2025-02-12

Метод главных компонент (Principal Component Analysis, PCA) (окончание)

Линейный метод снижения размерности (в идеале снижение до 2D или 3D) для предсказательных (регрессионных) моделей. Метод определяет новую систему координат, в которой наибольшая дисперсия по любой проекции в исходном наборе данных лежит на первой оси; вторая по величине дисперсия — на второй оси и т.д.

Т.е. преобразование любого большого числа переменных в меньшее число некоррелированных переменных, которые и называются главными компонентами.

Области применения метода главных компонент

1. Снижение размерности (снижение числа измерений в данных)
2. Выявление закономерностей и паттернов в наборе данных высокой размерности
3. Визуализация данных высокой размерности
4. Фильтрация шума в данных
5. Улучшение классификации в задачах на классификацию

Алгоритм

1. Нормализовать данные, например, через стандартизированную оценку (или Z-score: $z = \frac{x - \bar{X}}{S_x}$, где x — значение величины, $\bar{X}$ — среднее арифметическое этой величины, S_x — стандартное отклонение величины), см. также StandardScaler — scikit-learn 1.6.1 documentation
2. Построить матрицу ковариации $N \times N$
3. Диагонализировать матрицу
4. Отсортировать значения векторов в матрице от большего к меньшему
5. Оставить только K наибольших векторов
6. Модифицировать исходные данные
7. Вернуть набор данных меньшей размерности

Матрица ковариаций

$$S = \begin{pmatrix} \sigma_{x1x1} & \sigma_{x1x2} & \cdots & \sigma_{x1xq} & \sigma_{x1y1} & \cdots & \sigma_{x1yp} \\ \sigma_{x2x1} & \sigma_{x2x2} & \cdots & \sigma_{x2xq} & \sigma_{x2y1} & \cdots & \sigma_{x2yp} \\ \sigma_{x3x1} & \sigma_{x3x2} & \cdots & \sigma_{x3xq} & \sigma_{x3y1} & \cdots & \sigma_{x3yp} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \sigma_{ypx1} & \sigma_{ypx2} & \cdots & \sigma_{ypxq} & \sigma_{ypy1} & \cdots & \sigma_{ypyp} \end{pmatrix}$$

$$S = \left(\begin{array}{c|c} S_{11} & S_{12} \\ \hline S_{21} & S_{22} \end{array} \right)$$

	A	B	C	D	E	F	G	H
1	Y	X_1	X_2			*Y*	*X1*	*X2*
2	15	26	2,7		Y	59,33333		
3	17	33	2,9		X1	82,6	146	
4	19	39	3,6		X2	4,526667	6,993333	0,408
5	22	48	4		ковариационная матрица			
6	35	55	4,1					
7	8	25	2,4					
8	23	40	3,5			*Y*	*X1*	*X2*
9	11	31	3		Y	1		
10	6	22	2,2		X1	0,887471	1	
11	19	45	3,5		X2	0,920024	0,906103	1
12	17	41	2,9		корреляционная матрица			
13	9	23	2,3					
14	16	39	3					
15	23	60	3,6					
16	30	58	4,3					
17								

см. пример в Экселе https://real-statistics.com/multivariate-statistics/factor-analysis/principal-component-analysis/

1. Исходные данные:

	A	B	C	D	E	F	G	H	I	J
1	**Teacher Evaluation**									
2										
3		Expect	Entertain	Comm	Expert	Motivate	Caring	Charisma	Passion	Friendly
4	1	2	8	1	4	7	5	4	4	8
5	2	4	8	5	3	7	7	7	6	6
6	3	2	8	2	3	6	7	1	3	7
7	4	4	8	4	2	8	7	7	5	7
8	5	3	8	5	4	8	8	7	6	7
9	6	4	7	3	3	6	6	1	4	7
10	7	4	8	4	2	6	4	5	4	7
11	8	4	8	3	3	7	5	4	4	7
12	9	2	8	2	2	7	6	1	4	7
13	10	4	8	3	4	8	7	4	4	8

2. Статистические величины

	A	B	C	D	E	F	G	H	I	J
125		Expect	Entertain	Comm	Expert	Motivate	Caring	Charisma	Passion	Friendly
126	mean	3.708333	8.166667	3.45	3.383333	6.708333	6.008333	4.841667	4.725	7.3
127	stdev	1.266211	0.570026	1.649293	1.014047	0.956146	1.368897	2.589558	0.95233	0.588403
128	skew	0.138578	0.558794	0.609682	0.250119	0.030993	-0.23512	0.047501	-0.07387	0.322172
129	kurt	-0.07509	1.224573	-0.37097	0.790481	-0.09199	-0.42618	-0.92189	-0.24984	0.134885

