Метод главных компонент (PCA)

Математически обоснованный метод. Это классика, он присутствует как программа во всех без исключения пакетах и библиотеках для всех языков в том числе для Python, R.

Есть система координат в n-мерном пространстве

$$\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n$$

и эти координатные оси обладают свойствами:

1. перпендикулярны друг к другу;
2. $\vec{e}_i^T \vec{e}_j = 0, i \neq j$;
3. каждая из осей имеет длину = 1 $\vec{e}_i^T \vec{e}_i = 1$.

В этой системе координат у нас задано N точек, каждая точка задана вектором.

$$\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N$$

Мы эти векторы:

1. отцентрировали, т.е. вычли из них среднее по всем векторам, т.о. среднее стало равно 0, стрелка над нулем обозначает, что это n штук нулей, это вектор состоящий из нулевых координат.

$$\frac{1}{N} \sum_{i=1}^{N} \vec{x}_i = \vec{0}$$

2. отнормировали на длину, каждое $\vec{x}_i$ разделили на сумму квадратов и т.о. длина каждого векторочка стала равна 1

$$\vec{x}_i^T \vec{x}_i = 1$$

Вводится понятие веса каждой оси и под весом понимается дисперсия проекции на эту ось всех наших точек, т.е. суммирование идет по j от единицы до N, $\vec{e}_i$ – одна и та же ось для которой мы вычисляем вес, а $\vec{x}_j$ бегает по всем векторочкам.

$$\varphi(\vec{e}_i) = \sigma_i^2 = \frac{1}{N} \sum_{j=1}^{N} (\vec{e}_i^T \vec{x}_j)^2$$

Среднее для проекции будет равно 0, поэтому мы его не вычитаем из $\vec{e}_i^T \vec{x}_j$, остается сумма квадратов проекций.

Нас волнует на сколько эти веса неравномерно распределены или наоборот – равномерно по осям координат и мерой равномерности/неравномерности является энтропия, т.е. суммы весов умноженные на логарифмы весов в принципе, т.к. мы будем для разных систем координат считать энтропию, нам напревать по какому основанию логарифм берется, но т.к. энтропия считалась ранее для бинарных данных и связывалась информацией передаваемой по каналу и там был двоичный логарифм, то и мы для определенности будем использовать двоичный.

$$H\left\{\vec{e}_1,\vec{e}_2,\ldots,\vec{e}_n\right\}=-\sum_{i=1}^{n}\varphi(\vec{e}_i)\times\log_2\varphi(\vec{e}_i)$$

Чем меньше энтропия, тем более неравномерно распределены веса координат.

Переходим к решению.

Доказывается математически, что минимуму энтропии соответствует система координат составленная из собственных векторов ковариационной матрицы.

$C\vec{e}=\lambda\vec{e}$, где **C** – ковариационная матрица.

Если мы умножаем какой-то вектор на матрицу, то мы получаем другой вектор в общем случае, который не совпадает с первым, не только по длине, а самое главное по направлению.

Если речь идет о собственном векторе, то его умножение на матрицу эквивалентно умножению его на какой-то скаляр, т.е. у этого вектора останется то же направление, как у исходного, изменится длина в зависимости от того, кто такой **λ** и **λ** называется собственным числом, а $\vec{e}$ – собственный вектор.

Если мы вычислим у квадратной матрицы **n**x**n** имеется n штук разных собственных векторов, каждому из которых соответствует свое собственное число.

Когда все **n** собственных векторов определим $\vec{e}_1,\vec{e}_2,\ldots,\vec{e}_n$. Расположим их в порядке убывания собственных чисел $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_n$, а перед тем, как располагать, обратим внимание на то, что собственное число равно дисперсии проекций на соответствующую ось $\sigma_i^2=\frac{1}{N}\sum_{j=1}^{N}(\vec{e}_i^T\vec{x}_j)^2$, отбросим k штук с наименьшими весами, т.е. имеющие наименьшие собственные числа, то ошибка представления исходного множества точек может быть записано в процентах $\varepsilon^2=100\times\frac{\sum_{i=1}^{n-k}\lambda_i}{\sum_{i=1}^{n}\lambda_i}\%$ (сумму от 1 до (n-k) – сколько у нас осталось собственных чисел, делим на сумму всех собственных чисел, умножим на 100, вот это вот в процентах наша ошибочка представления исходных данных.)

