Нелинейные методы снижения размерности

Кантонистова Елена

elena.kantonistova@yandex.ru

1 ноября 2017

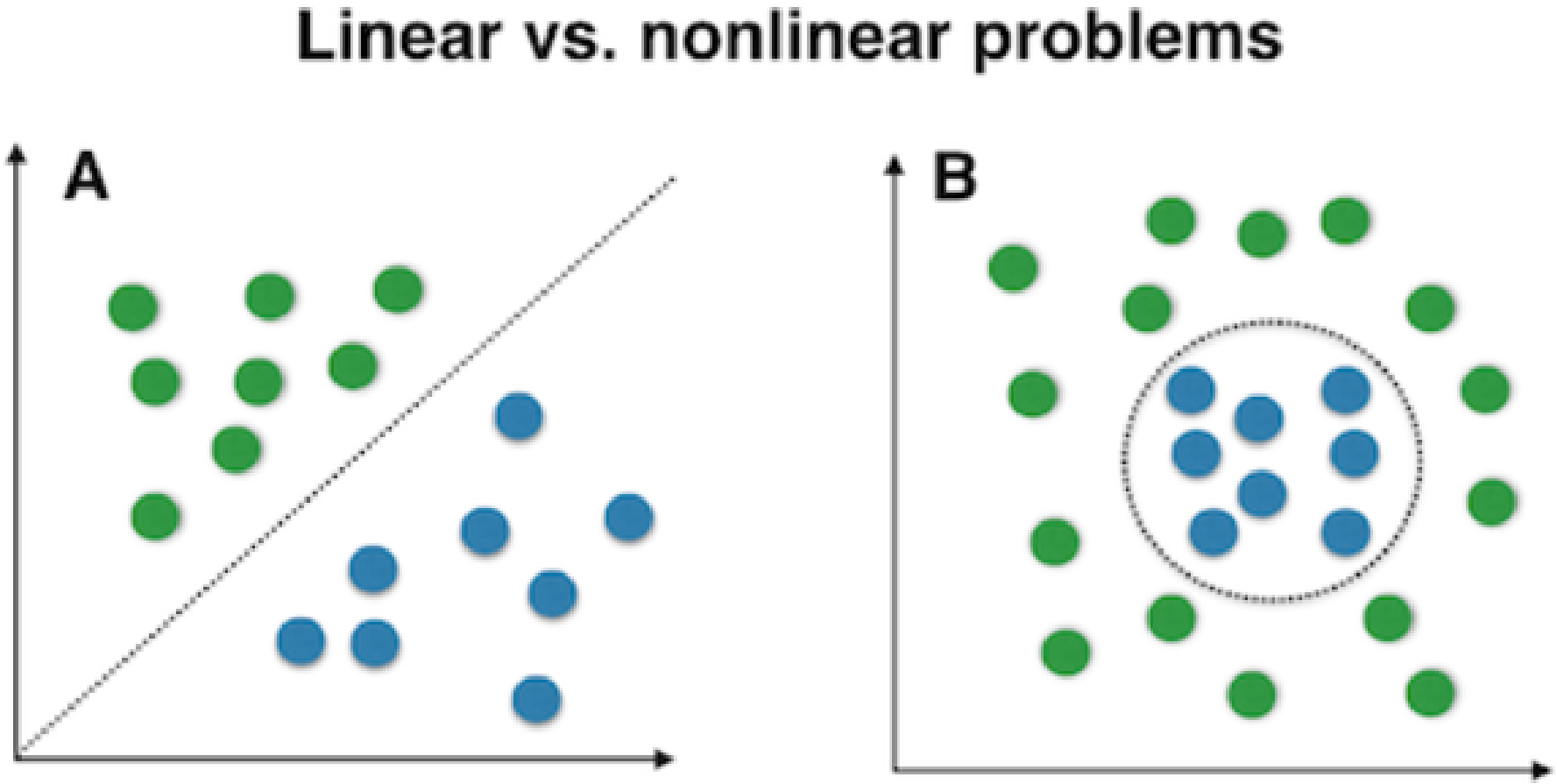
План лекции

1. Ядровой PCA
2. t-SNE

PCA (напоминание)