3. Матрица ковариаций на нормализованных данных

	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
3		Expect	Entertain	Comm	Expert	Motivate	Caring	Charisma	Passion	Friendly
4	Expect	1.603291	-0.12745	0.165966	0.154762	-0.17822	-0.18242	-0.07178	-0.15651	-0.2563
5	Entertain	-0.12745	0.32493	0.420168	0.028011	0.133053	0.091036	0.665266	0.205882	0.134454
6	Comm	0.165966	0.420168	2.720168	0.254622	0.451681	0.239916	3.197899	0.578571	0.006723
7	Expert	0.154762	0.028011	0.254622	1.028291	0.188375	0.064006	0.64944	0.24916	0.110924
8	Motivate	-0.17822	0.133053	0.451681	0.188375	0.914216	0.372199	0.852591	0.297269	0.021008
9	Caring	-0.18242	0.091036	0.239916	0.064006	0.372199	1.87388	0.362675	0.254412	-0.04454
10	Charisma	-0.07178	0.665266	3.197899	0.64944	0.852591	0.362675	6.705812	1.082143	0.157143
11	Passion	-0.15651	0.205882	0.578571	0.24916	0.297269	0.254412	1.082143	0.906933	0.04958
12	Friendly	-0.2563	0.134454	0.006723	0.110924	0.021008	-0.04454	0.157143	0.04958	0.346218

Диагонализированная матрица

	L	M	N	O	P	Q	R	S	T	U
3		Expect	Entertain	Comm	Expert	Motivate	Caring	Charisma	Passion	Friendly
4	Expect	1	-0.17658	0.079472	0.120531	-0.14721	-0.10525	-0.02189	-0.12979	-0.34401
5	Entertain	-0.17658	1	0.446921	0.048459	0.244122	0.116667	0.450687	0.37926	0.400869
6	Comm	0.079472	0.446921	1	0.152244	0.286424	0.106265	0.748758	0.368359	0.006927
7	Expert	0.120531	0.048459	0.152244	1	0.194286	0.046109	0.247318	0.258007	0.185906
8	Motivate	-0.14721	0.244122	0.286424	0.194286	1	0.284367	0.344343	0.326466	0.037342
9	Caring	-0.10525	0.116667	0.106265	0.046109	0.284367	1	0.102311	0.195155	-0.05529
10	Charisma	-0.02189	0.450687	0.748758	0.247318	0.344343	0.102311	1	0.438805	0.103132
11	Passion	-0.12979	0.37926	0.368359	0.258007	0.326466	0.195155	0.438805	1	0.088479
12	Friendly	-0.34401	0.400869	0.006927	0.185906	0.037342	-0.05529	0.103132	0.088479	1

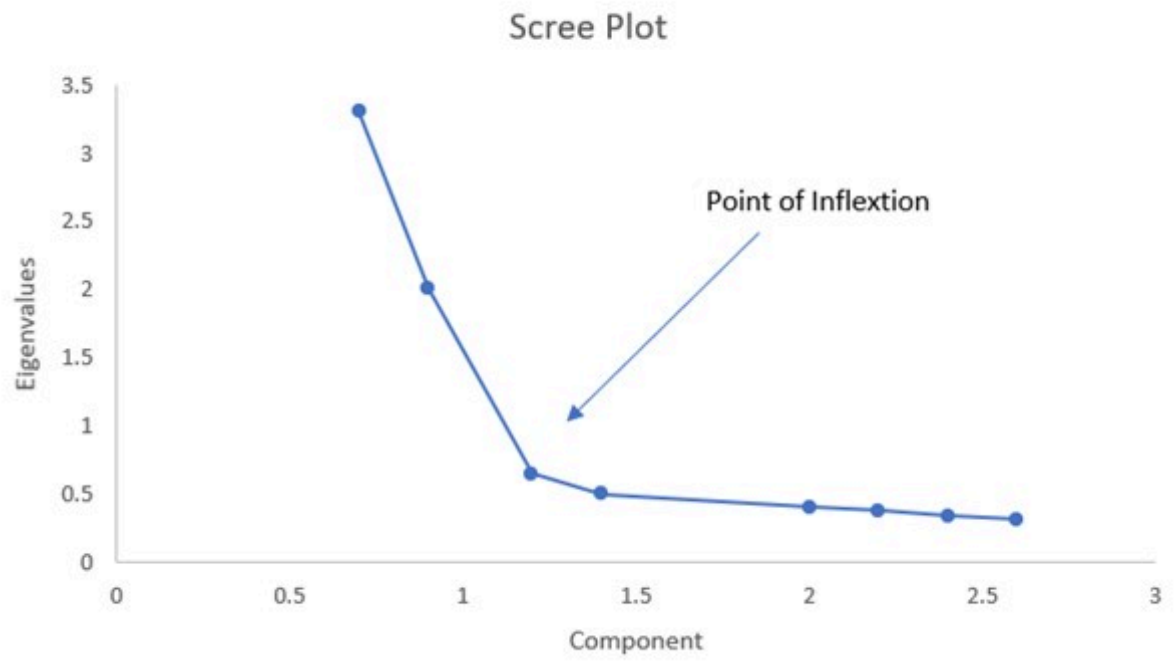
Векторы (Eigenvalues and eigenvectors - Wikipedia) для каждого признака

