

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



机器学习中的多分类问题

陈传涛 硕士

2019年04月21日

- 背景简介
- 基本概念
- 算法原理
 - 一对一 (One vs. One, OvO)
 - 一对多 (One vs. All, OvA)
 - 多对多 (Many vs. Many, MvM)
- 优劣分析
- 应用总结
 - 手写数字识别
 - CII识别
- 参考文献

- 预期收获
 - 1. 了解机器学习中多分类与多标签的区别
 - 2. 了解机器学习中针对多分类问题的常用算法
 - 3. 理解多分类问题的经典解决策略及应用

- 算法提出的原因
 - 现实中常遇到多分类学习任务；（手写数字识别）
 - 许多二分类学习器不支持多分类任务；（LR、SVM）
 - 对SVM改进，使其能够处理多分类问题；
- 算法的发展历史
 - one vs. rest classification, [Bottou et al. 1994]
 - ECOC, [Dietterich and Bakiri, 1995]
- 提出问题
 - 如何使二进制分类器处理多分类问题？

- 多分类与多标签
 - 二分类 (binary classification) , 判断是/否。
 - **多类别分类** (multi-class) / 多项式分类 (multinomial) , 例如: 图像识别, 判断水果种类。特点: 互斥性。
 - 多标签分类 (multi-label) , 新闻话题。
 - 多输出分类 (multi-output) / 多任务 (multi-task) , 性别和年龄, 风速和风向。

可能取值的 个数 标签个数	$K = 2$	$K > 2$
$L = 1$	binary	multi-class
$L > 1$	multi-label	multi-output

- 多分类问题解决策略
 - 直接法（直接求解多目标函数优化问题）
 - softmax回归模型（逻辑回归 + softmax函数）；
 - SVM变体，将多个分类面的参数求解合并到一个最优化问题中；
 - 拆解法（多分类问题拆解为多个二分类任务）
 - 一对一（One vs. One, OvO）
 - 一对多（One vs. All, OvA） / 一对余（One vs. Rest）
 - 多对多（Many vs. Many, MvM）
 - 扩展
 - 有向无环图，层级分类等等；
- 其它
 - 子模型，子学习器，二分类算法
 - K，类别数。

T	将机器学习中的多分类任务进行拆解，使二分类学习方法扩展到多分类学习任务中。
I	样本标签含有多个取值的数据集
P	For { 1. 多分类问题拆解 2. 使用二分类学习器进行训练 3. 多个分类器的集成 }
O	多分类问题的预测结果

P	如何对多分类任务进行拆解与集成？
C	多分类问题的数据集，已有的二分类学习算法
D	如何拆解任务，如何获得最终的预测结果
L	IEEE CS A类会议

- “一对多”算法流程图

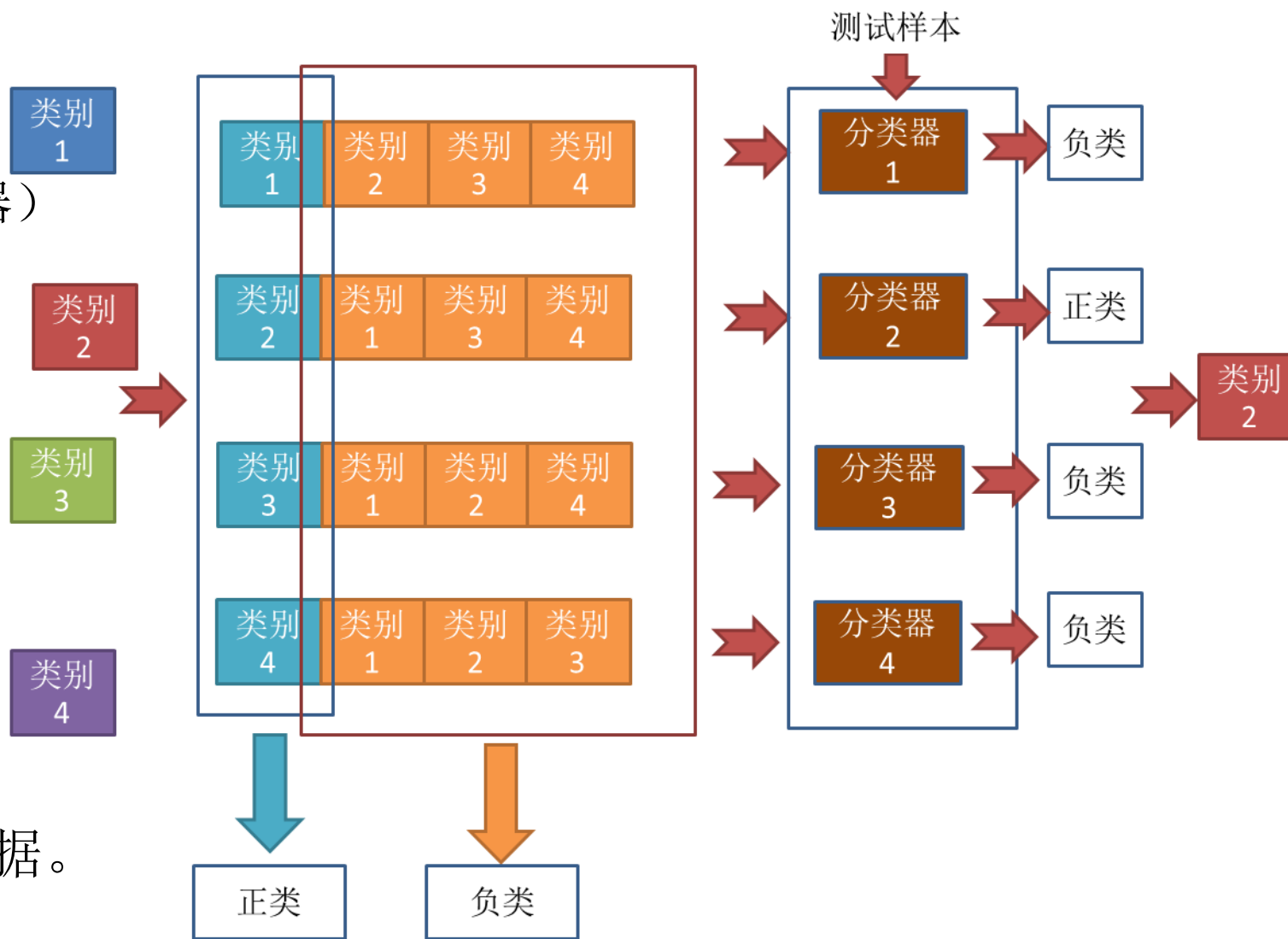
- 1. 数据集划分 (K次划分)
- 2. 训练阶段 (二分类学习器)
- 3. 测试阶段 (K次判别)
- 4. 决策阶段 (置信度)

