

# 慢性肾病辅助诊断系统的设计与实现

宋波, 宋同峰

(青岛科技大学 信息科学技术学院, 山东 青岛 266000)

**摘要:** 针对慢性肾病在全球占比之高、病情发展不可逆转、病情极易出现恶化的特点, 设计了慢性肾病辅助诊断系统; 利用数据挖掘随机森林算法的分类功能对病人化验数据进行处理, 判断病人是否患有慢性肾病; 设计并开发了基于 B/S 的慢性肾病辅助诊断系统, 该系统集慢性肾病辅助诊断、诊断信息查看、用户管理于一体; 该系统用于给经验不足的医生提供诊断参考, 助其提高诊断水平, 降低误诊率, 从而使慢性肾病患者尽早进行正确的治疗, 避免病情治疗延误带来的严重后果。

**关键词:** 慢性肾病; 辅助诊断系统; 数据挖掘; 随机森林; B/S

## Design and Implementation of an Assistant Diagnostic System for Chronic Nephropathy

Song Bo, Song Tongfeng

(College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266000, China)

**Abstract:** Aiming at the high proportion of chronic kidney disease in the world, the irreversible development of the disease and the easy deterioration of the disease, an auxiliary diagnosis system for chronic kidney disease was designed. The classification function of random forest algorithm of data mining is used to process the patient's laboratory data to determine whether the patient has chronic kidney disease. A B/S-based assistant diagnosis system for chronic nephropathy is designed and developed. The system integrates assistant diagnosis, diagnosis information viewing and user management of chronic nephropathy. The system is used to provide diagnostic reference for inexperienced doctors, help them improve the level of diagnosis, reduce the rate of misdiagnosis, so that patients with chronic kidney disease can be treated correctly as soon as possible, and avoid the serious consequences of delayed treatment.

**Keywords:** chronic kidney disease; assistant diagnosis system; data mining; random forest; B/S

## 0 引言

根据国际肾病协会、欧洲肾病学会—欧洲透析和移植学会、美国肾病学会估计, 全球至少有 8.5 亿人患有肾病, 占世界总人口的 11%<sup>[1]</sup>。在 2017 年全球发布的肾脏地图中, 仅慢性肾病每 10 个人就有一个人是患者, 由此可见慢性肾病占比之高。而慢性肾病的发展不可逆转, 因而患者只能通过长期药物治疗来控制病情发展。若患者不尽早接受治疗, 极易出现病情恶化导致终末期肾脏病<sup>[2]</sup>。因此尽早进行诊断慢性肾病对于后期治疗具有重大意义和价值。

随着医疗信息化建设的推进, 医疗数据量正在急剧膨胀, 导致数据挖掘技术在医疗领域的应用也更加广泛。目前数据挖掘应用于医疗领域主要是用来预防、诊断、治疗和预后。应用于预防是通过寻找疾病的诱因因子以及通过疾病早期筛查来预防疾病。应用于诊断是利用电子病历文本信息或图像信息辅助<sup>[3]</sup>医生对疾病进行诊断。应用于治疗是通过关联规则等算法挖掘出药物对疾病的治疗作用并且能制定临床路径, 方便医生开处方。应用于预后通过对

原始数据的分析, 能辅助预测患者手术后的身体指标状况, 从而起到预警作用<sup>[4]</sup>。本文研究的慢性肾病辅助诊断是数据挖掘应用于医疗诊断的体现, 并通过利用慢性肾病数据建立随机森林分类模型对慢性肾病进行分类来建立慢性肾病辅助诊断系统。

## 1 系统总体框架

本系统的主要工作是慢性肾病的辅助诊断, 对于该系统的主要流程就是将患者的化验信息以及患者的身份信息输入到该系统中, 通过已经训练好的慢性肾病诊断模型, 让该模型对该患者进行诊断, 判断是否患有慢性肾病。本系统采用 B/S 架构<sup>[5]</sup>进行设计, 分为显示层、业务层、服务层和数据层, 系统架构如图 1 所示。