Рассмотрим примеры:

Есть ковариационная матрица, чтобы не мучиться с коэффициентами ковариации, мы рассмотрим корреляционную матрицу $C=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$, т.е. у нас двумерное пространство e_1 и e_2, есть какой-то набор **X** и коэф. корреляции между x_1 и x_2 = 1, так получилось.

Записываем ур-е для собственного числа и вектора

$$C\vec{e} = \lambda\vec{e}$$

Переносим содержащее неизвестное в левую часть выносим $\vec{e}$, которое присутствует в обоих слагаемых за скобочки. Т.к. у нас остался скаляр **λ**, мы не можем его просто отнять от матрицы C, в линейной алгебре, если надо отнять скаляр, то он отнимается от диагональных элементов матрицы. Поэтому **λ** умножим на **I**, где **I** – единичная матрица(по диагонали единицы, вне диагонали нули) $(C - \lambda I)\vec{e} = 0$.

Если мы это дела раскроем, то получим систему линейных ур-ний, относительно неизвестных нам e_1 и e_2 и неизвестного **λ**, эта система однородная (правые части равны 0). Чтобы однородная система имела решение, ее определитель должен = 0, т.е. $\left|C - \lambda I\right| = 0$.

Расписываем:

$$\det\begin{bmatrix} 1-\lambda & 1 \\ 1 & 1-\lambda \end{bmatrix} = (1-\lambda)^2 - 1 = 0$$

(C-λI) в нашем случае = $\begin{matrix} 1-\lambda & 1 \\ 1 & 1-\lambda \end{matrix}$, теперь считаем определитель этой матрицы.

Произведение по основной диагонали это $(1-\lambda)^2$ минус произведение по дополнительной диагонали (это 1), и это все равно 0.

Решаем получившееся квадратное ур-е, видим два решения:

$$\lambda_1 = 2$$
$$\lambda_2 = 0$$

Сколько у нас разных неизвестных? Две штуки, потому что размерность пространства у нас 2, размер матрицы ковариационной 2x2. Отсюда два собственных числа получились.

Подставляем первое **λ** в систему ур-ний $(C - \lambda_1 I)\vec{e} = 0$ и решаем ее

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = 0$$

$$-e_1 + e_2 = 0$$
$$e_1 - e_2 = 0$$

Этой системе из двух ур-ний с двумя неизвестными удовлетворяет любое e_1 = e_2 =const.

Теперь обозначим **const** как **c** ($e_1 = e_2 = c$) и вот первый собственный вектор

$$\vec{e}_1 = \begin{bmatrix} c \\ c \end{bmatrix}.$$

Как определять как брать **C**? Глубоко плевать, но для определенности возьмем его таким, чтоб длина e_1 =1.

Разделим **C** на длину $\vec{e}_1 = \begin{bmatrix} c \\ c \end{bmatrix}$ =c^2+c^2=2c^2, извлекаем корень c2^.5, поделим на длину все компоненты, получим $\vec{e}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Проверка APL:

```
      (2 2⍴1)+.×,['']÷2 2*.5
1.414213562
1.414213562
      ÷2 2*.5
0.7071067812 0.7071067812
      2×÷2 2*.5
1.414213562 1.414213562
```

Теперь подставляем второе λ = 0, ищем второй собственный вектор аналогично первому.

$$(C - \lambda_2 I)\vec{e} = 0$$

Получаем два линейных ур-ния:

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = 0$$

$$e_1 + e_2 = 0$$

$$e_1 + e_2 = 0$$

Решение - та же константа, то с противоположными знаками

$$e_1 = -e_2$$

$$\vec{e}_2 = \begin{bmatrix} c \\ -c \end{bmatrix}$$

$$\vec{e}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

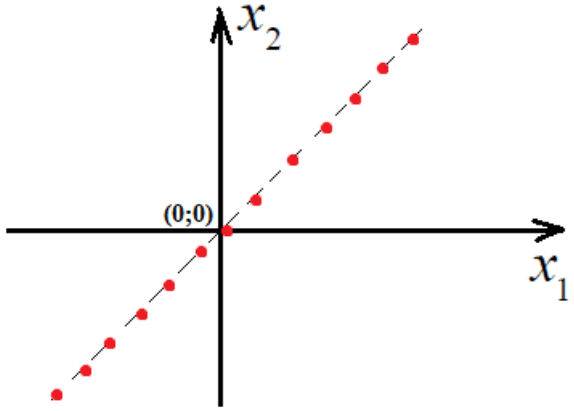
Проверка на APL:

```
      (2 2⍴1)+.×,['']1 ¯1×÷2 2*.5
0
```

Что такое единичная корреляционная матрица, и что такое собственные числа 2 и 0, и что такое собственные векторы $\begin{matrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{matrix}$ и $\begin{matrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{matrix}$?

$$C = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad \rho_{x_1 x_2} = 1$$

Как будут лежать точки x_1, x_2 для $\rho x_1 x_2$=1?



