



Визуализация многомерных пространств

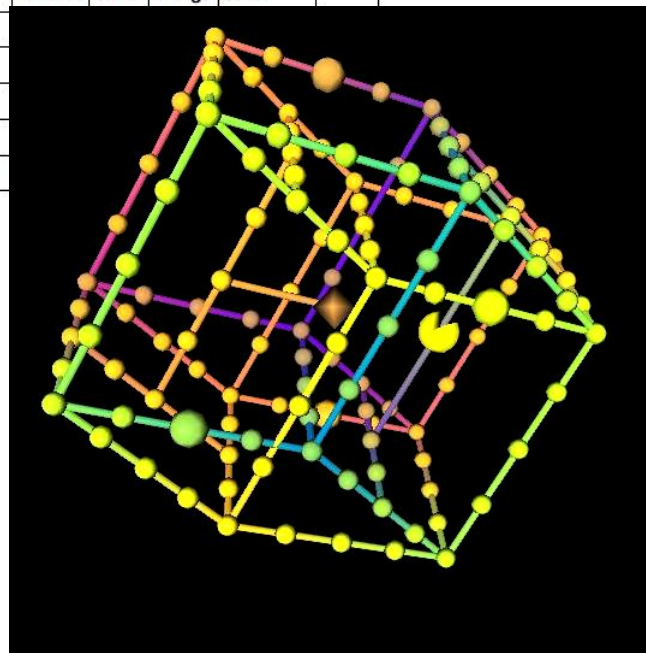


Автор: Сугоняев Андрей, гр. 331

Где мы встречаем многомерные пространства?

- Одна из самых распространенных областей - анализ данных:

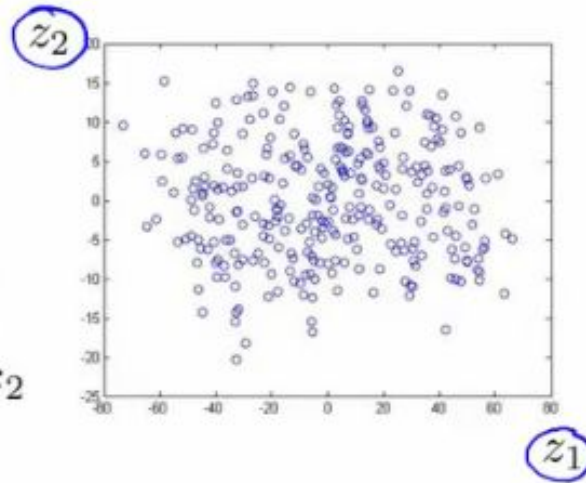
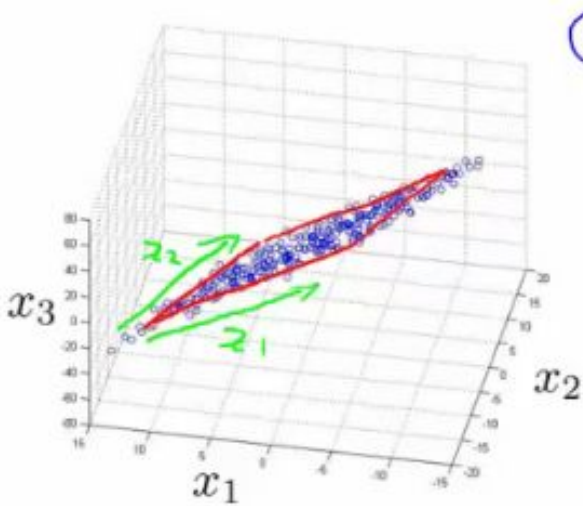
	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls	Total intl charge	Customer service calls	Churn
365	CO	154	415	No	No	0	350.8	75	59.64	216.5	94	18.40	253.9	100	11.43					
985	NY	64	415	Yes	No	0	346.8	55	58.96	249.5	79	21.21	275.4	102	12.39					
2594	OH	115	510	Yes	No	0	345.3	81	58.70	203.4	106	17.29	217.5	107	9.79					
156	OH	83	415	No	No	0	337.4	120	57.36	227.4	116	19.33	153.9	114	6.93					
605	MO	112	415	No	No	0	335.5	77	57.04	212.5	109	18.06	265.0	132	11.93					



Цель визуализации

- Цель – получить отображение данных в 2 или 3 мерном пространстве для дальнейшего изучения структурных особенностей и закономерностей этих данных.

Задача визуализации



Задача — найти такое отображение объектов выборки в пространство малой размерности, которое оптимизировало бы некоторый функционал качества.

"To deal with hyper-planes in a 14 dimensional space, visualize a 3D space and say 'fourteen' very loudly. Everyone does it." — Geoffrey Hinton

Методы

Рассмотрим методы, сопоставляющие точке в n -мерном пространстве точку в пространстве меньшей размерности:

1. Линейные:

