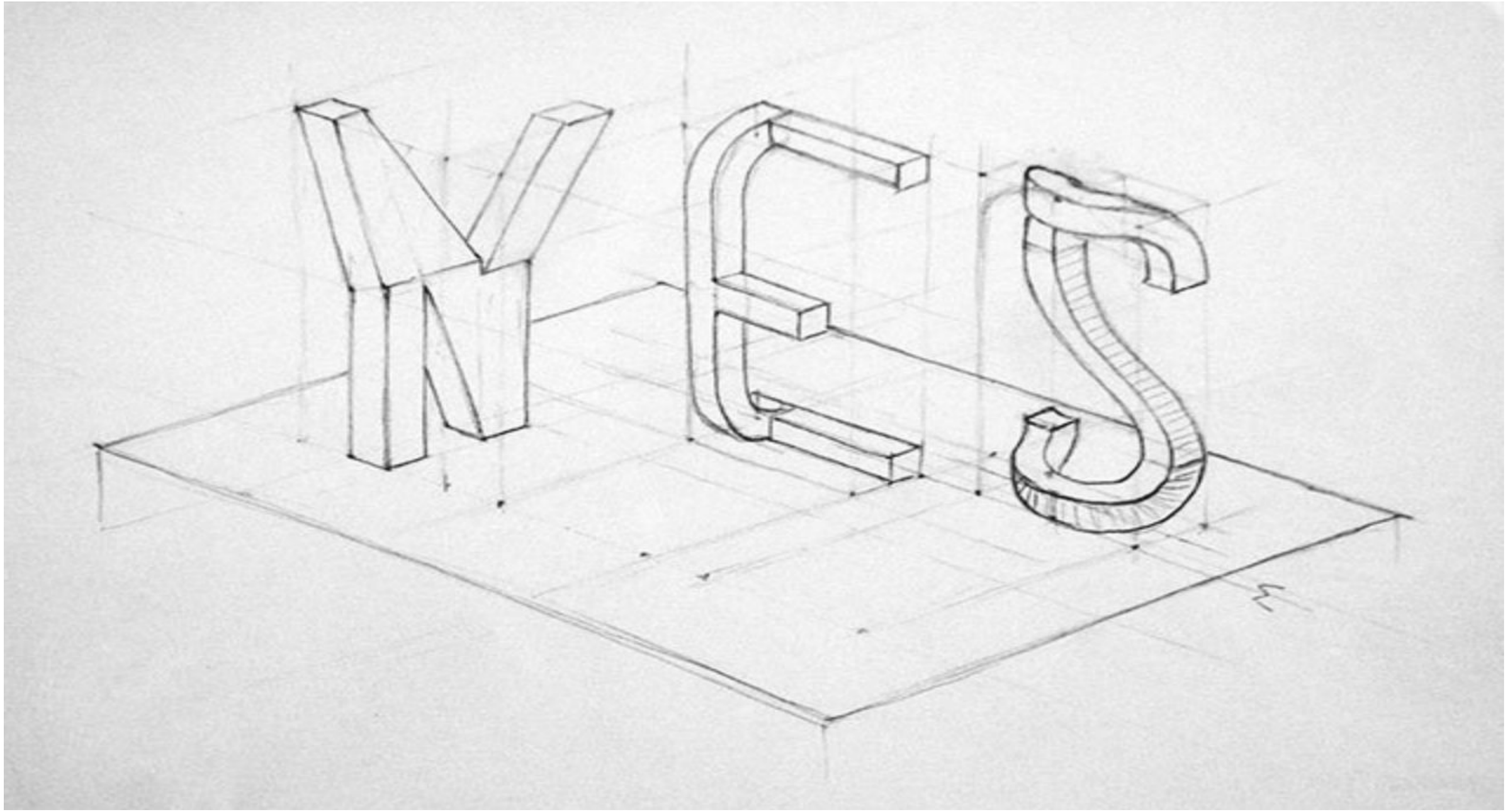
PCA



PCA (principal component analysis)

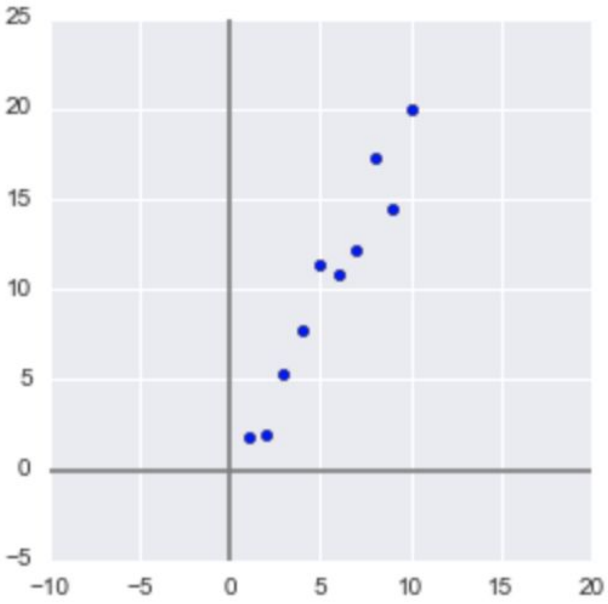
Часто хочется иметь упрощенную модель, максимально точно описывающую реальное положение дел (т.е. потерявшую минимальное количество информации)

```
x = pd.read_csv("train_housing.csv")
x.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoS
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

PCA на примере

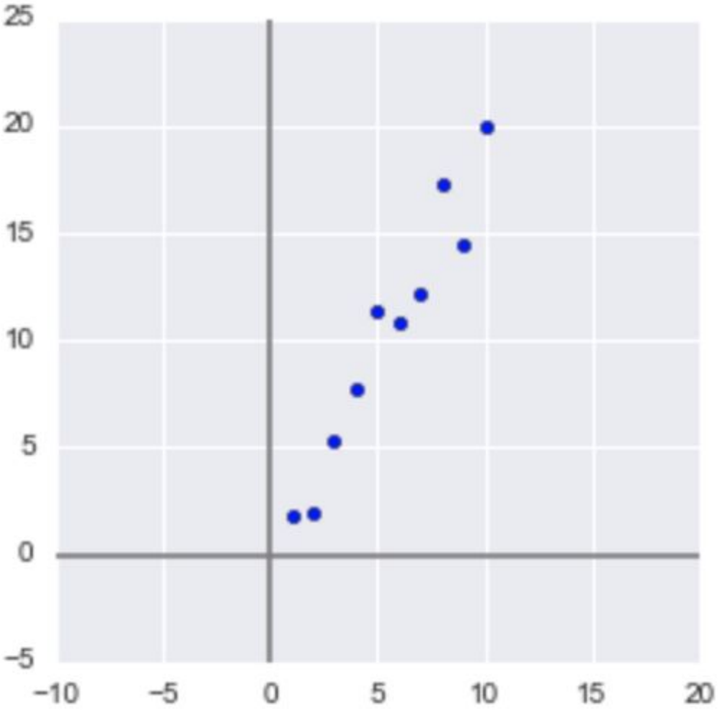
```
x = np.arange(1,11)
