

基于 OGC WPS 的 TVDI 算法实现与共享

张 钊 陈家赢 黄 魏

华中农业大学资源与环境学院, 武汉 430070

摘要 针对温度植被干旱指数(temperature vegetation dryness index, TVDI)计算复杂和重复实现等问题, 采用 Python 语言实现了 TVDI 模型算法, 并利用开源的网络地理信息系统(Web GIS)平台 52°North 包装成符合开放地理空间信息联盟(Open Geospatial Consortium, OGC)网络处理服务(WPS)规范的 WPS, 在网络上共享了基于 TVDI 算法的 WPS。以 2010 年湖北省中分辨率成像光谱仪(moderate-resolution imaging spectroradiometer, MODIS)数据为例, 对算法进行测试。结果表明, 干湿边线性拟合整体效果较好, 算法效率较高且运行稳定。服务的发布共享避免了科研中重复实现 TVDI 算法, 有利于研究人员专注于干湿状况的分析研究, 提高研究效率。

关键词 温度植被干旱指数; 网络处理服务; OGC; MODIS; 52°North

中图分类号 S 159.9 **文献标识码** A **文章编号** 1000-2421(2015)05-0063-07

土壤干湿状况是监测土地状况的重要指标之一, 是作物生长的决定性因素, 在水文、气候等领域都有广泛应用。近年来, 遥感技术迅猛发展, 遥感影像能够反映多时相、多光谱、高空间分辨率和大范围的地表信息, 这为大面积土壤水分实时、动态监测奠定了坚实的基础, 并已成为当前干旱监测的重要数据来源和技术手段^[1-2]。遥感能够方便快速地获取大尺度的地表信息, 因此, 在下垫面数据缺乏的情况下, 使用遥感进行大尺度区域旱情监测就显得尤为重要。温度植被干旱指数(temperature vegetation dryness index, TVDI)是典型的遥感干旱监测指数, 能通过反演土壤湿度来反映农业干旱状况。自 Sandholt 等^[3]利用简化的 NDVI-Ts 特征空间提出了温度植被旱情指数来估算土壤表层水分状况起, TVDI 在土壤水分和区域干旱遥感监测方面有了广泛的应用^[4-6]。

随着空间信息数据在共享、交换和用途方面的不断推进, 处理功能的共享与互操作也更加备受关注。在空间信息共享以及处理互操作方面, Web Service 提供了一个开放的平台, 通过网络访问服务资源, 所有的处理功能都可以以 Web Service 的方式通过网络提供给用户^[7]。Web Service 是一种分

布在网络上面向服务的架构技术, 通过标准的 XML 信息格式提供诸如空间信息分析这些可以通过网络访问的处理操作。但是通用的 Web Service 标准, 不能很好地解决具体某一学科领域的专业问题。例如, 对于地理空间信息, 标准的 Web Service 传输协议中没有包含空间信息元数据信息以及对空间信息数据的标准化, 使得其在解决 GIS 功能互操作方面存在不足。针对 Web Service 的不足以及日益增加的基于网络的空间数据处理需求, 开放地理空间信息联盟(Open Geospatial Consortium, OGC)制定了一个 Web 服务标准网络处理服务(web processing service, WPS), 该标准可以包装任何类型的对空间数据进行处理的功能^[8]。笔者以 2010 年湖北省中分辨率成像光谱仪(moderate-resolution imaging spectroradiometer, MODIS)遥感影像作为实验数据, 基于 TVDI 模型编程实现了 TVDI 算法, 利用开源的网络地理信息系统(Web GIS)平台 52°North 发布成符合 OGC WPS 标准规范的网络处理服务, 实现了 TVDI 算法的网络共享。服务的发布共享能避免科研中 TVDI 算法重复实现, 有效地提高现有服务的利用率, 利于研究人员专注于干湿状况的分析研究, 提高研究效率。

收稿日期: 2014-05-09

基金项目: 国家自然科学基金项目(41101192)