按图 1 所示, 表示层是系统用于界面交互的一层, 其重要利用 HTML 和 js 进行编写, 用 jsp 辅助传值, 实现数据交互; 业务层采用 SSM 框架<sup>[6]</sup>进行设计, 用于实现系统的功能逻辑; 服务层利用 tomcat 服务器进行项目部署工作; 数据层采用 oracle 数据库对数据进行存储和处理。

## 2 系统设计

### 2.1 系统功能设计

本系统的肾病辅助诊断功能如图 2 所示。

由图 2 可以看出, 本系统在功能上应具备以下几个功能模块。

收稿日期: 2019-04-03; 修回日期: 2019-04-20。

基金项目: 国家自然科学基金(61572268, 61303193, 61402246); 山东省重点研发计划项目(2017GSF18110, 2018GGX101029)。

作者简介: 宋波(1978-), 男, 山东青岛人, 博士, 教授, 主要从事领域为智能方法与软件开发技术方面的研究。

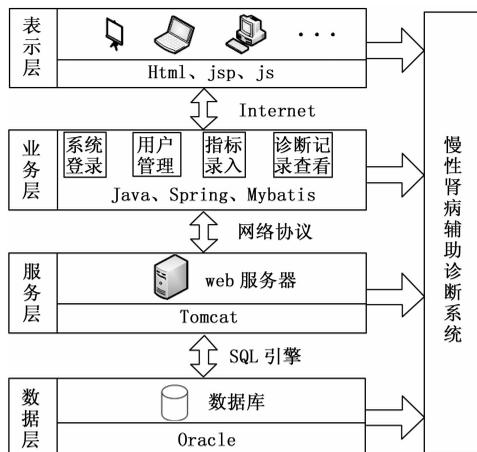


图 1 系统总体架构图

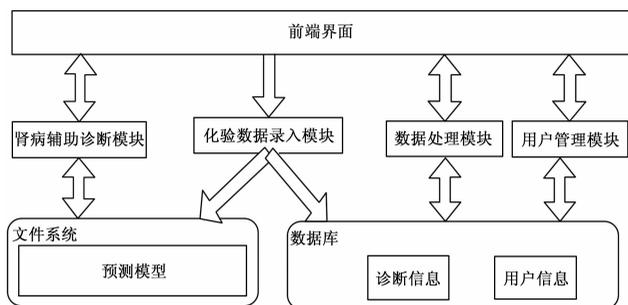


图 2 肾病辅助诊断系统框架图

### 2.1.1 化验数据录入模块

该模块的功能是将病人的基本信息和病人的本次化验数据录入系统, 提交诊断后进入后台由肾病辅助诊断模型进行数据处理和慢性肾病诊断, 最后将输出结果 0, 1 转换成是否患有慢性肾病并输出 (0 代表病人未患慢性肾病, 1 代表病人诊断为慢性肾病), 并即时将结果显示在页面醒目的位置。

### 2.1.2 肾病辅助诊断模块

该模块应具备两部分功能: 1) 是肾病诊断模型构建功能, 该模块利用原始数据对慢性肾病辅助诊断模型进行构建。首先将原始数据集分为训练数据集和测试数据集, 然后利用训练集训练慢性肾病辅助诊断分类模型, 利用测试集测试训练后的模型性能, 根据测试结果调节模型参数, 以使模型达到最佳性能; 2) 是利用医生输入进来的基本信息和化验项目参数封装成实例对象通过慢性肾病辅助诊断模型进行诊断, 并将诊断结果返回。

### 2.1.3 数据处理模块

该模块应具备的主要功能有: 存储诊断结果及化验项目信息的功能、能够即时查看患者历史诊断记录的功能、提供患者诊断记录增、删、改的功能。

### 2.1.4 用户管理模块

该模块的主要功能是提供添加、删除、更改用户信息的功能及密码修改重置功能。

## 2.2 系统模块设计

### 2.2.1 化验指标录入模块

该录入模块有病人基本信息 (如化验单号、姓名、年龄、性别等) 5 项, 病人化验指标 (如血压、比重、白蛋白、血尿素、血清肌酐等) 23 项, 用 v1~v23 表示。其中只有 v1~v23 作为模型训练的特征属性被使用, 病人基本信息只是为了方便医生查找和数据库记录而被添加的。23 项特征属性如表 1 所示。

