TÍNH TOÁN NỘI SUY IDW TRONG GIS DƯỚI GÓC NHÌN HỆ THỐNG VÀ ỨNG DỤNG MINH HỌA

Hoàng Thị Kiều Anh[1], Nguyễn Đức Tuấn[2], Nguyễn Duy Liêm[3], Nguyễn Thị Lan Phương[3], Khưu Minh Cảnh[4]
[1]. Đại học Thủy Lợi
[2]. Trung tâm Dịch vụ Phân tích Thí nghiệm và Tiêu chuẩn Đo lường Chất lượng TP.HCM
[3]. Trường Đại học Nông Lâm TP.HCM
[4]. Trung tâm Công nghệ thông tin và Địa không gian TP.HCM
Email: hoangkieuanh@tlu.edu.vn

Tóm tắt: *Trọng số nghịch đảo khoảng cách (IDW) mặc dù là phương pháp không mới nhưng luôn được quan tâm vì tính ứng dụng trong việc lập nhanh các bản đồ. IDW được hỗ trợ và cài đặt trên nhiều nền tảng công nghệ. Do vậy, việc giới thiệu và lựa chọn sử dụng cần được định hướng để người xây dựng hệ thống có bức tranh tổng thể. Bài báo này nêu một số thông tin về việc lựa chọn môi trường cài đặt thực thi IDW trong các ứng dụng.*

Từ khóa*: IDW, GIS, SQL, viễn thám.*

IMPLEMENTING IDW FROM SYSTEM PERSPECTIVE AND ILLUSTRATIVE APPLICATION

Abstract: *Inverse Distance Weighting (IDW), although not a new method, is always of interest due to its applicability in rapidly creating maps. IDW is supported and implemented on many technology platforms. Therefore, its introduction and selection need to be guided so that system developers have a comprehensive overview. This paper provides some information on selecting the environment for implementing IDW in applications.*

Keywords: *IDW, GIS, SQL, remote sensing*

Nhận bài: 20/04/2026 *Phản biện: 19/05/2026* *Duyệt đăng: 24/05/2026*

I. ĐẶT VẤN ĐỀ

Thông thường, trong các sự cố hoặc thải hóa chất gây ô nhiễm các nguồn không khí, nước. Và một vùng ảnh hưởng được xác lập với nguyên tắc cơ bản: khu vực gần nguồn sẽ bị tác động nhiều hơn. Đó là tư duy của nội suy IDW (Inverse Distance Weight), nghĩa là trọng số nghịch đảo khoảng cách. Tính toán IDW thường được thấy trong các hệ thống GIS với các ứng dụng xây dựng bản đồ để trực quan hóa mức độ ảnh hưởng của những nguồn phát sinh theo khoảng cách, như khả năng tác động từ nguồn gây ô nhiễm theo khoảng cách, cụ thể nghịch đảo khoảng cách thể hiện ở tính chất khoảng cách càng gần (càng nhỏ) thì mức độ "ảnh hưởng" càng lớn. Lưu ý: khác với phương pháp Kriging (sử dụng nền tảng thống kê), IDW chỉ thuần túy dựa trên khoảng cách và phân bố tuyến tính nên chỉ phù hợp các ứng dụng mang tính liên tục

Trong công nghệ, tính toán IDW chủ yếu được cài đặt như một thuật toán phân tích không gian trong các phần mềm GIS như họ ArcGIS,… hoặc các dạng thư viện của Python như scipy. Bên cạnh đó, IDW còn được cài đặt ở dạng tính toán SQL chuẩn trong các hệ quản trị cơ sở dữ liệu quan hệ như SQL Server, PosrgreSQL,… Bài báo này sẽ giới thiệu định hướng một số phương án cài đặt sử dụng IDW tổng hợp được và mô tả một ứng dụng cơ bản để người sử dụng có thể lựa chọn trong khi xây dựng hệ thống để tính toán nội suy IDW.

II. NỘI DUNG NGHIÊN CỨU

2.1. Cơ sở lý thuyết

Phần này giới thiệu về một số lý thuyết của tính toán nội suy IDW.

