智能软件开发
方向基础

第六章 集成学习
Ensemble Learning

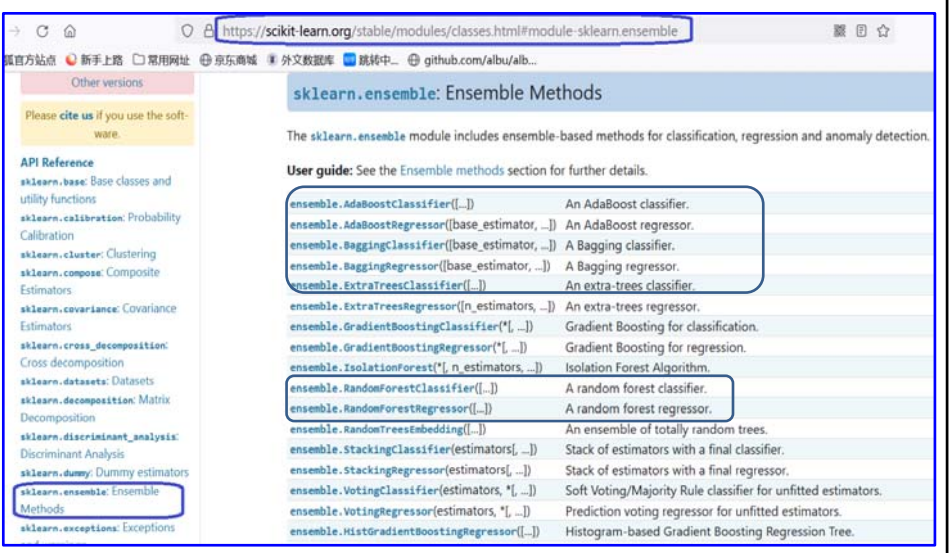
张朝晖

2022~2023学年第二学期

河北师范大学软件学院
Software College of Hebei Normal University

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble

https://scikit-learn.org/stable/modules/ensemble.html#ensemble



序号	内容
1	概述
2	机器学习的基本概念
3	模型的选择与性能评价
4	数据的获取、探索与准备
5	近邻模型------分类、回归
6	决策树模型------分类、回归
7	集成学习------分类、回归
8	(朴素)贝叶斯模型------分类
9	聚类
10	特征降维及低维可视化(PCA, t-SNE)
11	总复习