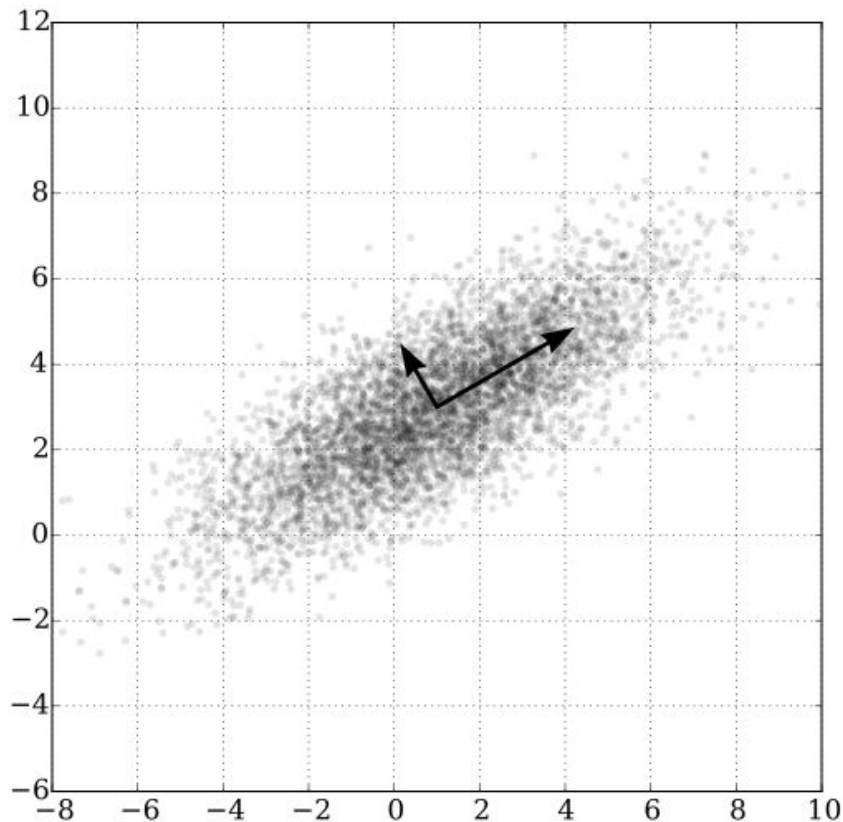
- Метод главных компонент

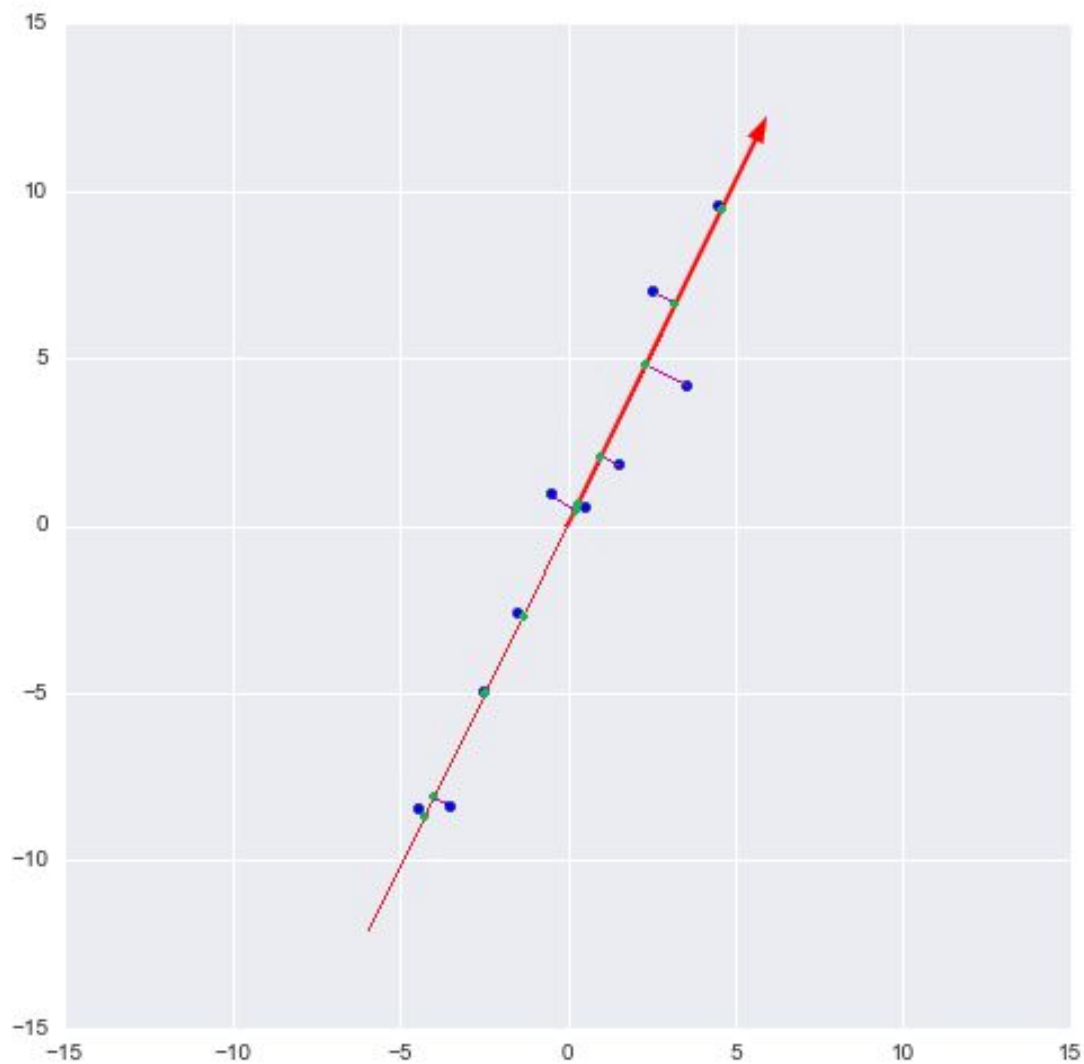
2. Нелинейные:

- Многомерное шкалирование
- t-SNE

Метод главных компонент (PCA)

Основной линейный метод понижения размерности – PCA – производит линейное сопоставление данных из n -мерного пространства пространству меньшей размерности так, чтобы максимизировать вариацию данных в их малоразмерном представлении.