- 优点:

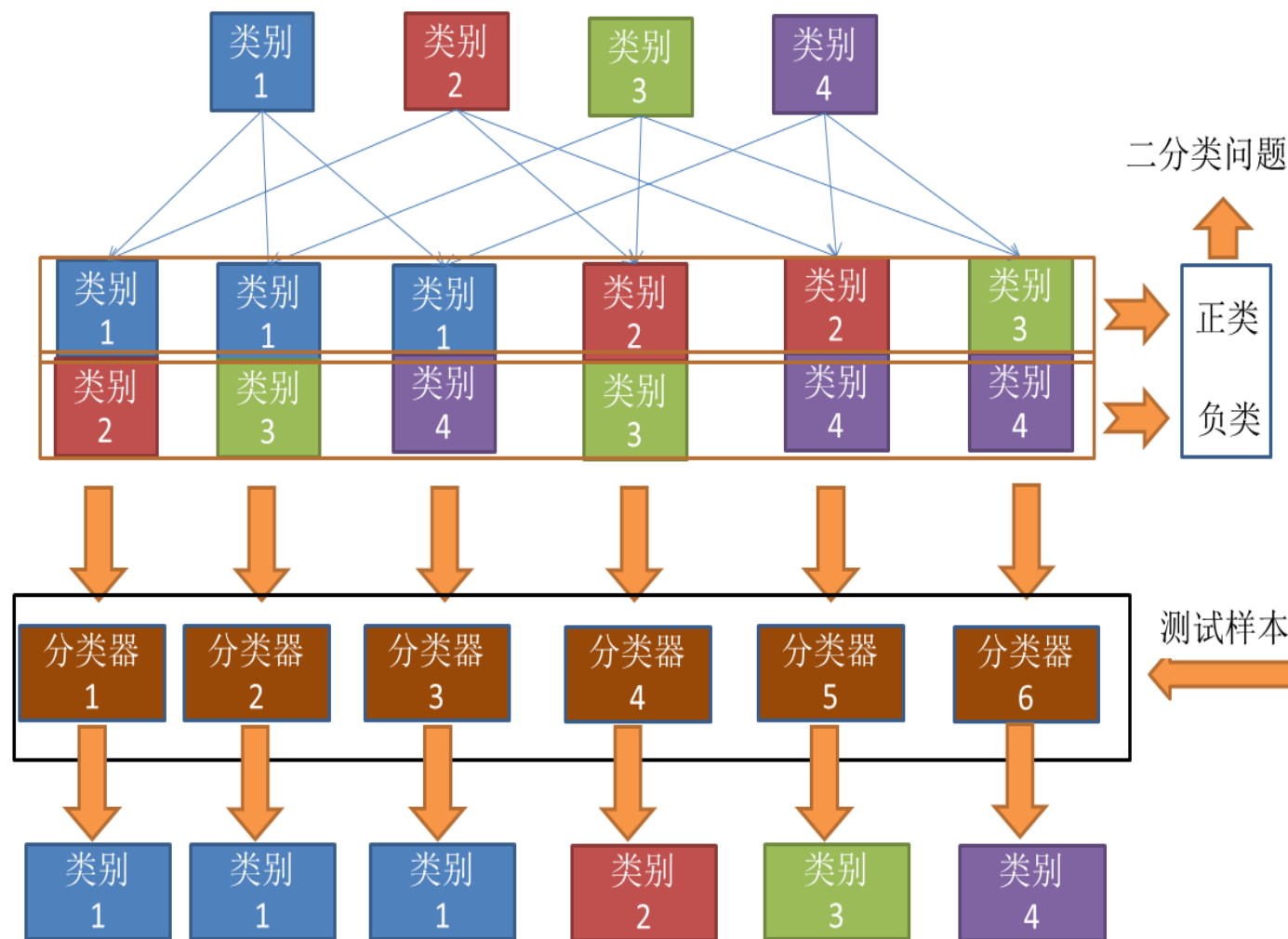
- 易实现

- 缺点:

- 引入数据不平衡问题
- 基分类器需要训练整体数据。



- “一对一”算法流程图
 - 1. 数据集划分 (C_k^2 次划分)
 - 2. 训练阶段 (二分类学习器)
 - 3. 测试阶段 (C_k^2 次判别)
 - 4. 决策阶段 (投票)
- 优点:
 - 没有引入数据不平衡问题;
- 缺点:
 - 子分类器数量随类别数 K 幂增长;



- “多对多” 算法
 - 核心思路：每次将若M个类别作为正类，若N类别作为负类。
 - “一对一”、“一对多”是“多对多”的特例：
 - 当 $M = 1$, $N = K-1$ 时，即为“一对多”策略；
 - 当 $M = N = 1$ 时，即为“一对一”策略；
 - 常用的MvM技术：
 - 纠错输出码 (Error Correcting Output Codes, ECOC)
 - 有向无环图 (Directed Acyclic Graph)
 - 层级分类 (hierarchical)

- 纠错输出码 (ECOC) 算法流程

- 1. 数据集划分 (M次划分)

- **编码** (码长M, 每条编码代表一个类别)

- 2. 训练阶段 (二分类学习器)

- 3. 测试阶段 (M次判别)

- 4. 决策阶段

- **解码** (比较编码的距离)

- 具体计算过程 (以测试样本和C1的编码为例) :

欧式距离: $\sqrt{[(-1+1)^2 + (+1+1)^2 + (-1-1)^2 + (+1-1)^2 + (+1-1)^2]} = 2\sqrt{3}$

海明距离: $(-1, +1, -1, +1, +1) \oplus (-1, -1, +1, -1, +1) \Rightarrow 01011 \oplus 00101 = 01110 \Rightarrow 3$

- 优点: 灵活 (M的设定); 纠错能力 (Error Correcting);

- 缺点: 纠错能力与训练效率不可兼得。

	f_1	f_2	f_3	f_4	f_5	海明距离	欧氏距离
	↓	↓	↓	↓	↓	↓	↓