y = 2 * x + np.random.randn(10)*2
X = np.vstack((x,y))
print X

OUT:
[[  1.          2.          3.          4.
5.
    6.          7.          8.          9.
10.        ]
 [  2.73446908   4.35122722   7.21132988  11.24872601
9.58103444
  12.09865079  13.78706794  13.85301221  15.29003911
18.0998018 ]]
```



PCA на примере

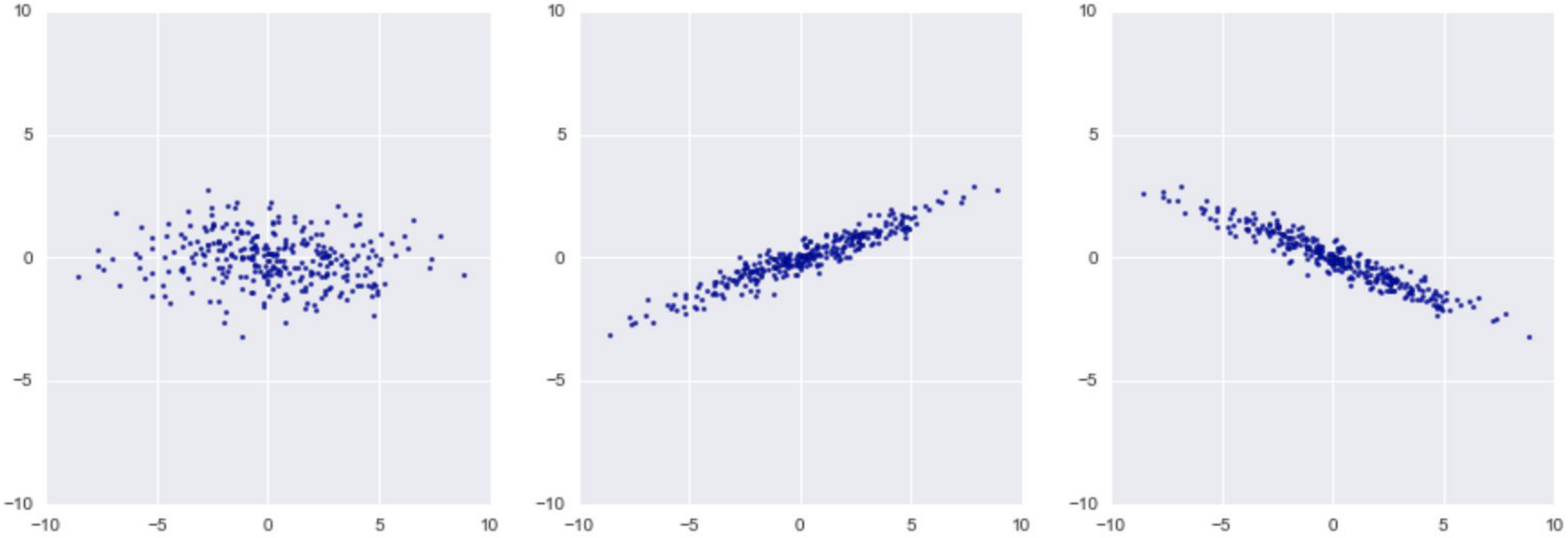
Хотим представить каждую точку вместо двух признаков одним, потеряв при этом минимальное количество информации

Но что такое информация?