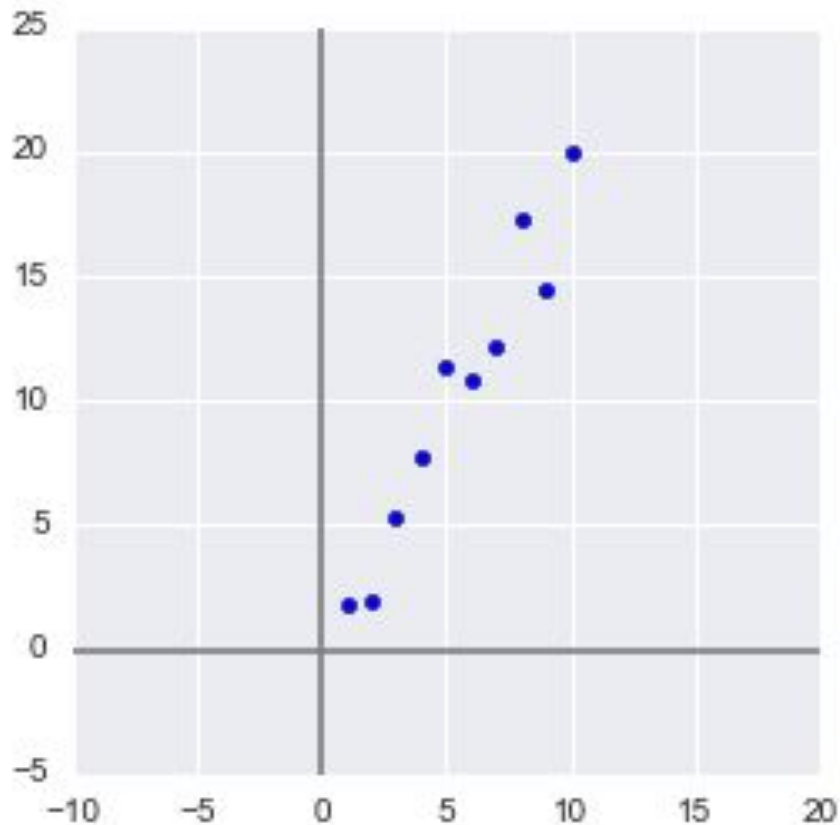


Максимизировать вариацию
по вектору



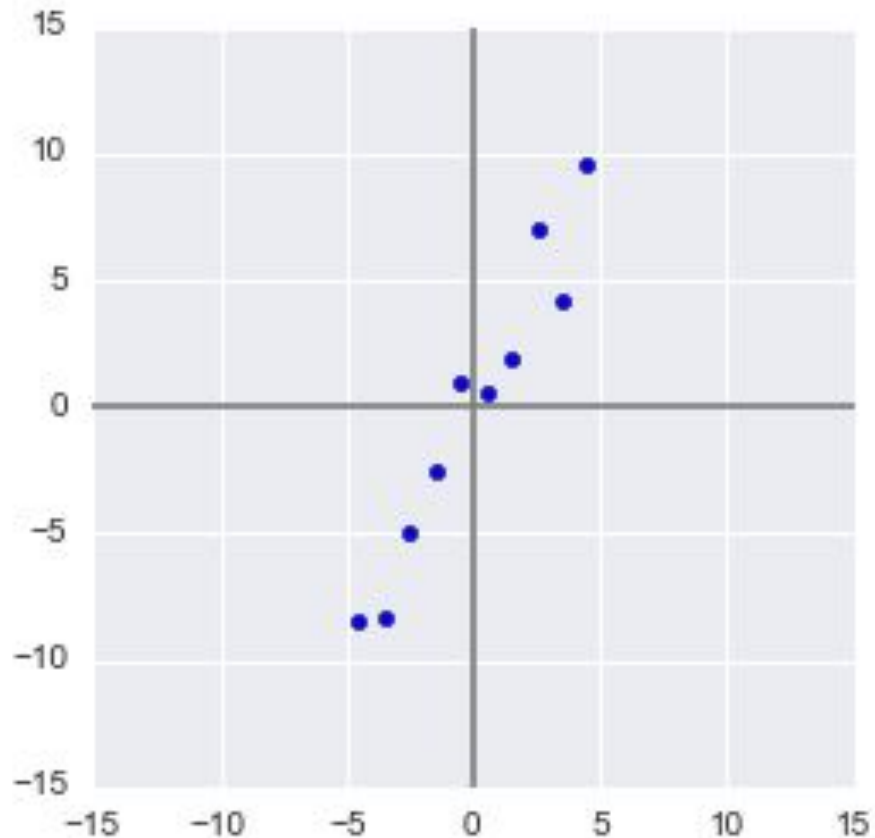
Минимизировать сумму
расстояний от точки до ее
проекции на данный вектор

Шаг 1: Организовать данные



- Записать $x_1 \dots x_n$ как вектор-строки
- Разместить вектор-строки в одной матрице X размером $m \times n$ (матрица объектов-признаков)

Шаг 2: Оцентрировать данные



- Найти среднее по каждой колонке
- Вычесть вектор средних из каждой строки матрицы объектов-признаков **X**

Шаг 3: Вычислить матрицу ковариации

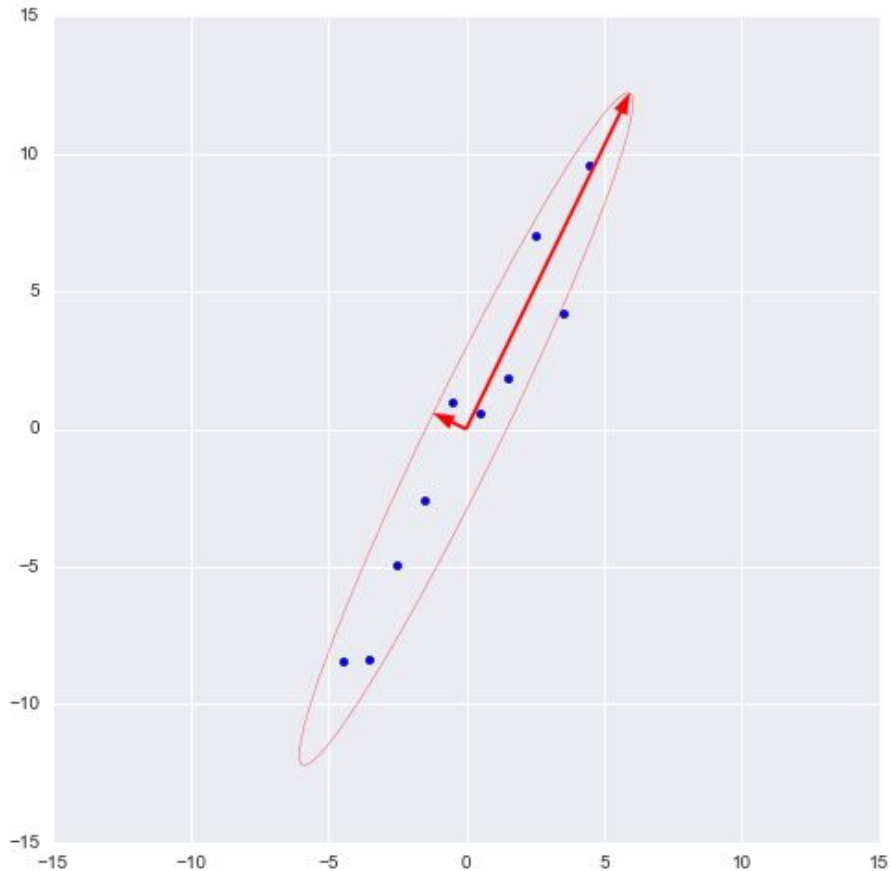
$$\text{cov}(X_i, X_j) = E\left[(X_i - E(X_i)) \cdot (X_j - E(X_j))\right]$$

$$\text{cov}(X_i, X_j) = E\left[X_i X_j\right]$$

$$C = \begin{pmatrix} \sigma_1^2 & \text{cov}(x_1, x_2) & \dots & \text{cov}(x_1, x_n) \\ \text{cov}(x_2, x_1) & \sigma_2^2 & \dots & \text{cov}(x_2, x_n) \\ \dots & \dots & \dots & \dots \\ \text{cov}(x_n, x_1) & \text{cov}(x_n, x_2) & \dots & \sigma_n^2 \end{pmatrix}$$