$C_1 \rightarrow$	-1	+1	-1	+1	+1	3	$2\sqrt{3}$
$C_2 \rightarrow$	+1	-1	-1	+1	-1	4	4
$C_3 \rightarrow$	-1	+1	+1	-1	+1	1	2
$C_4 \rightarrow$	-1	-1	+1	+1	-1	2	$2\sqrt{2}$
测试示例 $\rightarrow$	-1	-1	+1	-1	+1	↑	↑

(a) 二元 ECOC 码

- 纠错输出码 (ECOC) 的纠错能力

	f_1	f_2	f_3	f_4	f_5	海明距离	欧氏距离
$C_1 \rightarrow$	-1	+1	-1	+1	+1	3	$2\sqrt{3}$
$C_2 \rightarrow$	+1	-1	-1	+1	-1	4	4
$C_3 \rightarrow$	-1	+1	+1	-1	+1	1	2
$C_4 \rightarrow$	-1	-1	+1	+1	-1	2	$2\sqrt{2}$
测试示例 $\rightarrow$	-1	-1	+1	-1	+1		

(a) 二元 ECOC 码

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	海明距离	欧氏距离
$C_1 \rightarrow$	-1	-1	+1	+1	-1	+1	+1	4	4
$C_2 \rightarrow$	-1				+1	-1		2	2
$C_3 \rightarrow$	+1	+1	-1	-1	-1	+1	-1	5	$2\sqrt{5}$
$C_4 \rightarrow$	-1	+1		+1	-1		+1	3	$\sqrt{10}$
测试示例 $\rightarrow$	-1	+1	+1	-1	+1	-1	+1		

(b) 三元 ECOC 码

- 码长 M 与类别数量 K 的关系: $\log_2 K \leq M \leq 2^{k-1} - 1$
 - 码长 M = 数据集划分的次数 = 二分类学习器的个数;
 - M 越长, 纠错能力越强, 但所需训练的分类器更多。
 - 类别间的编码距离越长, 纠错能力越强。最优编码问题——NP难问题。
- 扩展: 三元ECOC码。