张 钊, 硕士研究生, 研究方向: 农业信息化。E-mail: giszhangke@163.com

通信作者: 陈家赢, 博士, 副教授, 研究方向: 数字土壤制图及网络地理信息系统。E-mail: chen.jiaying@mail.hzau.edu.cn

1 材料与方法

1.1 实验数据

以湖北省为研究区域。湖北省位于北纬 $29^{\circ}05' \sim 33^{\circ}20'$, 东经 $108^{\circ}21' \sim 116^{\circ}07'$, 地处亚热带, 属典型的季风性气候, 地貌类型多样, 山地、丘陵、岗地和平原兼备。研究所用的遥感数据来自美国国家航空航天局 (National Aeronautics and Space Administration, NASA) 地球观测系统数据和信息系统 (Earth Observing System Data and Information System, EOSDIS) 的 MODIS 数据产品, 包括 2010 年覆盖湖北省每 8 d 合成地表反射率 500 m 分辨率 L3 产品 MOD09A1, 8 d 合成地表温度 1 km 分辨率 L3 产品 MOD11A2。这些 L3 产品都经过大气校正、Bowtie 效应校正等处理, 可以直接用于科学研究。从 MOD09A1 中提取红光波段和近红外波段用于计算归一化植被指数 (normalized difference vegetation index, NDVI), 从 MOD11A2 中提取白天地表温度数据。

1.2 算法设计与实现

1) TVDI 原理。Sandholt 等^[3]基于特征空间存在的土壤湿度等值线经验, 提出了可以表示土壤湿

度的温度植被干旱指数 (temperature vegetation dryness index, TVDI), 定义为:

$$TVDI = \frac{T_s - T_{smin}}{f(NDVI)_{max} - T_{smin}} \quad (1)$$

$$f(NDVI)_{max} = a_{max} + b_{max} \times NDVI \quad (2)$$

其中, T_{smin} 是特征空间中的最小地表温度值, 对应特征空间的湿边, 即一条平行于 NDVI 轴的直线; T_s 是像元中的地表温度观测值, NDVI 是归一化植被指数观测值; $f(NDVI)_{max}$ 是相同 NDVI 对应的最大地表温度, 对应特征空间的干边, a_{max} 和 b_{max} 是干边 $f(NDVI)_{max}$ 的线性拟合参数 (图 1)。

研究表明, 湿边并不是平行于植被指数轴的一条直线^[2]。因此, 温度植被干旱指数应该适当修改为:

$$TVDI = \frac{T_s - f(NDVI)_{min}}{f(NDVI)_{max} - f(NDVI)_{min}} \quad (3)$$

$$f(NDVI)_{min} = a_{min} + b_{min} \times NDVI \quad (4)$$

式中, a_{min} 和 b_{min} 为湿边线性拟合参数。

2) TVDI 算法流程图。算法以 NDVI、 T_s 栅格数据集为输入, 经过异常值剔除、 T_s 最值提取、干湿边参数拟合以及地图代数等步骤, 计算得到 TVDI 数据, 并以 GEOTIFF 格式的栅格数据集输出结果。