表 1 慢性肾病辅助诊断模型特征属性

23 项特征属性	血压比重、白蛋白、葡萄糖、红细胞、白细胞、白细胞团、细菌、血糖随机、血尿素、血清肌酐、钠、钾、血红蛋白、包装细胞体积、白细胞计数、红细胞计数、高血压、糖尿病、冠心病、食欲不振、脚蹬水肿、贫血
----------	---

由于纯手工录入这 28 个指标会耗时较长, 为了解决耗时问题, 本实验设计了自动填充功能, 利用 AJAX<sup>[7]</sup> 传值如下:

```

/*
  自动填充病人信息
*/
function fillInfo(keyword, keydate) {
  .ajax({
    url: '{sysPath}patient/patientInfoSearch.do',
    type: 'POST', //GET
    async: true, //或 false, 是否异步
    data: {
      keyword: keyword,
      keydate: keydate
    },
    //timeout: 5000, //超时时间
    dataType: 'json', //返回的数据格式: json/xml/html/script/
    jsonp/text
    success: function(data, textStatus, jqXHR) {
      (" sampleNo").val(data[0]. sampleNo); //sampleNo, 化
      验单号
      (" patientId").val(data[0]. patientId);
      (" patientName").val(data[0]. patientName);
      (" sex").val(data[0]. sex);
      (" age").val(data[0]. age);
      (" v1").val(data[0]. v1);
      (" v2").val(data[0]. v2);
      (" v3").val(data[0]. v3);
      (" v4").val(data[0]. v4);
      (" v5").val(data[0]. v5);
      (" v6").val(data[0]. v6);
      ..... //省略部分为 v7~v23 的填充项
    },
    error: function(xhr, textStatus) {
      alert("系统错误");
    },
  });
}

```

```
complete,function(){
console.log(结束)
}
});
}
```

### 2.2.2 慢性肾病辅助诊断模块

此模块主要功能是利用输入的化验指标判断该患者是否患有慢性肾病。因此该诊断模块需要数据建模。本次慢性肾病辅助诊断建模利用随机森林分类算法的建模方案如下：

1) 随机森林中决策树的数目  $n$  是一个重要参数，需要根据设置数目的模型准确度反复调节，一般设置为 100~1000，多次调节选取最优参数；

2) 将慢性肾病数据按是否慢性肾病分别 1:1 的比例划分出训练集和测试集，将肾病训练数据进行有放回的随机抽样，抽取  $n$  次，得到  $n$  个样本，每个肾病训练数据样本容量约为总样本数的 2/3；

3) 属性子集的选取：慢性肾病数据集的特征属性有血压、比重、白蛋白、血尿素、血清肌酐等共 23 个，设每次随机选取的属性子集中属性的个数为  $x$ ， $\log_2 23$  约为 4.5，取 4，以 4 为属性子集的属性个数最小值，分别取 4, 5, 6, 7, 8, 9, 10 作为每次随机选取的属性子集中属性的个数。以  $x=4$  为例，随机抽取  $n$  个特征数目为 4 的特征子集进行建模实验，用慢性肾病测试数据测试建模准确率； $x=5$  时也进行相同实验，将所有  $x$  不同取值的模型准确率进行对比，选择最优  $x$  的取值；