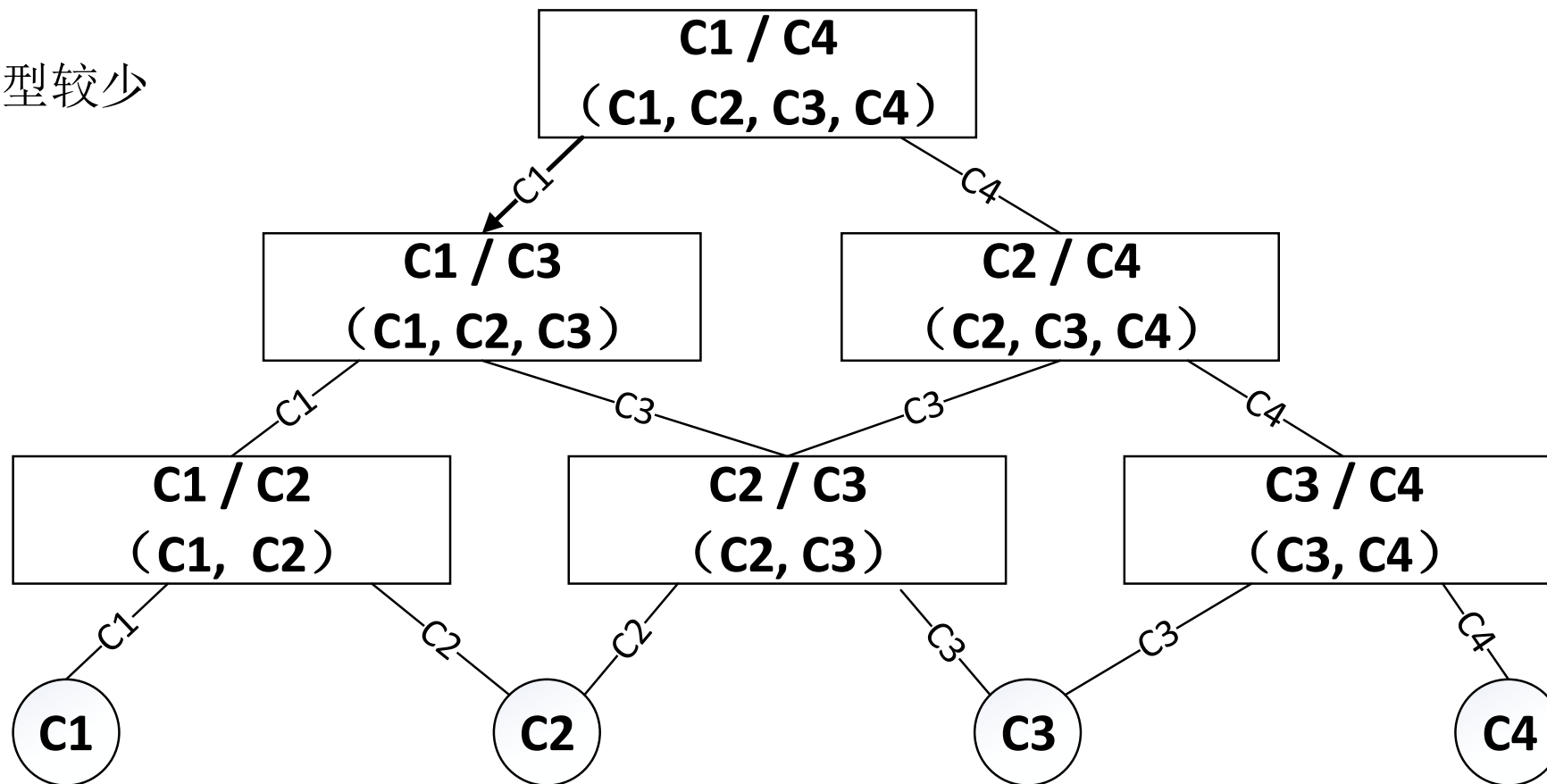
- 有向无环图 (Directed Acyclic Graph) 算法流程

- 优点

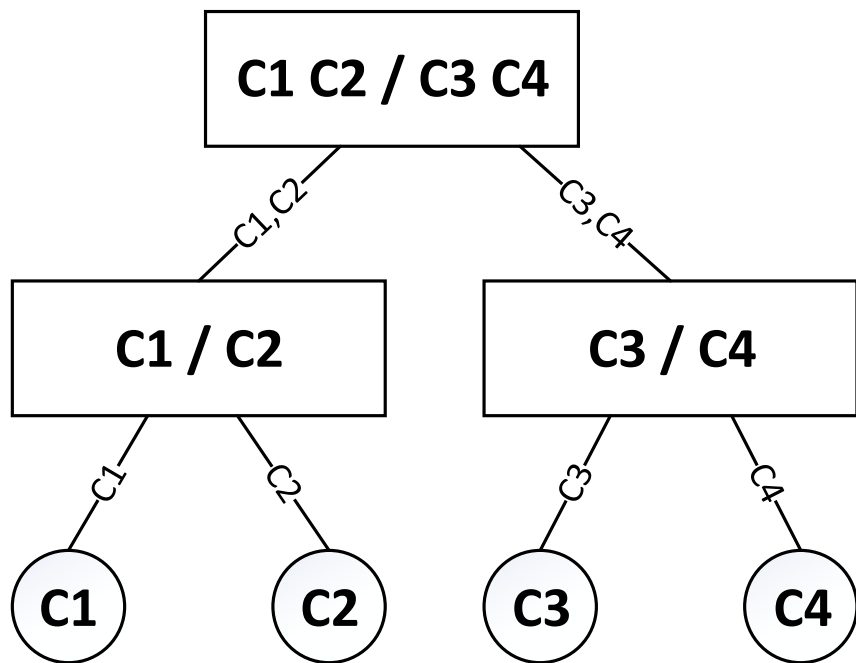