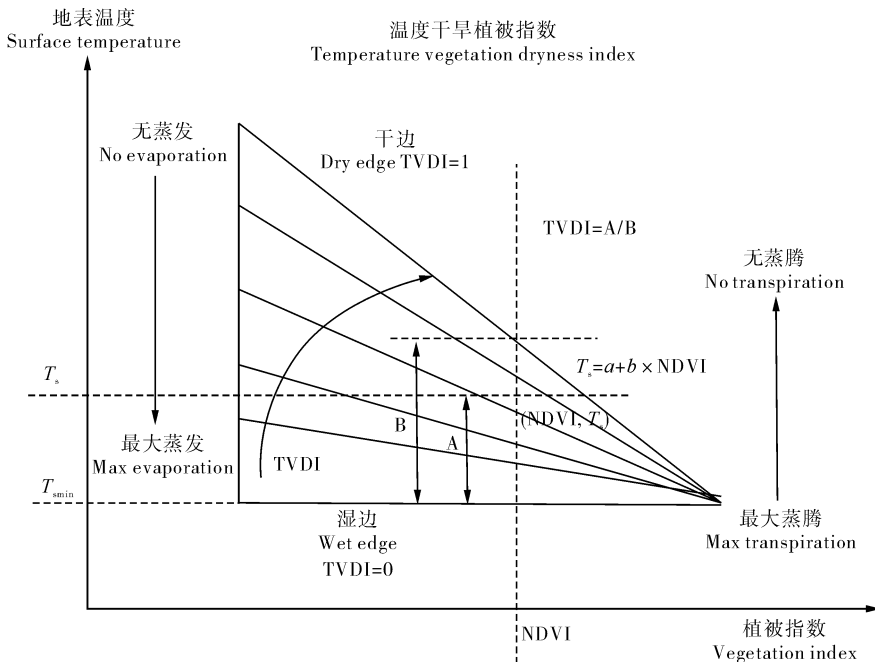


图 1 简化的 T_s /NDVI 特征空间与 TVDI 的定义

Fig.1 Simplified T_s /NDVI feature space and the definition of TVDI

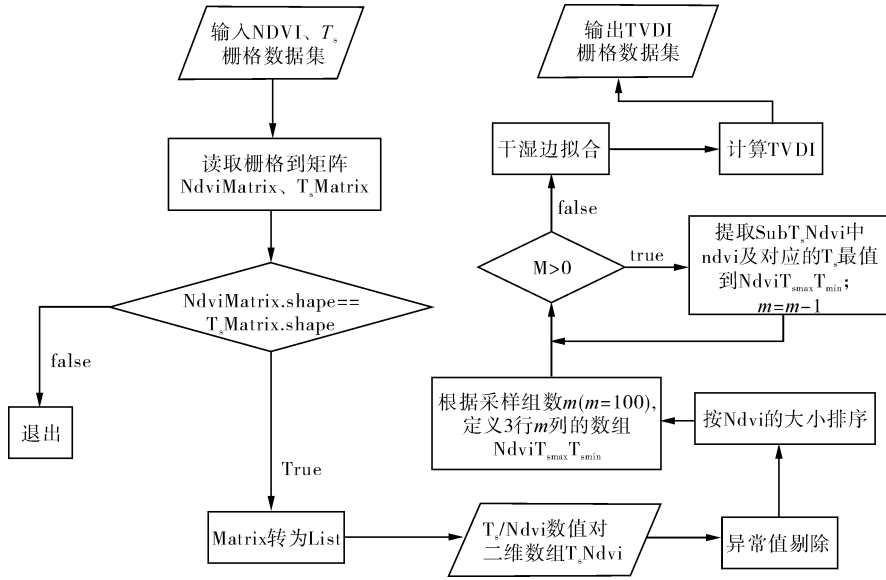


图 2 TVDI 算法流程图

Fig.2 TVDI algorithm flow chart

3) 算法设计与实现。TVDI 算法建立在网络环境中, 参数稳定, 输入和输出稳定, 具有一定的容错性。算法采用 Python 语言编程实现, Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言, 具有简洁、易读及可扩展性的特点, 提供了众多科学计算扩展库。算法利用 Numpy 科学计算模块的元组和切片技术操作矢量数组, 利用 Scipy 科学计算模块的最小二乘函数拟合干湿边。通过调用这两个科学计算扩展模块, 简化了源代码, 同时又保

证了较高的运算速度^[9]。TVDI 的计算主要包括以下 3 个步骤: 1) T_s 最值提取; 2) 干边和湿边的线性拟合; 3) 地图代数; 由 (3) 式可知, 干边和湿边的提取是 TVDI 算法的重点也是关键。

