GwMKnn:针对类属性数据加权的 MKnn 算法①

陈雪云[1,2], 郭躬德[1], 陈黎飞[1], 卢伟胜[1]

[1](福建师范大学 数学与计算机科学学院, 福州 350007)
[2](龙岩学院 数学与计算机科学学院, 龙岩 364012)

摘 要: 互 k 近邻 MKnn 算法是 k-近邻算法的一种有效改进算法, 但其对类属性数据通常采用属性值相同为 0, 不同为 1 的方法处理, 从而在类属性数据较多的数据集上分类效率受到一定程度的抑制. 针对 MKnn 对类属性数据处理方法的不足, 对类属性数据的处理引进类别基尼系数的概念, 对同类样本, 用基尼系数统计某一类属性中不同值分布对这个类的贡献度作为此类属性的权重, 并以此作为估算不同样本之间的相似性对 MKnn 进行优化, 扩宽 MKnn 的使用面. 实验结果验证了该方法的有效性.

关键词: 类属性数据; k-近邻; 互 k-近邻; 基尼系数

GwMKnn: MKnn algorithm for Nominal Data by Gini Weight

CHEN Xue-Yun[1,2], GUO Gong-De[1], CHEN Li-Fei[1], LU Wei-Sheng[1]

[1](School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)
[2](School of Mathematics and Computer Science, Longyan University, Longyan 364012, China)

Abstract: MKnn is an improved version of the k-nearest neighbor method, but it uses general approach to deal with nominal data, that is, if its value is the same then to 0, different to 1, thus the classification efficiency is suppressed a certain degree on the data sets with more nominal data. The concept of Category's Gini is introduced in this paper to deal with the shortage of the processing on nominal data, which statistics the contribution of samples in same class by its data distribution for its category and takes it as the attribute weight, used to estimate the similarity for different samples. It aims to optimize the MKnn method and promotes its applications. The experimental results demonstrate the effectiveness of the proposed method.

Key words: nominal data; k-nearest neighbor; mutual k-nearest neighbor ; Gini index

数据挖掘是指从大型数据库中, 挖掘潜在和未知模式的过程. 在过去的二十多年里, 研究人员在数据挖掘的众多领域取得了巨大成功, 其中, 分类是数据挖掘与决策支持领域中最根本的任务, 也是被广泛研究的课题之一. 分类是从已知样本类别的历史数据中通过学习建立模型, 用于预测或标记未知样本类别的过程. 许多传统的分类算法, 如朴素贝叶斯(NBC), k 近邻(Knn), 支持向量机(SVM)和决策树算法(C4.5)已被广泛应用于入侵检测、故障监测、信用卡欺诈分析等领域. 在 2006 年的 ICDM 会议上确认的数据挖掘领域中排名前 10 位最有影响力的挖掘算法中有 6 个是分类算法, 它们分别是 C4.5, SVM, AdaBoost 算法, Knn, NBC 和 CART[1].

k 近邻(Knn)算法以其实现的简单性及较高的分类准确性, 在许多领域得到广泛应用. 然而, Knn 分类器的性能在某种程度上依赖于数据集的样本量 n, 选择距离度量和 k.

近年来, 针对 Knn 相似度度量和 k 值的选取方面, 人们做了深入研究并提出许多卓有成效的改进方法. 例如:张著英等[2]将粗糙集理论应用到 Knn 算法中, 实现属性约简; 陆广泉等[3]用一种自训练的半监督分类模型算法改进 Knn 分类性能; Uchino 等[4]用待测样本

① 基金项目:国家自然科学基金(61070062);福建高校产学合作科技重大项目(2010H6007);福建省教育厅 B 类项目(JB12201)
收稿时间:2013-01-10;收到修改稿时间:2013-03-11

与训练样本向量间的距离加权策略代替多数表决策略; 余鹰等[5]引入变精度粗糙集模型改进 Knn 算法; 杨金福等[6]利用模板约简技术, 去除训练集中远离分类边界的样本; 在相似度度量改进上, Biau 等[7]利用加权版本的 k-近邻密度估计改进 Knn; Chu 和 David [8]用逆密度加权 K-近邻估计相关数据的类别, Guo 等[9]提出的基于 Knn 原理的模型方法较好地解决了传统的 Knn 算法存在的缺陷.