- 预测时使用模型较少

- 缺点

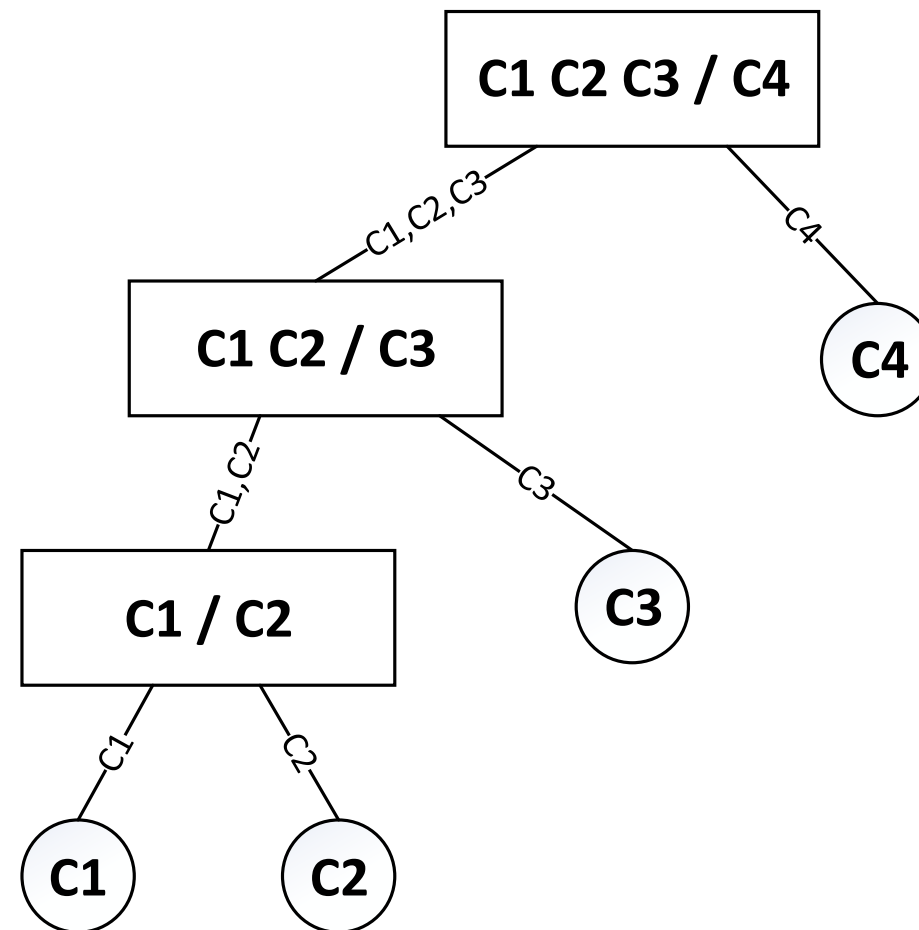
- 串行
 - 误差累积



- 层级分类 (Hierarchical) 算法流程



正态树 (H-Normal)