① T_s 最值提取。 T_s 最值提取, 即提取 NDVI 对应的地表温度的最大值和最小值。分别读取 T_s 和 NDVI 栅格数据集到 T_s Matrix, $Ndvi$ Matrix 2 个矩阵中, 并按图 3 所示的映射关系把 2 个矩阵转换成 2 个一维数组 T_s List 和 $Ndvi$ List。

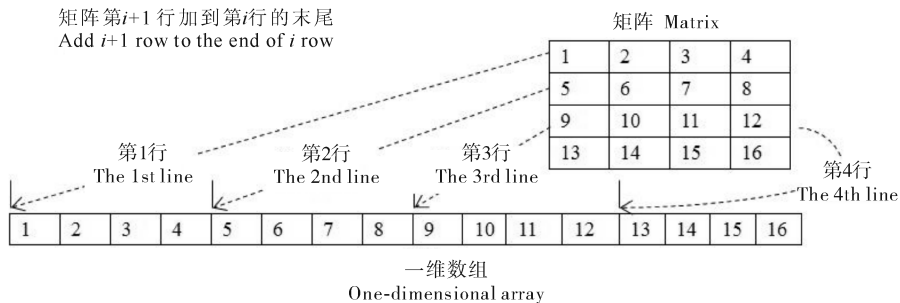


图 3 矩阵到一维数组的映射

Fig.3 Mapping matrix to one dimensional array

定义一个 2 行 N 列的二维数组 $T_sNdvi[2, N]$, 其中 $N = NdviList.size$ 或 $T_sList.size$; $NdviList.size, T_sList.size$ 分别为一维数组 $NdviList$ 和 T_sList 的长度, 二者之间存在 $NdviList.size =$

$T_sList.size$ 的关系。对 T_sNdvi 进行赋值, 令 $T_sNdvi[0, :] = NdviList, T_sNdvi[1, :] = T_sList$, 并按 $NdviList$ 的大小对 T_sNdvi 中的 $T_s/Ndvi$ 数值对排序(图 4)。

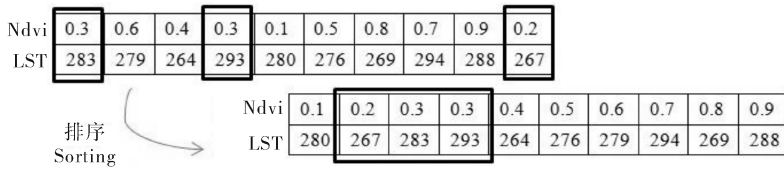


图 4 按 NdvI 大小对二维数组 T_sNdvi 排序,保持 NdvI 与 T_s 的对应关系不变

Fig.4 Sorting 2-D array T_sNdvi by NdvI and keep the correspondence between NdvI and T_s .

通过切片技术对排序后的 T_sNdvi 分组采样,按列下标把 T_sNdvi 分为 $m = 100$ 组,提取每组 T_s 的最值,对应的 NDVI 值为^[2]:

$$NDVI = NdvI_{Min} + NdvI_{Interval} \times i + \frac{NdvI_{Interval}}{2} \tag{5}$$

$$NdvI_{Interval} = \frac{NdvI_{Max} - NdvI_{Min}}{m} \tag{6}$$

其中, $NdvI_{Min}$, $NdvI_{Max}$ 分别是 $NdvI_{Matirx}$ 的最小值和最大值, i 是每组对应的编号, i 的取值范围 $[1, m]$ 。

②干湿边线性拟合。分组采样提取的 T_s 最大值和最小值,分别与其对应的 NdvI 值组成点对,最后采用最小二乘线性拟合的方式将点对拟合,形成干边和湿边。

1.3 基于 OGC WPS 的算法共享

1)OGC WPS 标准。OGC WPS 接口标准为地理信息处理服务提供了规范的输入和输出(请求和响应)规则,该标准还定义了客户端如何请求执行的一个处理过程,以及如何输出处理过程。OGC WPS 接口定义了 3 个操作,操作由客户端请求并在 WPS 服务器上执行,3 个操作分别是: GetCapabilities(允许用户请求并返回描述指定服务器执行能力的服务元数据文档)、DescribeProcess(允许用户请求和获取服务实例过程的详细信息,包括必要的输入、允许的输入格式和输出)、Execute(允许用户输入参数来执行 WPS 实现的一个具体过程,并得到返回结果)。