Метод главных компонент (PCA) преобразует базовые признаки в новые, каждый из которых является линейной комбинацией изначальных таким образом, что разброс данных (то есть среднеквадратичное отклонение от среднего значения) вдоль них максимален. Метод применяется для визуализации данных и для уменьшения размерности данных (сжатия).

Когда обычный(линейный) PCA бессилен

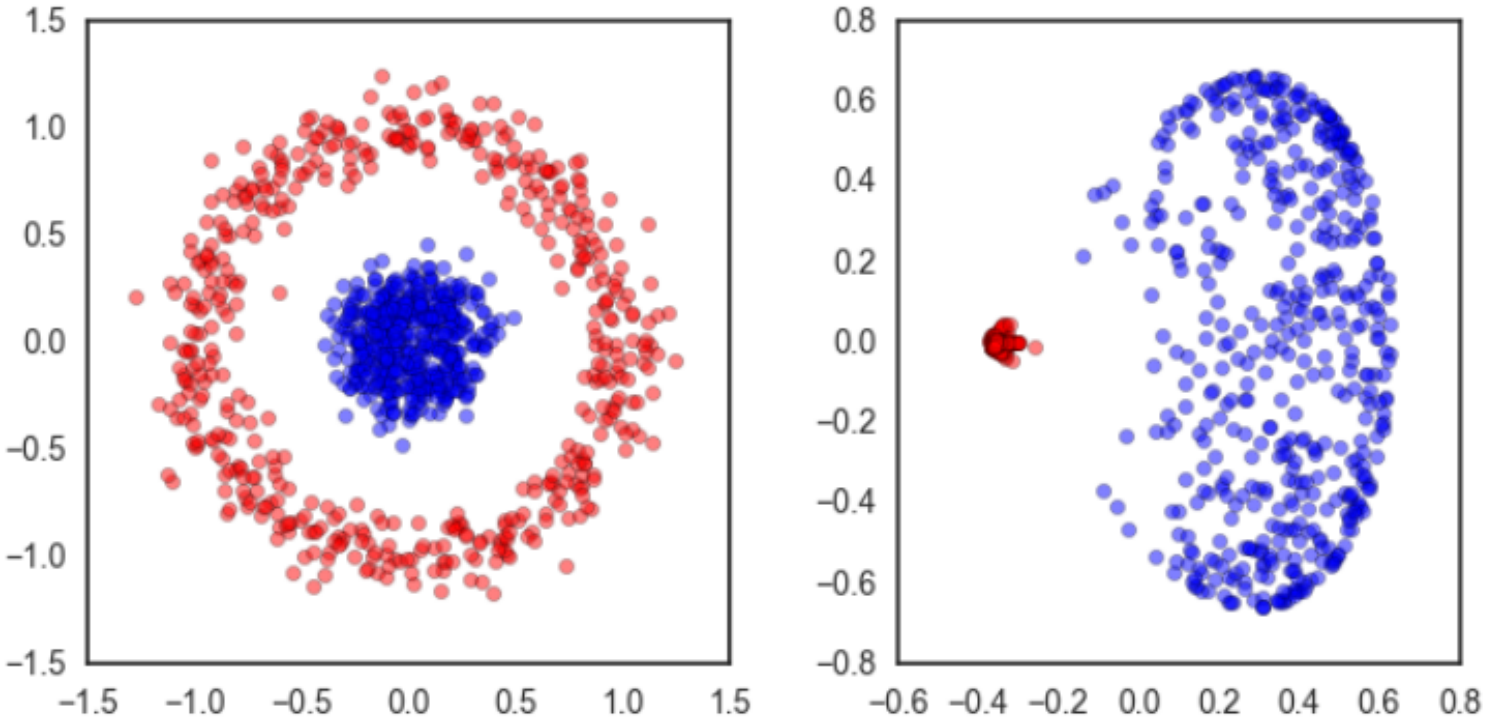


PCA с ядром

PCA с ядром (Kernel PCA) основан на том же принципе, что и обычный PCA, однако он позволяет создавать новые признаки не только линейной комбинацией исходных, но и более сложными преобразованиями.
Это происходит благодаря замене обычного скалярного произведения (x, y) , возникающего при решении задачи PCA, на некоторую функцию $K(x, y)$, обладающую всеми свойствами скалярного произведения.