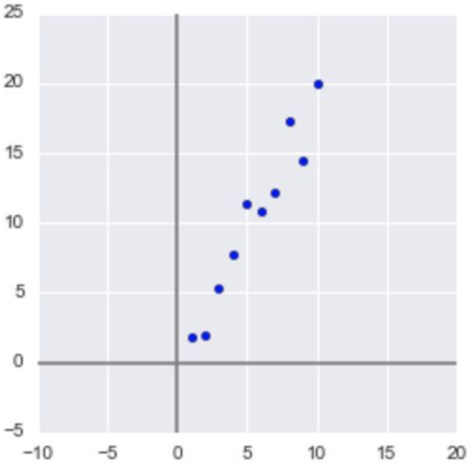
Дисперсия?

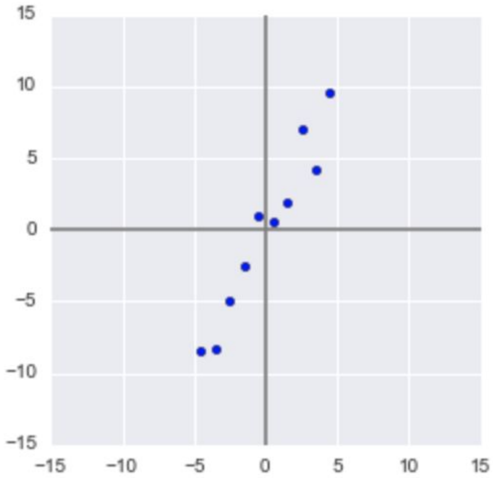
У этих трех распределений дисперсии равны



Ковариация!

$$Cov(X_i, X_j) = E\Big[\big(X_i - E(X_i)\big) \cdot \big(X_j - E(X_j)\big)\Big] = E(X_i X_j) - E(X_i) \cdot E(X_j)$$



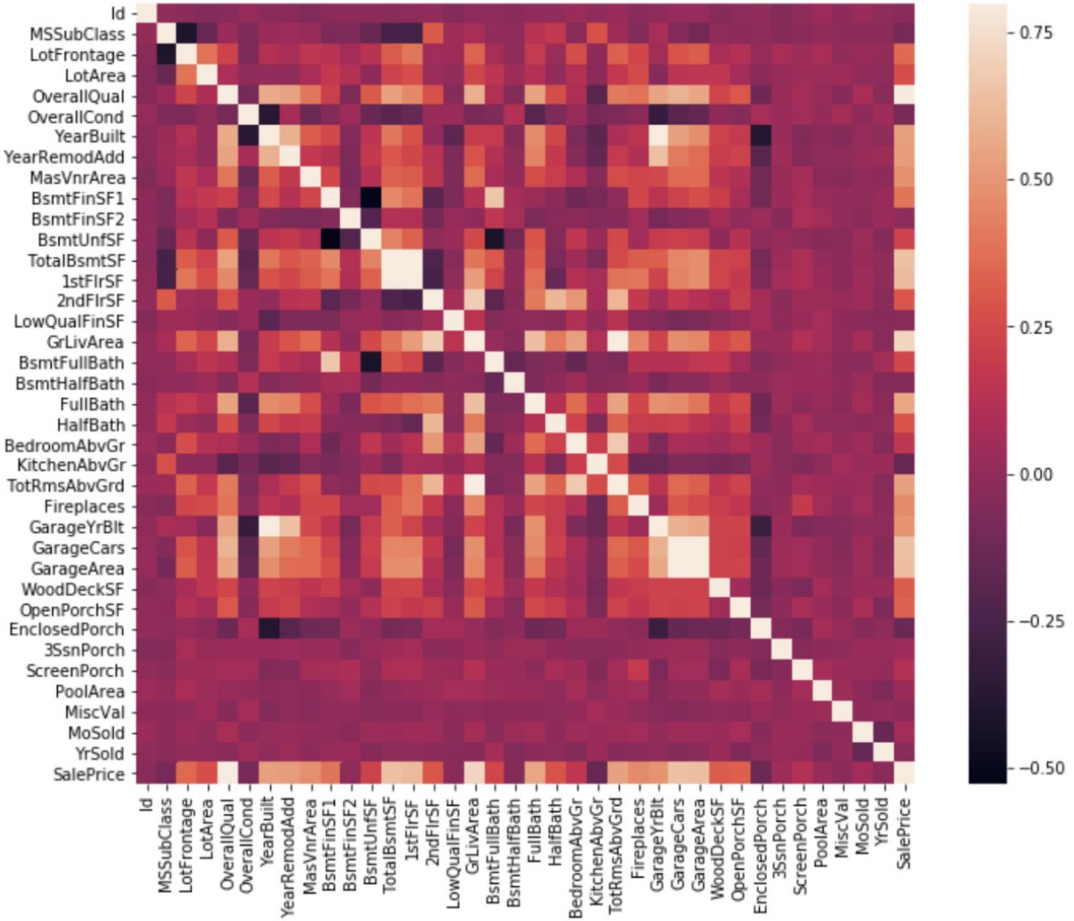


$$Cov(X_i, X_j) = E(X_i X_j)$$

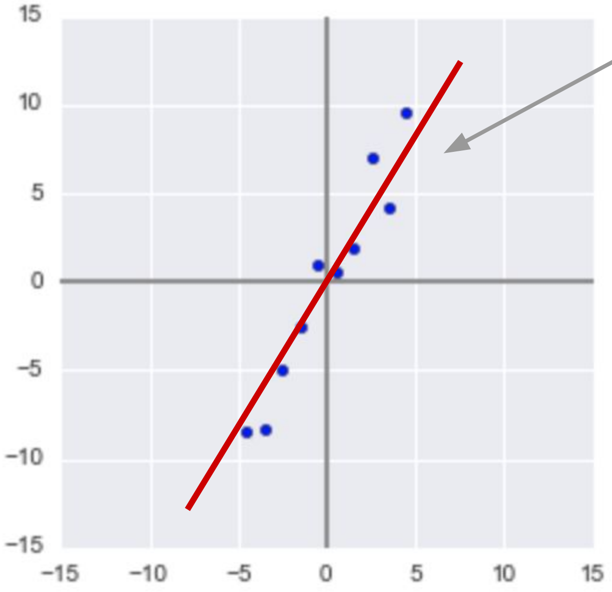
$$Cov(X_i, X_i) = Var(X_i)$$

Матрица ковариаци

$$D = \begin{pmatrix} \sigma_1^2 & \operatorname{cov}(v_1, v_2) & \dots & \operatorname{cov}(v_1, v_n) \\ \operatorname{cov}(v_2, v_1) & \sigma_2^2 & \dots & \operatorname{cov}(v_2, v_n) \\ \dots & \dots & \dots & \dots \\ \operatorname{cov}(v_n, v_1) & \operatorname{cov}(v_n, v_2) & \dots & \sigma_n^2 \end{pmatrix}$$



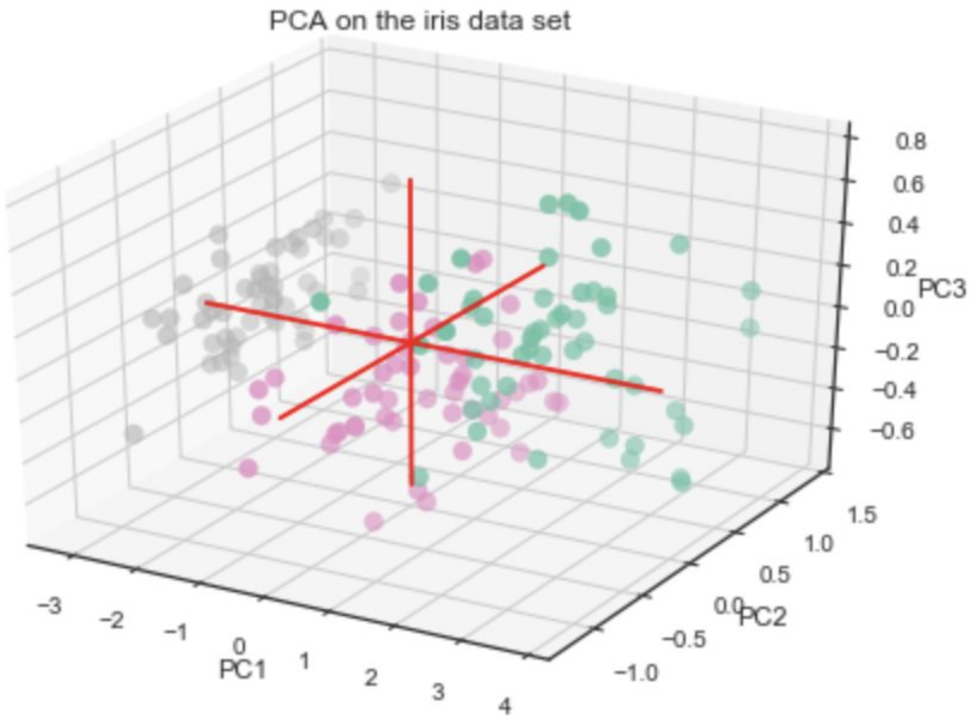
PCA



Хотим найти вот такой вектор

Чтобы дисперсия проекции точек на него была максимальной, т.е. чтобы сохранилось максимум информации.

PCA

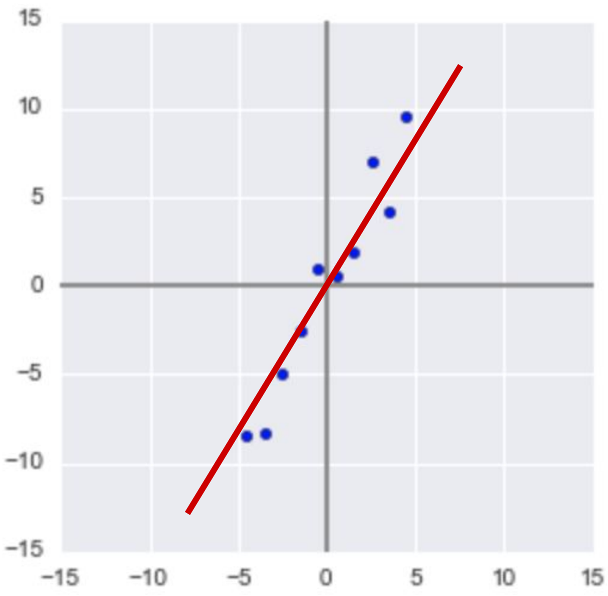


Хотим найти вот такие векторы