2)52°North WPS。52°North WPS 实现了地理

处理在网络上的标准化部署,具有处理和数据编码的可插拔体系结构,致力于建立一个可扩展的框架,以提供网络泛化算法,它完全基于 java 开源实现,支持 1.0.0 版本 WPS 规范的所有功能和操作,可插拔框架支持算法和 XML 数据的处理,建立在健壮库(包括 JT_s, geotools, XMLBeans, servlet API, derby)基础上,支持完整的服务活动日志记录;支持规范的异常处理,存储执行结果,全面支持 maven。

3)算法部署。研究采用 52° North ArcGIS Server Connector 扩展发布基于 Python Script 的 WPS,运行环境如下: ArcGIS Server 10.0, Tomcat 6.0, 52° North WPS 3.1.1, Maven 3.0.5。在 52° North WPS 中发布算法需要先注册算法。首先,建立 TVDI 算法 Python 脚本的映射文档和符号 OGC WPS 规范的处理描述文档。脚本映射文档描述了工作空间、算法的地址、脚本容器类型、必要的运行时组件等信息。处理描述文档为 WPS DescribeProcess 操作传送的元数据提供描述,这些元数据包括唯一处理标识符、可读的描述以及与输入输出参数相一致的详细格式信息。然后,在 wps_config.xml 文档中配置 WPS,添加元素(图 5A)。

其中, PROCESS_INVENTORY_DIR 为 Python 处理清单目录,包括脚本映射文档、处理描述文档和算法脚本目录; CONTAINER_URN 和 BACKEND_URN 分别用于指定机器提供的 Python 版本和机器提供的后端(AGS)版本。WORKSPACE_BASE 为算法执行工作空间。算法部署后,可以在 GetCapabilities 请求操作返回的页面中查看(图 5B)。

```

- <Repository active="true" className="org.n52.wps.python.PythonProcessRepository" name="PythonProcessRepository">
  <Property active="true" name="PROCESS_INVENTORY_DIR">E:\_TemporaryDATA\WPS\wpsBackendDirectory\pythonProcesses</Property>
  <Property active="true" name="CONTAINER_URN">urn:n52:python:2.6</Property>
  <Property active="true" name="BACKEND_URN">urn:n52:esri:arcgis:10.0</Property>
  <Property active="true" name="WORKSPACEBASE">E:\_TemporaryDATA\WPS\wpsBackendDirectory\executionWorkspace</Property>
</Repository>

- <wps:Process wps:processVersion="2">
  <ows:Identifier>giserzk.n52.wps.local.python.tvdi</ows:Identifier>
  <ows:Title>Python TVDI Algorithm</ows:Title>
</wps:Process>

```

图 5 WPS 配置文档添加的元素(A)和查看已注册的 TVDI 算法(B)

Fig.5 The increased elements of WPS configuration file(A) and check registered TVDI algorithm(B)

4)WPS 测试。52°North WPS 可以用直接浏览器作为简单客户端,另外还提供其他几种客户端,包括 Udig/Jumpg/OpenLayers,以及最新提供的 Arc-Map 客户端。研究采用对浏览器、系统或平台全面支持的 Web 调试代理 Fiddler2 测试 WPS,在 WPS 测试页面输入 Execute 操作的请求(图 6A)。Execute 操作请求文档指定 WPS 版本为 1.0.0,算法为 tvdi 算法,输入数据为 NDVI20100101_Resample51.tif、T_s20100101_Resample51.tif,并指定了输