MKnn[10]是一种基于 k-近邻 (Knn) 原理的分类算法. 它克服传统 Knn 分类算法可能存在伪近邻的缺陷. MKnn 通过判断是否互为近邻选择 k 近邻, 以此代替原来用于作数据表决的 k 近邻, 而把训练样本的可能的噪声数据丢弃, 以提高分类的精度. 然而 MKnn 对类属型数据的处理依然采用传统 Knn 的方法, 这在一定程度上限制了它在一些应用领域上的推广.

本文针对 MKnn 在类属性数据处理上的缺陷,引进类别基尼系数的概念来处理类属性数据, 用基尼系数统计某一类属性中不同值分布对这个类的贡献度作为此类属性的权重, 并以此作为估算不同样本之间的相似性度量对 MKnn 进行优化, 扩宽 MKnn 的使用面.

1 背景知识与相关工作

k 最近邻算法[11](k-Nearest Neighbor, Knn)是一个简单, 常用的分类算法. 但它的分类性能由于伪近邻的存在而大打折扣. 伪近邻, 即某个样本离其它样本很远时, 当它的近邻包含另一个样本, 而另一个样本的近邻不包含此样本的情况, 我们称此样本的近邻是它的伪近邻, 这在银行欺诈, 网络分析或入侵检测中通常被认为是离群的, 它将给经济投资或证券带来严重的影响. 伪近邻的存在是影响 Knn 算法分类精度的一个重要因素, 一些研究人员从反向最近邻和外壳最近邻入手展开研究, 例如, Gowda 等[12]获得了简明的基于互近邻表示法的训练集. Jin 等人[13]采用了对称近邻的关系, 需要考虑正向 NNS 和反向 NNS 以及孤立点检测. Gao 等[14]则用互近邻(MNN)去提高移动对象的查询系统的效率. 与此类似, Ding 和 He[15]采用 K-互近邻来提高 K-means 算法的性能, 并探讨用它完成聚类和孤立点检测任务. 最近,Liu 和 Zhang[10]提出的 MKnn, 较好地解决传统的 Knn 可能存在伪近邻的缺陷. MKnn 的基本思想是, 先用传统的方法求得给定样本的 k 近邻后, 再用同样的方法求得其 k 个近邻的 k 近邻. 以判断它们是否互为 k 近邻, 即互 k 近邻. 如果是, 则已知样本的类别将成为未知样本类别的一个候选类别标签. 最后, 通过使用多数表决策略在候选类别标签的基础上确定未知样本的类别标签集. 它考虑到传统 Knn 分类算法学习阶段可能存在伪近邻现象, 消除了异常数据, 剩下更多的真实信息, 加强了最终的预测结果的可信性或真实性. 因此, 用互 k 近邻作为分类依据比用 k 近邻更可取.

然而, MKnn 算法的性能很大程度上依赖于计算出来的 k 互近邻, 而互近邻的计算离不开所使用的相似性度量, 显然, 设计一个较好的处理类属型数据的相似性度量对包含多数类属型属性的数据集来说至关重要. 尽管存在大量针对类属型属性的相似性度量算法[16], 比如, Aha 和 Kibber 等[17]提出了重叠度量(Overlap Metric, OM), 对类属性数据采用属性值相同为 0, 不同为 1 的方法度量距离; Stanfill 和 Waltz[18]提出的值差度量(Value Difference Metric, VDM),利用如果一个属性的两个取值与输出类之间有更相似的关联, 则这两个值被认为靠得更近这个假设为类属性数据提供了适当的距离度量函数; Cost 和 Salzberg 等[19]利用在 VDM 距离基础上采用实例加权方案提出了修正的值差度量(Modified Value Difference Metric, MVDM)并应用于 PEBLS 系统; 为了最小化最近邻分类器在有限样本下的误分类风险和渐近的误分类风险之间的期望差,对最邻近方法下的二分类问题 Short 和 Fukunaga[20]提出了 Short-Fukunaga 度量(Short and Fukunaga Metric, SFM); Blanzieri 和 Ricci[21]直接最小化最近邻分类器在有限样本下的误分类风险提出了最小风险化度量(Minimum Risk Metric, MRM); Liu 和 Pan[22]等将基于熵的度量(Entropy-Based Metric, EBM) 引入到蚁群算法中而获得更快, 更准确的聚类结果; 而 Nie 和 Kambhampati[23]则利用基于频率的方法为经常被访问的属性值建立层并学习统计分类等等. 但这些方法很少同时考虑某一类别的某一类属型属性不同值分布对该类别的影响, 势必在一定程度上抑制了近邻的准确选择.