Они лежат на прямой под углом 45 градусов.

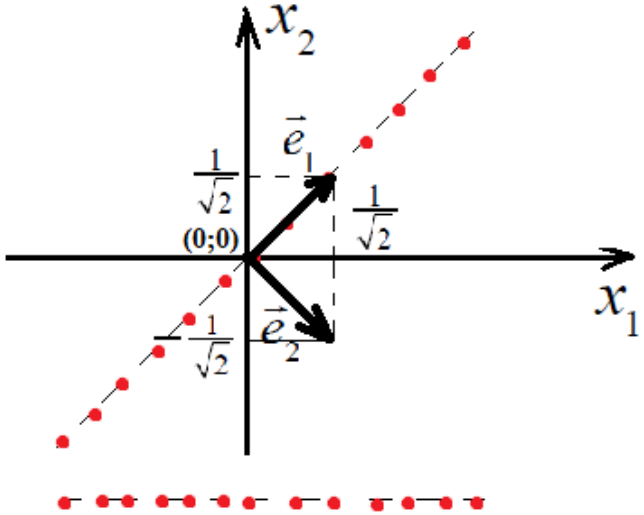
Т.е. x_2 =0+1(коэф.наклона) x x_1=x_1

Далее, среднее у них после центрирования = 0, соответственно коэф. корреляции равен

$$\frac{\frac{1}{N}\Sigma((x_1 - 0) \times (x_2 - 0))}{\Sigma x_1^2 \times \Sigma x_2^2} = \frac{\frac{1}{N}\Sigma (x_1 \times x_2)}{\Sigma x_1^2 \times \Sigma x_2^2} = 1$$ и это будет

$$\rho_{12} \frac{\frac{1}{N}\Sigma(x_1 - \overline{x}_1)(x_2 - \overline{x}_2)}{\Sigma(x_1 - \overline{x}_1)^2 \Sigma(x_2 - \overline{x}_2)^2}$$

Если мы посчитаем проекции на e_1, опустим их так, что если сначала они были под 45 градусов, теперь будут под 0 градусов. e_2 перпендикулярный к e_1. Все проекции на e_2 будут равны 0.



Значит мы можем, если **k** возьмем = 1, т.е. отбросим λ=0, то сумма от единицы до **(n-k)** будет равно 2, сумма от 1 до n =2+0=2, соответственно 2/2x100%=100%, то бишь оставляя первый собственный вектор соответствующий собственному числу мы сохраняем 100% информации о взаимном распределении точек, ничего не теряем. Действительно, накий пес нам играть в две координаты x_1 и x_2, если проекция на e_1- одномерная и все расстояния между точками сохраняются, мы видим всю структуру.

Пример 2:

Корреляционная матрица диагональная. Т.е. коэф. корреляции между x_1 и x_2 = 0. Соответственно, вычисление аналогично:

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C\vec{e} = \lambda\vec{e}$$
$$(C - \lambda I)\vec{e} = 0$$
$$\left|C - \lambda I\right| = 0$$

$$\det\begin{bmatrix} 1-\lambda & 0 \\ 0 & 1-\lambda \end{bmatrix} = (1-\lambda)^2 = 0$$