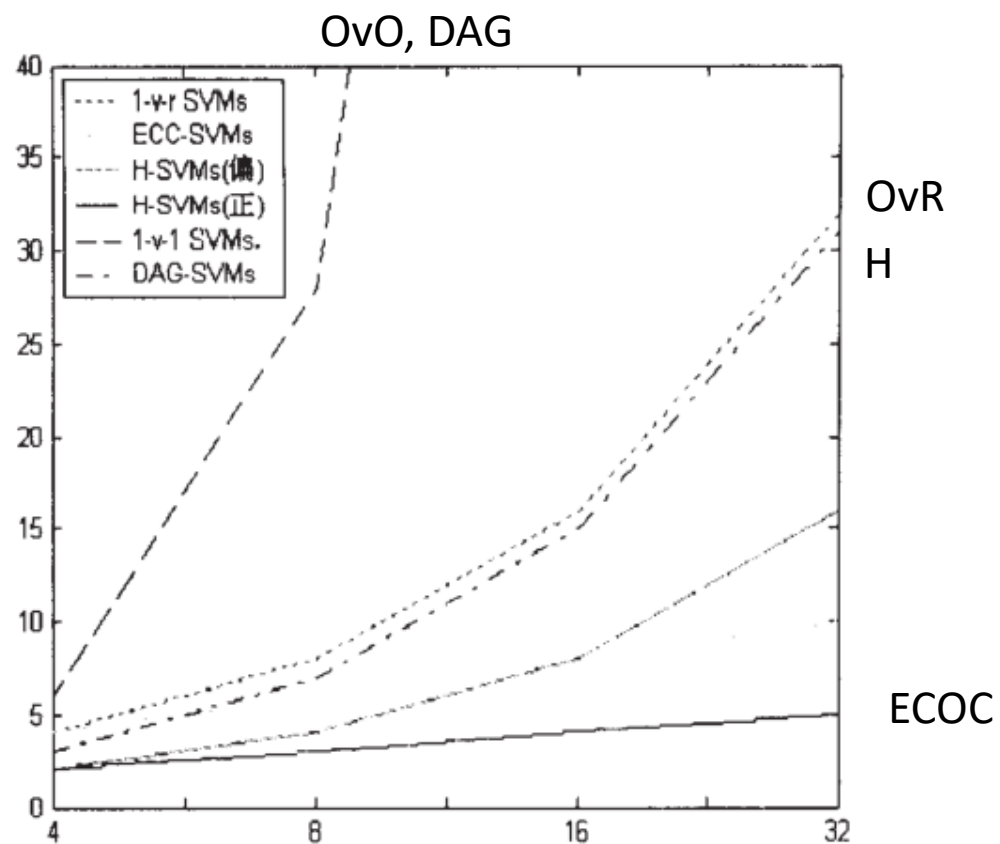


偏态树 (H-Skew)

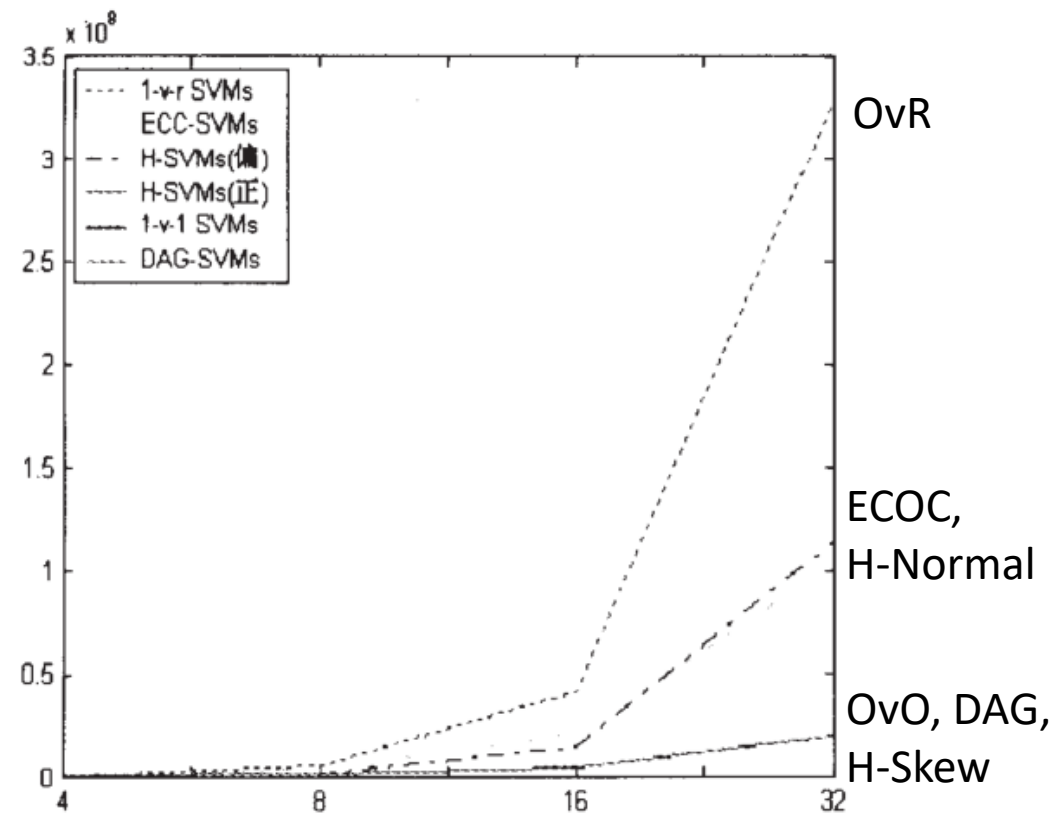
- 多分类问题拆解策略对比

策略	拆解、训练	决策	开销	优点	缺点
一对一	每次选出训练集中的某两类进行训练	简单投票	$C(k, 2)$ 个模型 每个模型的训练仅使用了2类样本	样本类别很多时，训练时间开销小于OvR	基分类器数量多
一对多	每次将某类作为正类，其他所有类作为负类进行训练	预测置信度最大	K个模型 每个模型的训练使用了全部样本	简单，易实现	数据不平衡问题
输出纠错码	每次选出M类作为正类，其他类作为负类	编码距离最小	nK个模型 默认n=1.5	灵活；纠错能力	最优编码问题
有向无环图	同OvO	“决策树”	$C(k, 2)$ 个模型	较OvO，分类时使用的模型较少	误差累积
层次分类	与OvR相似	“决策树”	K-1	较OvR，分类时使用的模型逐渐减少	误差累积