2 GwMKnn: 针对类属性数据加权MKnn算法

2.1 基本概念与定义

2.1.1 类别基尼系数和类别信息熵

训练集用 $D=\{(X_1,y_1),\cdots,(X_i,y_i),\cdots,(X_n,y_n)\}$ 表示,

$X_i=\{x_{i1},x_{i2},\cdots,x_{id}\}$是一个 d 维输入. y_i表示 X_i的预定义类别, $y_i\in\{y_1,y_2,\ldots,y_q\}$, q 代表训练集中的类别数量, k={1,2,. . . ,q}类中的样本数用 n_k表示. $f_k(x_{jl})$表示在 k 类中第 j ={1,2,⋯,d}个属性上不同取值 x_{jl} l={1,2,⋯, m}出现的次数. X_t为测试样本.

① 类别基尼系数[24]

$$\text{Gini(k, j)} = 1-\sum_{l=1}^{m} P_{jl}^2(k) \tag{1}$$

其中,

$$P_{jl}(k)=\frac{f_k(x_{jl})}{n_k} \tag{2}$$

这里, Gini(k,j)表示训练集 D 中 k 类样本在 j 属性上的基尼系数; $P_{jl}(k)$指 D 中属性 j 上属于 k 类的样本的某种值 x_{jl}出现的概率. 当 $f_k(x_{jl})=n_k$时, Gini(k,j)=0, 而 $P_{jl}(k)$为均值时, 即 $P_{jl}(k)$ =1/q, l=1,2, ⋯ ,m; k=1,2,⋯,q, 则基尼系数 Gini(k,j)=1- $\frac{q\times 1}{q^2}=\frac{q-1}{q}$, 此时值最大, 因此, 它的值域范围是[0, $\frac{q-1}{q}$]. 把基尼系数$1-\sum_{l=1}^{q} P_l^2$的值乘以 $\frac{q}{q-1}$ 之后, 数据样本的每个属性的基尼系数值都被映射到[0,1]区间范围内, 以让所有的基尼系数值相比权重一致而不会影响分类结果.

从类别基尼系数的定义可知, 对训练样本中某个属性 j, 把具有相同类别的数据进行统计: 如果某个类别在属性 j 上对于不同的属性值的基尼系数越大, 则表示属性 j 上属于此类的数据分布就越分散, 则其权重就越大, 样本属于这个类别的可能性就小; 反之, 如果基尼系数越小, 则属性 j 上属于该类的数据分布就越集中, 则其权重就越小, 则样本属于这个类别的可能性就大.

② 类别信息熵[25]

$\text{E(k, j)}= -\sum_{l=1}^{m} P_{jl}(k)\times\log P_{jl}(k)$, 其中 $P_{jl}(k)$计算同上. 当 $f_k(x_{jl})= n_k$时, E(k, j)=0. 对于类别信息熵的规范化, 采用对数 Logistic 模式公式如下:

$$\text{E}' = \frac{1}{1+e^{-E}}$$

2.1.2 伪近邻和互 k 近邻

如图 1 所示, 当 k =3 时样本 X1 的近邻为 X2, X3 和 X6. 同样的, X2, X3 和 X6 的近邻分别是{X3, X4, X5}, {X2, X4, X5}和{X7, X8, X9}. 人们可以观察到一个有趣的现象是当 k =3 时, X1 不是 X2, X3 和 X6 三个中任何一个的近邻, 尽管它把它们三个都作为它的最近邻. 原因是 X1 离它们太远. 从某种程度上说, X2, X3 和 X6 是 X1 的伪近邻.