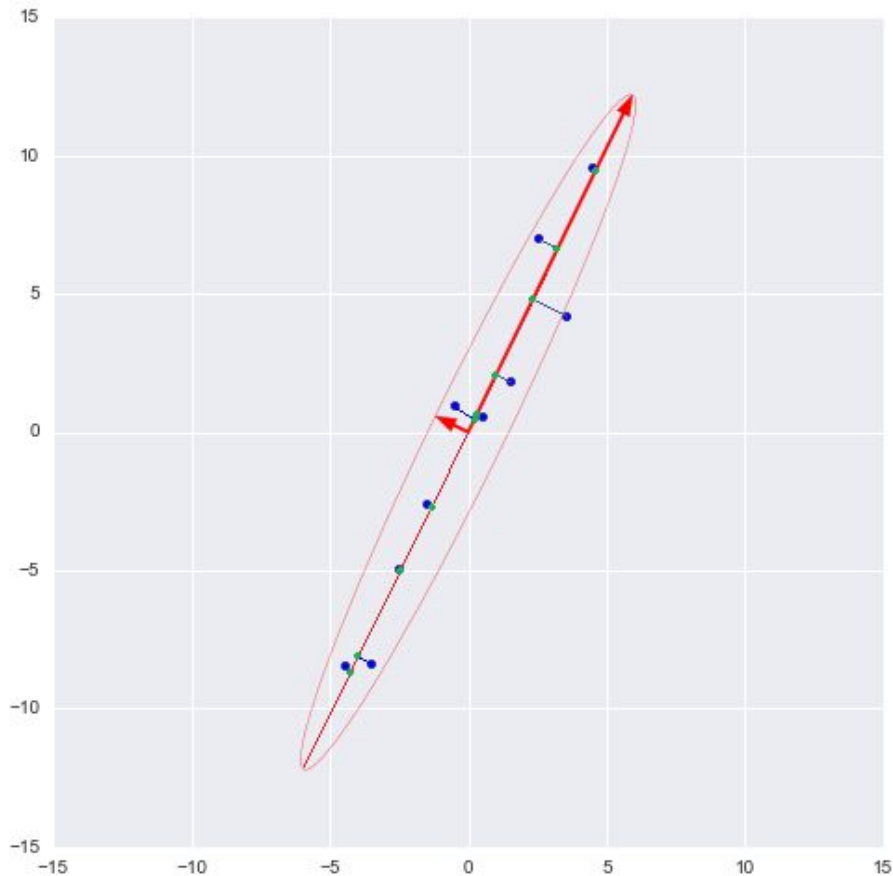
- Найти матрицу ковариации **C** размера $n \times n$ как:
$$C = \frac{1}{(n-1)} X^T X$$
- Использование $N - 1$ вместо N обусловлено поправкой Бесселя

Шаг 4: Найти собственные вектора и собственные числа матрицы C



- Вычислить матрицу V эйгенвекторов которая диагонализует ковариационную матрицу C :
 $C = V D V^{-1}$
- $D = \text{diag}\{ \lambda_1, \dots, \lambda_n \}$, где $\lambda_i, i = 1, \dots, n$ - собственные числа
- Матрица V размера $n \times n$ содержит n вектор-колонок, представляющие из себя собственные векторы
- Собственные числа и векторы упорядочены и идут парами
- Можно использовать сингулярное разложение
 $C = U S W^T$

Шаг 5: Проекция и реконструкция



- В матрицу $\mathbf{V}_{\text{reduced}}$ записать k вектор-колонок, соответствующих k наибольшим собственным числам.
- Умножить $\mathbf{V}_{\text{reduced}}$ на \mathbf{X} чтобы получить проекции на главные компоненты:

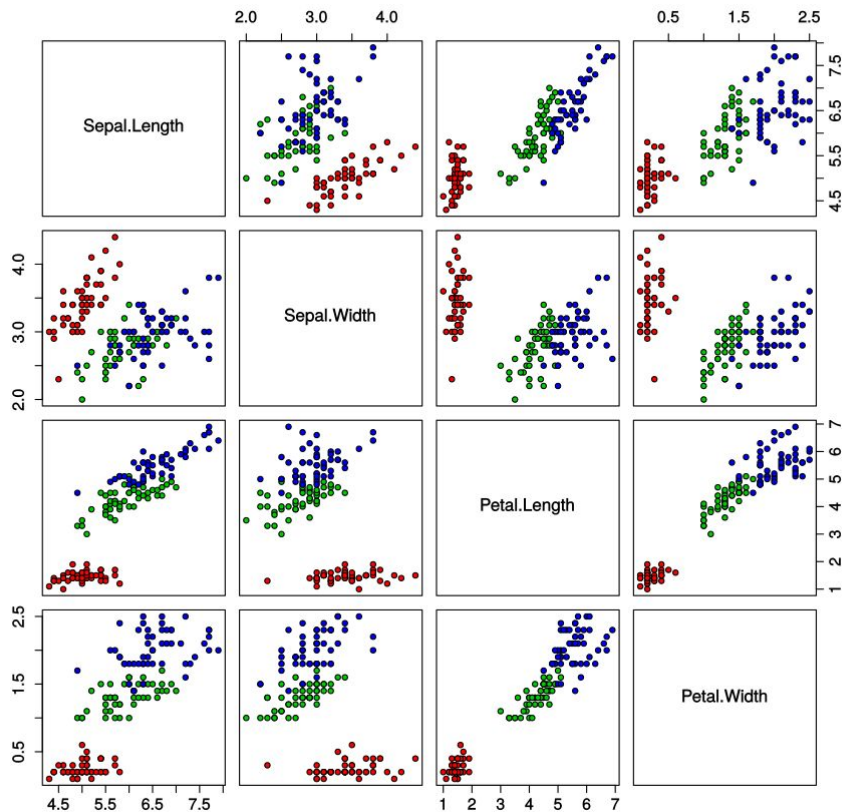
$$\mathbf{Z} = \mathbf{V}_{\text{reduced}} \cdot \mathbf{X}$$