- 多分类问题拆解策略对比

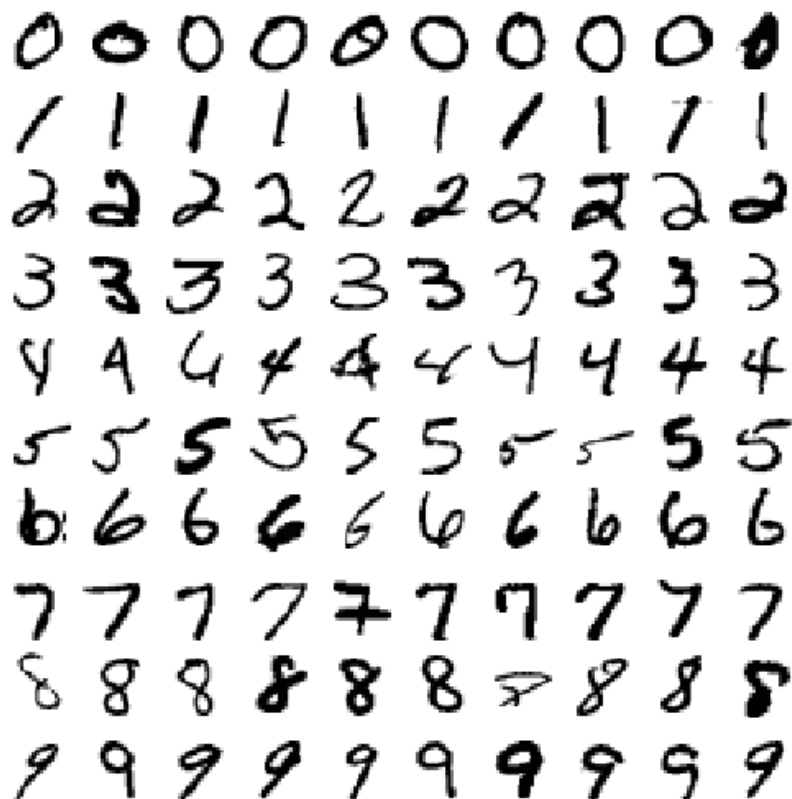


类别数 - 分类器数量 关系图



类别数 - 训练速度 关系图
 $T = cm^y$

- 手写数字识别



In [255]:

```

1 '''
2 线性SVM, 比较OVO, OVA, MVM 三种拆解策略
3 '''
4 from sklearn.svm import LinearSVC
5 from sklearn.multiclass import OneVsRestClassifier, OneVsOneClassifier, OutputCodeClassifier
6 # 划分训练集、测试集
7 X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target, test_size=0.3, random_state=1)
8 # 实验组
9 model_svc = svm.SVC(kernel='linear', decision_function_shape='ovo') # 创建模型
10 model_svc.fit(X_train, y_train) # 训练模型, X_train: 特征集, y_train: 标签。
11 y_pred = model_svc.predict(X_test) # 测试
12 print("Linear SVM Accuracy : %.3f" % metrics.accuracy_score(y_test, y_pred)) # 性能评估
13 print(metrics.classification_report(y_test, y_pred)) # 打印分类报告
14 # 对照组, 不同的任务分配策略
15 # y_pred_ovr = OneVsRestClassifier(svm.SVC(kernel='linear')).fit(X_train, y_train).predict(X_test)
16 # y_pred_ovo = OneVsOneClassifier(svm.SVC(kernel='linear')).fit(X_train, y_train).predict(X_test)
17 # y_pred_mvm = OutputCodeClassifier(svm.SVC(kernel='linear')).fit(X_train, y_train).predict(X_test)
18 print("Linear SVM ovr Accuracy : %.3f" % metrics.accuracy_score(y_test, y_pred_ovr))
19 print("Linear SVM ovo Accuracy : %.3f" % metrics.accuracy_score(y_test, y_pred_ovo))
20 print("Linear SVM mvm Accuracy : %.3f" % metrics.accuracy_score(y_test, y_pred_mvm))