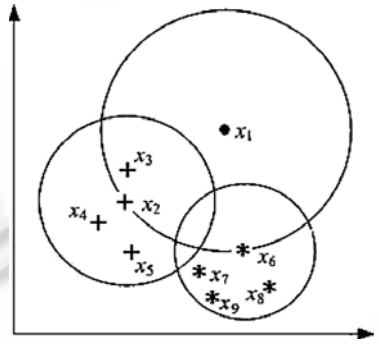

图 1 样本 x1 和它的 3 个近邻

给定一个数据集 D, 参数为 k,则 X_t的互 k 近邻 $M_k(X_t)$是:

$M_k(X_t)=\{X_i\in D|\ X_i\in N_k(X_t)\wedge X_t\in N_k(X_i)\}$

即当 X1 属于另一个样本 X2 的 k 近邻, 而 X2 同时属于 X1 的 k 近邻, 则样本 X1 是 X2 的互 k 近邻. 这里 $N_k(X_t)$是 X_t的 k 近邻的集合.

2.2 GwMKnn 的基本思想

GwMKnn 算法是采用类别基尼系数为权重, 利用类别中不同值出现的概率差作为其相似性度量的 MKnn 方法, 对于待分类样本 X_t, 采用传统的搜索技术从训练集 D 中获得 X_t的 k 近邻集 S1; 随后, 从集合 S1 中确定 X_t的互 k 近邻 S2, 即, 对于每个 X_t的近邻 $y_t\in$S1, 判断 X_t是否也是 y_t的 k 近邻之一; 如果是, y_t就是 X_t的互 k 近邻之一. 最后, 由确定的 X_t的互 k 近邻集 S2 中的标签使用多数表决策略确定 X_t的标签.

2.3 GwMKnn 的算法实现

2.3.1 计算类别基尼系数(Category's Gini, 简称 CG)

算法名: ComputeCG

输入: 训练数据集 D

输出: 第 j 个属性是 Nominal 类型时, 其不同类别的基尼系数 Gini(k, j),k=1,2,⋯,q. j 取所有类属型属性的下标;

Begin

Step1 选取 Ins={D 中类标号为 k 的样本};

Step2 统计 $f_k(x_{jl})$={Ins 中 x_{jl}在第 k 类样本中出现的次数};

统计 n_k={ D 中类标号为 k 的样本的个数};

Step3　根据公式(2)计算 $P_{jl}(k)$ ，即 Ins 中某种值 $f_k(x_{jl})$出现的概率;

Step4　根据公式(1)计算类别基尼系数 Gini(k, j), 并乘以 $\frac{q}{q-1}$ 规范化基尼系数;

Step5　以 Gini(k, j)作为权重返回;

End.

2.3.2 用CompGiniDist计算样本与样本间在j属性上的距离

算法名: CompGiniDist

输入: 样本 X_t 与 X_i, 属性 j, 权重 Gini(k, j)k=1, 2,…,q;

输出: 样本 X_t与样本 X_i在 j 属性上的距离 GiniDist(x_{jt},x_{ji});

Begin

Step1　初始化 GiniDist(x_{jt},x_{ji})=0;

Step2　判断 X_t 在属性 j 上的数据类型. 若数据类型为类属型且样本 X_t 与样本 X_i 在属性 j 上的值不相等, 则用 X_i 所在的类别基尼系数代替, 并使得 GiniDist(x_{jt}, x_{ji})= Gini(y_i, j);

Step3　如果相等, 则 GiniDist(x_{jt},x_{ji})=0;

Step4　如果数据类型为数值型,则 GiniDist(x_{jt},x_{ji})= (x_{jt} - x_{ji})^2;

Step5　返回 GiniDist(x_{jt},x_{ji});

End.

