

实验报告

141220132 银琦 141220132@smail.nju.edu.cn

一. 实验要求

Try J4.8 (C4.5), Naïve Bayes, SVM, Neural Network, kNN, and their ensemble version using Bagging on provided data sets based on 10-fold cross validation, can compare their performances w.r.t, accuracy, and AUC. Discuss on their performance and suggest how to improve Bagging of KNN (with necessary experimental evidence).

二. 问题及数据集描述

给定十个数据集，具体如下表，分别用 C4.5 决策树、朴素贝叶斯、支持向量机、神经网络、K-近邻以及每个算法的集成学习算法对数据集进行十折交叉验证，比较它们的精确度及 AUC，并改进 KNN 的集成学习算法。


数据集	breast-w	colic	credit-a	credit-g	diabetes
样本总数	699	368	690	1000	768
属性数量	9	22	15	20	8
分类数量	2	2	2	2	2
数据集	hepatitis	mozilla4	pc1	pc5	waveform-5000
样本总数	155	15545	1109	17186	5000
属性数量	20	5	21	38	40
分类数量	2	2	2	2	3

三. 方法描述

本次实验使用 Weka 进行数据分类，对每个数据集都用规定的五种算法及其集成算法进行分类。

1. J48

决策树是对数据进行分类，以达到预测的目的。先根据训练集数据形成决策树，如果该树不能对所有对象给出正确的分类，那么选择一些例外加入到训练集数据中，重复该过程一直到形成正确的决策集。决策树代表着决策集的树形结构。决策树由决策结点、分支和叶子组成。决策树中最上面的结点为根结点，每个分支是一个新的决策结点，或者是树的叶子。每个决策结点代表一个问题或决策，通常对应于待分类对象的属性。每一个叶子结点代表一种可能的分类结果。沿决策树从上到下遍历的过程中，在每个结点都会遇到一个测试，对每个结点上问题的不同的测试输出导致不同的分支，最后会到达一个叶子结点，这个过程就是利用决策树进行分类的过程，利用若干个变量来判断所属的类别。

在 Weka 中 J48 的参数为 ，其中 -C 为剪枝的阈值，-M 为叶子上的最小实例数，如果某一个叶子节点小于该值，则判定其为噪声或错误数据将其剪去。

2. Naïve Bayes

贝叶斯分类器的分类原理是通过某对象的先验概率，利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。而朴素贝叶斯分类器在贝叶斯分类器的基础上增加了特征条件独立，通过计算先验概率来得到后验概率（条件概率），也就是某个条件成立的情况下，某个情况发生的可能性，所谓“朴素”就是假设各个属性之间没有关系。

3. SVM

SVM 方法是通过一个非线性映射 p ，把样本空间映射到一个高维乃至无穷维的特征空间中（Hilbert 空间），使得在原来的样本空间中非线性可分的问题转化为在特征空间中的线性可分的问题。

在 Weka 中使用 LibSVM 实现 SVM，参数设置如下，其中 -S 是向量机的种类，-K 是核函数的类型，-D 是核函数中的 degree 设置，-G 是核函数中 gamma 函数设置，-R 是核函数中 coef0 的设置，-N 设置 v-SVC，-M 设置 cache 内存大小，-C 设置 C-SVC，-E 设置允许的终止判据，-P 设置 e-SVR 中损失函数 p 的值。

```
LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model D:\program\install\weka\Weka-3-8 -seed 1
```

4. Neural Network

在 Weka 中用 multilayer perception 作为神经网络的分类器，参数设置如下，其中 -L 是 Weights 被更新的数量，-M 为更新 weights 时的动量，-N 是训练的迭代次数，-V 是 Validation set 的百分比，训练将持续直到其观测到在 validation set 上的误差已经一直在变差，或者训练的时间已经到了，如果 validation set 设置的是 0 那么网络将一直训练直到达到迭代的次数，-S 用于初始化随机数的生成，随机数被用于设定节点之间连接的初始 weights，并且用于 shuffling 训练集，-E 用于终止 validation testing，这个值用于决定在训练终止前在一行内的 validation set error 可以变差多少次，-H 定义神经网络的隐层。

```
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
```

5. KNN

如果一个样本在特征空间中的 k 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。KNN 算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

在 Weka 中 IBK 代表 KNN 算法，其中 -K 为邻居的个数。

```
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A "weka.core.EuclideanDistance -R first-last"
```

6. Bagging

Bagging 的基本思路如下：给定一个弱学习算法和一个训练集，单个弱学习算法准确率不高，将该学习算法使用多次，得出预测函数序列，进行投票，最后结果准确率将得到提高。Bagging 要求不稳定的分类方法，如决策树，神经网络等。

Bagging 算法训练流程如下：从样本集中有放回的抽样 M 个样本；用这 M 个样本训练基分类器 C ；重复这个过程 x 次，得到若干个基分类器。

Bagging 算法的预测流程如下：对于新传入实例 A ，用这 x 个新分类器得到一个分类结果的列表；若待分类属性是数值型（回归），求这个列表的算数平均值作为结果返回；若待分类属性是枚举类型（分类），按这个列表对分类结果进行投票，返回票数最高的。

在 Weka 中 Bagging 参数设置如下，其中-P 是对大小的设置，-S 用于初始化随机数的生成，-I 为迭代次数。

```
Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
```

四. 结果

说明：每个表中第 2,3 行是每个算法的结果，第 4,5 行是每个算法在 Bagging 下的结果。

1. 数据集一：breast-w

breast-w	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	94.5637%	95.9943%	95.7082%	95.2790%	95.1359%
AUC	0.955	0.986	0.964	0.986	0.973
accuracy	96.2804%	95.8512%	95.4220%	95.9943%	95.8512%
AUC	0.985	0.989	0.973	0.989	0.987

