

Машинное обучение

Семинар 14

Понижение размерности и метод главных компонент

В машинном обучении часто возникает задача уменьшения размерности признакового пространства. Для этого можно, например, удалять признаки, которые слабо коррелируют с целевой переменной; выбрасывать признаки по одному и проверять качество модели на тестовой выборке; перебирать случайные подмножества признаков в поисках лучших наборов.

Ещё одним из подходов к решению задачи является поиск новых признаков, каждый из которых является линейной комбинацией исходных признаков. В случае использования квадратичной функции ошибки при поиске такого приближения получается *метод главных компонент* (principal component analysis, PCA), о котором и пойдет речь.

Существует несколько разных постановок метода главных компонент. Мы разберём некоторые из них.

1 Максимизация дисперсии

Задача 1.1. Рассмотрим многомерную случайную величину (X_1, \dots, X_D) . Пусть у неё нулевое математическое ожидание. Мы собрали для такой случайной величины ℓ наблюдений и записали их в виде матрицы $X \in \mathbb{R}^{\ell \times D}$. Найдите выборочную ковариационную матрицу для выборки X .

Решение. Нам надо найти ковариационную матрицу между столбцами матрицы X . На её диагонали будут стоять дисперсии. Обычно $\text{Var}(X_j) = \mathbb{E}(X_j^2) - \mathbb{E}^2(X_j)$. Так как математическое ожидание каждого столбца равно нулю, тогда $\text{Var}(X_j) = \mathbb{E}(X_j^2)$, и оценку таких дисперсий можно найти по формуле

$$\hat{\sigma}_j^2 = \frac{1}{\ell} \sum_{i=1}^{\ell} x_{ij}^2.$$

В остальных клетках матрицы будут стоять выборочные ковариации. Обычно $\text{Cov}(X_j, X_k) = \mathbb{E}(X_j X_k) - \mathbb{E}(X_j)\mathbb{E}(X_k)$. Так как математическое ожидание каждого столбца равно нулю, тогда $\text{Cov}(X_j, X_k) = \mathbb{E}(X_j X_k)$, и их оценки можно найти по формуле

$$\hat{\rho}_{jk} = \frac{1}{\ell} \sum_{i=1}^{\ell} x_{ij} \cdot x_{ik}.$$

Получается, что выборочную ковариационную матрицу можно записать как $\frac{X^T X}{\ell}$. В вычислениях ниже нам будет часто встречаться матрица $X^T X$. Будем держать в голове, что она пропорциональна выборочной ковариационной матрице. Отметим, что если бы математическое ожидание было бы неизвестно, для получения несмещённой оценки дисперсии и ковариации мы должны были бы делить на $\ell - 1$. ■

Пусть $X \in \mathbb{R}^{\ell \times D}$ — матрица «объекты-признаки», где ℓ — число объектов, а D — число признаков. Будем считать, что данные являются центрированными — то есть среднее в каждом столбце матрицы X равно нулю. Мы хотим уменьшить размерность пространства до d . При этом каждый новый признак будет линейно выражаться через исходные с какими-то весами

$$z_{ij} = \sum_{k=1}^D x_{ik} u_{kj}.$$

Чем больше дисперсии выборки мы сохранили, тем больше сохраняется информации. Будем искать веса $u_1, \dots, u_D \in \mathbb{R}^D$ так, чтобы сохранить максимальную дисперсию:

1. $z_1 = Xu_1$: выборочная дисперсия z_1 должна быть максимальной при условии $\|u_1\|^2 = 1$.
2. $z_2 = Xu_2$: выборочная дисперсия z_2 должна быть максимальной при условии $\|u_2\|^2 = 1$ и $u_1 \perp u_2$.
3. $z_3 = Xu_3$: выборочная дисперсия z_3 должна быть максимальной при условии $\|u_3\|^2 = 1$ и $u_1 \perp u_3, u_2 \perp u_3$
4. и так далее

В матричном виде мы можем записать это как $Z = XU$. При этом матрица U будет ортогональной, то есть $U^T U = I$. Нам это нужно, чтобы задача оптимизации имела единственное решение.