2.3.3 用 GwMKnn 进行分类

算法名: GwMKnn

输入: 训练数据集 D, 样本与样本间在 j 属性上的距离 GiniDist(x_{jt},x_{ji}), 待分类样本 X_t, 近邻数 k;

输出: 预测 X_t的类别 y_t;

Begin

Step1　初始化相关变量, 如, $N_k(X_t)=\phi$, $Y(X_t)=\phi$;

Step2　计算样本 X_t与各训练样本 X_i, i=1,2,…,n 间的距离;

$$dist(X_t, X_i)=\sqrt{\sum_{j=1}^{d} GiniDist\ (x_{jt}, x_{ji})}$$

Step3　根据距离从数据集 D 中找出 X_t的 k 个近邻 $N_k(X_t)$;

Step4　对于 X_t的每个近邻 $X_i \in N_k(X_t)$重复执行下面两个步骤;

Step4.1　从数据集 D 中获得 X_i的 k 近邻 $N_k(X_i)$;

Step4.2　如果 X_t也是 X_i的 k 近邻 $N_k(X_i)$之一则添加 X_i的类别信息 y_i到 $Y(X_t)$;

Step5　统计 $Y(X_t)$中各个类别信息的数目, 并把个数最多的类别标签作为 X_t的类别标签;

Step6　返回 X_t的类别标签 y_t;

End.

3　实验与结果分析

为了验证 GwMKnn 的有效性. 考虑到属性对类别的贡献度也可以用信息熵来衡量, 本文以下用类别信息熵作为相似性度量的 MKnn, 简记为 EwMKnn,作为数据结果比较.

我们选择 Knn、MKnn、EwKnn、EwMKnn、GwKnn 和 GwMKnn 这 6 种基于最近邻思想的分类器作为对比对象. 其中 EwKnn 和 GwKnn 是只把类别信息熵和类别基尼系数应用于 Knn 上处理方法. 各种算法使用的样本间相似度度量都采用欧氏距离. 由于估计 Knn 算法参数 k 的取值较困难, 实验中都设定 k= 5. 另外, 实验采用的计算机配置如下: CPUPentium (R)D CPU 2.40GHZ, 内存 1.0GB, 操作系统 Windows XP. 实验中的所有数据均是在上述配置的计算机上运行取得.

3.1 实验数据

实验采用 10 个数据集对 6 种分类器进行测试, 考虑到本文所用方法主要是针对类属性数据的, 因此所选的 10 个数据集中, 类属型数目比较多的数据集占 8 个, 与数值型属性数目相等的有 1 个, 数值型属性数目比较多的有 1 个. 有关这些数据集的基本信息见表 1. 它们均来自 UCI 机器学习公共数据集[26]. 为了在相同的实验环境中进行比较,我们自己设计编写所有的算法, 感兴趣的读者可以发邮件给作者. 由于使用欧氏距离, 为减少不同取值范围对相似度度量的影响, 所有数据集的类别信息熵和类别基尼系数最后都经标准化处理变换到[0, 1]区间.

3.2 实验结果

实验采用 10-折交叉验证方法, 在选择的 10 个数据集上进行训练、测试. 通过随机抽样将每个数据集均分为 10 个子集, 每次选择其中的 9 个子集为训练数据, 剩余的第 10 个子集为测试数据. 为了更客观的评价分类器的性能, 我们采用 F1-Measure 来进行评价, 并做了 10 次 10-折交叉验证, 结果取均值. F1-Measure 的定义如下[27]:

$$F_\beta = \frac{(\beta^2+1)pr}{\beta^2 p + r}$$

其中, p 是正确率, r 是召回率, β 是调整正确率和召回率在评价函数中所占比重的参数. 通常 β 取 1, 即 F1-Measure,这时评价函数变成

$$F1 = \frac{2pr}{p+r}$$

表 1 实验数据集的基本信息

数据集	实例个数	属性数目	数值型属性数目	类属型属性数目	类别数目	类分布
Balloons	76	4	0	4	2	35：41
Dermatology	366	34	1	33	6	112:61:72:49:52:20
hayes-roth	132	4	0	4	3	51:51:30
House	91	17	0	16	2	54:37
house-votes-84	435	16	0	16	2	168:267
Labor	57	16	8	8	2	20:37
Monks	124	7	0	6	2	62:62
Splice	3190	61	0	61	3	767:768:1655
Voting	435	17	0	16	2	267:168
Vowel	990	14	10	3	11	90:90…:90