Решением будет:

$$\lambda_1 = 1$$
$$\lambda_2 = 1$$

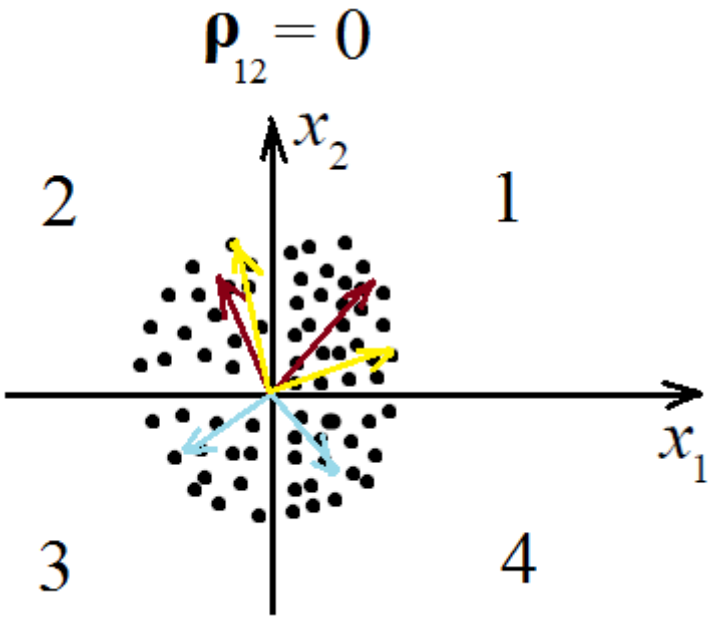
Находим собственные векторы, подставляя первое λ, матрица становится нулевой:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = 0$$

Соответственно любые e_1 и e_2 будут удовлетворять:

$$0 \times e_1 + 0 \times e_2 = 0$$
$$0 \times e_1 + 0 \times e_2 = 0$$

Единственное, что нас волнует, это то, что эти вектора e_1 и e_2 должны быть перпендикулярны между собой:



ρ=0, т.е. точки распределены, грубо говоря, внутри какого-то круга и когда мы считаем 1,2,3,4-квадранты, получаем сумму (X1i x X2i), т.к. у нас $\overline{x_1} = \overline{x_2} = 0$, то в первом квадранте у нас будет ''+'' на ''+'', получим большое положительное, в третьем квадранте будет ''–'' на ''–'' тоже большое положительно число, во втором и четвертом квадрантах ''+'' на ''–'' получим большое отрицательное число.

Итого: положительное(1квадрант)+отрицательное(2квадрант)+ положительное (3квадрант) + отрицательное(4квадрант) = 0

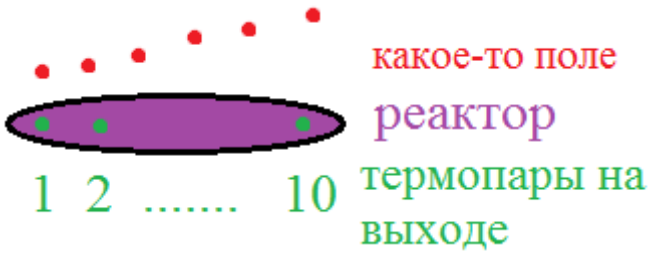
И относительно этого нуля, как не крути систему координат, ничего не изменится!

И если снова обратимся к нашему ур-нию в выкинем теперь второе λ = 1, то 1/(1+1) x 100% =50%

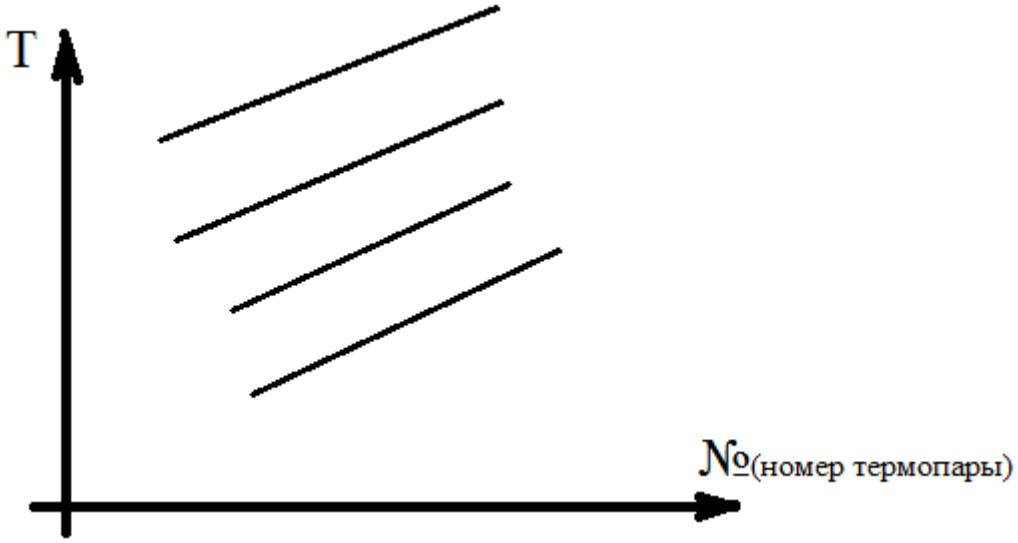
Выкидывая любую из координат x_1 и x_2 , мы теряем 50% информации о распределении наших точек. Например проектируем на красную ось, во все, кто выше или ниже, а вот самые удаленные друг от друга точки сольются в одну точку.