Давайте попробуем найти первую главную компоненту. Сведём все требования к ней в оптимизационную задачу. Данные нормированы. Преобразование предполагает, что мы взвесим исходные случайные величины с какими-то весами. Получается, что математическое ожидание по-прежнему будет нулевым, и выборочная дисперсия будет совпадать со вторым моментом

$$\begin{cases} \|Xu_1\|^2 = u_1^T X^T X u_1 \rightarrow \max_{u_1} \\ \|u_1\|^2 = u_1^T u_1 = 1. \end{cases}$$

Запишем лагранжиан (знак минус перед λ используется для удобства):

$$L(u_1, \lambda) = u_1^T X^T X u_1 - \lambda(u_1^T u_1 - 1).$$

Продифференцируем его и приравняем нулю:

$$\frac{\partial L}{\partial u_1} = 2X^T X u_1 - 2\lambda u_1 = 0.$$

Получается, что

$$X^T X u_1 = \lambda u_1$$

Отсюда получаем, что u_1 должен быть собственным вектором матрицы $X^T X$. Учтём это и преобразуем функционал:

$$\|X u_1\|^2 = u_1^T X^T X u_1 = \lambda u_1^T u_1 = \lambda \rightarrow \max_{u_1}$$

Значит, собственный вектор u_1 должен соответствовать максимальному собственному значению матрицы $X^T X$. Напомним, что эта матрица пропорциональна ковариационной матрице. То есть именно она характеризует дисперсию выборки.

Для следующих компонент к оптимизационной задаче будут добавляться требования ортогональности предыдущим компонентам. Решая эти задачи, мы получим, что главная компонента u_i равна собственному вектору, соответствующему i -му собственному значению.

После того как найдены главные компоненты, можно проецировать на них и новые данные. Если нам нужно работать с тестовой выборкой X^{test} , то её проекции вычисляются как $Z^{\text{test}} = X^{\text{test}} U$.

Задача 1.2. Давайте убедимся, что мы поняли метод главных компонент и попробуем ответить на следующие вопросы:

1. Чему будет равно среднее z_1 , если исходные данные центрированы? А z_2 ?
2. Докажите, что из того, что $u_1 \perp u_2$ следует, что $z_1 \perp z_2$. Чему будет равна $\text{Cov}(z_1, z_2)$?
3. Докажите, что $\text{Var}(x_1) + \dots + \text{Var}(x_D) = \text{Var}(z_1) + \dots + \text{Var}(z_D)$
4. Выразите $\sum_{k=1}^D \|z_k\|^2$ через собственные числа. Как с помощью собственных чисел оценить долю объяснённой компонентами дисперсии?

Решение.

1. Все исходные переменные центрированы, их математическое ожидание равно нулю. Мы взвешиваем случайные величины с какими-то весами. Математическое ожидание от этого никак не меняется. Оно остаётся нулевым.
2. Из ортогональности u_1 и u_2 следует ортогональность z_1 и z_2

$$z_1^T z_2 = u_1^T X^T X u_2 = \lambda_2 \cdot u_1^T u_2 = 0.$$

Тут мы воспользовались тем, что u_2 это собственный вектор матрицы $X^T X$. Выходит, что

$$\text{Cov}(z_j, z_k) = \mathbb{E}(z_j \cdot z_k) - \mathbb{E}(z_j) \cdot \mathbb{E}(z_k) = \mathbb{E}(z_j \cdot z_k) = z_j^T z_k = 0.$$

3. С одной стороны

$$\text{Var}(x_1) + \dots + \text{Var}(x_D) = \frac{1}{\ell - 1} \text{tr}(X^T X),$$

так как все дисперсии в ковариационной матрице находятся на диагонали, а след представляет из себя сумму диагональных элементов.

С другой стороны

$$\text{Var}(z_1) + \dots + \text{Var}(z_D) = \frac{1}{\ell - 1} \text{tr}(Z^T Z) = \frac{1}{\ell - 1} \text{tr}(U^T X^T X U) = \frac{1}{\ell - 1} \text{tr}(X^T X),$$

так как $U^T U = I$, под следом матрицы можно переставлять по циклу и, снова, все дисперсии находятся в матрице $Z^T Z$ на диагонали. Обратите внимание, что все недиагональные элементы в $Z^T Z$ будут нулевыми. Мы доказали это в предыдущем пункте.