Чтобы матрица ковариаций проекций точек на них была максимальной, т.е. чтобы сохранилось максимум информации.

PCA

Проекция наших двумерных данных X на вектор v есть $\mathbf{v^T X}$



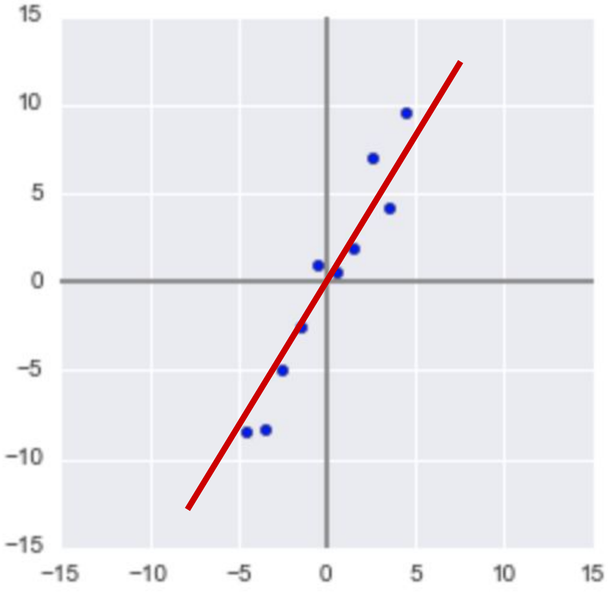
$$Var(X) = \Sigma = E(X \cdot X^T)$$

$$Var(X^*) = \Sigma^* = E(X^* \cdot X^{*T}) = E\Big((\vec{v}^T X) \cdot (\vec{v}^T X)^T\Big) =$$
$$= E(\vec{v}^T X \cdot X^T \vec{v}) = \vec{v}^T E(X \cdot X^T)\vec{v} = \vec{v}^T \Sigma \vec{v}$$

Итак, дисперсия максимизируется при максимальном значении $\mathbf{v^T \Sigma v}$.

