

Собственные значения и собственные векторы

Основные определения

Ненулевой вектор v называется **собственным вектором (eigenvector)** квадратной матрицы A , если выполняется равенство

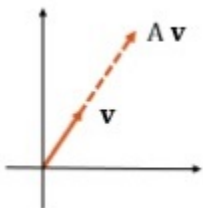
$$A \cdot v = \lambda \cdot v,$$

где λ – некоторый скаляр, называемый **собственным значением** (или **собственным числом, eigenvalue**) матрицы A , соответствующим собственному вектору v .

Если v – некоторый собственный вектор, то вектор kv ($k \neq 0$) также является собственным вектором.

Геометрический смысл собственного вектора: умножение собственного вектора v на матрицу A дает вектор λv , коллинеарный вектору v . При этом его длина изменяется в λ раз.

Если A рассматривать как матрицу некоторого линейного оператора, то **собственный вектор** этого оператора – любой ненулевой вектор v , который оператором A отображается в коллинеарный ему вектор λv , а соответствующий скаляр λ называется **собственным значением** оператора.



Подставляя $A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$ и $v = \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{pmatrix}$ в соотношение $Av = \lambda v$, получим однородную систему линейных уравнений относительно неизвестных v_1, v_2, \dots, v_n :

$$\begin{cases} (a_{11} - \lambda)v_1 + a_{12}v_2 + \dots + a_{1n}v_n = 0 \\ a_{21}v_1 + (a_{22} - \lambda)v_2 + \dots + a_{2n}v_n = 0 \\ \dots \\ a_{n1}v_1 + a_{n2}v_2 + \dots + (a_{nn} - \lambda)v_n = 0 \end{cases}.$$

Эта система имеет ненулевое решение, если ее определитель равен нулю:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{vmatrix} = 0.$$

Это уравнение называется **характеристическим уравнением**. Раскрывая определитель, получаем многочлен n -й степени относительно λ :

$$\lambda^n + p_1\lambda^{n-1} + \dots + p_{n-1}\lambda + p_n = 0.$$

Корни характеристического уравнения и являются собственными значениями матрицы. Для нахождения собственных векторов нужно каждое собственное значение λ_k подставить в однородную систему уравнений и решить ее. Эти системы будут иметь бесконечное множество решений, но для каждого λ_k достаточно найти только одно ненулевое решение.

Собственные векторы и собственные значения широко используются в различных приложениях линейной алгебры. Это связано с тем, что многие соотношения существенно упрощаются в системе координат, построенной в базисе из собственных векторов. А множество собственных значений линейного оператора характеризует важные свойства оператора без привязки к какой-либо системе координат.

Нахождение собственных значений и собственных векторов в Engeer

В **Engeer** собственные значения и собственные векторы матрицы A вычисляются с помощью функции `eigen`, содержащейся в библиотеке `LinearAlgebra`. Эта функция имеет два выходных параметра: вектор собственных значений матрицы A `val` и матрица `vec`, столбцы которой являются собственными векторами матрицы A .

Пример. Найдем в **Engeer** собственные значения и собственные векторы матрицы $A = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix}$:

```
[ ]: using LinearAlgebra;
A = [0 1; -2 -3];
val, vec = eigen(A);
println("eigenvalues =");
display(val);
println("eigenvectors =");
display(vec);
```

Таким образом, матрица A имеет два собственных значения $\lambda_1 = -2$ и $\lambda_2 = -1$ и два соответствующих им собственных вектора $v_1 = \begin{pmatrix} -0,447214 \\ 0,894427 \end{pmatrix}$ и $v_2 = \begin{pmatrix} 0,707107 \\ 0,707107 \end{pmatrix}$.

Сделаем проверку. Вычислим разности $Av_1 - \lambda_1 v_1$ и $Av_2 - \lambda_2 v_2$ и убедимся в том, что они равны нулевым векторам.

```
[ ]: display(A * vec[:,1] - val[1] * vec[:,1]);
display(A * vec[:,2] - val[2] * vec[:,2]);
```

⇒Задание 1

Найдите в **Engge** собственные значения и собственные векторы матрицы $A =$

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

Сделайте проверку.

[]:

####

Решение

[]:

```
using LinearAlgebra;
A = [1 -1 0; -1 2 -1; 0 -1 1];
val, vec = eigen(A);
println("eigenvalues =");
display(val);
println("eigenvectors =");
display(vec);
display(A * vec[:,1] - val[1] * vec[:,1]);
display(A * vec[:,2] - val[2] * vec[:,2]);
display(A * vec[:,3] - val[3] * vec[:,3]);
```

⇒Задание 2

Найдите в **Engge** собственные значения и собственные векторы диагональной

матрицы $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}.$

Сделайте проверку.

[]:

####

Решение

[]:

```
using LinearAlgebra;
A = [1 0 0; 0 2 0; 0 0 3];
val, vec = eigen(A);
println("eigenvalues =");
display(val);
println("eigenvectors =");
```

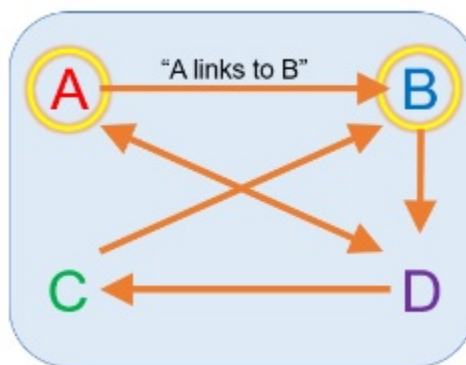
```
display(vec);
display(A * vec[:,1] - val[1] * vec[:,1]);
display(A * vec[:,2] - val[2] * vec[:,2]);
display(A * vec[:,3] - val[3] * vec[:,3]);
```

Как мы видим, собственные значения диагональной матрицы совпадают с элементами, лежащими на ее главной диагонали. А собственные векторы образуют линейно независимую систему векторов.