Зачем нужны ядра

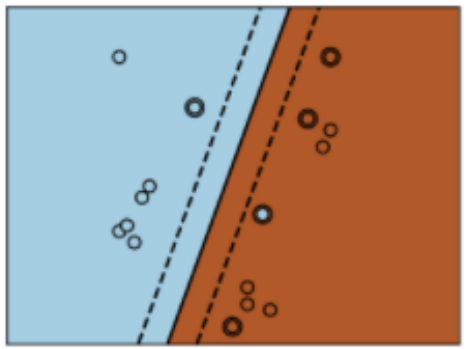
Иногда данные в задаче не являются линейно разделимыми. Однако, после некоторого нелинейного преобразования признаков данные можно разделить прямой. Введение ядер - это как раз и есть то самое нелинейное преобразование признаков, которое помогает нам впоследствии разделить данные прямой.

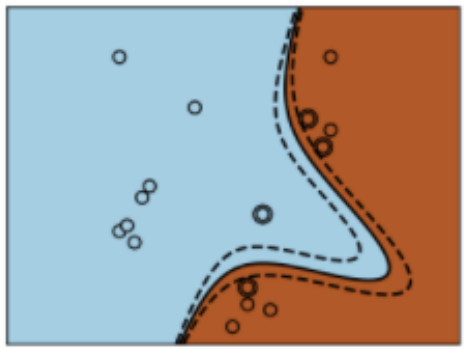


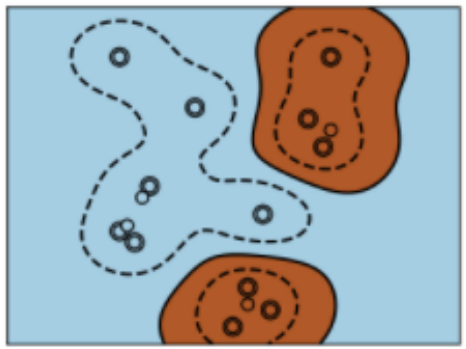
Распространенные ядра

- Линейное
- Полиномиальное
- RBF $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$
-

Ядра







t-SNE

t-SNE = t-distributed stochastic neighbor embedding

Данный метод нужен для визуализации многомерных данных в двумерном или трехмерном пространстве.