4) 将  $n$  个样本和  $n$  个属性子集一对一组成  $n$  个决策树；

5) 将肾病测试样本实例输入到  $n$  个决策树中，以投票法确认该测试实例的肾病类别即为随机森林模型测得的肾病类别。

利用随机森林算法建立慢性肾病辅助诊断模型的建模流程如图 3 所示。

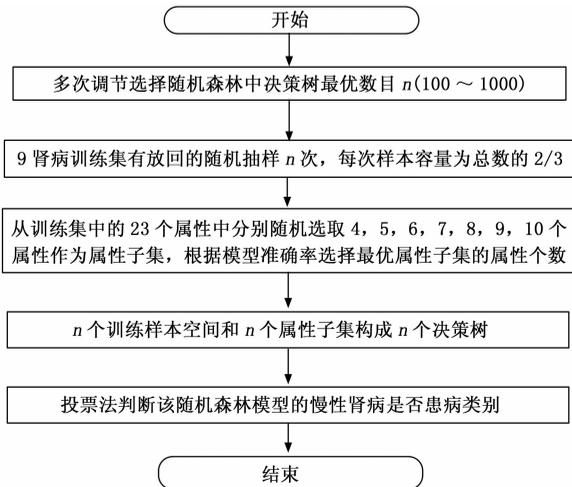


图 3 慢性肾病辅助诊断的随机森林建模流程图

本实验按照上述随机森林的肾病辅助诊断建模流程，

利用 weka<sup>[8]</sup> 包里的随机森林 (randomForest) 类建立分类模型，模型的训练数据和测试数据来源于机器学习库 UCI 的慢性肾病<sup>[9]</sup>数据集。该数据集共 400 条数据、24 个特征属性，2 个诊断类别。本文将其分为 200 条训练数据，200 条测试数据对随机森林模型进行训练<sup>[10]</sup>。模型训练测试设计如下：

```
public static void main(String[] args) throws Exception {
    Classifier m_classifier = new WriteRandomForest();
    File inputFile = new File("C:\Users\Administrator\Desktop\uci\Chronic_Kidney_Disease\train.csv.arff");
    ArffLoader atf = new ArffLoader();
    atf.setFile(inputFile);
    Instances instancesTrain = atf.getDataSet(); // 读入训练文件
    inputFile = new File("C:\Users\Administrator\Desktop\uci\Chronic_Kidney_Disease\test.csv.arff");//A 类下的测试语料文件
    atf.setFile(inputFile);
    Instances instancesTest = atf.getDataSet(); // 读入测试文件
    instancesTest.setClassIndex(instancesTest.numAttributes()-1); //设置最后一行为分类类别行。instancesTest.numAttributes()可以取得属性总数
    double sum = instancesTest.numInstances(),//取实例数
    right = 0.0f;
    instancesTrain.setClassIndex(instancesTest.numAttributes()-1);
    m_classifier.buildClassifier(instancesTrain); //训练
    System.out.println(m_classifier);
    ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("C:\Users\Administrator\Desktop\1\RandomForest.model"));
    oos.writeObject(m_classifier);
    oos.flush();
    oos.close();
    // 保存模型
    SerializationHelper.write("RandomForest.model", m_classifier); //参数一为模型保存文件,classifier4 为要保存的模型
    for(int i = 0;i<sum;i++)//测试分类结果
    { System.out.println(m_classifier.classifyInstance(instancesTest.instance(i)) + "," + instancesTest.instance(i).classValue());
    if(m_classifier.classifyInstance(instancesTest.instance(i)) == instancesTest.instance(i).classValue())
    {
        right++; //正确值加 1
    }
    }
    // 获取上面保存的模型
    Classifier classifier8 = (Classifier) weka.core.SerializationHelper.read("RandomForest.model");
}
```

```

double right2 = 0.0f;
for(int i = 0;i<sum;i++)//测试分类结果 2 (通过)
{
    if(classifier8. classifyInstance( instancesTest. instance
(i)) == instancesTest. instance(i). classValue())
    {
        right2++; //正确值加 1
    }
}

Evaluation eval = new Evaluation(instancesTrain); //构造
评价器
eval. evaluateModel(m_ classifier, instancesTest); //用测试
数据集来评价 m_ classifier
System. out. println( eval. toClassDetailsString( " == =
ClassDetailsString==="));
System. out. println( eval. toSummaryString( " == = Sum-
mary ===", false));
//输出信息
System. out. println( eval. toMatrixString( " == = Confusion
Matrix ===")); //Confusion Matrix
}

```

该随机森林模型的主要参数是  $m\_numTrees$  和  $m\_KValue$ 。其中  $m\_numTrees$  代表的是随机森林模型中树的数目,  $KValue$  代表的是每个结点建造时选择随机特征的数量。本实验默认树数目 100, 默认随机特征数量为  $\log_2(\text{data. numAttributes}() - 1) + 1$ 。为了确定最优参数, 本文进行了多次实验, 将树的数量分别设置为 100, 200, 300, 400, 500, 600, 700, 800, 所得结果以折线图的形式呈现, 见图 4。在树的数量取到最优的情况下, 随机特征的数量设置为 4, 5, 6, 7, 8, 9, 10, 所得结果见图 5。