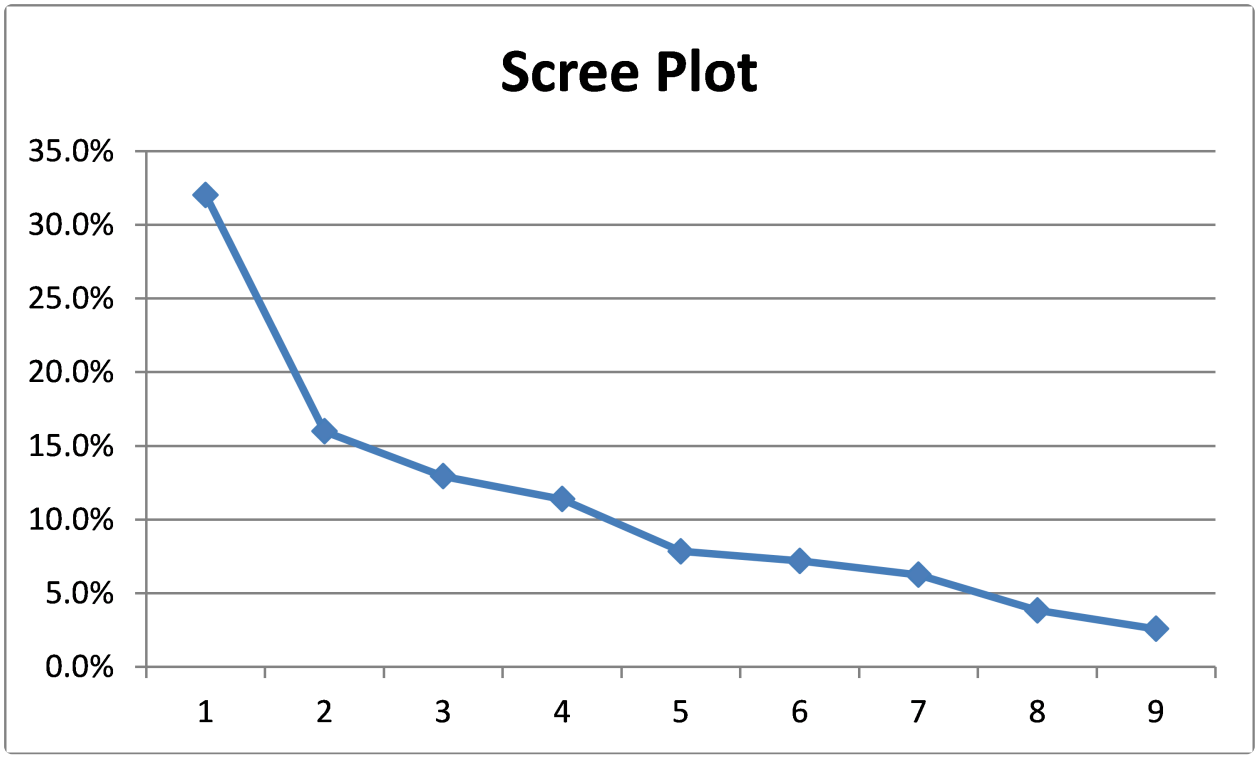
	L	M	N	O	P	Q	R	S	T	U
18		2.880437	1.438654	1.16393	1.024453	0.705209	0.647623	0.562393	0.345248	0.232053
19	Expect	0.108673	0.639248	-0.2573	0.113776	0.417101	-0.09591	0.480829	-0.28424	-0.07242
20	Entertain	-0.41156	-0.25314	-0.1811	-0.26162	0.339636	6.8E-05	0.436453	0.592532	-0.09143
21	Comm	-0.44432	0.290679	-0.18728	-0.3022	0.030103	-0.13252	-0.3009	-0.06395	0.691799
22	Expert	-0.21564	0.135501	-0.18288	0.839641	0.028153	-0.00824	-0.22139	0.378794	0.057765
23	Motivate	-0.34066	0.036795	0.419631	0.156	-0.42352	-0.55994	0.424548	-0.08822	0.035156
24	Caring	-0.17657	0.025335	0.714513	0.062974	0.632544	0.029678	-0.21613	-0.07297	-0.0257
25	Charisma	-0.4815	0.199554	-0.1803	-0.16444	-0.08907	-0.09916	-0.36144	-0.16784	-0.70158
26	Passion	-0.40531	0.029434	0.107551	0.116215	-0.21505	0.782759	0.281837	-0.25913	0.06513
27	Friendly	-0.17762	-0.61785	-0.31904	0.23117	0.274429	-0.19028	0.041652	-0.55867	0.078217