2.1.1. Về IDW

Trong không gian 1 chiều, phương pháp IDW với 2 điểm lân cận gần nhất là phương pháp nội suy tuyến tính. Nghĩa là phương pháp ước lượng một giá trị chưa biết nằm giữa hai giá trị đã biết trên đường thẳng. Đây là kỹ thuật thường để tính toán hoặc tra cứu, tính toán màu, ước lượng các số liệu có tính chất theo xu hướng tuyến tính. Bài toán thường được thấy dưới dạng phát biểu sau:

Giả sử cần tìm giá trị y tương ứng với x, nghĩa giá trị của cặp tọa độ (x,y), biết rằng giá trị hai điể $x_1 < x < x_2$) thỏa:

Và điểm (x,y) nằm trên đoạn thẳng nối 2 điểm (x_1,y_1) và (x_2,y_2). Công thức nội suy sẽ là:

$y=y_1+((x-x_1)(y_2-y_1))/(x_2-x_1)$

Đối với không gian nhiều chiều (từ 2 trở lên), về tính toán chi tiết, phương pháp nội suy IDW là:

$$Z_0 = \frac{\sum_{i=1}^{N} Z_i \times w_i}{\sum_{i=1}^{N} w_i}$$

Trong đó:

- $w_i = \frac{1}{d_i^p}$

Z_0: giá trị ước tính của biến z tại vị trí cần tính
Z_i: giá trị mẫu tại các điểm i lân cận
N: số lượng điểm lân cận cần tính toán
d_i: khoảng cách giữa điểm mẫu và điểm cần ước tính; số mũ p được xác định là hàm độ đo

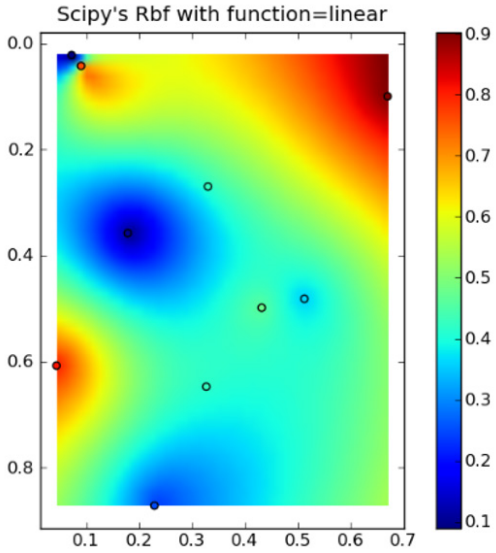
2.1.2. Độ đo sử dụng trong IDW

IDW sử dụng độ đo là khoảng cách Euclide, một cách nói quen thuộc là "đường chim bay". Trong tính toán khoảng cách Euclide sẽ có số mũ là 2 (p=2).

Lưu ý: với các p mang giá trị khác, độ đo sẽ có ý nghĩa khác. Ví dụ: p=1 là độ đo theo các tòa nhà trong đô thị, tên gọi là Manhattan. Chú ý thêm, về tổng quát, cả độ đo Euclide và Manhattan thuộc nhóm chuẩn vector, có tên gọi tổng quát hóa là khoảng cách Minkowski với công thức như sau:

$$D(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p}$$

Khi p=1 là độ đo Manhattan, khi p=2 là độ đo Euclide và khi p=∞ thì gọi là độ đo Chebyshev.



Hình 1: Một minh họa về kết quả nội suy IDW bằng gói scipy

2.1.3. Nền tảng Python và SQL

Python là một ngôn ngữ lập trình bậc cao đa năng được Guido van Rossum nghiên cứu và phát hành lần đầu vào năm 1991. Với các tư duy gọn, nhẹ, ngày nay Python được sử dụng trong nhiều hệ thống của nhiều ngành vì tính linh hoạt. Trước đó, giữa những năm 1970, SQL (Structured Query Language – ngôn ngữ truy vấn có cấu trúc) do Edgar F.Codd xây dựng được chọn là mô hình tiêu chuẩn dung cho hệ thống quản lý cơ sở dữ liệu quan hệ. Mặc dù Python và SQL không phải "đối thủ" trực tiếp xây dựng hệ thống, thậm chí bổ sung cho nhau vì những ưu điểm của 2 công nghệ này. Tuy vậy, cả 2 đều có thể cài đặt IDW để xử lý. Do vậy, việc hoạch định lựa chọn sử dụng để xử lý tính toán IDW là một yếu tố trong hệ thống. Trong lập trình ứng dụng hiện đại, việc lựa chọn phương pháp cài đặt cho hệ thống là quan trọng vì các tiêu chí liên quan đến tốc độ, hiệu suất và các yếu tố khác về nguồn tài nguyên khi thực hiện nhiều tính toán, đặc biệt khi các yêu cầu được gửi gần thời gian với nhau.