主要内容

(以决策树为)个体模型的集成学习

1 Bootstrap Aggregating(bagging)

个体模型是决策树，也可以是是其它分类模型

2 Random Forest(RF)

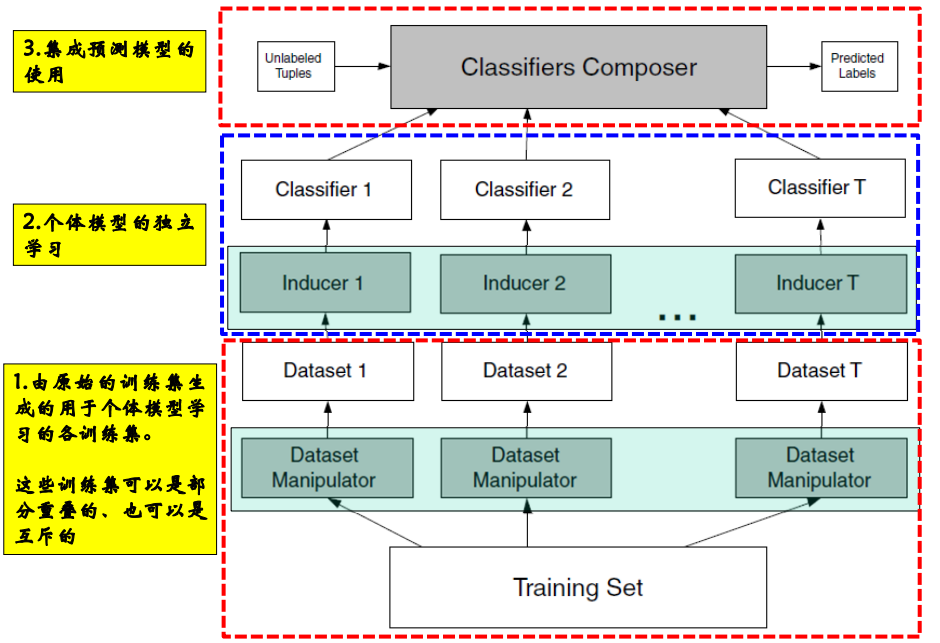
个体模型是决策树

分类—简单投票；

回归—简单平均

3 AdaBoost

例：面向分类任务的独立个体模型集成



主要内容

1 Bootstrap Aggregating(bagging)

个体模型是决策树，也可以是是其它分类模型

2 Random Forest(RF)

个体模型是决策树

分类—简单投票；

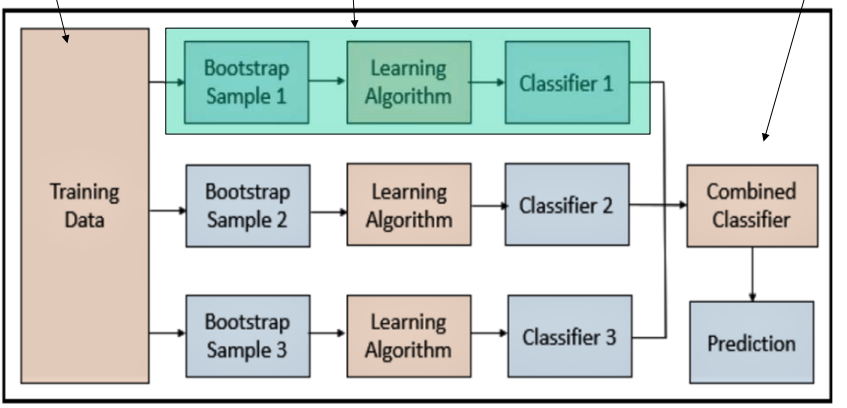
回归—简单平均

3 AdaBoost

总的训练集应有效捕获更多的样本多样性

每个个体模型都是基于总训练集的某个子集进行学习。

将所有个体模型的预测输出有效组合，所得到的预测性能优于在整个数据集上学得的单个模型的预测性能



集成学习方式之一--混合训练数据(Mix training data)

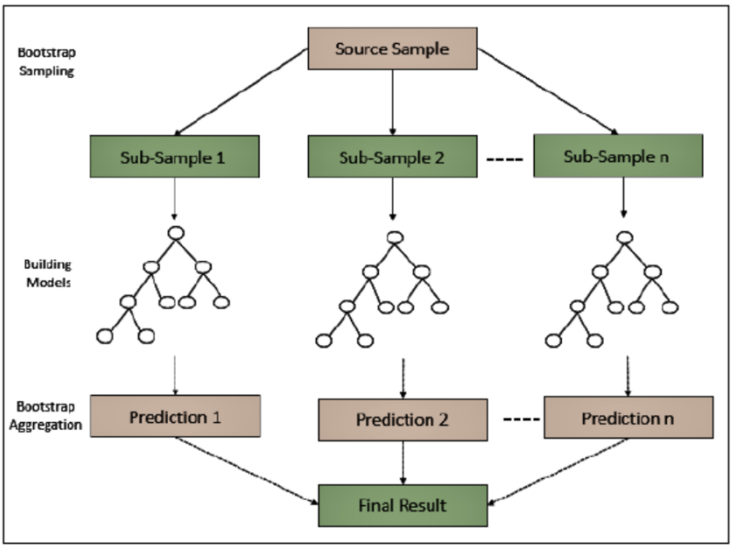
典型代表：bagging(Bootstrap Aggregating)

基于bagging的分类
----投票法决策

$Bagging(S=((x_1,y_1),\ldots,(x_m,y_m)))$
1 $for\ t \leftarrow 1\ to\ T\ do$
2 $S_t \leftarrow Bootstrap(S) \triangleright i.i.d.\ Sampling\ with\ replacement\ from\ S.$
3 $h_t \leftarrow TrainClassifier(S_t)$
4 $return\ h_* = x \rightarrow MajorityVote((h_1(x),\ldots,(h_T(x)))$

