### Центральность (CENTRALITY)

В теории графов и в теории сетей центральностью называют меру важности вершины (или узла). В сетях этот параметр можно рассматривать как общее количество связей данного узла по отношению к другим.

[сюда дописать одно-два предложения с какой-нибудь хренью]

#### Betweenness centrality

Описание.

Betweenness является одним из подвидов центральности вершины (или ребра) и показывает количество кратчайших путей, проходящих через эту вершину (или ребро). В русскоязычной литературе нет единого перевода этого термина, каждый исследователь переводит его, делая поправку на контекст своей работы. «Betweenness» интерпретируется как “максимальная взвешенная загруженность” или “нагрузка узла”, “центрированность промежуточных связей” или просто “промежуточная центральность”. Поскольку в данной работе нет контекста для этого термина, то и переводить его не будем.

Значение “betweenness” для вершины определяется по формуле:

где – число кратчайших путей из вершины *i* в вершину *j* через вершину *v,* а – общее число кратчайших путей между всеми парами вершин *i* и *j.*

Значение betweenness” для ребер:

где – число кратчайших путей между всеми парами вершин *i* и *j,*  проходящими через ребро *e*.

Сервис состоит из нескольких функций и позволяет найти:

* «Betweenness», найденная для вершин (или одной вершины);
* Оценочное значение «betweenness» для вершин;
* «Betweenness», найденная для ребер (или для одного ребра);
* Оценочное значение «betweenness» для ребер.

Оценочное значение «betweenness» учитывает только кратчайшие пути длиной, указанной в параметре cutoff или менее. Для больших графов, в которых для подсчета значения «betweenness» требуется слишком много времени, функция, находящая оценочное значение становится весьма полезной.

Для вычисления «betweenness» используется алгоритм, подобный алгоритму, предложенному Брандесом. [Brandes]

Полученное распределение значений для вершин или ребер можно построить в виде графика с помощью сервиса *Plot distribution*, объединив их в workflow-редакторе.

Функция.

1. betweenness(graph, v, directed, weights, nobigint, normalized)
2. edge.betweenness(graph, e, directed, weights)
3. betweenness.estimate(graph, vids, directed, cutoff,

weights,nobigint)

1. edge.betweenness.estimate(graph, e, directed, cutoff,

weights)

Входные параметры.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Иденти-фикатор** | **Имя параметра** | **Описание** | **Тип** | **Допусти-мые значе-ния** | **Значе-ние по умол-чанию** | **Обяза-тель-ность параметра** |
| function | Function | Одна из четырех функций для произведения необходимых вычислений:   * betweenness (1), * edge.betweenness (2), * betweenness.estimate (3), * edge.betweenness.estimate (4) | string | Указа-ны в описа-нии | - | Да |
| graph | Graph (1), (2), (3), (4) | Граф, загруженный в систему, для последующей работы с ним | input file | - | - | Да |
| v | Vertex for calculation (1), (3) | Вершина, для которой нужно вычислить betweenness | integer | - | V(g) – множество всех вершин в графе | Нет |
| e | Edge for calculation (2), (4) | Ребро (его номер), для которого нужно вычислить betweenness | integer | - | E(g) – множество всех ребер в графе | Нет |
| directed | Consider directed paths or not (ignored for undirected graph) (1), (2), (3), (4) | Логический параметр, учитывать ориентацию путей или считать граф неориентированным | logical | true, false | true | Нет |
| normalized | Normalize the betweenness scores or not (1) | Нормировать полученные значения betweenness или нет | logical | true, false | false | Нет |
| weights | Vector with weights for each of edges (1), (2), (3), (4) | Вектор, в котором содержатся веса для каждого из ребер графа. | numeric vector | vector, NULL | NULL | Нет |
| nobigint | Use big integers in calculations or not? (1), (3) | Использовать ли большие числа в вычислениях, если true, то таковые не используются. Параметр применим только для решеток, где имеется огромное количество кратчайших путей между парой вершин. | logical | true, false | true | Нет |
| cutoff | The constraint of maximum path length (3), (4) | Максимальная длина, при которой считается значение “betweenness”.  Если значение = 0 или < 0, то будет вычисляться не оценочное, а обычное значение “betweenness”. | number | - | - | Да |

Комментарии.

* Если полученное значение “betweenness” нужно нормировать (normalized = true), то результаты будут отнормированы в соответствии со следующим выражением:

где В – исходное значение, а Вn – нормированное, n – число вершин в графе.

[Freeman]

Выход (output).

1. Число или вектор чисел, в зависимости от количества вершин, для которых надо провести вычисления.
2. Число или вектор чисел, в зависимости от количества ребер, для которых надо провести вычисления.
3. Число или вектор чисел с оценочными значениями “betweenness” для вершин.
4. Аналогично (3) – для ребер.

#### Closeness centrality

Описание.

Чаще всего этот термин переводится как «центральность по близости», «плотность», «компактность» или попросту «близость», т.е. то, как близко расположена вершина по отношению к другим вершинам, как быстро может распространиться информация из одного узла по всем другим.

Closeness centrality для вершины вычисляется по следующей формуле:

Т.е. она равна обратному значению к сумме длин кратчайших путей ко всем вершинам графа. [Sabidussi] Если же между вершинами *v* и *i* нет пути, то вместо длины пути в формуле будет использовано общее количество вершин.

Сервис состоит из нескольких функций и позволяет найти:

* Близость, найденная для вершин (или одной вершины);
* Оценочное значение близости для вершин (учитывает только кратчайшие пути длиной, указанной в параметре cutoff или менее).