Применение собственных значений и собственных векторов для вычисления рейтинга веб-страниц

Поисковые системы рассчитывают рейтинг веб-страниц в соответствии с популярностью каждой страницы.

Рассмотрим четыре веб-страницы A, B, C и D. Они содержат ссылки друг на друга. Например, страница A содержит ссылки на B и D. Взаимосвязь между веб-страницами можно представить в виде графа. Стрелка между A и B означает, что страница A содержит ссылку на страницу B.



Этот граф можно представить в виде **матрицы переходов A**:

	A	B	C	D
A	0.1	0	0	0.45
B	0.45	0.1	0.9	0
C	0	0	0.1	0.45
D	0.45	0.9	0	0.1

Каждый элемент матрицы a_{ij} – это вероятность перехода пользователя со страницы i на страницу j . Сумма вероятностей в каждом столбце всегда равна единице. Вероятность 0 означает, что ссылка между страницами i и j отсутствует.

Диагональные элементы, равные 0,1, означают, что существует вероятность 0,1 того, что пользователь останется на той же странице.

Пусть вектор-столбец $c = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ с единицей в третьей строке означает пользователя, который начинает свою навигацию со страницы C.

Зададим в **Engge** матрицу A и вектор c . Умножив матрицу A на вектор c , получим вектор $cProb$ вероятностей перехода пользователя со страницы C на другие страницы:

```
[ ]: A = [0.1 0 0 0.45;0.45 0.1 0.9 0;0 0 0.1 0.45;0.45 0.9 0 0.1];  
c = [0; 0; 1; 0];  
cProb = A * c
```

Вектор $cProb$ показывает, что пользователь, находящийся на странице C, с вероятностью 90% перейдет на страницу B и с вероятностью 10% останется на странице C.

Если повторить эту операцию много раз ($A \cdot A \cdot \dots \cdot A \cdot c$), то в результате получится **доминирующий собственный вектор** (соответствующим наибольшему по модулю собственному значению) матрицы A .

Элементы этого вектора дают вероятность того, что пользователь окажется на определенной странице после навигации по страницам в течение бесконечного времени.

Найдем собственные значения и собственные векторы матрицы A :

```
[ ]: val, vec = eigen(A)
```

Два собственных значения комплексные; в данной задаче их можно игнорировать.

Два других собственных значения равны 1 и 0,1. Доминирующий собственный вектор (4-й столбец матрицы vec) соответствует наибольшему собственному значению (1).

Сохраним доминирующий собственный вектор в переменной $v1$:

```
[ ]: v1 = vec[:,4]
```

Чем больше значения элементов собственного вектора, тем более высокий рейтинг будет у этой страницы. Т.е. у пользователя больше всего шансов оказаться на странице D.

Напомним, что произведение собственного вектора на число также является собственным вектором. Если промасштабировать вектор $v1$ так, чтобы сумма его элементов равнялась 1, то в результате получится вектор вероятностей того, что пользователь окажется на каждой из четырех страниц по прошествии бесконечного времени.

Разделив вектор `v1` на сумму его элементов, получим вектор вероятностей `v1Prob`:

```
[ ]: v1Prob = v1 / sum(v1)
```

Таким образом, пользователь, переходящий между 4 страницами, имеет вероятность приблизительно 18% оказаться на страницах А и С, 27% – на странице В и 36% – на странице D.

⇒Задание 3

Найдите в **Engge** вероятности оказаться на каждой из 4 страниц по прошествии бесконечного времени, если матрица переходов имеет вид: $A =$

$$\begin{pmatrix} 0,1 & 0 & 0 & 0 \\ 0,5 & 0,5 & 0,5 & 0,5 \\ 0 & 0 & 0,5 & 0,45 \\ 0,4 & 0,5 & 0 & 0,05 \end{pmatrix}.$$

```
[ ]:
```

####

Подсказка

Сначала задайте матрицу переходов A . Затем найдите ее собственные значения и собственные векторы с помощью выражения `val, vec = eigen(A)`. Сохраните в переменной `v1` доминирующий собственный вектор – тот столбец матрицы `vec`, который соответствует наибольшему по модулю собственному значению (т.е. наибольшему по модулю элементу вектора `val`). Затем разделите доминирующий вектор на сумму его элементов, в результате получится вектор искомых вероятностей.

#####

Решение

```
[ ]: A = [0.1 0 0 0; 0.5 0.5 0.5 0.5; 0 0 0.5 0.45; 0.4 0.5 0 0.05];  
val, vec = eigen(A)
```

Как мы видим, наибольшее по модулю собственное значение равно единице (4-й элемент вектора `val`). Ему соответствует 4-й собственный вектор (доминирующий вектор). Сохраним его в переменной `v1` и пронормируем, разделив на сумму его элементов:

```
[ ]: v1 = vec[:,4];  
v1Prob = v1 / sum(v1)
```

Таким образом, пользователь имеет вероятность 50% оказаться на странице В, приблизительно 24% – на странице С, 26% – на странице D и нулевую вероятность – на странице А.

Тест для получения сертификата

Пройти тест по разделу »