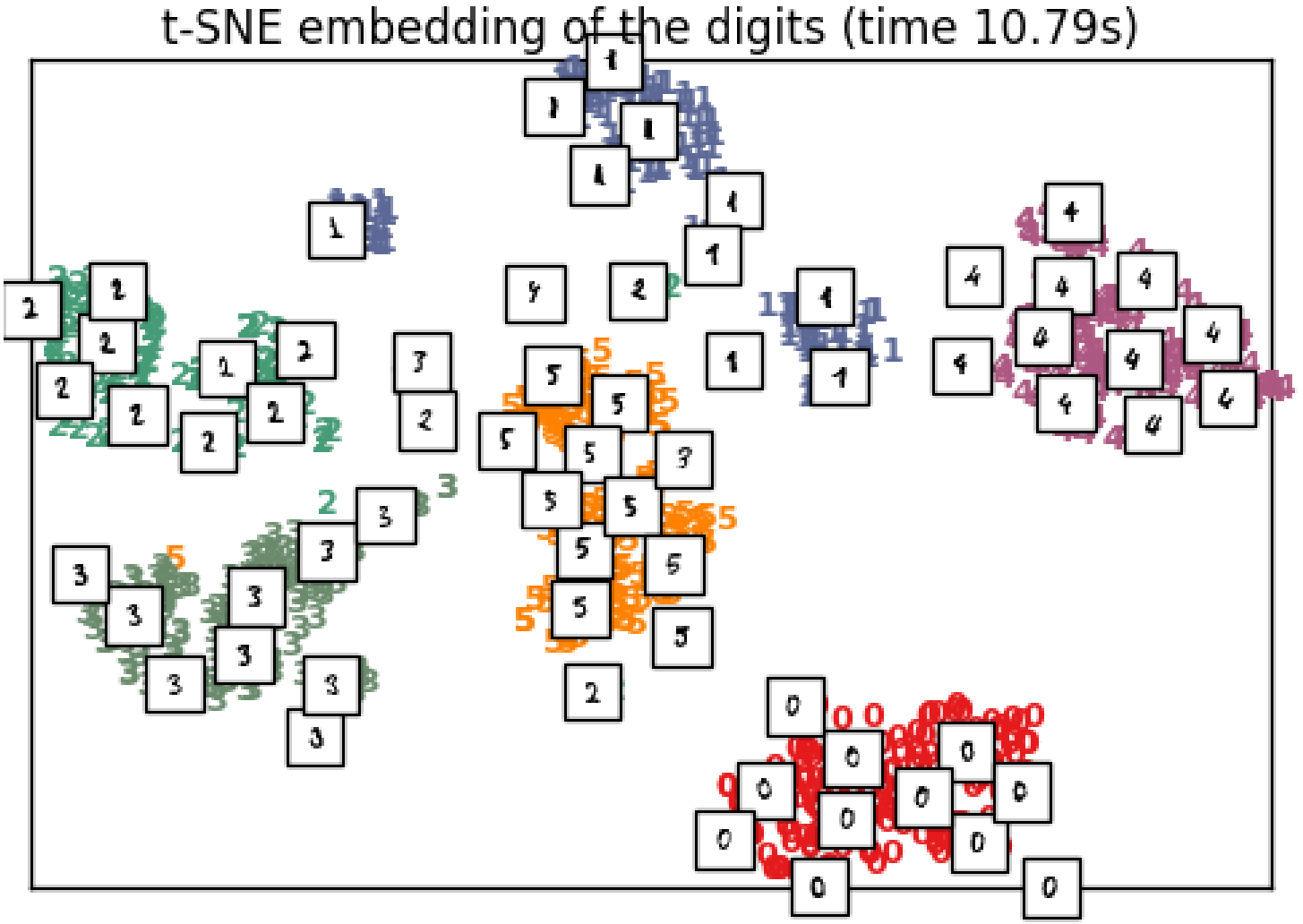
Идея метода

У нас есть набор данных с точками, описываемыми многомерной переменной с размерностью пространства существенно больше трех. Необходимо получить новую переменную, существующую в двумерном или трехмерном пространстве, которая бы в максимальной степени сохраняла структуру и закономерности в исходных данных.
Другими словами, точки, которые в исходном пространстве находятся далеко друг от друга, в новом 2-х или 3-хмерном тоже должны оказаться далеко друг от друга (и наоборот, близкие - близко).