- Умножить $\mathbf{V}_{\text{reduced}}^T$ на проекции \mathbf{Z} чтобы реконструировать данные:

$$\mathbf{X} = \mathbf{V}_{\text{reduced}}^T \cdot \mathbf{Z}$$

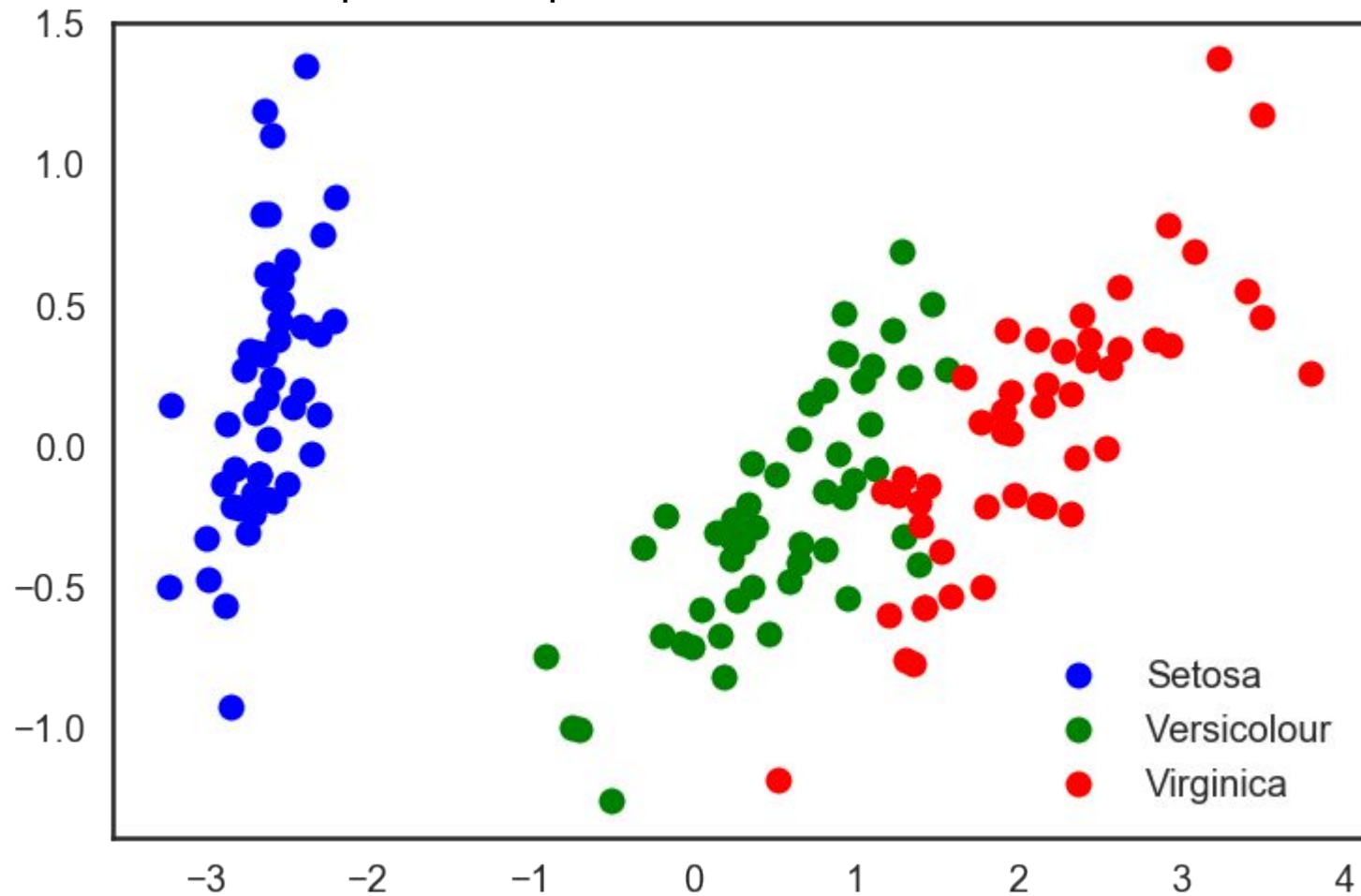
Ирисы Фишера

Iris Data (red=setosa, green=versicolor, blue=virginica)