4. Так как все математические ожидания нулевые, $\|z_k\|^2 = z_k^T z_k = \text{Var}(z_k)$. Выходит, что $\sum_{k=1}^D \|z_k\|^2 = \sum_{k=1}^D \text{Var}(z_k) = \sum_{k=1}^D \text{Var}(x_k)$. Выразим эту сумму через собственные значения

$$\sum_{k=1}^D \|z_k\|^2 = \sum_{k=1}^D \|X u_k\|^2 = \sum_{k=1}^D u_k^T X^T X u_k = \sum_{k=1}^D u_k^T \lambda_k u_k = \sum_{k=1}^D \lambda_k.$$

Выходит, что если мы оставляем в данных d компонент, то дробь $\frac{\lambda_1 + \dots + \lambda_d}{\lambda_1 + \dots + \lambda_D}$ показывает, какая доля дисперсии сохранилась после проецирования выборки на главные компоненты.

■

Задача 1.3. Фил придумал метод бесполезных компонент. Как и в методе главных компонент, бесполезные компоненты являются линейными комбинациями исходных переменных. Бесполезные компоненты также ортогональны между собой. Вектор весов, с которыми исходные переменные входят в бесполезную компоненту, всегда имеет единичную длину. В отличие от метода главных компонент, первая бесполезная компонента обладает наименьшей выборочной дисперсией. Вторая бесполезная компонента ортогональна первой и обладает наименьшей выборочной дисперсией при условии ортогональности. И так далее.

Как связаны метод бесполезных компонент и метод главных компонент?

Решение. Компоненты из метода бесполезных компонент — это ровно главные компоненты, но только перечисленные в обратном порядке.

■

2 Минимизация ошибки приближения

Пусть $X \in \mathbb{R}^{\ell \times D}$ — матрица «объекты-признаки», где ℓ — число объектов, а D — число признаков. Поставим задачу уменьшить размерность пространства до d . Будем считать, что данные являются центрированными — то есть среднее в каждом столбце матрицы X равно нулю.

Мы хотим, чтобы каждый новый признак линейно выражался через исходные

$$z_{ij} = \sum_{k=1}^D x_{ik} u_{kj}.$$

В матричном виде это можно записать как

$$Z = XU^T.$$

Чтобы у этого уравнения существовало единственное решение, нужно ввести на матрицу весов ограничения. Пусть она будет ортогональной $U^T U = I$. Если это требование выполнено, можно получить формулу для матрицы X

$$X = ZU.$$

В матрице Z будет меньше признаков, чем в X , так как $d < D$. Это равенство нельзя выполнить строго. Нам придётся потребовать, чтобы отклонение матрицы признаков X от ZU было как можно меньше. Размер этого отклонения будем вычислять с помощью нормы Фробениуса:

$$\begin{cases} \|X - ZU\|_F \rightarrow \min_{Z,U} \\ U^T U = I \end{cases}$$

Норма Фробениуса матрицы — это сумма квадратов ее значений. Получившаяся задача — это задача матричного разложения. Необходимо представить матрицу X в виде произведения двух матриц U и W , которые будут иметь меньший ранг. То есть нужно уменьшить ранг матрицы, при этом потеряв как можно меньше информации в ней.

Решением данной задачи также являются собственные векторы ковариационной матрицы.

3 Задачи на метод главных компонент¹

Задача 3.1. У бесстрашного исследователя Ильдуса есть набор точек $(x_1, y_1), \dots, (x_n, y_n)$. Ильдус находит прямую, сумма расстояний от которой до каждой точки минимальна. Верно ли, что прямая проходит через точку (\bar{x}, \bar{y}) ?

Решение. Да. Например, прямую можно задать вектором w единичной длины и числом b , $w^T \begin{pmatrix} x \\ y \end{pmatrix} = b$. Обозначим вектор $(x_i, y_i)^T$ буквой v_i . Тогда нам нужно минимизировать функцию

$$\sum (w^T v_i - b)^2.$$

¹Задачи взяты из коллекции Бориса Демешева: https://github.com/bdemeshev/mlearn_pro

Дифференцируем по b и получаем $\sum w^T v_i = n \cdot b$, что и означает, что прямая проходит через среднюю точку,

$$w^T \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = b.$$

■

Задача 3.2. Есть две переменных, $x_1 = (1, 0, 0, 3)^T$, $x_2 = (3, 2, 0, 3)^T$. Найдите первую и вторую главные компоненты.