入输出的数据格式都为 geotiff。服务器响应 Execute 请求,并返回运行结果(图 6B)。

2 结果与分析

选取 4 月到 10 月受云影响小的 9×2 幅影像测试算法与 WPS,干湿边线性拟合参数及其随时间的变化如表 1 和图 7 所示。表 1 中, a_{max} 和 a_{min} 分别是干湿边拟合方程的常数项, b_{max} 和 b_{min} 是干湿边的斜率, R^2_{max} 和 R^2_{min} 干湿边拟合方程的判定系数。

```

<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <wps:Execute xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:wps="http://www.opengis.net/wps/1.0.0" version="1.0.0" service="WPS">
  <ows:Identifier>giserzk.n52.wps.local.python.tvdi</ows:Identifier>
  <wps:DataInputs>
  - <wps:Input>
    <ows:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">inputNDVI</ows:Identifier>
    <ows:Reference method="GET" mimeType="application/geotiff" xlink:href="http://localhost:8080/wpsinput/NDVI20100101_Resample51.tif"/>
  </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
  - <wps:ResponseDocument>
    <ows:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">inputTS</ows:Identifier>
    <ows:Reference method="GET" mimeType="application/geotiff" xlink:href="http://localhost:8080/wpsinput/LST20100101_Resample51.tif"/>
  </wps:ResponseDocument>
  </wps:ResponseForm>
  </wps:Execute>

<?xml version="1.0" encoding="UTF-8"?>
- <ns:ExecuteResponse xml:lang="en-US" version="1.0.0" service="WPS" serviceInstance="http://localhost:8080/wps/WebProcessingService?
  REQUEST=GetCapabilities&SERVICE=WPS" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsExecute_response.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns="http://www.opengis.net/wps/1.0.0">
  <ns:Process ns:processVersion="2">
    <ns1:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">giserzk.n52.wps.local.python.tvdi</ns1:Identifier>
    <ows:Title xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ows="http://www.opengis.net/ows/1.1"
      xmlns:wps="http://www.opengis.net/wps/1.0.0">Python TVDI Algorithm</ows:Title>
  </ns:Process>
  <ns>Status creationTime="2013-06-11T14:48:15.822+08:00">
    <ns:ProcessSucceeded>Process successful</ns:ProcessSucceeded>
  </ns>Status>
  <ns:ProcessOutputs>
  - <ns:Output>
    <ns1:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">outputTVDI</ns1:Identifier>
    <ows:Title xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ows="http://www.opengis.net/ows/1.1"
      xmlns:wps="http://www.opengis.net/wps/1.0.0">output file</ows:Title>
    <ns:Reference href="http://localhost:8080/wps/RetrieveResultServlet?id=0431c32e-39e9-44b7-bb7f-f29aed2397?outputTVDI.ea1aa1a6-a2ff-
      472e-aa3c-34e7d75c518e" mimeType="application/geotiff" schema="">
  </ns:Output>
  </ns:ProcessOutputs>
</ns:ExecuteResponse>

```

图 6 WPS Execute 请求 XML 示例(A) 和 Execute 请求响应结果(B)

Fig.6 The XML example of the request of WPS Execute(A) and response result of Execute request(B)

表 1 干湿边拟合参数表

Table 1 The fitting parameters of dry edge and wet edge

年积日/d Day of year	b_{max}	a_{max}	R^2_{max}	b_{min}	a_{min}	R^2_{min}
97	-2.216	301.735	0.758	2.672	278.236	0.421
121	-1.986	306.476	0.763	3.409	288.189	0.760
129	-2.207	304.431	0.805	3.425	281.089	0.673
145	-3.921	308.077	0.775	1.735	285.788	0.489
161	-6.962	310.513	0.872	8.461	279.633	0.454
185	-4.703	311.931	0.744	5.956	287.293	0.731
241	-2.705	305.819	0.760	4.148	283.125	0.508
281	-2.421	303.018	0.791	1.922	283.68	0.475
297	-3.457	298.303	0.890	5.090	277.863	0.636

