温度植被干旱指数（TVDI）的计算及其Python程序实现

**一、温度植被干旱指数基本概念**

温度植被干旱指数（Temperature Vegetation Dryness Index，TVDI）是一种基于光学与热红外遥感通道数据进行植被覆盖区域表层土壤水分反演的方法。

作为同时与归一化植被指数(NDVI)和地表温度(LST)相关的温度植被干旱指数(TVDI)可用于干旱监测，尤其是监测特定年内某一时期整个区域的相对干旱程度，并可用于研究干旱程度的空间变化特征。

温度植被干旱指数的公式如下：



其中，湿边方程*LST*smin = a \* NDVI + b；干边方程*LST*smax = c \* NDVI + d。

TVDI的值域为[0, 1]。

TVDI的值越大，表示土壤湿度越低，TVDI越小，表示土壤湿度越高。

**二、干湿边拟合**

在TVDI指数计算的过程中，需要拟合干边方程和湿边方程，其中还需要用户自己设置参数，是比较重要的计算步骤。拟合的关键是怎么找到有效的NDVI（相当于自变量X）和LST（相当于因变量Y）这两组相对应的数据，本文针对干湿边的拟合过程做一下详细的说明。

根据输入的NDVI和LST数据，以NDVI为横坐标，LST为纵坐标，可以得到NDVI-LST的散点图。根据NDVI-LST的散点图，就能得到每个NDVI值对应的LST的最大最小值，也就是我们要求的干边和湿边。具体步骤如下：

1）将NDVI等分为100份，分别为0~0.01，0.01~0.02，，，，，0.99~1.0（可根据具体情况进行修改）；

2）在NDVI影像中查找每个范围内的NDVI值的索引，然后根据这些索引获取对应的LST数据的栅格值，求出每个范围内LST栅格值中的最大值和最小值；

3）将100份的LST栅格值分别求出最大值和最小值，就能得到NDVI-LST对应的干湿边的散点图，如图1所示；

4）设置NDVI的有效值范围（通常为0.2~0.8），选取有效范围内的数值进行线性拟合，得到干湿边方程。

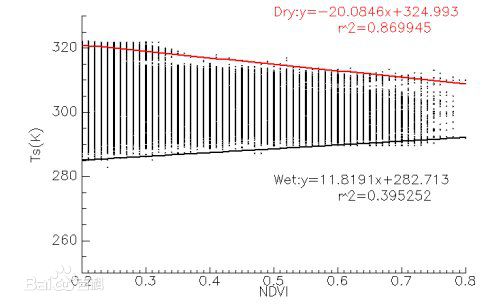


图1 干湿边的散点图及线性拟合

**三、TVDI计算的Python实现**

TVDI模型对干旱监测方面有重要的意义，但是在实际应用中却比较少见，主要原因是目前的遥感软件中缺少计算TVDI的工具，要实现这个模型需要编程，做起来十分麻烦。有学者利用ENVI/IDL来计算TVDI，开发出了TVDI计算补丁，但用起来很不灵活，有时还会出现未知错误，给用户造成很多困扰。

因此，本文基于Python语言实现了TVDI的计算，详细代码如下：

# -\*- coding: utf-8 -\*-

## Calculate TVDI

**import** os

**import** math

**from** osgeo **import** gdal

**import** numpy

**from** util **import** **\***

## 计算TVDI

**def** tvdi**(**dataDir**,** ndviMin**,** ndviMax**):**

abcd **=** edge**(**ndviMin**,** ndviMax**)**

a **=** abcd**[**0**][**0**]**

b **=** abcd**[**0**][**1**]**

c **=** abcd**[**1**][**0**]**

d **=** abcd**[**1**][**1**]**

**print** "干边拟合：y = %.3fx + %.3f，r2 = %.3f" **%** **(**a**,** b**,** abcd**[**0**][**2**]** **\*** abcd**[**0**][**2**])**

**print** "湿边拟合：y = %.3fx + %.3f，r2 = %.3f" **%** **(**c**,** d**,** abcd**[**1**][**2**]** **\*** abcd**[**1**][**2**])**

TVDI **=** numpy**.**zeros**((**nRows**,** nCols**))**

**for** m **in** range**(**nRows**):**

**for** n **in** range **(**nCols**):**

TVDI**[**m**][**n**]** **=** **((**abcd**[**0**][**0**]** **\*** NDVI**[**m**][**n**]** **+** abcd**[**0**][**1**])** **-** LST**[**m**][**n**])** **/** **((**abcd**[**0**][**0**]** **\*** NDVI**[**m**][**n**]** **+** abcd**[**0**][**1**])** **-** **(**abcd**[**1**][**0**]** **\*** NDVI**[**m**][**n**]** **+** abcd**[**1**][**1**]))**

**if(**TVDI**[**m**][**n**]** **>** 9999 **or** TVDI**[**m**][**n**]** **<** **-**9999**):**

TVDI**[**m**][**n**]** **=** 0.