Сортируем

	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW
60	B										X		X'		Y
61	0.108673	0.639248	-0.2573	0.113776	0.417101	-0.09591	0.480829	-0.28424	-0.07242		2		-1.34917		0.782502
62	-0.41156	-0.25314	-0.1811	-0.26162	0.339636	6.8E-05	0.436453	0.592532	-0.09143		8		-0.29238		-1.96758
63	-0.44432	0.290679	-0.18728	-0.3022	0.030103	-0.13252	-0.3009	-0.06395	0.691799		1		-1.48548		-0.23406
64	-0.21564	0.135501	-0.18288	0.839641	0.028153	-0.00824	-0.22139	0.378794	0.057765		4		0.608124		1.123701
65	-0.34066	0.036795	0.419631	0.156	-0.42352	-0.55994	0.424548	-0.08822	0.035156		7		0.305044		-0.76562
66	-0.17657	0.025335	0.714513	0.062974	0.632544	0.029678	-0.21613	-0.07297	-0.0257		5		-0.7366		-0.66149
67	-0.4815	0.199554	-0.1803	-0.16444	-0.08907	-0.09916	-0.36144	-0.16784	-0.70158		4		-0.32502		-0.22281
68	-0.40531	0.029434	0.107551	0.116215	-0.21505	0.782759	0.281837	-0.25913	0.06513		4		-0.76129		0.149636
69	-0.17762	-0.61785	-0.31904	0.23117	0.274429	-0.19028	0.041652	-0.55867	0.078217		8		1.18966		-0.56694

Для определения оптимального количества компонент можно воспользоваться методом локтя:





Лабораторная работа №2. Реализация метода главных компонент

1. Загрузить из наборов данных Scikit-learn набор `breast cancer wisconsin dataset` (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html) размерностью 30
2. Загрузить в отдельные переменные входные и выходные данные (`data`, `target`)

3. Реализовать метод главных компонент в функции, принимающей 2 аргумента: входные данные и искомое число главных компонент K (рекомендуется визуализировать данные по мере возможности) — перед реализацией на Питоне рекомендую изучить пример на Экселе)
    1. Нормализовать данные, вычитая для каждого значения в колонке среднее значения этой колонки (`X.mean()`)
    2. Построить матрицу ковариации, используя метод NumPy `cov()`
    3. Диагонализировать матрицу методом NumPy `linalg.eig()`
    4. Отсортировать векторы, используя встроенный метод `argsort()`
    5. Используя синтаксис слайсинга, оставить K главных компонент
    6. Используя метод `dot()`, модифицировать исходные нормализованные данные
    7. Вернуть данные сниженной размерности
4. Загрузить те же данные во встроенную в Scikit-learn реализацию метода главных компонент (`from sklearn.decomposition import PCA`)
5. Сравнить результаты вашей реализации и готовой
6. Найти оптимальное число главных компонент методом локтя