```

executed in 178ms, finished 19:11:52 2019-04-19

Linear SVM Accuracy : 0.981

precision recall f1-score support

0	1.00	1.00	1.00	59
1	0.98	1.00	0.99	49
2	1.00	1.00	1.00	49
3	0.98	0.98	0.98	64
4	1.00	0.98	0.99	61
5	0.96	0.98	0.97	47
6	1.00	1.00	1.00	51
7	1.00	0.98	0.99	57
8	0.96	0.93	0.95	46
9	0.93	0.95	0.94	57

micro avg	0.98	0.98	0.98	540
macro avg	0.98	0.98	0.98	540
weighted avg	0.98	0.98	0.98	540

Linear SVM ovr Accuracy : 0.948

Linear SVM ovo Accuracy : 0.981

Linear SVM mvm Accuracy : 0.917

- 手写数字识别-拆解法

In [294]:

```
1 """对MVM拆解法的码长M进行调优
2 """
3 from sklearn.model_selection import GridSearchCV
4 from sklearn.linear_model import LogisticRegression
5
6 # 实验组
7 y_pred_lr_mvm = OutputCodeClassifier(LogisticRegression()).fit(X_train, y_train).predict(X_test)
8 print(metrics.accuracy_score(y_test, y_pred_lr_mvm))
9
10 # 对照。对code_size进行网格搜索:
11 params = {'code_size':range(2, 30,4)}
12 gsearch = GridSearchCV(estimator = OutputCodeClassifier(LogisticRegression()),
13                        param_grid = params, cv=3, scoring='accuracy')
14 gsearch.fit(X_train, y_train)
15 y = gsearch.predict(X_test) # 自动选择最优参数, 对测试集进行预测
16
17 print('best params, score: \n ', gsearch.best_params_, gsearch.best_score_, '\n')
18 print('test score: ', metrics.accuracy_score(y_test, y), '\n')
19
20 for i in zip(gsearch.cv_results_['params'], gsearch.cv_results_['mean_test_score']):
21     print(i)
```

executed in 128ms, finished 19:48:44 2019-04-19

0.9092592592592592

best params, score:

{ 'code_size': 22} 0.94351630867144

test score: 0.9555555555555556

({'code_size': 2}, 0.9220365950676214)

({'code_size': 6}, 0.939538583929992)

({'code_size': 10}, 0.9427207637231504)

({'code_size': 14}, 0.9411296738265712)

({'code_size': 18}, 0.9411296738265712)

({'code_size': 22}, 0.94351630867144)

({'code_size': 26}, 0.9427207637231504)

- CII识别 - 64分类

	Models	accuracy	precision_ macro	precision_ micro	recall_ma cro	recall_mic ro	time
直接法	LogisticRegression	9.1%	0.1%	9.1%	1.6%	9.1%	1.754
	DecisionTree	8.1%	0.1%	8.1%	1.6%	8.1%	23.050
	LGBM	13.2%	14.9%	13.2%	6.0%	13.2%	8.560
拆解法	OneVsOne(DecisionTree)	84.0%	73.2%	84.0%	68.3%	84.0%	13.616
	OneVsRest(DecisionTree)	81.5%	74.5%	81.5%	66.9%	81.5%	7.389
	MvM(DecisionTree)	84.8%	72.4%	84.8%	70.8%	84.8%	17.674
	OvO(lightGBM)	79.2%	62.1%	79.2%	53.4%	79.2%	105.602
	OvR(lightGBM)	74.4%	55.4%	74.4%	56.1%	74.4%	18.223
	MvM(lightGBM)	76.0%	70.7%	76.0%	58.7%	76.0%	26.343
多对多对照组	MvM(DecisionTree) n=0.5K	83.4%	64.2%	83.4%	68.3%	83.4%	6.335
	MvM(DecisionTree) n=1.0K	84.9%	71.9%	84.9%	70.8%	84.9%	24.868
	MvM(DecisionTree) n=2K	85.4%	75.4%	85.4%	71.8%	85.4%	50.673
	MvM(DecisionTree) n=4K	85.3%	75.4%	85.3%	71.7%	85.3%	66.140
	MvM(DecisionTree) n=5K	85.3%	74.6%	85.3%	71.0%	85.3%	78.146
	MvM(DecisionTree) n=6K	85.2%	75.0%	85.2%	71.2%	85.2%	93.723

- 算法的应用领域
 - 二分类学习算法处理多分类任务
- 未来的发展
 - ECOC的最优编码问题;
 - 有向无环图、层级分类方法中的最优化分类别问题。

- [1] 《机器学习》周志华. 线性模型-多分类学习: P63~P66.
- [2] Bottou L, Cortes C, Dcnker et al. Comparison of classifier methods: a case study in handwritten digit recognition[C]. In: International Conference on Pattern Recongition, IEEE Computer Society Press, 1994:77~87
- [3] Multiclass and multilabel algorithms — scikit-learn 0.20.3 documentation
<https://scikit-learn.org/stable/modules/multiclass.html#multiclass>



谢谢！

大成若缺，其用不弊。大盈
若冲，其用不穷。大直若屈。
大巧若拙。大辩若讷。静胜
躁，寒胜热。清静为天下正。

