

Описание на част от функциите, имплементирани на езика Matlab

- `d_L(a,b)`

Аргументи: a и $b \rightarrow$ пермутации на първите n естествени числа, където $n > 0$ е произволно цяло число;

Описание: като резултат от изпълнението на функцията се връща разстоянието на Лий (Lee distance) между пермутациите a и b .

```
function [res]=d_L(a,b)
    n=length(a);
    res=sum(min(abs(a-b),n-abs(a-b)));
end
```

- `max_d(n,d_name)`

Аргументи: $n \rightarrow$ естествено число ($n > 0$);

$d_name \rightarrow$ наименование на разстояние (метрика) между пермутации;

възможни стойности: `@d_C` \leftrightarrow Cayley's distance

`@d_F` \leftrightarrow Spearman's footrule

`@d_H` \leftrightarrow Hamming distance

`@d_K` \leftrightarrow Kendall's tau

`@d_L` \leftrightarrow Lee distance

`@d_M` \leftrightarrow Chebyshev metric

`@d_R` \leftrightarrow Spearman's rho

`@d_U` \leftrightarrow Ulam's distance

Описание: пресмята възможно най-голямото разстояние между две пермутации на първите n естествени числа за дадена метрика d_name .

```
function [res]=max_d(n,d_name)
    res=0;
    p=perms(1:n);
    for i=1:factorial(n)
        res=max(res,d_name(1:n,p(i,:)));
    end
end
```

- `pmf(n,d_name)`

Аргументи: $n \rightarrow$ естествено число ($n > 0$);

$d_name \rightarrow$ наименование на разстояние (метрика) между пермутации;

възможни стойности: както при `max_d(n,d_name)`

Описание: пресмята вектора от вероятностите (probability mass function) на случайната величина, получена от метриката d_name при случайно избрани две пермутации на първите n естествени числа.

```
function [res]=pmf(n,d_name)
    res=zeros(1,max_d(n,d_name)+1);
    p=perms(1:n);
    for i=1:factorial(n)
        d=d_name(1:n,p(i,:));
        res(d+1)=res(d+1)+1;
    end
    res=res./factorial(n);
end
```

- mgf(n,t,d_name)

Аргументи: $n \rightarrow$ естествено число ($n > 0$);

$t \rightarrow$ реално число;

$d_name \rightarrow$ наименование на разстояние (метрика) между пермутации;

възможни стойности: както при $max_d(n,d_name)$

Описание: пресмята стойността на пораждащата моментите функция (moment generating function) в точката t на случайната величина, получена от метриката d_name при случайно избрани две пермутации на първите n естествени числа.

```
function [res]=mgf(n,t,d_name)
    res=sum(pmf(n,d_name).*exp((0:max_d(n,d_name)).*t));
end
```

- psi_t(n,t,d_name)

Аргументи: $n \rightarrow$ естествено число ($n > 0$);

$t \rightarrow$ реално число;

$d_name \rightarrow$ наименование на разстояние (метрика) между пермутации;

възможни стойности: както при $max_d(n,d_name)$

Описание: пресмята стойността на нормиращата константа в точката t за модела на Малоус (Mallows' model), породен от случайната величина, получена от метриката d_name при случайно избрани две пермутации на първите n естествени числа.

```
function [res]=psi_t(n,t,d_name)
    m=max_d(n,d_name);
    prob=pmf(n,d_name);
    res=log(factorial(n)*sum(prob.*exp((0:m).*t)));
end
```

- k_cluster_data(data,k,d_name)

Аргументи: $data \rightarrow$ набор от наредени данни; вектор от пермутации

$k \rightarrow$ естествено число ($k > 0$); брой кълстери в данните;

$d_name \rightarrow$ наименование на разстояние (метрика) между пермутации;

възможни стойности: както при $max_d(n,d_name)$

Описание: обхождайки всички възможни случаи (brute-force search) намира k пермутации (cluster centroids), за които се минимизира сумата от разстоянията между наблюденията (data) и най-близкия център, т.е. резултатът от изпълнението на функцията съответства на прилагането на k -means clustering за наредените данни data.

```
function [res,modes,count]=k_cluster_data(data,k,d_name)
    [~,n1]=size(data);
    n=n1-1;
    p=perms(1:n);
    c=nchoosek(1:factorial(n),k);
    res=sum_min_data(data,p([c(1,:)],:),d_name);
    count=1;
    for i=2:nchoosek(factorial(n),k)
        res0=sum_min_data(data,p([c(i,:)],:),d_name);
        if res0<res
            count=1;
            modes=p([c(i,:)],:);
        end
        if res0==res
            count=count+1;
        end
        res=min(res,res0);
    end
end
```

- LRS_latent(data,modes,proportions,theta,d_name)

Аргументи: data → набор от наредени данни; вектор от пермутации

modes → вектор от пермутации; модални наредби;

proportions → вектор от числа между 0 и 1, чиито елементи се сумират до 1;

theta → вектор от реални числа;

d_name → наименование на разстояние (метрика) между пермутации; възможни стойности: както при $max_d(n,d_name)$

Описание: намира статистиката на отношение на правдоподобия (likelihood ratio statistic) на наблюденията data за обобщения модел на Малоус (Latent Mallows' model) с параметри modes, proportions и theta.

```
function [res]=LRS_latent(data,modes,proportions,theta,d_name)
    [data_size,n1]=size(data);
    n=n1-1;
    [~,k]=size(proportions);
    res=0;
    psi_k=zeros(1,k);
    for j=1:k
```

```

    psi_k(j)=psi_t(n,theta(j),d_name);
end
for i=1:data_size
    y=data(i,1:n);
    s=0;
    for j=1:k
        s=s+p(j)*exp(theta(j)*d_name(modes(j,:),y)-psi_k(j));
    end
    res=res+data(i,n1)*log(s);
end
res=2*(res+sum(data(:,n1))*log(factorial(n)));
end

```

- EM_latent(data,d_name,error,m_initial,p_initial,t_initial)

Аргументи: data → набор от наредени данни; вектор от пермутации

error → реално положително число;

d_name → наименование на разстояние (метрика) между пермутации;

въможни стойности: както при $max_d(n,d_name)$

m_initial → вектор от пермутации;

p_initial → вектор от числа между 0 и 1, чиито

елементи се сумират до 1;

t_initial → вектор от реални числа;

Описание: намира максимално правдоподобни оценки за параметрите на обобщения модел на Малоус (Latent Mallows' model), базирани на наблюденията data. Функцията прилага обобщения ЕМ-алгоритъм с първоначални оценки m_initial, p_initial и t_initial и критерий за прекратяване на итерациите, при който най-голямата разлика в две последователни оценки на параметрите не надвишава error.

```

function [m,p,t]= EM_latent(data,d_name,error,m_initial,p_initial,t_initial)
    m=m_initial;
    p=p_initial;
    t=t_initial;
    [~,k]=size(p_initial);
    e=zeros(1,3*k)+1;
    while max(e)>error
        [m_new,p_new,t_new]=EM_latent_iteration(data,m,p,t,d_name);
        e(1:k)=abs(p-p_new);
        e((k+1):2*k)=abs(t-t_new);
        e((2*k+1):3*k)=abs(m-m_new);
        m=m_new;
        p=p_new;
        t=t_new;
    end
end

```