Решение. Матрица с центрированными столбцами имеет вид: $\tilde{X} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \\ -1 & -2 \\ 2 & 1 \end{pmatrix}$

$$\text{Тогда } \tilde{X}^T \tilde{X} = \begin{pmatrix} 6 & 4 \\ 4 & 6 \end{pmatrix}.$$

Её собственные числа: $\lambda_1 = 10$, $\lambda_2 = 2$, собственные вектора $u_1 = (1/\sqrt{2} \quad 1/\sqrt{2})^T$, $u_2 = (1/\sqrt{2} \quad -1/\sqrt{2})^T$. Найдём главные компоненты:

$$Z = XU = \begin{pmatrix} 0 & 1 \\ -1 & 0 \\ -1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ -1/\sqrt{2} & -1/\sqrt{2} \\ -3/\sqrt{2} & 1/\sqrt{2} \\ 3/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

Первая и вторая главные компоненты — это первый и второй столбцы матрицы Z соответственно.

■

4 Ядровой переход в методе главных компонент

Если внимательно посмотреть на обычный PCA, можно заметить, что, преобразование данных $X = ZU$ линейно. Как известно, признаковое пространство может быть устроено куда сложнее, зависимости могут быть и нелинейными. В методе главных компонент это выльется в то, что ошибка приближения никогда не упадет ниже какого-то числа — потери дисперсии будут в любом случае, любая ось в исходном евклидовом пространстве потеряет часть дисперсии при проекции на нее.

Для каких-то задач легко придумать трансформацию данных, которая упростит задачу. Понять, какой на самом деле размерности задача, бывает непросто, но в ряде случаев хватает картинки. Посмотрим на примеры ниже

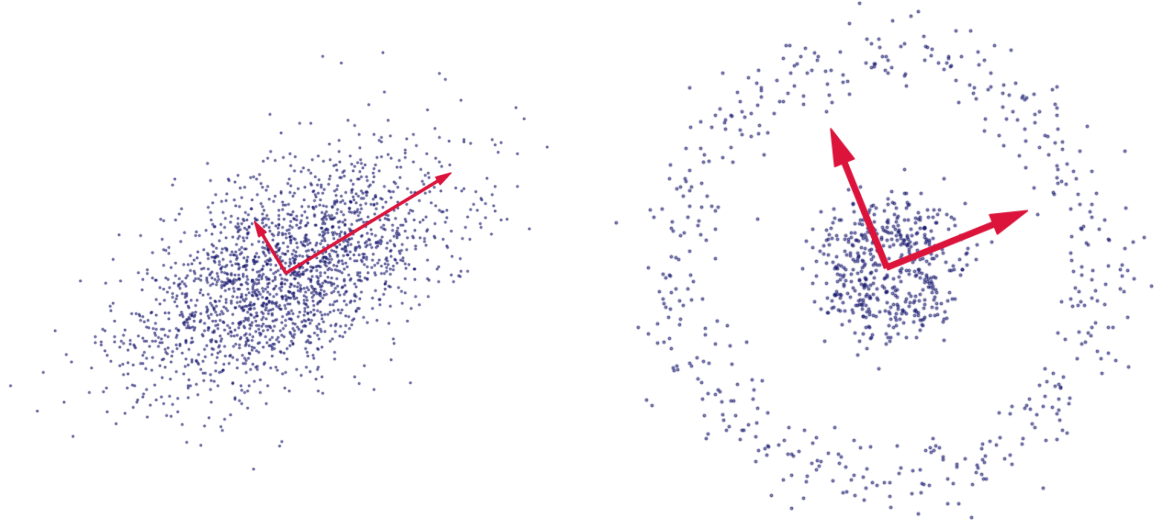


Рис. 1. Примеры главных осей

Если в первом случае главные оси выделяются хорошо, к тому же дисперсия явно больше распределена вдоль одной из них, то на второй картинке оси равнозначны – при любой проекции потеряем примерно половину информации. Истинная размерность окружности равна 1. Если перейти в полярные координаты, достаточно одной координаты (радиуса), чтобы задать ее. Либо можно добиться того же, добавив признаки вида x_i^2 , чья сумма даст квадрат радиуса. Конечно, заранее нужная трансформация неизвестна, вместо этого можно использовать знакомый нам ядровой переход.