由表 1 和图 7 可知,拟合的干边斜率为负值,湿边斜率为正值,所有不同覆盖率的地表温度均在“植被指数—地表温度”所构成的梯形框架中,这与 Moran 等^[10]的研究结果相符合。从线性拟合的优度来看,干边判定系数都大于 0.7,线性拟合的效果很好;

湿边线性拟合整体效果也较好,判定系数有 5 幅大于 0.5,有 2 幅非常接近 0.5,另外 2 幅也都大于 0.4。在某些时段拟合精度较低的原因主要是 MODIS 产品计算不准带来的精度误差会形成噪声点,从而影响线性拟合的精度^[2]。

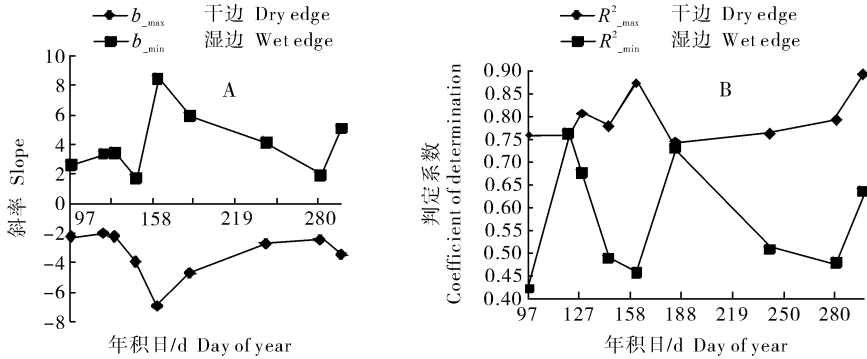


图 7 干湿边斜率(A)和判定系数(B)随时间变化情况

Fig.7 The variation trend of solp(A) and coefficient of determination(B) of dry edge and wet edge

WPS 各次测试运算时间如下:第 97 天数据用时 20.59 s,第 121 天数据用时 21.40 s,第 129 天数据用时 18.99 s,第 145 天数据用时 19.17 s,第 161 天数据用时 19.71 s,第 185 天数据用时 18.16 s,第 241 天数据用时 19.70 s,第 281 天数据用时 19.92 s,第 297 天数据用时 20.80 s,各次测试运算时间的期望为 19.826 7 s,标准差为 0.997。算法效率较高,且运行时间稳定。

间信息领域,由于其在传输协议中没有包含空间信息元数据信息以及对空间数据的标准化使得 Web Services 在解决 GIS 功能互操作方面存在不足^[12]。OGC WPS 是一个标准化的方法,专注于 Internet 上可共享的地理处理^[13]。一个典型的 WPS 支持所有 OGC WPS 规范的功能,能在应用服务器内运行本地程序和算法。



图 8 WPS 运算时间变化趋势图

Fig.8 The WPS operation time trend

本研究设计并实现了 TVDI 算法,算法中干湿边提取方法将特征空间上下边缘的点都加入到拟合过程中,可以在一定程度上克服噪声点的影响,能更好地接近真实的干湿边。以湖北省 2010 年的 MODIS 数据为例对算法进行测试,结果表明,TVDI 干湿边线性拟合整体效果较好,且算法运行稳定,效率较高。最后,基于 OGC WPS 标准接口利用 52°North WPS 平台实现 TVDI 算法的网络共享与调用,避免了科研中 TVDI 算法重复实现,让研究人员专注于干湿状况的分析研究,提高研究效率。目前,本研究还存在两个问题:一是用于测试算法的影像数据质量对计算结果有一定影响,从而会影响对 TVDI 算法效果的评估;二是 WPS 的 Execute 请求和返回结果都是 XML 文档,对于使用者要求较高。研究的后续工作应该进一步优化 TVDI 算法,实现并行处理和网格计算,提高批量影像处理的效率;同时对 WPS 的 Execute 请求和返回结果进行包装,以提供一个简单易操作平台。