Рассмотрим искусственный примерчик о температурном поле:

Есть реактор, где у нас 1, 2,...., 10 термопары на выходе и какое-то поле, слева холоднее, справа горячее.



Если мы в разные моменты времени проведем измерения, то это поле сохраняя распределение, что слева холоднее, справа горячее, может еще быть все больше, все меньше, т.е. оно не меняя распределения по термопарам, а средняя температура может быть то больше, то меньше.



Функция которая на APL возвращает три результата:

a1 – собственные числа

b1 – матрица собственных векторов

c1 – матрица проекций на собственные вектора исходных данных, в данном случае x1.

```
      x1←⍳10
      x1←⊃(⊂x1)+⍳16
      (a1 b1 c1)←mds.(COVM SELFIC)x1
```

Во-первых, нам очень интересно, какие получились первые 10 чисел

```
      a1
212.5 0 0 0 0 0 0 0 0 0
```

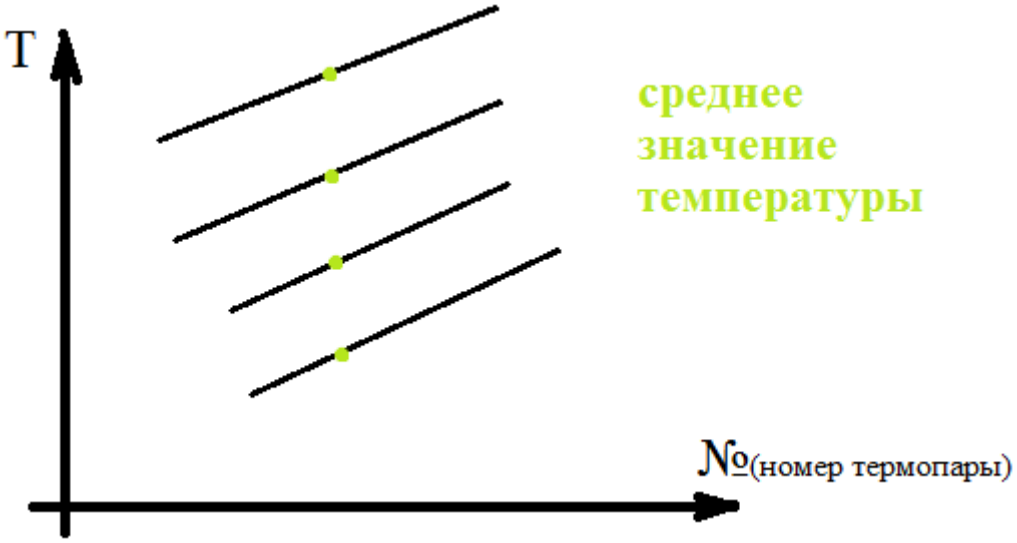
Т.к. матрица ковариационная, то собственные числа нецелочисленные и получилось, что одно собственное число 212.5 отличается от 0, все остальные равны 0.

Если мы отбросим 9 собственных векторов и оставим один, который соответствует первому собственному числу, которое не нулевое. Мы сохраним 100% информации(212.5/(212.5+0+0+0+0+0+0+0+0+0)x100%).