基于bagging的回归
----所有个体模型预测输出的算术均值

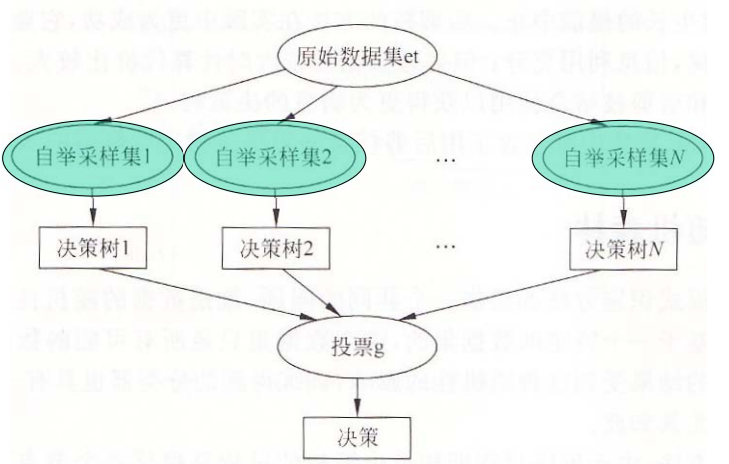
$Bagging(S=((x_1,y_1),\ldots,(x_m,y_m)))$
1 $for\ t \leftarrow 1\ to\ T\ do$
2 $S_t \leftarrow Bootstrap(S) \triangleright i.i.d.\ Sampling\ with\ replacement\ from\ S.$
3 $h_t \leftarrow TrainRegression(S_t)$
4 $return\ h_* = x \rightarrow Mean((h_1(x),\ldots,(h_T(x)))$



基于已知标签样本集的**自举采样集**，构建**多棵决策树**；

分类--多棵决策树**投票决策**

回归--多棵决策树**预测结果的平均**



Algorithm 1 Bagging

Inputs: Training data S; supervised learning algorithm, BaseClassifier, integer T specifying ensemble size; percent R to create bootstrapped training data.

Do $t = 1, \ldots, T$

1. Take a bootstrapped replica S_t by randomly drawing $R\%$ of S.
2. Call **BaseClassifier** with S_t and receive the hypothesis (classifier) h_t.
3. Add h_t to the ensemble, $\mathcal{E} \leftarrow \mathcal{E} \cup \quad h_t$.

End

Ensemble Combination: Simple Majority Voting—Given unlabeled instance x

1. Evaluate the ensemble $\mathcal{E} = \{h_1, \ldots, h_T\}$ on x.
2. Let $v_{t,c} = 1$ if h_t chooses class ω_c, and 0, otherwise.
3. Obtain total vote received by each class

$$V_c = \sum_{t=1}^{T} v_{t,c}, \quad c = 1, \ldots, C$$

Output: Class with the highest V_c.

***Bagging*（*Bootstrap AGGregatING*）算法**

输入：训练样本集$\boldsymbol{D} = \{(x_i, y_i), i = 1, \ldots, m\}$；

监督式基学习器算法ℓ；基学习器的数目N

模型的学习阶段：

初始化**基学习模型的集合**E为空集.

Do t = 1, …, N

由数据集D自举重采样得容量为m的数据集D_t;

基于数据集D_t，调用基学习器算法ℓ，得个体模型$h_t(x)$;

更新E: $E \leftarrow E \cup \{h_t(x)\}$

End

模型的使用阶段：

对于任意观测x，集成预测

- 若为实值函数回归，则 $\hat{y} = \frac{1}{N}\sum_{t=1}^{N} \hat{h}_t(x)$
- 若为分类，则 $\hat{y} = \underset{j \in \{1,2,\ldots,|Y|\}}{\operatorname{argmax}} \sum_{t=1}^{N} I\left(\hat{h}_t(x) = j\right)$

输出：$\hat{y}$　　注意：$\hat{h}_t(x)$为个体模型$h_t(\cdot)$在x处产生的预测输出.