Dưới đây là Bảng về tính năng khác biệt giữa truy vấn SQL và Python cho dữ liệu không gian:

Đặc điểm so sánh	Truy vấn SQL	Python
Mục tiêu chính	Để giao tiếp và quản lý cơ sở dữ liệu quan hệ	Là ngôn ngữ lập trình đa năng, tự động hóa và các mô hình cao cấp
Môi trường triển khai tiêu biểu	Các hệ quản trị CSDL không gian như PostGIS/PostgreSQL, SpatiaLite, SQL Server, Oracle Spatial	Python scripts, Jupyter notebooks, các ứng dụng GIS (như ArcGIS, QGIS)

Kiểu luận lý	***Dạng khai báo (declarative):*** cho hệ thống biết muốn gì và hệ thống sẽ quyết định cách thức lấy thông tin tương ứng	***Dạng bắt buộc (imperative):*** phải viết rõ mỗi bước cần làm gì trong xử lý
Phương thức hoạt động	Thông qua các hàm SQL không gian như ST_Distance, ST_Buffer,.. được các hệ quản trị tương thích với chuẩn OGC/ISO	Thông qua thư viện như: GeoPandas, Shapely, Rasterio, PySAL để xử lý hình học, raster hoặc thống kê/hiển thị
Phạm vi dữ liệu	Phù hợp cho các tập dữ liệu có cấu trúc, có sự liên kết, kết nối (join) hoặc các quan hệ với nhau và có khả năng lọc cao.	Dữ liệu cho các loại dữ liệu phức hợp, máy học, phù hợp các dữ liệu không cấu trúc.
Tính hiệu quả	Nhanh khi xử lý trực tiếp nhiều triệu dòng dữ liệu trên máy chủ.	Nhanh cho việc phân tích dữ liệu thống kê phức tạp và định dạng (reshape) dữ liệu trong bộ nhớ
Đề xuất các trường hợp (nên) sử dụng đối với dữ liệu GIS và viễn thám	- Trích xuất/truy vấn dữ liệu từ cơ sở dữ liệu; - Quản lý nhiều người dùng trên dữ liệu không gian; - Phân tích cho tập dữ liệu lớn bên trong tập dữ liệu	- Phân tích không gian và mô hình hóa ngoài CSDL; - Xử lý dữ liệu Raster; - Tự động hóa các tác vụ GIS; - Học máy bằng dữ liệu không gian.

Từ bảng so sánh trên, nhận thấy việc lựa chọn cài đặt IDW bằng Python hoặc SQL là vấn đề cần cân nhắc cho các hệ thống hiện có từ gốc nhìn tài nguyên cũng như ứng dụng và cách đáp ứng các yêu cầu.

2.2. Cài đặt và ứng dụng mẫu triển khai

IDW có nhiều biến thể cài đặt ở các môi trường khác nhau. Điều này làm nên sự linh động trong xử lý nhưng cũng cần phải lựa chọn đối với các hệ thống nhất định. Ngoài các phương pháp tính toán cao cấp hơn như cài đặt xử lý song song, dưới đây là một số cài đặt thường được nhắc đến:

2.2.1. Cài đặt IDW bằng đoạn mã Python

Thông thường, việc cài đặt IDW bằng việc xác định ma trận khoảng cách. Trong mặt phẳng 2 chiều, khi cho danh sách tọa độ các điểm và mong muốn tính toán nội suy giá trị tại 1 điểm, ma trận khoảng cách được lập ra để tìm các khoảng cách nhỏ nhất.

Hàm tính toán ma trận khoảng cách được dễ dàng xây dựng

```
def distance_matrix(x0, y0, x1, y1):
    obs = np.vstack((x0, y0)).T
    interp = np.vstack((x1, y1)).T

    # Tạo ma trận khoảng cách giữa các cặp điểm
    d0 = np.subtract.outer(obs[:,0], interp[:,0])
    d1 = np.subtract.outer(obs[:,1], interp[:,1])

    return np.hypot(d0, d1)
```

Sau đó, tính toán IDW được cài đặt:

```
def simple_idw(x, y, z, xi, yi):
    dist = distance_matrix(x,y, xi,yi)

    # Trọng số trong IDW là nghịch đảo khoảng cách
    weights = 1.0 / dist

    # Chuẩn hóa trọng số
    weights /= weights.sum(axis=0)

    # Nhân trọng số cho mỗi điểm nội suy với giá trị Z
    zi = np.dot(weights.T, z)
    return zi
```