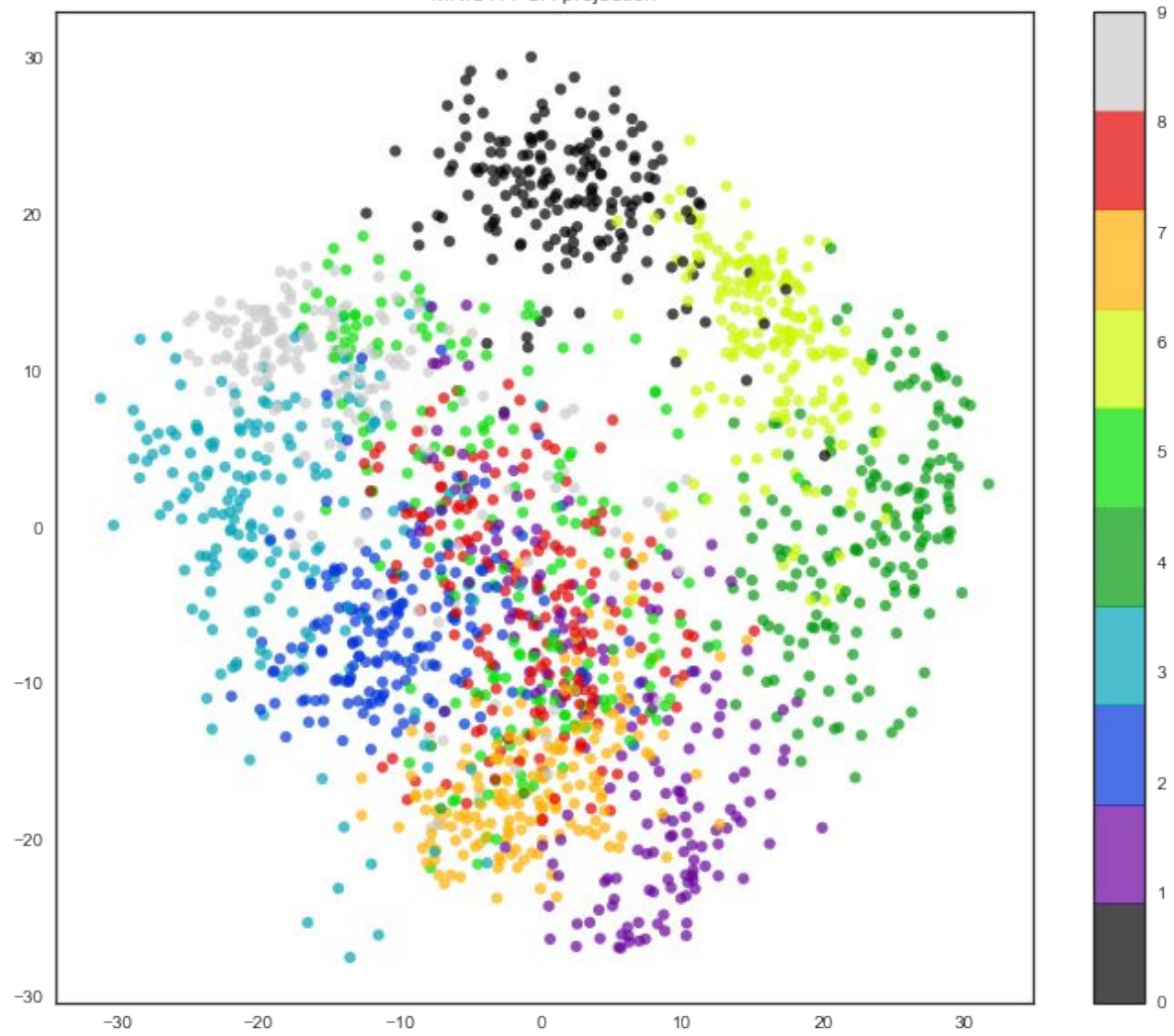


Длина чашелистика	Ширина чашелистика	Длина лепестка	Ширина лепестка	Вид ириса
5.1	3.5	1.4	0.2	<i>setosa</i>
4.9	3.0	1.4	0.2	<i>setosa</i>
4.7	3.2	1.3	0.2	<i>setosa</i>
5.0	2.3	3.3	1.0	<i>versicolor</i>
5.6	2.7	4.2	1.3	<i>versicolor</i>
5.7	3.0	4.2	1.2	<i>versicolor</i>
5.7	2.9	4.2	1.3	<i>versicolor</i>
6.2	2.9	4.3	1.3	<i>versicolor</i>
5.1	2.5	3.0	1.1	<i>versicolor</i>
5.7	2.8	4.1	1.3	<i>versicolor</i>
6.3	3.3	6.0	2.5	<i>virginica</i>
5.8	2.7	5.1	1.9	<i>virginica</i>
7.1	3.0	5.9	2.1	<i>virginica</i>


Проекция ирисов на главные компоненты



MNIST. PCA projection



Почему такой плохой результат?

$$0.5 \cdot \text{0} + 0.5 \cdot \text{3} = \text{0}$$


Линейная комбинация объектов датасета не является рукописной цифрой.

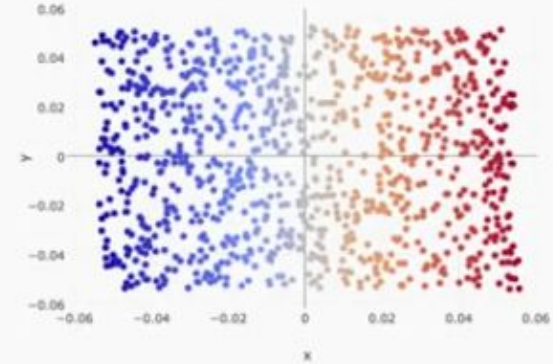
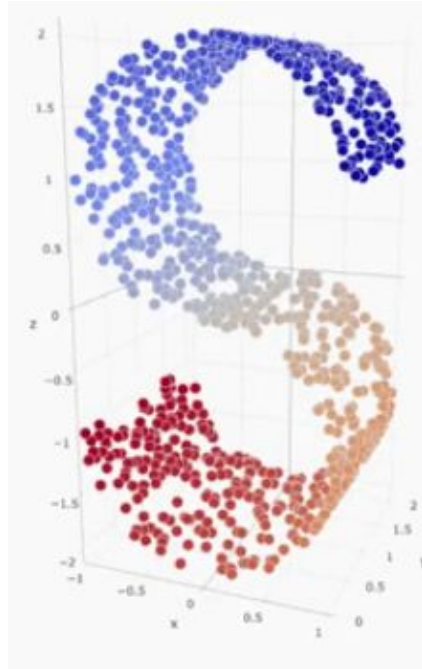
Значит объекты расположены в подпространстве, не являющемся линейным.

Нелинейные методы

Рассмотрим более простую модель и поставим задачу нелинейного понижения размерности:

Задача — найти отображение объектов выборки в пространство малой размерности, которое оптимизировало бы функционал качества.