3 讨论

TVDI 在土壤干湿状况研究中的应用日益广泛,但是 TVDI 的计算过程复杂,并且不同研究人员的研究过程中存在重复实现 TVDI 算法,这耗费了大量的时间和精力。Web Service 是新一代的基于互联网的分布式计算技术,是可以通过 Web 发布、查找和调用的自包含、自描述的模块化应用^[11]。Web Services 所有的标准都是通用的,因此,不能很好地解决具体某一学科领域的专业问题。在地理空

参 考 文 献

- [1] 王海,杨祖祥,王麟,等.TVDI在云南2009/2010年干旱监测中的应用[J].云南大学学报:自然科学版,2014,36(1):59-65.
- [2] 杨曦,武建军,闫峰,等.基于地表温度-植被指数特征空间的区域土壤干湿状况[J].生态学报,2009,29(3):1205-1216.
- [3] SANDHOLT I,RASMUSSEN K,ANDERSEN J.A simple interpretation of the surface temperature/vegetation index space for assessment of surface moisture status[J].Remote Sensing of Environment,2002,79(2/3):213-224.
- [4] 沈彦军,李红军,雷玉平.干旱指数应用研究综述[J].南水北调与水利科技,2013,11(4):128-133,186.
- [5] 刘立文,张吴平,段永红,等.TVDI模型的农业旱情时空变化遥感应用[J].生态学报,2014,34(13):3704-3711.
- [6] 张喆,丁建丽,鄢雪英,等.基于温度植被干旱指数的土库曼斯坦典型绿洲干旱遥感监测[J].生态学杂志,2013,32(8):2172-2178.
- [7] 孙雨,李国庆,黄震春.基于OGC WPS标准的处理服务实现研究[J].计算机科学,2009,36(8):86-88,137.
- [8] 姜超,徐青,孙庆辉,等.基于OGC WPS的GIS空间分析方法[J].测绘科学,2011,36(6):198-200.
- [9] 沈殊璇,薄亚明.适合于科学计算的脚本语言Python[J].微计算机应用,2002,23(5):289-291.
- [10] MORAN M S,CLARK T R,INOUE Y,et al.Estimating crop water deficit using the relation between surface-air temperature and spectral vegetation index[J].Remote Sensing of Environment,1994,49:246-263.
- [11] SAHINA K,GUMUSAYB M U.Service Oriented Architecture (SOA) Based Web Services For Geographic Information Systems [C]// CHEN J,JIANG J,KAINZ W.The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.Beijing: ISPRS Congress,2008(37):625-630.
- [12] 张仁华,孙晓敏,王伟民,等.一种可操作的区域尺度地表通量定量遥感二层模型的物理基础[J].中国科学,2004,34(增刊2):200-216.
- [13] SUSANA L,MIGUEL R L,OSCAR P,et al.A Toponym Resolution Service following the OGC WPS Standard[C]// MICHELA B,CYRIL R, XIANG L.Web and Wireless Geographical Information Systems. Shanghai: Springer Berlin Heidelberg, 2008,5373:75-85.

Implementation and sharing of TVDI algorithm based on OGC WPS

ZHANG Ke CHEN Jia-ying HUANG Wei

College of Resources and Environment, Huazhong Agricultural University, Wuhan 430070, China

Abstract A web processing service based on Open Geospatial Consortium (OGC) web processing service (WPS) specification was designed and shared by the open source network geospatial information system platform 52°North to figure out the problems of temperature vegetation drought index (TVDI) including complicated calculation and repeat to achieve. TVDI model algorithm was realized using the Python language programming. The MODIS data in 2010 in Hubei Province were used to test the algorithm. The results showed that the ensemble of dry and wet edges linear fitting was good. The operation of algorithm was stable with high efficiency. The release and sharing of TVDI WPS avoided the problems of complicated calculation and repeat to achieve. It will help the researchers to focus on analyzing/studying the dry and wet conditions, and improve the efficiency of study.

Key words TVDI; WPS; OGC; MODIS; 52°North

(责任编辑:陆文昌)