微平均(Micro-F1)是对所有的类别计算一个 F1-Measure; 宏平均(Macro-F1)是对每个类别计算一个 F1-Measure, 再对求得的 F1-Measure 进行算术平均即得宏平均. 微平均更多的受分类器对一些常见类分类效果的影响, 而宏平均可以更多的反映对一些特殊类的分类效果. 本实验采用 Micro-F1 和 Macro-F1 指标分别衡量分类器的分类精度. 表 2 为 5 种分类器对每个数据集分别进行分类所取得的分类精度.

表 2 6 种分类器的分类精度对比

数据集	考核方法	Knn	MKnn	EwKnn	EwMKnn	GwKnn	GwMKnn
Balloons	Micro-F1	72.37	72.37	74.87	75.39	73.55	75.66
	Macro-F1	72.36	72.36	74.34	74.99	72.89	75.22
Dermatology	Micro-F1	97.02	95.87	97.84	97.57	96.91	96.39
	Macro-F1	89.61	86.12	92.2	91.33	89.25	87.66
Hayes-roth	Micro-F1	62.2	64.02	70.23	71.36	71.44	73.41
	Macro-F1	47.36	50.24	59.74	62.18	61.34	64.69
House	Micro-F1	94.07	96.37	95.6	96.81	92.64	94.29
	Macro-F1	93.91	96.25	95.45	96.68	92.5	94.15
House-votes-84	Micro-F1	92.53	94.28	93.89	94.05	88.6	86.64
	Macro-F1	92.23	93.98	93.56	93.69	88.35	86.34
Labor	Micro-F1	87.89	89.65	86.49	87.89	89.65	92.28
	Macro-F1	86.54	88.82	85.94	87.49	88.7	91.83
Monks	Micro-F1	77.18	77.42	70.56	73.63	78.71	81.69
	Macro-F1	77.16	77.39	70.38	73.57	78.55	81.66
Splice	micro-F1	81.6	85.83	84.54	85.6	79.56	85.03
	macro-F1	73.86	78.63	74.24	76.14	71.39	77.49
Voting	Micro-F1	92.92	94.14	94.6	94.37	94.11	94.07
	macro-F1	92.65	93.84	94.31	94.07	93.85	93.75
Vowel	Micro-F1	93.84	97.53	94.52	97.69	94.1	97.54
	macro-F1	73.43	87.75	75.77	88.47	74.32	87.79

如表 2 所示, 在数据集 vowel 中, 虽然类属型属性数目只有 3 个, 而数值型属性数目有 10 个之多, 但因为数据在这 3 个属性中属于某个类别的元素相对集中, 而使得其在 EwKnn 和 GwKnn 上的性能能得到较大的提升, 再加上互 k 近邻的引入而使得 EwMKnn 和 GwMKnn 上的精度要远远高于 Knn. 对于类属型属性数目与数值型属性数目一样多的 labor 数据集而言, 因为类属型属性数目占一半的比重, 加上互 k 近邻的影响而使得后面四种分类器的性能都高过 Knn. 从总体上看, 在 10 个数据集的平均结果中, MKnn、EwKnn、EwMKnn、GwKnn 和 GwMKnn 的性能都高过 Knn, 而这其中尤其以 GwMKnn 最高.