*Bagging*模型的性能评价$(out-of-bag\ \ estimate)$

对于$\forall(x,y)\in D$，x的包外预测类别

$$H^{oob}(x)=\arg\max_{k\in\{1,2,...,C\}}\sum_{j=1}^{T}I(h_j(x)=k)\cdot I(x\notin D_j)$$

x作为部分个体模型的包外样本，得到的类别预测输出

第k类的投票结果

包外预测错误率

$$Err^{oob}=\frac{1}{|D|}\sum_{(x,y)\in D}I(H^{oob}(x)\neq y)\times 100\%$$

主要内容

1 Bootstrap Aggregating(bagging)

个体模型是决策树，也可以是是其它分类模型

2 Random Forest(RF)

个体模型是决策树

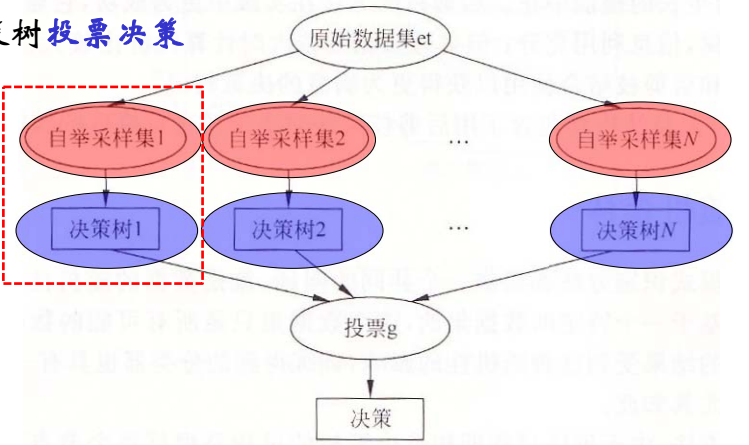
分类—简单投票；

回归—简单平均

3 AdaBoost

随机森林(bagging + 扰动版的基学习器)

- 基于样本集自举重采样+特征子集随机抽取，构建多棵决策树，组成决策树的“森林”；
- 多棵决策树投票决策



2. 基本步骤

关键：样本采样、特征采样，各树彼此独立；投票无偏。

(1)模型学习--构造N棵决策树(不剪枝)。

每棵树的构建，需要：

A--对样本数据进行“自举法(bootstrapping,或自助法)”重采样，得到1个样本集

出发点：使用相同特征空间的不同数据点

B—为该树的每个节点的特征选择，生成备选特征子集。

从特征集内随机抽取m个特征，形成该节点的学习所需要的特征子集。

(2)模型的使用--决策：

输入未知样本，得到多个决策树的输出：

分类--投票，胜者为王；回归—平均，得输出。

Algorithm 2 Random Forests

Let $\mathscr{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ denote the training data, with $x_i = (x_{i,1}, \ldots, x_{i,p})^T$. For $j = 1$ to J:

1. Take a bootstrap sample $\mathscr{D}_j$ of size N from $\mathscr{D}$.
2. Using the bootstrap sample $\mathscr{D}_j$ as the training data, fit a tree using binary recursive partitioning
   a. Start with all observations in a single node.
   b. Repeat the following steps recursively for each unsplit node until the stopping criterion is met:
      (i) Select m predictors at random from the p available predictors.
      (ii) Find the best binary split among all binary splits on the m predictors from Step (i).
      (iii) Split the node into two descendant nodes using the split from Step (ii).

模型的学习

To make a prediction at a new point x,

- $\hat{f}(x) = \frac{1}{J}\sum_{j=1}^{J} \hat{h}_j(x)$ for regression
- $\hat{f}(x) = \arg\max_y \sum_{j=1}^{J} I\left(\hat{h}_j(x) = y\right)$ for classification

where $\hat{h}_j(x)$ is the prediction of the response variable at x using the jth tree (Algorithm 1).