Функция.

1. closeness(graph, vids, mode, weights, normalized)
2. closeness.estimate(graph, vids, mode, cutoff, weights)

Входные параметры.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Иденти-фикатор** | **Имя параметра** | **Описание** | **Тип** | **Допусти-мые значе-ния** | **Значе-ние по умол-чанию** | **Обя-за-тель-ность параметра** |
| function | Function | Одна из двух функций для произведения необходимых вычислений:   * closeness (1), * closeness.estimate (2) | string | Указа-ны в описа-нии | - | Да |
| graph | Graph (1), (2) | Граф, загруженный в систему, для последующей работы с ним | input file | - | - | Да |
| vids | Vertex for calculation (1), (2) | Вершина, для которой нужно вычислить близость | integer | - | V(g) – множество всех вершин в графе | Нет |
| mode | Type of paths to consider (only for directed graph) (1), (2) | Типы путей в графе, которые будем учитывать при нахождении близости. Учитывается только в ориентированных графах. | string | all, out, in, total | total | Нет |
| normalized | Normalize the closeness scores or not (1) | Нормировать полученные значения близости или нет | logical | true, false | false | Нет |
| weights | Vector with weights for each of edges (1), (2) | Вектор, в котором содержатся веса для каждого из ребер графа. | numeric vector | vector, NULL | NULL | Нет |
| cutoff | The constraint of maximum path length (2) | Максимальная длина пути, при которой считается оценочное значение близости.  Если значение = 0 или < 0, то будет вычисляться не оценочное, а обычное значение. | number | - | - | Да |

Комментарии.

* В параметре mode нужно указать типы путей в графе, которые нужно учитывать. Всего возможно три варианта для ориентированного графа: путь к вершине должен содержать ребро, входящее в эту вершину (“in”), выходящее из вершины (“out”) или любое из этих двух (“all”). “total” – значение для неориентированных графов.
* Нормирование производится при помощи умножения на (*n-1*), где *n* – число вершин.

Выход (output).

1. и (2) Число (для одной вершины) или вектор чисел со значениями близости.

#### Eigen-vector centrality

Описание.

[…]

Функция.

evcent (graph, directed, scale, weights)

Входные параметры.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Иденти-фикатор** | **Имя параметра** | **Описание** | **Тип** | **Допусти-мые значе-ния** | **Значе-ние по умол-чанию** | **Обя-за-тель-ность параметра** |
| graph | Graph | Граф, загруженный в систему, для последующей работы с ним | input file | - | - | Да |
| directed | Consider directed paths or not (ignored for undirected graph) | Логический параметр, учитывать ориентацию путей или считать граф неориентированным | logical | true, false | false | Нет |
| scale |  |  | logical | true, false | true | Нет |
| weights | Vector with weights for each of edges | Вектор, в котором содержатся веса для каждого из ребер графа. | numeric vector | vector, NULL | NULL | Нет |

#### Alpha centrality

Описание.

[…]

Функция.

alpha.centrality(graph, nodes, alpha, loops, exo, weights,

tol=1e-7, sparse=TRUE)

Функция по умолчанию использует допустимое для системы значение “tolerance” для линейных зависимостей в столбцах исследуемой матрицы. А также использует разреженные матрицы для вычислений.

Входные параметры.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Иденти-фикатор** | **Имя параметра** | **Описание** | **Тип** | **Допусти-мые значе-ния** | **Значе-ние по умол-чанию** | **Обя-за-тель-ность параметра** |
| graph | Graph | Граф, загруженный в систему, для последующей работы с ним | input file | - | - | Да |
| nodes | Vertex for calculation | Вершина, для которой нужно вычислить alpha centrality. По умолчанию значения считаются для каждой из вершин в графе. | integer | - | V(g) – множество всех вершин в графе | Нет |
| alpha |  |  | number | - | 1 | Нет |
| loops |  |  | logical | true, false | false | Нет |
| weights | Vector with weights for each of edges | Вектор, в котором содержатся веса для каждого из ребер графа. | numeric vector | vector, NULL | NULL | Нет |
| exo1 | Exogenous factor (if it is one number) |  | number | - | 1 | Нет |
| exo2 | Exogenous factor (if it is a vector) | (Функция выбирает один из параметров exo1 или exo2, в зависимости от того какой параметр: число или вектор чисел был задан) | numeric  vector | - | - | Нет |

#### Bonacich power centrality

Описание.

[…здесь нужно понять что это за именная центральность и зачем он ее выдумал, в каких случаях полезна]

Функция.

bonpow(graph, nodes, loops, exponent, rescale,

tol=1e-7, sparse=TRUE)

Функция по умолчанию использует допустимое для системы значение “tolerance” для линейных зависимостей в столбцах исследуемой матрицы. А также использует разреженные матрицы для вычислений.

Входные параметры.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Иденти-фикатор** | **Имя параметра** | **Описание** | **Тип** | **Допусти-мые значе-ния** | **Значе-ние по умол-чанию** | **Обя-за-тель-ность параметра** |
| graph | Graph | Граф, загруженный в систему, для последующей работы с ним | input file | - | - | Да |
| nodes | Vertex for calculation | Вершина, для которой нужно вычислить alpha centrality. По умолчанию значения считаются для каждой из вершин в графе. | integer | - | V(g) – множество всех вершин в графе | Нет |
| loops |  |  | logical | true, false | false | Нет |
| exponent |  |  | number | - | 1 | Нет |
| rescale |  |  | logical | true, false | false | Нет |