4 结束语

文中提出针对类属性数据加权的互 k 近邻学习算法(GwMKnn), 通过对类属性数据引进类别基尼系数的概念, 用基尼系数统计类属性数据中某个元素对这个类别的贡献度作为此类的权重, 并以此作为估算不同样本之间的相似性度量方法对 MKnn 进行优化, 以扩宽 MKnn 的使用面. 在 10 个 UCI 机器学习公共数据集上的实验结果表明, GwMKnn 可有效进行互 k 近邻分类, 其分类精度优于 MKnn 算法, 更优于传统的 Knn 算法及其本文所提的其它改进算法, 并大幅提升分类效率. 下一步的工作重点是深入分析影响模型期望风险的其它因素, 提出更有效的优化方法, 以进一步提高分类性能.

参考文献

1 Wu X, Kumar V, Quinlan J, Ghosh J, Yang Q, Motoda H, McLach-lan G., Ng A, Liu B, Yu P, Zhou Z, Steinbach M, Hand D, Stein-berg D. Top 10 algorithms in data mining. Knowledge and Infor-mation Systems,2008,14:1–37.

2 张著英,黄玉龙,王翰虎.一个高效的 KNN 分类算法.计算机科学,2008,35(3):170–172.

3 陆广泉,谢扬才,刘星,张师超.一种基于 KNN 的半监督分类改进算法.广西师范大学学报(自然科学版),2012,30(1): 45–49.

4 Uchino E, Tokunaga K, Tanaka H, Suetake N. IVUS-Based Co-ronary Plaque Tissue Characterization Using Weighted Multi-ple k-Nearest Neighbor.Engineering Letters,2012,20: Advance online publication.

5 余鹰,苗夺谦,刘财辉,王磊.基于变精度粗糙集的 KNN 分类改进算法.模式识别与人工智能,2012,25(4):617–623.

6 杨金福,宋敏,李明爱.一种新的基于距离加权的模板约简 K 近邻算法.电子与信息学报,2011,33(10):2378–2383.

7 Biau G, Chazal F, David C, Devroye L, Rodr´ıguez C. A Weighted k-Nearest Neighbor DensityEstimate for Geometric Inference. Electron.J.Statist, 2011,5:204–237.

8 Chu B, David T. k nearest-neighbor estimation of inverse density eighted expectations. Economics Bulletin, 2008, 3(48):1–6.

9 Guo G, Wang H, Bell D. kNN Model-based Approach in Classification. Proc. of ODBASE'03, Lecture Notes in Computer Science, 2003, LNCS2888: 986–996.

10 Liu H, Zhang S. Noisy data elimination using mutual k-nearest neighbor for classification mining. The Journal of Systems and Software, 2012,85:1067–1074.

11 Cover TM, Hart PE. Nearest Neighbor Pattern Classific. IEEE Trans on Information Theory, 1967,13(1):21–27.

12 Gowda KC, Krishna G. The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. IEEE Transactions on Information Theory,1979,25(4):488–490.

13 Jin W, Tung AKH, Han J, Wang W. Ranking outliers using symmetric neighborhood relationship. Proc. of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining(PAKDD), 2006:577–593.

14 Gao Y, Zheng B, Chen G, Li Q, Chen C, Chen G. Efficient mutual nearest neighbor query processing for moving object trajectories. Information Sciences, 2010,180: 2170–2195.

15 Ding CHQ, He X. K-nearest-neighbor consistency in data clustering: incorporating local information into global opti-mization. Proc. of ACM Symposium on Applied Comput-ing(SAC), 2004: 584–589.

16 Huang Y, Guo GD, Neagu D. A Partial Coverage Based Approach to Classification. Proc. of ICTAI(1) 2007: 275–280.

17 Aha DW, Kibber D, Albert MK. Instance-Based Learning Algorithms. Machine Learning,1991,6(1):37–66.

18 Stanfill C, Waltz D. Toward memory-based reasoning. Communications of the ACM,1986,12(29):1213–1228.

19 Cost S, Salzberg S. A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features.Machine Learning,

(下转第 158 页)

程序编译进 U-Boot.