从p个特征中随机抽取的特征数目m经验值：

$m = \sqrt{p}$　　$m = \log_2 p$　　$m = \frac{p}{3}$

主要内容

1 Bootstrap Aggregating(bagging)

个体模型是决策树，也可以是是其它分类模型

2 Random Forest(RF)

个体模型是决策树

分类—简单投票

回归—简单平均

3 AdaBoost

Boosting方法

Boosting　本意----通过增压，加大发动机功率

引申—提升分类器性能

AdaBoost 最为广泛，Adaptive Boosting

主要内容

Boosting方法

Boosting　本意----通过增压，加大发动机功率

引申—提升分类器性能

AdaBoost 最为广泛, Adaptive Boosting

二分类、多分类、实值函数回归

AdaBoost* — *Adaptive Boosting

Y. Freund and ***R. E. Schapire***. ***A decision-theoretic generalization of on-line learning and an application to boosting***. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

2003年，Schapire 和 Freund 被授予“***the Godel Prize***”
-- one of the most prestigious awards in theoretical computer science

基于*AdaBoost*算法的强分类器训练

输入：(1)训练样本集 $\mathcal{D}=\{(x_i,y_i)\mid i=1,...,N\}$

$y_i\in\{-1,+1\}$

其中 $\begin{cases}y_i=-1,\text{训练样本}x_i\text{为负样本}\\ y_i=+1,\text{训练样本}x_i\text{为正样本}\end{cases}$

(2)**弱分类器**的学习算法 L

(3)弱分类器的数目 M

注意：此处的弱分类器模型中常用的就是“决策树树桩”，也就是只使用了一个特征生成的决策树.

输出：一个由 M 个**弱分类器**构成的**强分类器**

训练过程：

A. **初始化训练样本** x_i **权重** $\mathcal{D}_1(i)\quad i=1,...,N$

(1)若正负样本数目一致，则 $\mathcal{D}_1(i)=\frac{1}{N}$

(2)若正负样本数目分别为 N_+,N_-，则 $\begin{cases}\text{正样本}\mathcal{D}_1(i)=\frac{1}{2N_+}\\ \text{负样本}\mathcal{D}_1(i)=\frac{1}{2N_-}\end{cases}$

B. *for* $m=1,...,M$

(1)**训练弱分类器** $f_m(x)=L(\mathcal{D},\mathcal{D}_m)\in\{-1,+1\}$

(2)估计弱分类器 $f_m(x)$ 的分类错误率 e_m

注意：如果该弱分类器的错误率不是小于0.5，则只是结合其预测情况，更新样本概率分布，但不会参与集成。

如：$e_m=\sum_{i=1}^{N}\mathcal{D}_m(i)I(f_m(x_i)\neq y_i)$ [注：$e_m<0.5$]

B. *for* $m = 1,\ldots,M$ (续前)

(3) 估计弱分类器 $f_m(x)$ 的权重 $c_m = \log\dfrac{1-e_m}{e_m}$ (注意此处为自然对数)

(4) 基于弱分类器 $f_m(x)$ 调整各样本权重，并归一化

调　整：$\mathcal{D}_{m+1}(i) = \mathcal{D}_m(i)\cdot\exp\left[c_m\cdot 1_{(f_m(x_i)\neq y_i)}\right]$

$$= \begin{cases} \mathcal{D}_m(i) & \text{若 } f_m(x_i) = y_i \\ \mathcal{D}_m(i)\cdot\dfrac{1-e_m}{e_m} & \text{若 } f_m(x_i) \neq y_i \end{cases}$$

归一化：$\mathcal{D}_{m+1}(i) \leftarrow \dfrac{\mathcal{D}_{m+1}(i)}{\sum_{j=1}^{N}\mathcal{D}_{m+1}(j)}$　$i = 1,\ldots,N$

“分治策略”调整训练样本的概率分布，使后续弱分类器的学习更关注难度大的训练样本。

C. 强分类器

$$H(x) = \operatorname{sgn}\left[\sum_{m=1}^{M} c_m f_m(x)\right]$$

若分类器加权组合，得到强分类器。预测性能好的弱分类器在预测中，起较大作用。

河北师范大学软件学院
Software College of Hebei Normal University

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$;
Base learning algorithm L;
Number of learning rounds T.