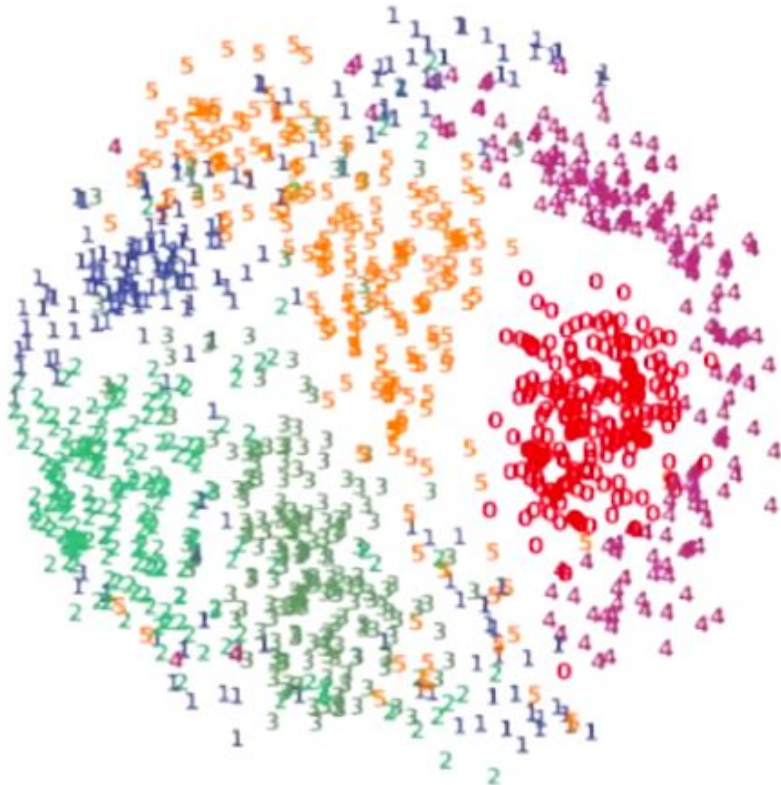
При этом мы не ограничены линейными отображениями.



Многомерное шкалирование

Гипотеза: малоразмерное представление сохраняет попарные расстояния между объектами.

d_{ij} - расстояние между x_i и x_j
 $\tilde{d}_{ij} = \|\tilde{x}_i - \tilde{x}_j\|$ - евклидово расстояние между малоразмерными представлениями



Функционал качества:

Ищем представления, аппроксимирующие d_{ij} :

$$\sum_{i < j}^{\ell} (\|\tilde{x}_i - \tilde{x}_j\| - d_{ij})^2 \rightarrow \min_{(\tilde{x}_i)_{i=1}^{\ell} \subset \mathbb{R}^d}$$

- стресс-функция

Алгоритм: SMACOF (Scaling by MAjorizing a COmplicated Function)

$$\sigma(X) = \sum_{i < j \leq n} w_{ij} (d_{ij}(X) - \delta_{ij})^2 = \sum_{i < j} w_{ij} \delta_{ij}^2 + \sum_{i < j} w_{ij} d_{ij}^2(X) - 2 \sum_{i < j} w_{ij} \delta_{ij} d_{ij}(X)$$



$$\begin{aligned} \sigma(X) &= C + \text{tr } X' V X - 2 \text{tr } X' B(X) X \\ &\leq C + \text{tr } X' V X - 2 \text{tr } X' B(Z) Z = \tau(X, Z) \end{aligned}$$



Repeat

$$Z \leftarrow X^{k-1}$$

$$X^k \leftarrow \min_X \tau(X, Z)$$

until $\sigma(X^{k-1}) - \sigma(X^k) < \epsilon$

Stochastic Neighbour Embedding (SNE)

Гипотеза: В точности воспроизвести расстояния – слишком сложно.
Достаточно сохранения пропорций.

$$\rho(x_1, x_2) = c\rho(x_1, x_3) \Rightarrow \rho(\tilde{x}_1, \tilde{x}_2) = c\rho(\tilde{x}_1, \tilde{x}_3).$$

Опишем объекты нормированными расстояниями до остальных объектов:

$$p(x_j | x_i) = \frac{\exp(\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(\|x_i - x_k\|^2 / 2\sigma^2)} \quad q(\tilde{x}_j | \tilde{x}_i) = \frac{\exp(\|\tilde{x}_i - \tilde{x}_j\|^2)}{\sum_{k \neq i} \exp(\|\tilde{x}_i - \tilde{x}_k\|^2)}$$

Функционал качества:

Минимизируем разницу между распределениями расстояний с помощью дивергенции Кульбака-Лейблера:

$$\sum_{i=1}^{\ell} \sum_{j \neq i} p(x_j | x_i) \log \frac{p(x_j | x_i)}{q(\tilde{x}_j | \tilde{x}_i)} \rightarrow \min_{(\tilde{x}_i)_{i=1}^{\ell} \subset \mathbb{R}^d}$$

Алгоритм: (Стохастический) градиентный спуск

Repeat

$$y_i = y_i - \lambda \frac{\partial KL(P || Q)}{\partial y_i}$$

until convergence

$$KL(P || Q) = \sum_j \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$\frac{\partial KL(P || Q)}{\partial y_i} = 4 \sum_{j \neq i} Z(p_{ij} - q_{ij}) q_{ij} (y_i - y_j),$$

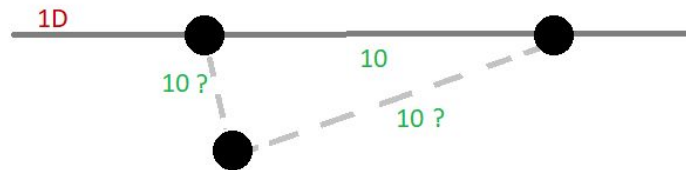
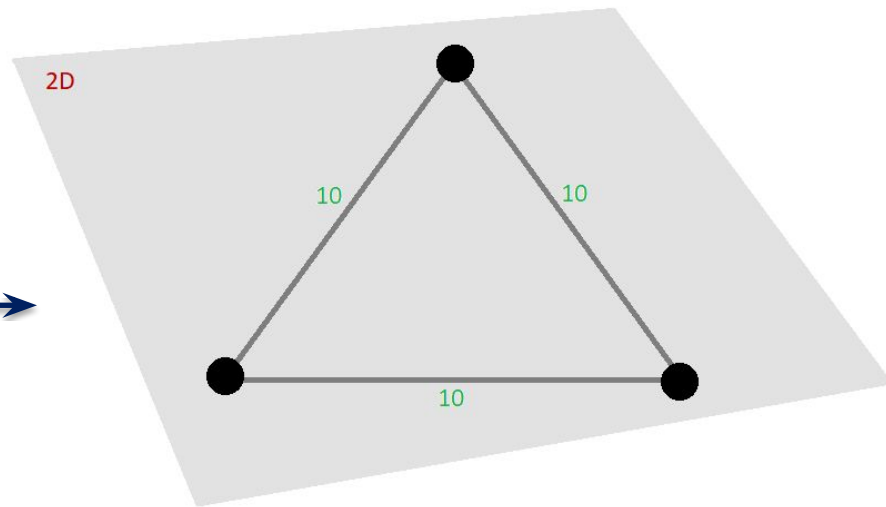
$$Z = \sum_k \sum_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1}$$

t-distributed SNE

Чем выше размерность пространства, тем меньше расстояния между парами точек отличаются друг от друга (проклятие размерности).