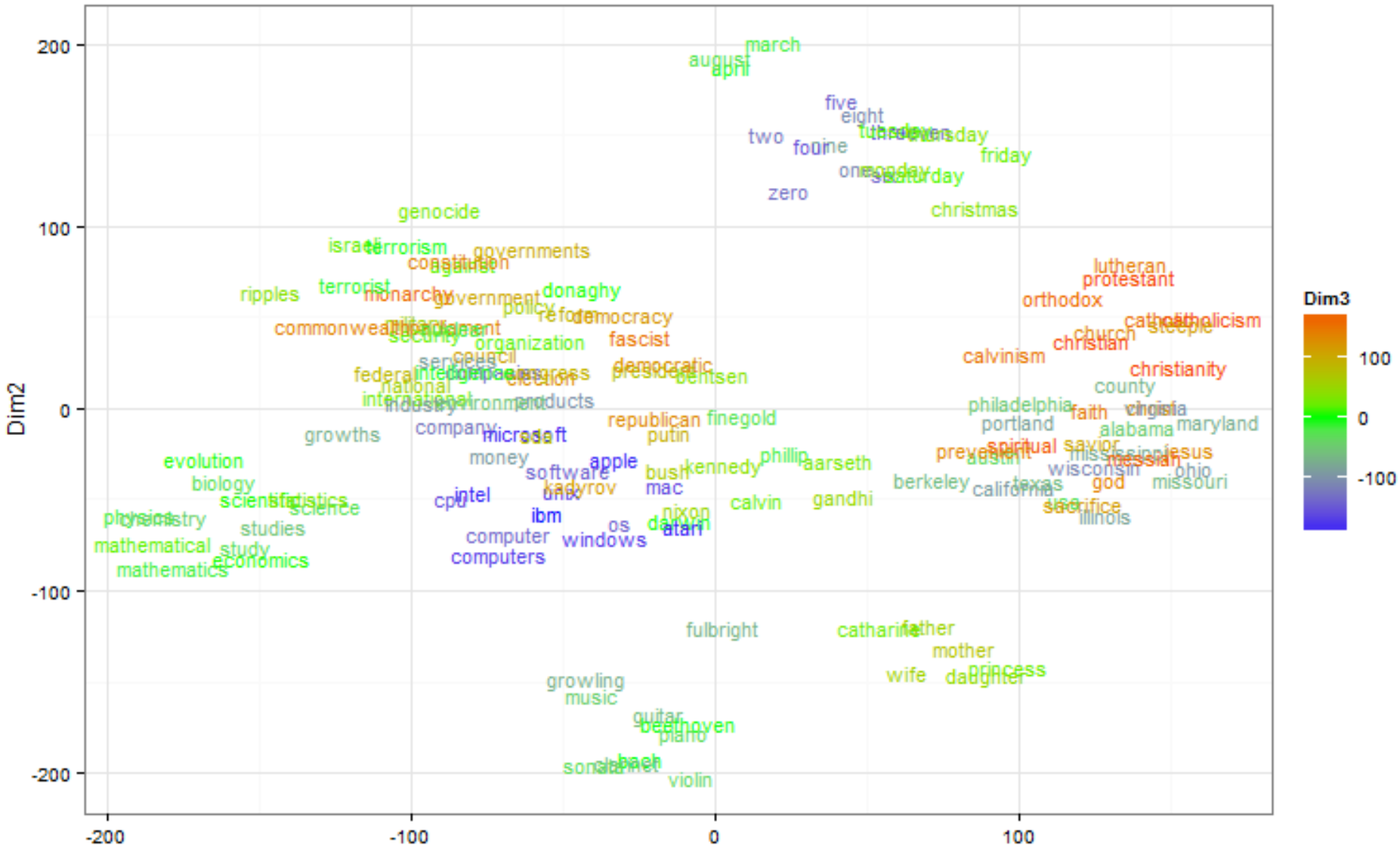
Пример использования: MNIST

A selection from the 64-dimensional digits dataset

Проекция в двумерное пространство с помощью t-SNE



Визуализация близости слов с помощью t-SNE



Плюсы и минусы t-SNE

- Основной плюс: с помощью t-SNE можно получить очень хорошее наглядное представление о том, как выглядят многомерные данные
- Основной минус: метод работает очень долго, даже реализация из sklearn