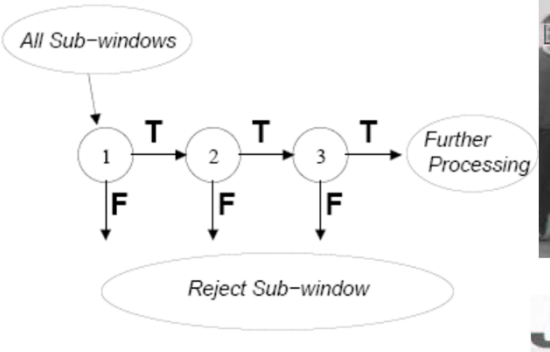
AdaBoost 二分类模型学习的另一种描述算法

Process:

1. $\mathcal{D}_1(i) = 1/m$. % Initialize the weight distribution
2. **for** $t = 1, \cdots, T$:
3. $h_t = L(D, \mathcal{D}_t)$; % Train a learner h_t from D using distribution $\mathcal{D}_t$
4. $\epsilon_t = \Pr_{x\sim\mathcal{D}_t}[h_t(x) \neq y]$; % Measure the error of h_t
5. **if** $\epsilon_t > 0.5$ **then break**
6. $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$; % Determine the weight of h_t
7. $\mathcal{D}_{t+1}(i) = \dfrac{\mathcal{D}_t(i)}{Z_t}\times\begin{cases}\exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i\end{cases}$
$= \dfrac{\mathcal{D}_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ % Update the distribution, where
% Z_t is a normalization factor which
% enables $\mathcal{D}_{t+1}$ to be distribution
8. **end**

Output: $H(x) = \operatorname{sign}\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$

典型应用：*real-time face detection*



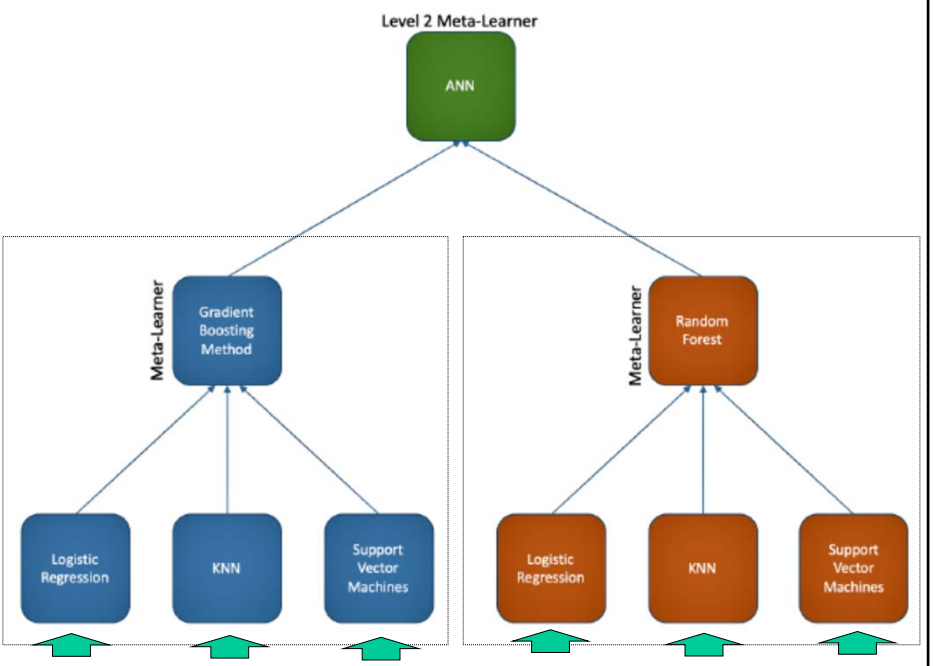
1 Bootstrap Aggregating(bagging)
2 Random Forest(RF)
3 AdaBoost

4. GBDT(Gradient Boosting Decision Tree)
5. XGBoost (eXtreme Gradient Boosting)
6. LightGBM
7. CatBoost(Categorical Boost)