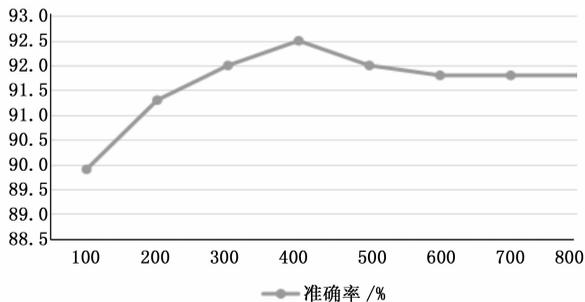


图 4 随机森林建模参数对照图 1

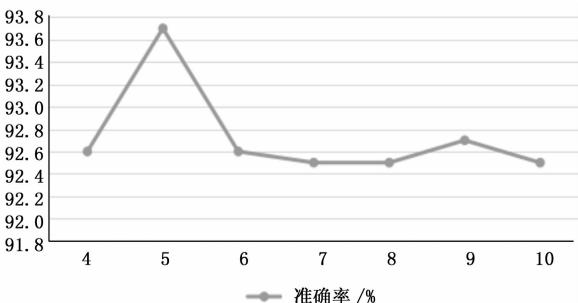


图 5 随机森林建模参数对照图 2

由以上两折线图可以看出, 当树的数量取 400 时, 分类准确率达到最优, 此时准确率为 92.5%。在树数量取 400 的前提下, 随机特征数量为 5 时模型达到最大值, 准确率为 93.7%。因此当  $m\_numTrees = 400$  且  $m\_KValue = 5$  时, 本次慢性肾病辅助诊断实验的随机森林模型达到最优准确率, 为 93.7%。本次慢性肾病辅助诊断系统设计将应用最优参数进行系统设计实现。

### 2.2.3 数据处理模块

该模块主要有两个功能: (1) 是让医生可以查看病人诊断历史记录, 可以通过病人 ID、病人姓名查看某患者在该系统的所有历史诊断记录, 通过化验单编号可查患者唯一记录。在数据列表展示区点击化验单编号可以看到患者此次诊断的详细化验信息和诊断结果信息, 并且具备修改和删除以及导出功能; (2) 是数据库设计<sup>[11]</sup>, 本文数据库表结构如图 6 所示。

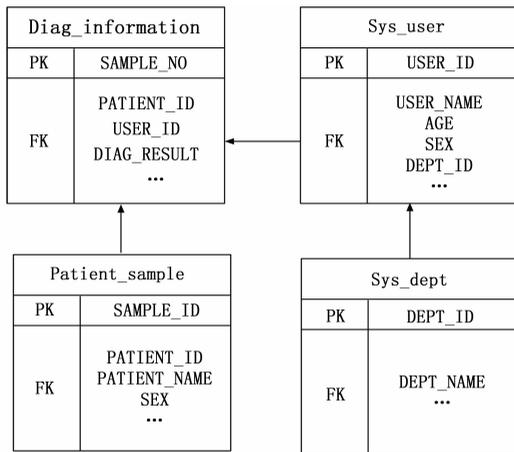


图 6 肾病辅助诊断系统数据库结构

用户管理及密码修改模块: 用户管理模块为仅系统管理员可见模块, 该模块主要有用户添加、编辑功能、用户科室配置功能、用户修改密码功能。修改密码功能通过输入原密码、新密码和确认新密码后, 点击确定修改即可。该部分有原密码校验功能、新密码和确认新密码是否一致功能。密码校验功能代码如下:

```

/* *
 * 校验新旧密码
 */
function checkContent() {
    var oldPWD = . trim((" oldPWD"). val()); //旧密码
    var newPWD1 = . trim((" newPWD1"). val()); //新密码
    var newPWD2 = . trim((" newPWD2"). val()); //确认新
密码

    if (oldPWD == null || oldPWD == undefined || oldPWD
== "") {
        alert("旧密码不能为空!");
        return false;
    }
}