Это затрудняет точное сохранение пропорций в двух- или трехмерном пространстве.

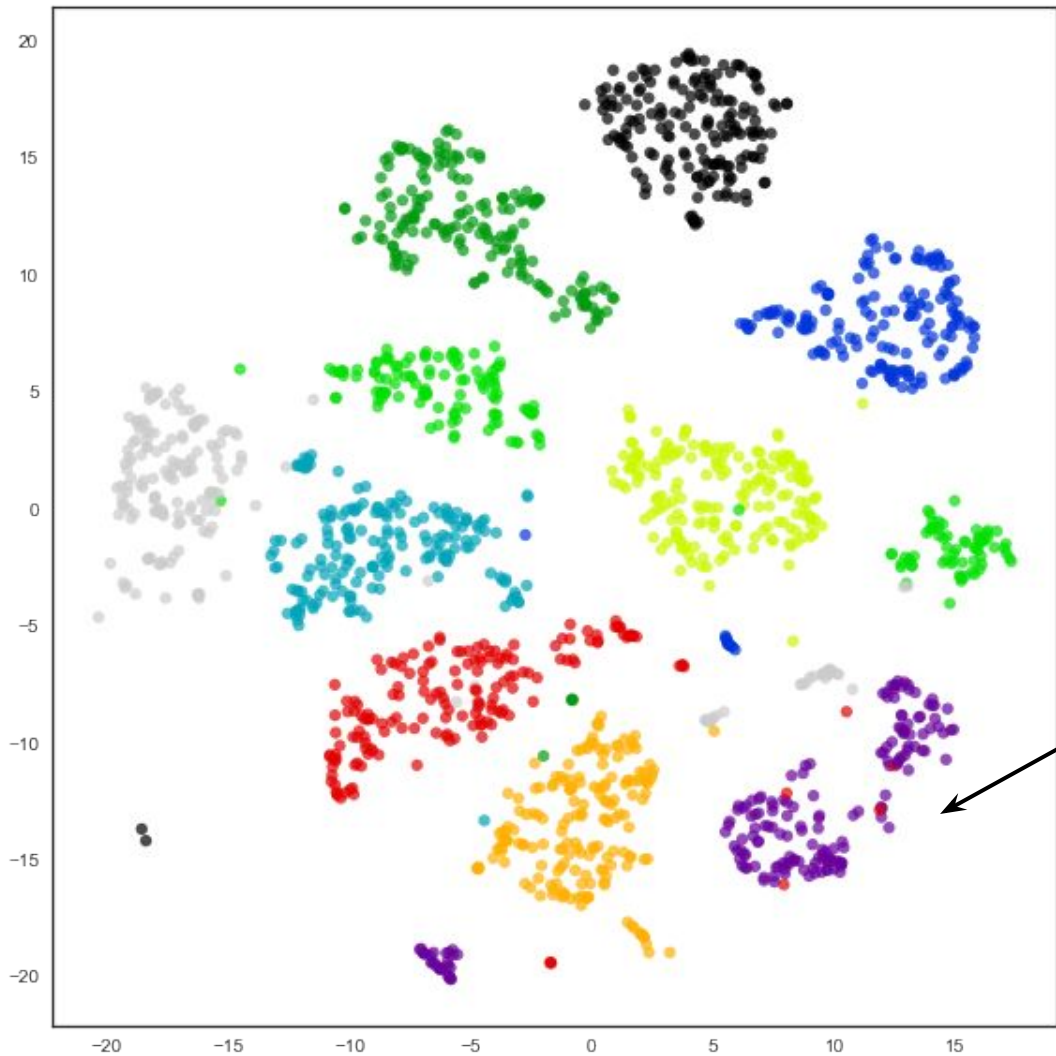


Значит нужно меньше штрафовать за увеличение пропорций в маломерном пространстве.

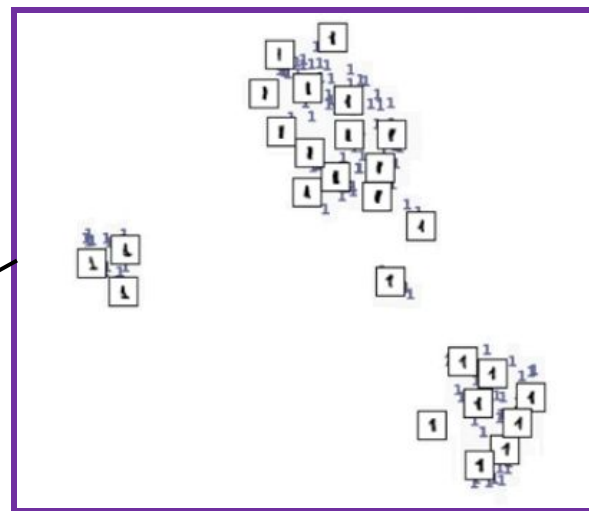
Изменим распределение:

$$q(\tilde{x}_j | \tilde{x}_i) = \frac{(1 + \|\tilde{x}_i - \tilde{x}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\tilde{x}_i - \tilde{x}_k\|^2)^{-1}}$$





Сохраняет кластерную структуру самих классов



Сравнение методов

Метод главных компонент

- | | |
|---------------------------------------|--|
| + быстро работает | - Находит только линейные комбинации признаков |
| + в общем сохраняет больше информации | - Чувствителен к масштабированию признаков |
| + можно восстановить исходные данные | |

Многомерное шкалирование

- | | |
|--|---------------------------------------|
| + лучше визуализирует структуру данных | - Страдает от «проклятья размерности» |
| | - Сложный алгоритм оптимизации |

t-SNE

- | | |
|---|---|
| + лучше всего отображает кластерную структуру данных | - Долго работает |
| + можно оптимизировать стохастическим градиентным спуском | - Невозможно восстановить исходные данные |

Выводы

- Существует множество методов визуализации многомерных данных
- Выбор метода сильно зависит от конкретной задачи
- Ключевым фактором при выборе метода является балансирование между большей потерей информации и лучшей визуализацией структуры данных

Спасибо за внимание