2.2.2. Cài đặt IDW trên hệ thống SQL

Một cách cơ bản, mã SQL để cài đặt tính toán IDW được thể hiện bằng các lệnh SQL chuẩn ISO. Đoạn mã SQL dưới đây cài đặt tìm K điểm gần với vị trí (x,y) và tính toán trọng số trung bình

```
WITH DistanceData AS (
    SELECT
        id,
        value,
        -- Tính theo khoảng cách Euclide.
        SQRT(POWER(source_x - 10.0, 2) + POWER(source_y - 20.0, 2)) AS
distance
    FROM known_points
),
WeightedData AS (
    SELECT
        id,
        value,
        distance,
        -- Nghịch đảo khoảng cách (tham số mũ p = 2)
        -- Giá trị epsilon (0.00001) để tránh chia khoảng cách là 0
        CASE WHEN distance = 0 THEN 1.0
             ELSE 1.0 / POWER(distance, 2)
        END AS weight
    FROM DistanceData
    ORDER BY distance ASC
    -- Giới hạn K điểm lân cận nhất (Ví dụ: K = 5)
    LIMIT 5
)
SELECT
    SUM(value * weight) / SUM(weight) AS idw_interpolated_value
FROM WeightedData;
```

Lưu ý cho đoạn mã trên:

- DistanceData: tính toán khoảng cách giữa các vị trí theo công thức Pythago

- WeightedData: Tính toán trọng số cho mỗi điểm

Bên cạnh đó, ở các hệ quản trị cơ sở dữ liệu không gian, mỗi điểm được xem là các đối tượng. Và hệ thống thiết kế dạng cơ sở dữ liệu quan hệ hướng đối tượng. Từ đó, các tính toán về không gian được tính toán theo phương thức/hàm hóa. Ví dụ: hàm khoảng cách được chuẩn hóa thành hàm: ST_Distance(geom1, geom2) với geom1 và geom2 là 2 đối tượng hình học cần tính khoảng cách. Từ đó, đoạn code trên khi cài đặt ở hệ cơ sở dữ liệu không gian được gợi ý viết như sau:

```
WITH nearest_samples AS (
    SELECT
        s.id,
        s.measured_value,
        ST_Distance(s.geom, t.geom) AS distance
    FROM sample_points s
    CROSS JOIN target_points t
    WHERE t.id = 1  -- giả định 1 là ID của điểm ta muốn dự đoán giá trị
    ORDER BY ST_Distance(s.geom, t.geom) ASC
    LIMIT 5 - Sử dụng lân cận 5 điểm gần nhất
)
SELECT
    SUM(measured_value / POWER(distance, 2)) /
    SUM( 1/POWER(distance, 2)) AS interpolated_value
FROM nearest_samples;
```

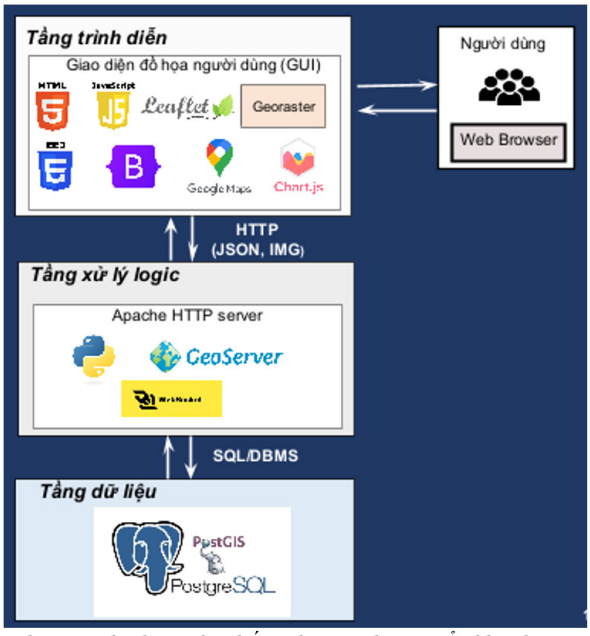
2.2.3. Sử dụng IDW trên hệ thống PostGIS

Ngoài ra, trong các hệ thống cơ sở dữ liệu không gian như PostGIS, việc tính toán IDW được cài đặt sẵn thành các hàm. Cụ thể hàm được sử dụng là: ST_InvDistWeight4ma. Bên cạnh đó, các hệ thống PostGIS cũng hỗ trợ tính toán cho loại định dạng dữ liệu như raster bằng hàm ST_InterpolateRaster.