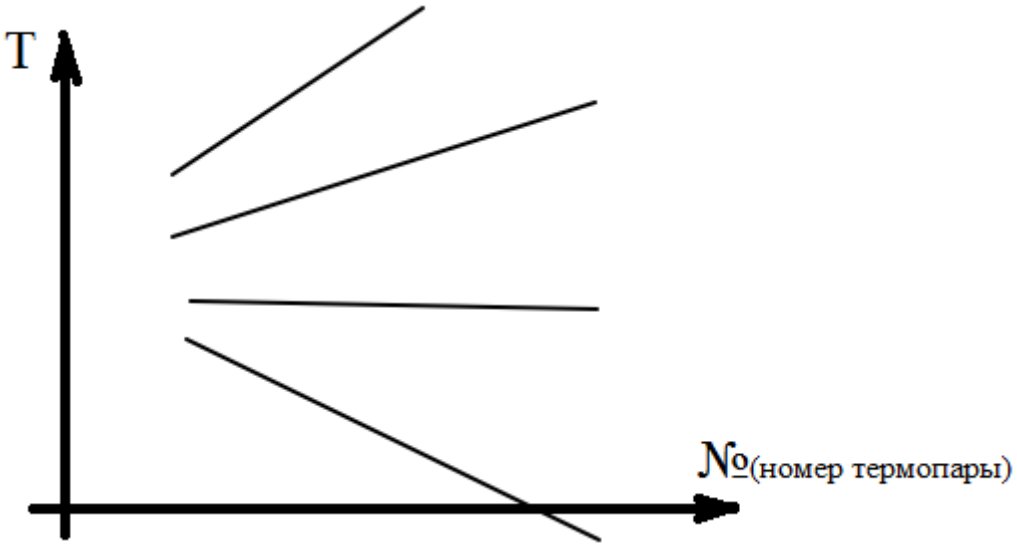
Во-вторых, нам интересно посмотреть на первый собственный вектор, что это за голубь на который мы должны проектировать?

```
      b1[;1]
¯0.316227766 ¯0.316227766 ¯0.316227766 ¯0.316227766 ¯0.316227766
¯0.316227766 ¯0.316227766 ¯0.316227766 ¯0.316227766 ¯0.316227766
```

Первое, что мы должны, что все компоненты этого собственного вектора в 10-мерном пространстве равны между собой по модулю и просто равны. Второе, они все отрицательные, ну и плевать, то бишь, проекции на этот собственный вектор будет (-0.316227766)xT1+(-0.316227766)xT2+...+(-0.316227766)xT10, т.е., если рассматривать эти 0.316227766 несмотря на минус, как веса, то получаем, что в качестве проекции у нас взвешенное с этими весами, одинаковыми сумма всех наших термопар, которые определяют на сколько у нас отличается среднее значение температуры. Т.е. на самом деле у нас не 10-мерное пространство, а одномерное, где средний ур-нь меняется.



Теперь рассмотрим видоизмененный этот пример, где у нас меняется не только средний уровень, но и распределение, т.е.:



В каком-то эксперименте мы одно наблюли, в другом — другое и т.д.

Т.е. у нас меняются две вещи:

1. средняя температура;
2. распределение температур (слева — холодно, справа — горячо или наоборот).

Для этого случая посчитаем собственные числа и векторы.

Во-первых, мы видим, что у нас два собственных числа не равны 0 (первое число и второе число), остальные = 0. Мы можем безболезненно, не теряя информации, понизить нашу размерность пространства с 10-мерной до 2-мерной.

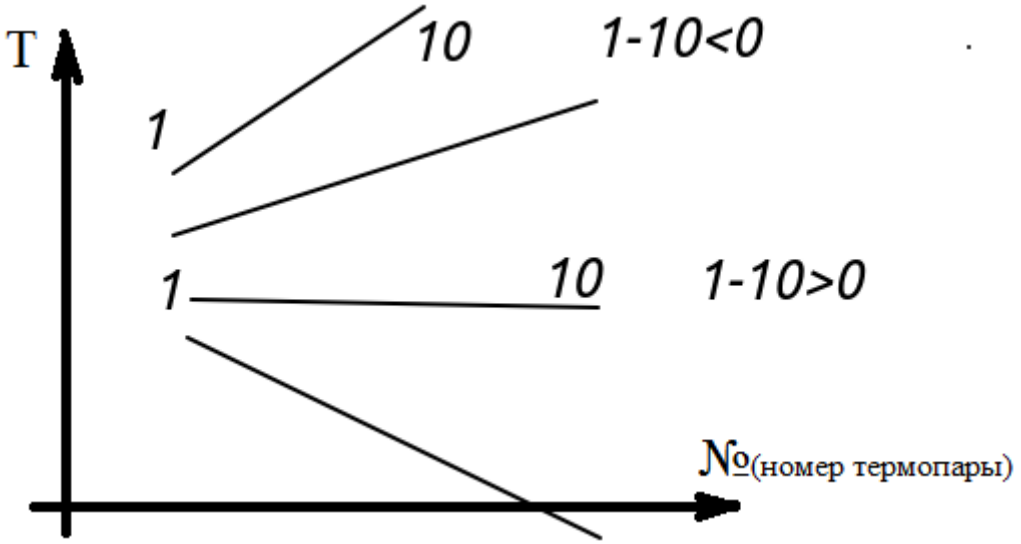
```
      x2←⍳10
      x2←⊃(⊂x2)×∈2⍴⊂⍳8
      x2[⍳8;]←⌽x2[⍳8;]
      (a2 b2 c2)←mds.(COVM SELFIC)x2
      a2
2103.75 1588.125 2.479131558E¯13 8.187234765E¯14 6.036530033E¯14 1.97029597E¯14
1.239658888E¯14 ¯3.340844541E¯14 ¯5.933993521E¯14 ¯1.555394639E¯13
```

