



Código de registro

python – ¿Calculando SimRank usando NetworkX?

Tiempo: 2019-04-08 03:10:01.0 Etiqueta : python graph-algorithm networkx

Traducción: Fuente  Corrección de traducción

Me preguntaba ¿cómo podemos usar el módulo [python networkX](#) para implementar [SimRank](#) para comparar la similitud de 2 nodos? Entiendo que networkX proporciona métodos para buscar vecinos y algoritmos de análisis de enlaces como PageRank y HITS, pero ¿hay alguno para SimRank?

Ejemplos, tutoriales son bienvenidos también!

Mejor respuesta

Actualizar

Implementé una librería `networkx_addon`. SimRank está incluido en la biblioteca. Echa un vistazo a: https://github.com/hhchen1105/networkx_addon para más detalles.

Uso de la muestra:

```
>>> import networkx
>>> import networkx_addon
>>> G = networkx.Graph()
>>> G.add_edges_from([('a','b'), ('b','c'), ('a','c'), ('c','d')])
>>> s = networkx_addon.similarity.simrank(G)
```

Puede obtener la puntuación de similitud entre dos nodos (por ejemplo, nodo 'a' y nodo 'b') mediante

```
>>> print s['a']['b']
```

SimRank es una medida de similitud de vértice. Calcula la similitud entre dos nodos en un gráfico basado en la topología, es decir, los nodos y los enlaces del gráfico. Para ilustrar el SimRank, consideremos el siguiente gráfico, en el que a, b, c se conectan entre sí y d está conectado a c. Cómo un nodo a es similar a un nodo d, se basa en cómo los nodos vecinos de a, b y c, similares a los vecinos de d, c.

```
+-----+
|       |
a---b---c---d
```



Código de registro

importancia relativa entre vecinos directos y vecinos indirectos. La ecuación formal de SimRank se puede encontrar [here](#).

La siguiente función toma una gráfica networkx \$ G \$ y el parámetro de importancia relativa \$ r \$ como entrada, y devuelve la simulación del valor de similitud entre dos nodos cualquiera en \$ G \$. La simulación del valor de retorno es un diccionario de diccionario flotante. Para acceder a la similitud entre el nodo \$ a \$ y el nodo \$ b \$ en el gráfico \$ G \$, uno puede simplemente acceder a \$ sim[a][b] \$.

```
def simrank(G, r=0.9, max_iter=100):
    # init. vars
    sim_old = defaultdict(list)
    sim = defaultdict(list)
    for n in G.nodes():
        sim[n] = defaultdict(int)
        sim[n][n] = 1
        sim_old[n] = defaultdict(int)
        sim_old[n][n] = 0

    # recursively calculate simrank
    for iter_ctr in range(max_iter):
        if _is_converge(sim, sim_old):
            break
        sim_old = copy.deepcopy(sim)
        for u in G.nodes():
            for v in G.nodes():
                if u == v:
                    continue
                s_uv = 0.0
                for n_u in G.neighbors(u):
                    for n_v in G.neighbors(v):
                        s_uv += sim_old[n_u][n_v]
                sim[u][v] = (r * s_uv / (len(G.neighbors(u)) * len(G.neighbors(v))))
    return sim

def _is_converge(s1, s2, eps=1e-4):
    for i in s1.keys():
        for j in s1[i].keys():
            if abs(s1[i][j] - s2[i][j]) >= eps:
                return False
    return True
```

Para calcular los valores de similitud entre nodos en el gráfico anterior, puede intentar esto.



Código de registro

```
>>> simrank(G)
```

Obtendrás

```
defaultdict(<type 'list'>, {'a': defaultdict(<type 'int'>, {'a': 0, 'c': 0.6260762680740786
```

Verifiquemos el resultado calculando la similitud entre, por ejemplo, nodo a y nodo b, denotado por $S(a, b)$.

$S(a, b) = r * (S(b, a) S(b, c) S(c, a) S(c, c)) / (2 * 2) = 0.9 * (0.6538 0.6261 0.6261 1) / 4 = 0.6538$,

que es lo mismo que nuestro $S(a, b)$ calculado anteriormente.

Para más detalles, es posible que desee revisar el siguiente documento:

G. Jeh y J. Widom. SimRank: una medida de similitud de contexto estructural. En KDD'02 páginas 538-543. Prensa ACM, 2002.

Artículos relacionados

- [¿Calculando un tamaño de directorio usando Python?](#)
- [¿Cómo hago para que Linux reconozca una nueva unidad SATA/dev/sda que intercambié en caliente sin reiniciar?](#)
- [calculando el ángulo entre dos líneas en python](#)
- [calculando la raíz cúbica en python](#)
- [Python calculando números catalanes](#)
- [Calculando Pi con decimal en Python](#)
- [python - Numpy está calculando mal \[duplicado\]](#)
- [calculando el promedio usando awk desde múltiples archivos](#)

[Haga clic para ver más artículos relacionados](#)

Artículo anterior: SQL Server Join In Order

Siguiente artículo: ruby-on-rails – Rails – cómo obtener la URL de dominio de búsqueda en un diseño

Derechos de autor © 2016-2018 Código de registro. Todos los derechos reservados.
Adoptando con orgullo WordPress | Acerca de Blog