2.3. Minh họa ứng dụng nhỏ về IDW

Theo đó, để xây dựng một ứng dụng, hiện tại, thông thường mô hình kiến trúc sẽ được tách thành 3 tầng khác nhau, cụ thể:

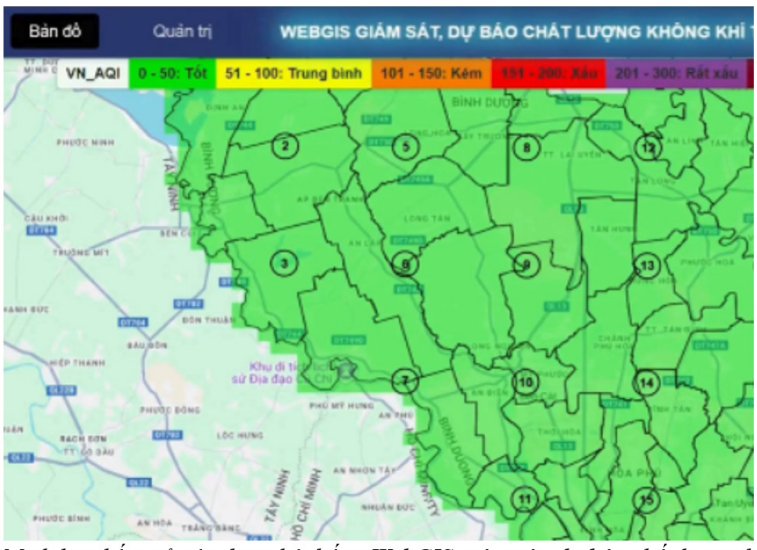
- Tầng dữ liệu: là nơi lưu trữ, quản lý và có thể xử lý một số dữ liệu của hệ thống
- Tầng xử lý logic chứa các ứng dụng chạy, tiếp nhận các yêu cầu và lấy dữ liệu để xử lý
- Tầng trình bày tiếp nhận các yêu cầu của người dùng thông qua giao diện đồ họa.



Hình 2: Mô hình các hệ thống thường được triển khai hiện nay

Như vậy, nhìn vào hệ thống, IDW có thể cài đặt ở tầng xử lý logic hoặc ở tầng dữ liệu. Từ đó, chúng ta có thể lựa chọn phương án tính toán cho phù hợp. Đề xuất sử dụng tầng cơ sở dữ liệu để tính toán khi việc tính toán liên quan đến nhiều trạm đo và việc tính toán cần nhanh. Trong khi đó, việc lựa chọn Python trong tầng xử lý logic để tính toán khi cần sự tích hợp, phối hợp với nhiều yếu tố khác để rà soát, phân tích, thống kê dữ liệu trực tiếp.

Minh họa kết quả của ứng dụng về giám sát, dự báo chất lượng không khí. Khi đó, do số lượng trạm phân tích không nhiều và tỉ lệ bản đồ không quá chi tiết nên hệ thống lựa chọn tính toán bằng Python. Kết quả như sau với các tọa độ các trạm được và giới hạn tính toán nội suy nằm trong phạm vi địa lý:



Hình 3: Minh họa kết quả xây dựng hệ thống WebGIS giám sát, dự báo chất lượng không khí

III. KẾT LUẬN

Nội suy IDW có những ưu điểm tính toán nhanh và thiết lập bản đồ ngay có thể hỗ trợ linh hoạt cho việc dựng nhanh các hệ thống bản đồ trực quan. Ngoài ra, đó cũng là một trong những công cụ để người quản lý có cơ sở để nghi ngờ những hoạt động bất thường diễn ra trong một "khung cảnh" mà có thể dự đoán được trong tính toán nội suy. Do đó, từ những tiện ích trên, bài báo này nói về các hướng cài đặt có thể cho IDW để người nghiên cứu có thể lựa chọn và tùy biến trên hệ thống. Một hệ thống có thể cài đặt nhiều phương pháp tính toán IDW và sau đó tùy theo mức độ yêu cầu mà hệ thống có thể linh hoạt sử dụng từng tài nguyên để tính toán và trả về kết quả.

TÀI LIỆU THAM KHẢO

Thông tin về SQL: https://vi.wikipedia.org/wiki/SQL

Lệnh nội suy trong PostGIS: https://postgis.net/docs/RT_ST_InterpolateRaster.html

Lệnh tính toán IDW trong PostGIS: https://postgis.net/docs/RT_ST_InvDistWeight4ma.html

Về độ đo: https://viblo.asia/p/distance-measure-trong-machine-learning-ByEZkopYZQ0

Các cài đặt IDW bằng Python: https://stackoverflow.com/questions/3104781/inverse-distance-weighted-idw-interpolation-with-python