Во-вторых, очень интересно на какие собственные векторы мы будем проектировать:

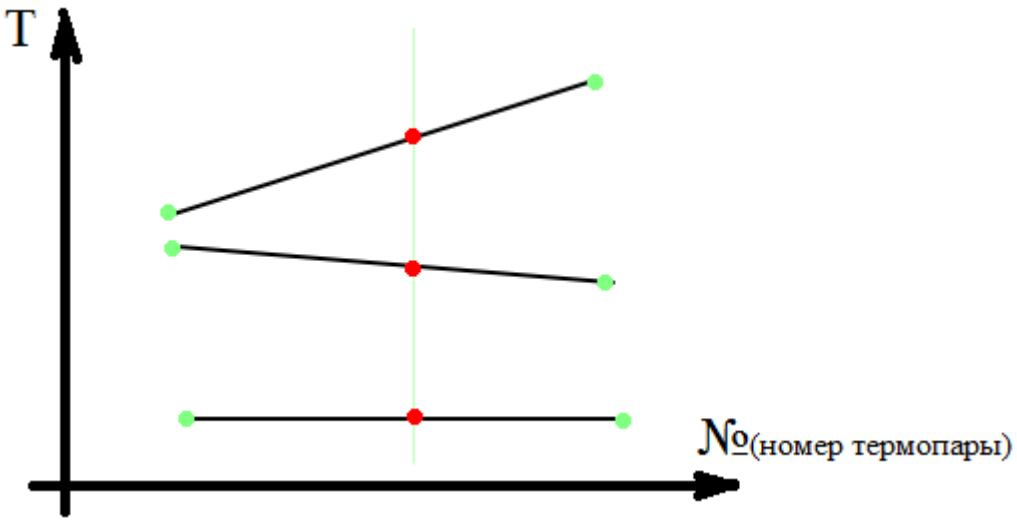
```
      b2[;,2]
0.4954336943        ¯0.316227766
0.3853373178        ¯0.316227766
0.2752409413        ¯0.316227766
0.1651445648        ¯0.316227766
0.05504818826       ¯0.316227766
¯0.05504818826      ¯0.316227766
¯0.1651445648       ¯0.316227766
¯0.2752409413       ¯0.316227766
¯0.3853373178       ¯0.316227766
¯0.4954336943       ¯0.316227766
```

Начнем со второго. Второй нам даст, при проектировании 10-ти термопарок, опять видим одинаковые множители, т.е. он даст аналог средней температуры на выходе.

А вот первый поинтереснее, мы видим, что первая компонента, соответствующая первой термопаре имеет какое-то положительное значение, а последняя компонента (10-ая термопара) имеет точно такое же по модулю значение, но отрицательное, т.е. когда мы начнем проектировать, мы из первой термопары умноженной на 0.4954336943 вычтем 10-ую умноженную на 0.4954336943, т.е. получим первое минус десятое, эта разность будет характеризовать перекос поля:



Теперь идя к центру реактора, сохраняется одинаковость весов и противоположность знаков, но по модулю эти веса убывают, это связано с тем, что в середине реактора не зависимо от наклона у нас маленькие изменения, а по краям — большие изменения в зависимости от направления поля. 5 и 6 термопары будут иметь минимальный вес, а крайние — максимальный, это перекос. Первое собственное число = перекос, второе собственное число = среднее значение поля.


T
№(номер термопары)