outputPath **=** dataDir **+** os**.**sep **+** outputfile

WriteGTiffFile**(**outputPath**,** nRows**,** nCols**,** TVDI**,** geotrans**,** srs**,** **-**999**,** gdal**.**GDT\_Float32**)**

**print** "Calculate TVDI finished!\nSave as '%s'\n" **%** outputPath

## 计算干边和湿边

**def** edge**(**ndviMin**,** ndviMax**):**

**print** "nRows = %d, nCols = %d" **%** **(**nRows**,** nCols**)**

edgeArr **=** **[[]** **for** i **in** range**(**99**)]**

**for** i **in** range**(**nRows**):**

**for** j **in** range **(**nCols**):**

**if(**NDVI**[**i**][**j**]** **>** 0 **and** NDVI**[**i**][**j**]** **<** 1**):**

k **=** str**(**NDVI**[**i**][**j**])[**2**:**4**]**

**if(**LST**[**i**][**j**]** **>** **-**9999 **and** LST**[**i**][**j**]** **<** 9999**):**

edgeArr**[**int**(**k**)].**append**(**LST**[**i**][**j**])**

## 排序，寻找干边湿边

eMax **=** **[]**

ndviSort **=** **[]**

eMin **=** **[]**

**for** n **in** range**(**len**(**edgeArr**)):**

edgeArr**[**n**]** **=** numpy**.**sort**(**edgeArr**[**n**])**

**if(**len**(**edgeArr**[**n**])** **>** 1**):**

eMax**.**append**(**edgeArr**[**n**][**len**(**edgeArr**[**n**])** **-** 1**])**

eMin**.**append**(**edgeArr**[**n**][**0**])**

ndviSort**.**append**(**float**(**n**)** **/** 100.**)**

**print** "%s, %s, %s" **%** **(**eMax**[**n**],** eMin**[**n**],** ndviSort**[**n**])**

nMax **=** int**(**ndviMax **\*** 100**)**

nMin **=** int**(**ndviMin **\*** 100**)**

**if(**nMax **>** len**(**ndviSort**)):**

nMax **=** len**(**ndviSort**)** **-** 1

fiteMax **=** eMax**[**nMin**:**nMax**]**

fiteMin **=** eMin**[**nMin**:**nMax**]**

fitndviSort **=** ndviSort**[**nMin**:**nMax**]**

fitMax **=** linefit**(**fitndviSort**,** fiteMax**)**

fitMin **=** linefit**(**fitndviSort**,** fiteMin**)**

**return** **[**fitMax**,** fitMin**]**

## 线性拟合，返回系数，截距和相关系数值

**def** linefit**(**x **,** y**):**

N **=** float**(**len**(**x**))**

sx**,** sy**,** sxx**,** syy**,** sxy **=** 0**,** 0**,** 0**,** 0**,** 0

**for** i **in** range**(**0**,**int**(**N**)):**

sx **+=** x**[**i**]**

sy **+=** y**[**i**]**

sxx **+=** x**[**i**]** **\*** x**[**i**]**

syy **+=** y**[**i**]** **\*** y**[**i**]**

sxy **+=** x**[**i**]** **\*** y**[**i**]**

a **=** **(**sy **\*** sx **/** N **-** sxy**)** **/** **(** sx **\*** sx **/** N **-** sxx**)**

b **=** **(**sy **-** a**\***sx**)/**N

r **=** abs**(**sy **\*** sx **/** N **-** sxy**)** **/** math**.**sqrt**((**sxx **-** sx **\*** sx **/** N**)** **\*** **(**syy **-** sy **\*** sy **/** N**))**

**return** **[**a**,** b**,** r**]**

**if** \_\_name\_\_ **==** "\_\_main\_\_"**:**

## 定义文件路径

dataDir **=** r'#文件夹路径#'

## 输入文件

NDVIfile **=** r'NDVI.tif'

LSTfile **=** r'LST.tif'

## 设置有效NDVI

ndviMax **=** 0.8

ndviMin **=** 0.2

## 输出文件

outputfile **=** 'TVDI.tif'

## 读取栅格数据

NDVIpath **=** dataDir **+** os**.**sep **+** NDVIfile

NDVI **=** ReadRaster**(**NDVIpath**).**data

LSTpath **=** dataDir **+** os**.**sep **+** LSTfile

LST **=** ReadRaster**(**LSTpath**).**data

nCols **=** ReadRaster**(**NDVIpath**).**nCols

nRows **=** ReadRaster**(**NDVIpath**).**nRows

geotrans **=** ReadRaster**(**NDVIpath**).**geotrans

srs **=** ReadRaster**(**NDVIpath**).**srs

## 计算TVDI

tvdi**(**dataDir**,** ndviMin**,** ndviMax**)**

**print** "Finished!"