Как и прежде, для этого понадобится функция $\varphi : \mathbb{X} \rightarrow H$, которая переводит исходные признаки в новое спрямляющее пространство, и само ядро $K(x, z) = x^T z$. Также напомним, что новое признаковое описание будем обозначать Φ , а за $K = \Phi\Phi^T$ возьмем матрицу Грама в спрямляющем пространстве. Следующие предположения аналогичны рассуждениям для обычного РСА

1. Данные должны быть центрированы, $\sum_{i=0}^D \varphi(x_i) = 0$.

После ядрового перехода нет никакой гарантии, что среднее останется равным 0. Как обычно, это можно сделать руками, если вычесть среднее. А дальше, путем пары алгебраических преобразований, можно получить выражения для центрированной матрицы \hat{K}

$$\begin{aligned}\hat{\varphi}(x) &= \varphi(x) - \frac{1}{D} \sum_{j=1}^D \varphi(x_j) \\ \hat{K}(x, z) &= \hat{\varphi}(x)^T \hat{\varphi}(z) \\ \hat{K}(x, z) &= K(x, z) - \frac{1}{D} \sum_{n=1}^D (K(x, x_n) + K(x_n, z)) + \frac{1}{D^2} \sum_{n,m=1}^D K(x_n, x_m) \\ \hat{K} &= K - 2 \cdot \mathbb{1}_{1/D} K + \mathbb{1}_{1/D} \cdot K \cdot \mathbb{1}_{1/D}\end{aligned} \quad (4.1)$$

Матрица $\mathbb{1}_a$ это матрица, на всех элементах которой стоит a . Матрица K пригодится для дальнейших махинаций

2. Хочется, чтобы уменьшение дисперсии в новом пространстве было как можно меньше.

Будем исходить из того же предположения, что и в исходном РСА – главные компоненты можно найти при помощи собственных векторов матрицы ковариации в новом пространстве

Ковариационная матрица в H имеет вид

$$C = \frac{1}{D} \sum_{i=0}^D \varphi(x_i) \varphi(x_i)^T = \frac{1}{D} \Phi^T \Phi$$

Понятно, что найти ее в явном виде можно не всегда, матрица Φ может быть бесконечномерной, но это и не требуется. Заметим, что $\varphi(x_i)^T v$ это скаляр, для любого вектора v , а значит

$$Cv = \lambda v$$

$$Cv = \frac{1}{D} \sum_{i=0}^D \varphi(x_i) (\varphi(x_i)^T v) = \frac{1}{D} \sum_{i=0}^D \varphi(x_i) a_i = \lambda v$$

Отсюда следует, что любой собственный вектор матрицы C будет представим в виде линейной комбинации в новом пространстве, а значит главные компоненты можно найти по формуле, зная ядро K и векторы a

$$y = \varphi(x_i)^T u = \sum_{j=1}^d a_{ji} \varphi(x_i)^T \varphi(x_j) = \sum_{j=1}^d a_j K(x_i, x_j) \quad (4.2)$$

Здесь же можно сразу отрегулировать, сколько главных компонент хочется оставить – достаточно просуммировать до индекса d – новой уменьшенной размерности пространства

3. Оказывается, чтобы найти векторы a выше, достаточно посчитать собственные значения матрицы K

$$Cv = \frac{1}{D} \sum_{i=0}^D \varphi(x_i) \varphi(x_i)^T v = \frac{1}{D} \sum_{i=0}^D \varphi(x_i) \varphi(x_i)^T \sum_{j=0}^D a_j \varphi(x_j) = \frac{1}{D} \Phi^T \Phi \Phi^T a$$

$$\lambda v = \lambda \sum_{j=0}^D a_j \varphi(x_j) = \lambda \Phi^T a$$

$$\Phi^T (\Phi \Phi^T a - D \lambda a) = 0$$

Сделаем одно допущение – будем считать, что новое представление данных Φ не будет нулевым. Иначе ядровой переход кажется бессмысленным. В таком случае, можно избавиться от Φ^T и перейти к выражению

$$Ka = D \lambda a \quad (4.3)$$

Таким образом, необходимости считать ковариационную матрицу напрямую нет, для уменьшения размерности хватит информации про само ядро.