PCA

Итак, дисперсия максимизируется при максимальном значении $\mathbf{v}^T\,\mathbf{\Sigma v}$.



Как найти это максимальное значение и соответствующий вектор v?

Ответ прост:
максимальное значение -- это максимальное собственное значение матрицы **Σ = Var(X)** а v -- соответствующий этому собственному значению собственный вектор

PCA

Итак,

направление максимальной дисперсии у проекции всегда совпадает с собственным вектором, имеющим максимальное собственное значение, равное величине этой дисперсии.

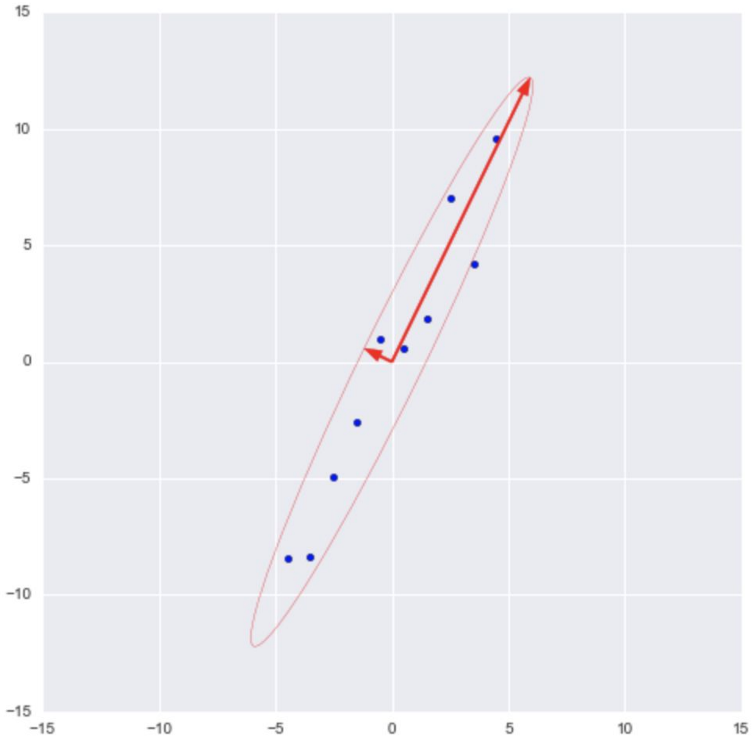
PCA

Итак,

направление максимальной дисперсии у проекции всегда совпадает с собственным вектором, имеющим максимальное собственное значение, равное величине этой дисперсии.

И это справедливо также для проекций на большее количество измерений – дисперсия (ковариационная матрица) проекции на m-мерное пространство будет максимальна в направлении m собственных векторов, имеющих максимальные собственные значения.

PCA



Две главные компоненты нашей двумерной системы

SVD (Singular Value Decomposition)

Любую матрицу A размера nxm можно представить в виде произведения трех матриц U, Σ и V:

$$\underset{n \times m}{\boldsymbol{A}} = \underset{n \times n}{\boldsymbol{U}} \times \underset{n \times m}{\boldsymbol{\Sigma}} \times \underset{m \times m}{\boldsymbol{V}^T}$$

При этом:

U и V -- ортогональные матрицы

Σ -- диагональная

https://habr.com/company/yandex/blog/241455/

SVD: пример

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{bmatrix}$$

$$M = U\Sigma V^{T}$$

$$U = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad V^* = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sqrt{0.2} & 0 & 0 & 0 & \sqrt{0.8} \\ 0 & 0 & 0 & 1 & 0 \\ -\sqrt{0.8} & 0 & 0 & 0 & \sqrt{0.2} \end{bmatrix},$$

SVD: применение

- Нахождение псевдообратной матрицы
- PCA
- рекомендательные системы