```

```

if (newPWD1 == null || newPWD1 == undefined || newPWD1 == "") {
    alert("新密码不能为空!");
    return false;
}
if (newPWD2 == null || newPWD2 == undefined || newPWD2 == "") {
    alert("确认新密码不能为空!");
    return false;
}
if (oldPWD == newPWD1) {
    alert("新密码与原密码一致!");
    return false;
}
if (newPWD1 != newPWD2) {
    alert("新密码与确认密码不一致!");
    return false;
}
}

```

### 3 系统效果

#### 3.1 系统登录模块

该模块在用户通过正确输入用户名和密码，登录成功后进入系统界面如图 7 所示。



图 7 肾病辅助诊断系统主页

从主界面上看该系统一共分为 3 部分，分别是指标录入平台、选择查询平台以及系统帮助平台。指标录入平台 and 选择查询平台分别是对患者化验项目指标的录入和对已经诊断的患者记录进行查询。系统帮助平台的本地用户管理仅系统管理员可见，修改密码功能普通用户可使用。

#### 3.2 肾病辅助诊断指标录入及诊断模块

指标录入模块是医生用来将患者的基本信息和化验指标输入到系统中，该部分利用了 AJAX 传值通过输入患者化验单号后使用自动填充功能，无需医生再输入患者的其他信息就可以将该患者的所有信息自动填充到页面中。

慢性肾病辅助诊断指标录入页面如图 8 所示。

该模块通过录入病人基本信息及化验结果后点击“诊断”按钮，数据就会被后台诊断功能接收并处理，并将诊断结果以 JS 弹框输出，方便医生查看辅助诊断结果。同时，

肾病辅助诊断指标录入			
患者ID: 0001	化验单号: 20190522CH006	患者姓名: 张三	患者性别: 男
患者年龄: 23	白蛋白: 0	血压: 50	葡萄糖: 0
比重: 1.01	白蛋白: abnormal	白蛋白: present	
红细胞: abnormal	血肌酐: 268	血压表: 96	
白细胞: present	钙: 134	钾: 5.1	
血尿酸: 4	白蛋白总量: 29	白蛋白计数: 15200	
血红蛋白: 10	高血压: yes	糖尿病: yes	
红细胞计数: 4.3	食欲下降: yes	肾功能: yes	

图 8 指标录入模块页面

诊断数据和诊断医生也被保存到数据库中，方便医生查看历史诊断记录。

#### 3.3 肾病诊断记录查询模块

该模块主要为用作医生查看病人诊断历史记录，可以通过病人 ID、病人姓名查看某患者在该系统的所有历史诊断记录，通过化验单编号可查患者唯一记录。在数据列表展示区点击化验单编号可以看到患者此次诊断的详细化验信息和诊断结果信息，并且具备修改和删除以及导出功能。该模块如图 9 所示。



图 9 诊断结果查询页面

#### 3.4 用户管理及密码修改模块

用户管理模块为仅系统管理员可见模块，该模块主要有用户添加、编辑功能、用户科室配置功能、用户重置密码功能。

用户科室配置功能页面如图 10 所示，该功能在用户点击进入该页面时先从 session 中调取用户的科室，然后查询除用户所在科室以外的所有科室，将结果放入“未在科室”的控件中。通过“>>”可将用户“所在科室”删掉，并且“未在科室”会增加所删内容；通过“<<”可将“未在科室”中选中的内容放入到用户“已在科室”，即添加用户“已在科室”，同时“未在科室”作相应删减。配置科室功能是为以后研究其他疾病的辅助诊断功能预留了空间<sup>[12]</sup>。

### 4 结束语

本文利用数据挖掘的随机森林分类技术和 web 开发技术<sup>[13]</sup>构建了慢性肾病辅助诊断系统<sup>[14]</sup>。该系统通过对病患的基本信息数据和化验指标数据的录入，利用数据挖掘随机森林分类模型对患者是否患有慢性肾病进行分类，并将

(下转第 59 页)