河北师范大学软件学院
Software College of Hebei Normal University

主要内容

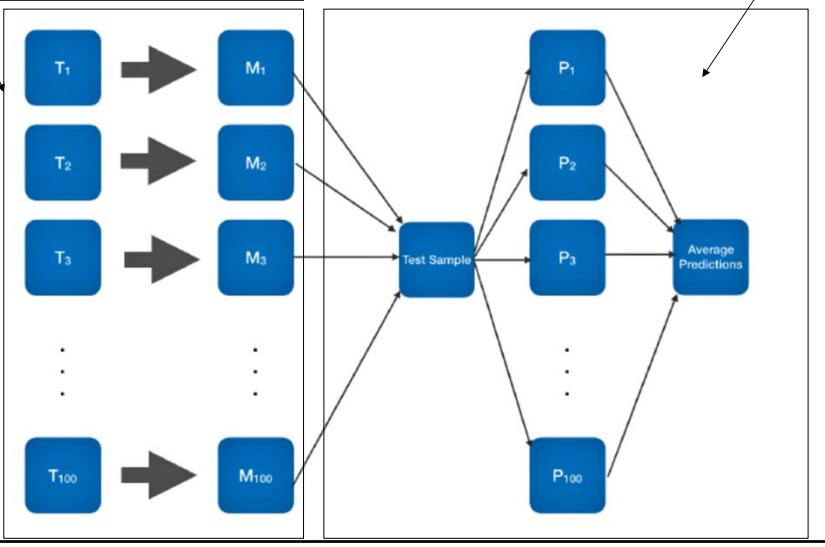
2. 集成学习模型的其它形式

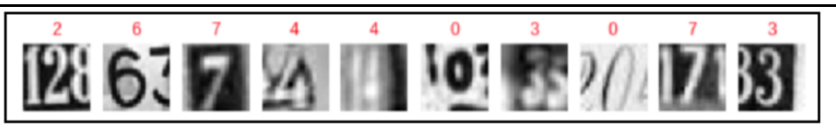


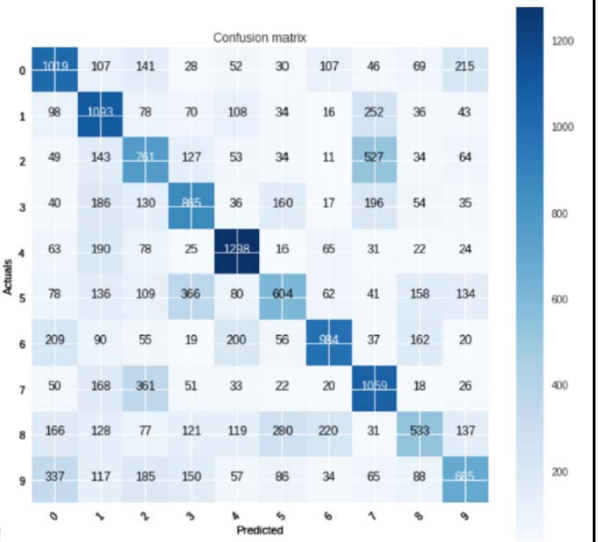
例：以神经网络为个体模型的街景门牌号码识别

模型的训练：基于不同训练集得到的彼此独立的神经网络模型

模型的使用：每个测试样本分别送入各模型，产生关于所有类别预测概率，计算平均，得到关于每个类别的平均概率；由概率最大，得到相应的预测类别。







河北师范大学软件学院
Software College of Hebei Normal University

思考题

以决策树作为个体模型，采用BAGGING以及Random Forest实现个体模型的并行集成.

给定已知答案的训练集，请对两种集成模型的实现步骤进行详细描述.

(1) 分类

(2) 回归

河北师范大学软件学院
Software College of Hebei Normal University

关于特征的重要性评价的讨论

基于置换重要性算法(permutation importance algorithm)的特征重要性评分计算

输入： 学习得到的预测模型m，用于特征评价的数据集D(训练集或验证集)

STEP1. 计算模型m在数据集D上的参考分值S

例：分类模型的预测正确率、回归模型的决定系数R^2

STEP2. 对于每个特征 j ∈ {1,2, … , d}:

2.1 对于每次随机置乱$k\epsilon\{1,2,\dots,K\}$:

----随机打乱数据集D中第j个特征的取值，产生该特征取值乱序后的污染数据集$\tilde{D}_{k,j}$

----计算模型m在数据集$\tilde{D}_{k,j}$上的预测分值$s_{k,j}$。

2.2 计算特征j的重要性得分：

$$i_j = s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j}$$

STEP3. 将所有特征重要性得分归一化，得最终评价结果。

https://scikit-learn.org/stable/modules/permutation_importance.html#permutation-importance