本均分n份。对每份，用其他样本作训练集，该份作验证集。



k-近邻法(KNN)

- 基于10-fold CV的平均表现选择K
- 先留下测试集

In [147]:

```
import numpy as np
from sklearn import datasets
# 随机数组初始化
np.random.seed(30)

#利用random()函数将iris数据集顺序打乱，因为iris数据集是按照花卉种类顺序采集的
iris = datasets.load_iris()
x = iris.data
y = iris.target
i = np.random.permutation(len(iris.data))

#将数据集中前120条作为训练集+验证集，后30条作为测试集
x_t = x[i[:-30]]
y_t = y[i[:-30]]
x_test = x[i[-30:]]
y_test = y[i[-30:]]
```

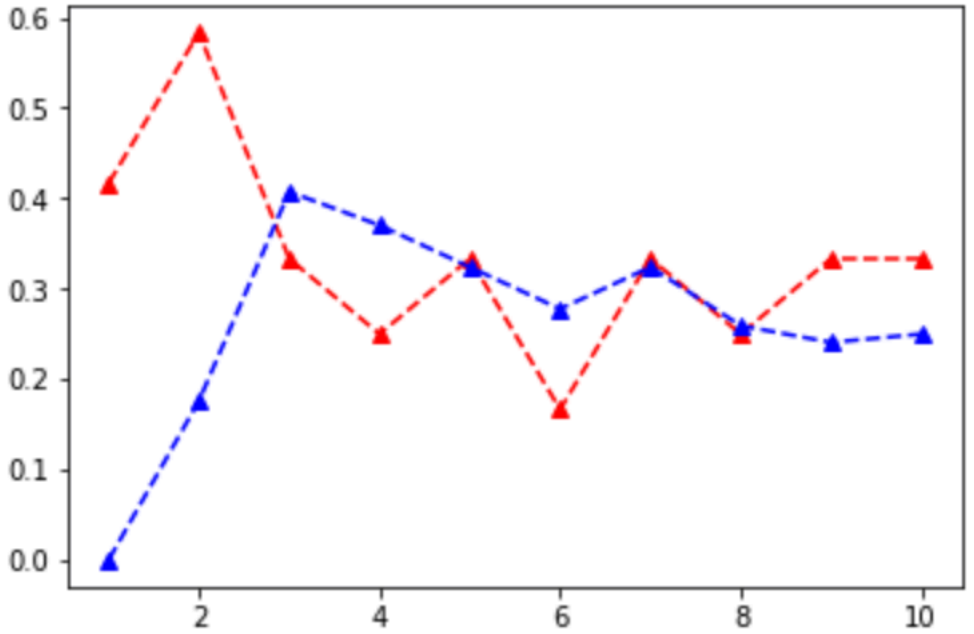
k-近邻法(KNN)

- 再利用10-fold CV：
- 对k设置一个取值范围
- 对每个固定的k：
  - 从10份样本中固定一份作验证集
  - 剩余9份样本作训练集
  - 对每个验证集中的样本点x，
    - 找训练集中与之最近的k个样本，
    - 根据它们所属类别预测x的类别
  - 计算对验证集的预测误差
  - 类似地，可计算对训练集的预测误差

```
# 导入分类器并用fit()函数训练；n-fold CV
from sklearn.neighbors import KNeighborsClassifier
k = 10
n = 10
validation_error_sum = [0]*k
train_error_sum = [0]*k
for j in range(1,k+1):
    for h in range(0,n):
        nn = int(120/n)
        x_validation = x_t[h*nn:(h+1)*nn]
        y_validation = y_t[h*nn:(h+1)*nn]
        a=np.arange(0,120)
        b = np.arange(h*nn,(h+1)*nn)
        x_train = x_t[list(set(a)-set(b))]
        y_train = y_t[list(set(a)-set(b))]
        knn = KNeighborsClassifier(n_neighbors=j)
        knn.fit(x_train,y_train)
        train_com = list(knn.predict(x_train) - y_train)
        train_error = 1-train_com.count(0)/(120-nn)
        train_error_sum[j-1] = train_error_sum[j-1] + train_error
        validation_com = list(knn.predict(x_validation) - y_validation)
        validation_error = 1-validation_com.count(0)/nn
        validation_error_sum[j-1] = validation_error_sum[j-1] + validation_error
```

k-近邻法(KNN)

- 根据预测误差确定k
- 本例可取 k = 6。
- 至此，模型确定！



- 最后，基于 k = 6 对测试集进行预测，得到预测误差，作为最终模型的评估

In [149]:
```
knn = KNeighborsClassifier(n_neighbors=6)
knn.fit(x_test,y_test)
test_com = list(knn.predict(x_test) - y_test)
test_error = 1-test_com.count(0)/30
print(test_error)
```

```
0.03333333333333326
```

k-近邻法(KNN)总结

- 想法：用周围的邻居的类别进行预测
- 非参数方法
- 优点：对异常值不敏感、无需对数据进行分布假设、操作简单
- 缺点：计算复杂度高、空间复杂度高、样本不平衡时易误差较大、无法给出数据的结构信息、维度过高时不易实现