修改 U-Boot 目录下的 Makefile 文件, 在第 1882 行处模仿 smdk2410_config 目标增加新目标 mini2440_config.

mini2440_config : unconfig

@$(MKCONFIG) $(@:_config=) arm arm920t mini2440 NULL s3c24x0

3.3 u-boot.bin 文件的生成与测试

从终端进入到 u-boot-1.1.6 目录后, 依次执行以下命令:

make mini2440_config

make

等待几分钟后, 会在 u-boot-1.1.6 目录下生成 u-boot.bin 文件. 使用 H-JTAG 软件将开发板的 Nor Flash 全部擦除, 然后将 u-boot.bin 烧写到 Nor Flash 中. 烧写成功后, 复位开发板, 可以从超级终端中看到如图 2 所示的信息.

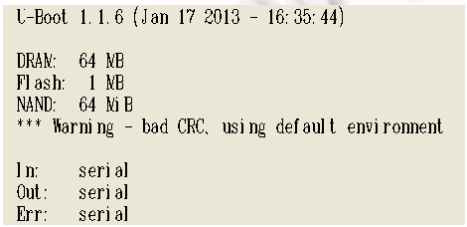

图 2 U-Boot 成功启动后超级终端的输出信息

超级终端输出的信息表明, CPU 和串口能够正常工作, 此时可以按下任意键进入 U-Boot 控制界面如图 3 所示. 在控制界面用 U-Boot 的相关命令测试其功能, 测试结果表明 U-Boot 可以在 mini2440 上稳定地运行, 并能实现其功能. 至此, U-Boot 的移植工作结束.

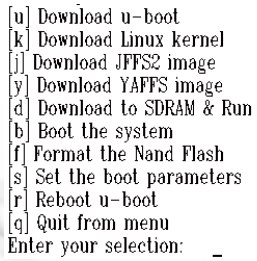

图 3 U-Boot 控制界面

4 结语

利用上述的 U-Boot 移植方法, 可以使 U-Boot 稳定地运行在以 S3C2440 为处理器的 mini2440 开发板上, 并且能够实现利用 U-Boot 下载内核、加载根文件系统等功能, 最终可以正确引导 Linux 内核启动, 为后续嵌入式系统开发打下了坚实的基础.

参考文献

1 韦东山.嵌入式 Linux 应用开发完全手册.北京:人民邮电出版社,2008.22–30.

2 田泽.嵌入式系统开发原理与实践.北京:清华大学出版社,2005.38–62.

3 友善之臂.MINI2440 用户手册.友善之臂,2009.10–12.

4 詹荣开.嵌入式系统 BootLoader 技术内幕. [2012-12-18]. http://www.ibm.com/developerworks/cn/linux/l-btloader/

5 孙弋,等.ARM-Linux 嵌入式系统开发基础.西安:西安电子科技大学出版社,2008.102–115.

(上接第 108 页)

1993(10):57–78.

20 Short RD, Fukunaga K. The optimal distance measure for nearest neighbour classification. IEEE Trans.Inform.Theory, 1981(27),622–627.

21 Blanzieri E, Ricci F. Probability Based Metrics for Nearest Neighbor Classification and Case-Based Reasoning. Proc. of the 3rd International Conference on Case-Based Reasoning Research and Development (ICCBR'99),1999: 14–28.

22 Liu B, Pan J, McKay RI. Entropy-based metrics in swarm clustering. International Journal of Intelligent Systems, 2009(24): 989–1011.

23 Nie Z, Kambhampati S. A Frequency-based Approach for Mining Coverage Statistics in Data Integration.http://www.public.asu.edu/~zaiqingn/freqbased.pdf.

24 Samet S, Miri A. Privacy preserving ID3 using Gini Index over horizontally partitioned data. International Conference on Computer Systems and Applications,2008(1-3):645–651.

25 Li R, Tao X, Tang L, Hu Y. Using Maximum Entropy Model for Chinese Text Categorization. Computer Science, 2004, 3007: 578–587.

26 UCI Repository of Machine Learning Databases. [2012-12-12]. http://repository.seasr.org/Datasets/UCI/arff/

27 郭躬德,黄杰,陈黎飞.基于 KNN 模型的增量学习算法.模式识别与人工智能,2010,5:86–94.