2. 数据集二：colic

colic	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	85.3261%	77.9891%	72.5543%	80.4348%	81.2500%
AUC	0.813	0.842	0.670	0.857	0.802
accuracy	85.5978%	77.9891%	69.5652%	84.5109%	81.2500%
AUC	0.864	0.842	0.692	0.876	0.824

3. 数据集三：credit-a

credit-a	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	86.0870%	77.6812%	55.5072%	83.6232%	81.1594%
AUC	0.887	0.896	0.513	0.895	0.808
accuracy	86.8116%	77.8261%	55.7971%	85.0725%	81.3043%
AUC	0.928	0.896	0.535	0.908	0.886

4. 数据集四：credit-g

credit-g	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	70.5000%	75.4000%	68.7000%	71.5000%	72.0000%
AUC	0.639	0.787	0.491	0.730	0.660
accuracy	73.3000%	74.8000%	68.6000%	76.1000%	72.1000%
AUC	0.753	0.787	0.490	0.776	0.694

5. 数据集五：diabetes

diabetes	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	73.8281%	76.3021%	65.1042%	75.3906%	70.1823%
AUC	0.751	0.819	0.500	0.793	0.650
accuracy	74.6094%	76.5625%	65.1042%	76.8229%	71.0938%
AUC	0.798	0.817	0.500	0.822	0.725

6. 数据集六: hepatitis

hepatitis	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	83.8710%	84.5161%	79.3548%	80.0000%	80.6452%
AUC	0.708	0.860	0.500	0.823	0.653
accuracy	83.8710%	85.8065%	79.3548%	84.5161%	81.2903%
AUC	0.865	0.890	0.492	0.846	0.782

7. 数据集七: mozilla4

mozilla4	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	94.7958%	68.6394%	69.5400%	91.1869%	88.9932%
AUC	0.954	0.829	0.537	0.940	0.877
accuracy	95.1110%	68.7424%	69.8231%	91.2834%	88.8582%
AUC	0.976	0.830	0.549	0.945	0.928

8. 数据集八: pc1

pc1	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	93.3273%	89.1794%	93.5077%	93.5978%	92.0649%
AUC	0.668	0.650	0.563	0.723	0.740
accuracy	93.5978%	88.9089%	93.8683%	93.3273%	91.0730%
AUC	0.855	0.628	0.574	0.835	0.793

9. 数据集九: pc5

pc5	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	97.4631%	96.4157%	97.2536%	97.1023%	97.2943%
AUC	0.817	0.833	0.548	0.941	0.932
accuracy	97.5271%	96.4797%	97.2187%	97.3118%	97.3700%
AUC	0.959	0.845	0.552	0.954	0.953

10. 数据集十: waveform-5000

waveform-5000	J4.8	Naïve Bayes	SVM	Neural Network	KNN
accuracy	75.0800%	80.0000%	86.4200%	83.5600%	73.6200%
AUC	0.830	0.956	0.898	0.963	0.802
accuracy	81.2000%	79.9800%	86.0200%	85.6800%	74.4600%
AUC	0.949	0.956	0.939	0.969	0.900

五. 结果分析

J4.8 分类器在这 10 个数据集上分类的精确度最高, 而且所需要的时间很短, 该分类器在样本数据较多的时候容易获得较高的准确度, 但是很不稳定。

朴素贝叶斯分类器分类器较为稳定, 精确度中等, 所需要的时间也较短, 在数据量小或者属性值少的时候也可以提高精确度。

SVM 在测试中精确度最低, 但是理论上 SVM 应该有很高的准确率, 所以对 SVM 的参数设置以及核函数的选择很重要, 如果设置得当应该准确率较高, 但如果设置的不合适就

会导致精确度大大下降。

神经网络分类的精确度较高，接近 J4.8，但是代价是需要花费大量的时间和空间，神经网络类似 J4.8，具有不稳定性，总体来讲数据量增大会提高精确度，但是在数据集 9 中神经网络的精确度较低，所以数据量过大可能导致达不到学习的目的。

KNN 花费的时间空间不大，在测试中精确度较低，但较为稳定，增大样本数量后 KNN 的精确度会有所提升，如第 9 个数据集中 KNN 算法得到的分类器的精确度在所有算法中仅次于 J4.8 分类器。

AUC 所显示的准确度与 Accuracy 不完全相同，因为 AUC 加入了阈值的限制。

Bagging 适用于不稳定的分类算法，如决策树、神经网络等，而 KNN、SVM、Neural Network 都是较稳定的分类算法，所以 Bagging 对它们没有太大提高甚至会有下降，比如在 KNN 算法下，使用 Bagging 算法后只有数据集 7 和数据集 8 得到的精确度略有提升，其它数据集得到的精确度都有所下降。

六. KNN 改进

1. Langley 等人的研究表明，KNN 算法对属性的增减是敏感的，因此可以利用属性的重新可重复抽样来得到不同的训练集，并使用这些训练集训练出来的分类器进行学习。尝试手动实现，但是一直有错未能成功，且数据集处理较为复杂，所以无法用实验验证，在参考文献 2 中论文作者设计实验进行了比较，从给出的结果中可以看出该方法对 KNN 的性能有所提升。
2. 调整 KNN 的 K 值可以提高准确度，但是我认为这是对 KNN 的改进，而不是基于 Bagging 的 KNN 算法的改进。

七. 参考文献

1. <http://www.doc88.com/p-1902125160802.html>
2. <http://www.docin.com/p-940636505-f6.html>
3. <http://blog.csdn.net/davidie/article/details/50434130>
4. 数据挖掘概念与技术 原书第三版