Теперь обсудим, как применять KernelPCA. Во-первых, обратим внимание, что A строится по объектам через $\Phi\Phi^T$, то есть содержит n строк. Нулевые собственные значения так или иначе не интересны для проекции, поэтому брать размерность векторов больше, чем D не имеет смысла. В обычном PCA U строилась по признакам, т.е. $X^T X$, где будет не более D строк.

Затем вспомним, что из сингулярного разложения следует, что $Z = XU = V\Sigma$. Поскольку теперь напрямую известно V , оно же A – собственные векторы K , можем этим воспользоваться, чтобы найти трансформацию в матричном виде

$$\begin{aligned} z &= \sigma_i a_i = \sqrt{\lambda_i} a_i = \frac{1}{\sqrt{\lambda_i}} K a_i \\ K^{\text{test}} &= (K(x_j^{\text{test}}, x_i^{\text{train}}))_{i=1, j=1}^{m, n} \\ \Lambda^{-\frac{1}{2}} &= \text{diag}(\sqrt{\lambda_i})_i \\ Z^{\text{test}}_{m \times d} &= K^{\text{test}}_{m \times n} A_{n \times d} \Lambda^{-\frac{1}{2}}_{d \times d} \end{aligned} \quad (4.4)$$

4. Векторы главных осей u в новом признаковом пространстве должны быть единичной длины

Как ни странно, для этого тоже достаточно знать лишь матрицу K и векторы a

$$u^T u = 1 \implies \sum_{i,j} a_i a_j \varphi(x_i)^T \varphi(x_j) = a^T K a = 1 \quad (4.5)$$

Таким образом, итоговый алгоритм очень похож на исходный PCA за парой деталей, которые были обозначены выше. Фактически это применение трансформации для нахождения нового признакового пространства и одновременно понижение размерности. Подведем итог и распишем алгоритм для двух случаев PCA

Этап	PCA	KernelPCA
Обучение	Работаем с исходным пространством \mathbb{X}	Задаем ядро $K(x, z)$, находим Φ (явно или, например, при помощи RFF)
	Центрируем пространство \mathbb{X}	Центрируем и находим \hat{K}
	Составляем матрицу U_d размера $D \times d$ из с.в. $X^T X$	Составляем матрицу A_d размера $\ell \times d$ из с.в. \hat{K} и матрицу с.з. Λ_d
Применение	Центрируем X^{test}	Находим центрированную \hat{K}^{test}
	Ищем $Z^{\text{test}} = X^{\text{test}} U_d$	Ищем $Z^{\text{test}} = \hat{K}^{\text{test}} A_d \Lambda_d^{-\frac{1}{2}}$

Главные преимущества ядерного метода главных компонент:

- в новом спрямляющем пространстве может сохраниться больше дисперсии, из-за чего новые представления будут более осмысленными
- не нужно придумывать трансформацию, которая сохранит максимальное количество дисперсии
- можно уловить нелинейные зависимости и, например, превратить неразделимую задачу классификации в разделимую, либо же кластеризовать точки, между которыми нет линейной связи, пример ниже

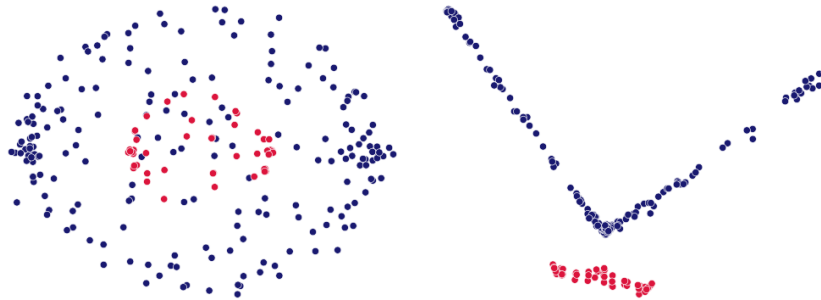


Рис. 2. Результат PCA и KernelPCA на трехмерной сфере

Его минусы следующие:

- ядро нужно выбрать
- у этого метода больше гиперпараметров, чем у обычного варианта, что усугубляется тем, что единой метрики качества понижения размерности нет (ошибка приближения будет считаться уже в H , не получится сравнить с ошибкой в исходном пространстве)
- при большом объеме данных ядро считается достаточно долго
- теряется интерпретируемость – главные компоненты более не соответствуют исходным признакам
- нет аналитического обратного преобразования из H (но можно посчитать численно)