

# Машинное обучение (Machine Learning)

## Визуализация данных с использованием tSNE

Уткин Л.В.

Санкт-Петербургский политехнический университет Петра Великого



# Содержание

- 1 Методы визуализации и понижения размерности
- 2 Метод t-SNE

# Методы визуализации и понижения размерности

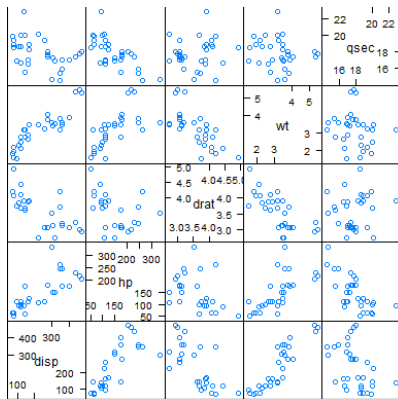
# “Хорошая” визуализация (требования)

- Каждый объект с большой размерностью представляется объектом с малой размерностью
- Сохранение “соседства” двух объектов в разных пространствах
- Удаленные точки соответствуют отличающимся объектам
- Масштабируемость

# Более формально

- Точка данных - это точка  $x_i$  в исходном пространстве  $\mathbb{R}^D$
- Образ - точка  $y_i$  в пространстве  $\mathbb{R}^2$  или  $\mathbb{R}^3$ . Каждый образ соответствует одной исходной точке
- Алгоритм визуализации выбирает положение образов в  $\mathbb{R}^2$  или  $\mathbb{R}^3$  в соответствии с определенными правилами (в основном для сохранения пространственной структуры данных)

# Диаграммы рассеяния



# Методы сокращения размерности



# Линейный дискриминантный анализ (LDA)

- LDA является параметрическим, так как предполагает унимодальное нормальное распределение данных
- Если распределения существенно не Гауссовы, то LDA не сохраняет сложную структуру данных
- LDA может быть ошибочным, когда разделяющая информация содержится не столько в средних, сколько в дисперсии данных



# Многомерное масштабирование (MDS: Multi-Dimensional Scaling)

Многомерное масштабирование упорядочивает точки с малой размерностью так, чтобы минимизировать несходство между попарными расстояниями в исходном пространстве и пространстве малой размерности

$$Cost = \sum_{i < j} (d_{ij} - \hat{d}_{ij})^2, \quad d_{ij} = \|x_i - x_j\|^2, \quad \hat{d}_{ij} = \|y_i - y_j\|^2$$

# Отображение, которое сохраняет локальную геометрию (LLE)

- **LLE (Locally Linear Embedding) метод:**
- Идея - сделать локальные конфигурации точек в пространстве малой размерности похожими на локальные конфигурации в пространстве высокой размерности

$$Cost = \sum_i \left\| x_i - \sum_{j \in N(i)} w_{ij} x_j \right\|^2, \quad \sum_{j \in N(i)} w_{ij} = 1$$

- Фиксированные веса

$$Cost = \sum_i \left\| y_i - \sum_{j \in N(i)} w_{ij} y_j \right\|^2$$

- Найти  $y$ , который минимизирует потери при ограничениях:  $y$  имеют единичную дисперсию для каждой размерности

# Метод t-SNE

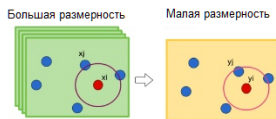
# Вероятностная версия локального MDS: Stochastic Neighbor Embedding (SNE)

L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008.

*Основная идея SNE - конвертировать близость каждой пары точек в исходном пространстве  $\mathbb{R}^D$  большой размерности в вероятность того, что одна точка данных связана с другой точкой как с ее соседом.*

*SNE сохраняет локальную структуру данных в  $\mathbb{R}^2$  или  $\mathbb{R}^3$*

# Мера сходства точек в исходном пространстве



- Преобразование многомерного Евклидова расстояния между точками в условные вероятности, отражающие сходство точек  $x_i$  в  $\mathbb{R}^D$ :

$$p_{j|i} = \frac{e^{-\|x_i - x_j\|^2 / 2\sigma_i^2}}{\sum_k e^{-\|x_i - x_k\|^2 / 2\sigma_i^2}}$$

- $p_{j|i}$  показывает, насколько точка  $x_j$  близка к точке  $x_i$  при гауссовом распределении вокруг  $x_i$  с заданным отклонением  $\sigma_i$ .

# Мера сходства точек в исходном пространстве

$$p_{j|i} = \frac{e^{-\|x_i - x_j\|^2 / 2\sigma_i^2}}{\sum_k e^{-\|x_i - x_k\|^2 / 2\sigma_i^2}}$$

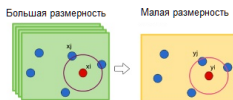
- Отклонение  $\sigma_i$  для каждой точки выбирается так, чтобы точки в областях с большей плотностью имели меньшую дисперсию или фиксированную оценку перплексии (оценка эффективного количества «соседей» для  $x_i$ ):

$$2^{H(p_{j|i})} = 2^{-\sum_x p_{j|i} \log_2 p_{j|i}}$$

- Симметричная мера сходства точек (для упрощения вычисления градиента):

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \quad p_{ii} = 0.$$

# Мера сходства точек в пространстве малой размерности



- Условные вероятности, отражающие сходство точек  $y_i$  в  $\mathbb{R}^2$  или  $\mathbb{R}^3$ :

$$q_{j|i} = \frac{e^{-\|y_i - y_j\|^2 / 2\sigma_i^2}}{\sum_k e^{-\|y_i - y_k\|^2 / 2\sigma_i^2}}, \quad \sigma_i = 1/\sqrt{2}$$

- Симметричная мера сходства точек:

$$q_{ij} = \frac{q_{j|i} + q_{i|j}}{2n}, \quad q_{ii} = 0$$

# Кульбак-Лейблер дивергенция

- Если точки  $y_i$  и  $y_j$  корректно моделируют сходство между точками  $x_i$  и  $x_j$ , то соответствующие условные вероятности  $p_{ij}$  и  $q_{ij}$  будут эквивалентны
- Распределение  $(p_{ij})$  фиксировано,  $(q_{ij})$  может меняться
- Мы хотим, чтобы  $(p_{ij})$  и  $(q_{ij})$  были бы как можно ближе.
- Используем дивергенцию Кульбака-Лейблера

$$Cost = \sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{ji} \log \frac{p_{ij}}{q_{ij}}$$



# Дивергенция Кульбака-Лейблера и градиент

Градиент:

$$\frac{\partial \text{Cost}}{\partial y_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij}) (y_i - y_j)$$

# Распределение Стьюдента

- t-распределение Стьюдента с одной степенью свободы (Коши распределение)

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq i} \left(1 + \|y_i - y_k\|^2\right)^{-1}}, \quad q_{ii} = 0$$

- Тяжелые хвосты: решается проблема скученности (расстояние между двумя точками в  $\mathbb{R}^2$ , соответствующими двум среднеудаленным точкам в  $\mathbb{R}^D$ , должно быть существенно больше, чем расстояние, которое позволяет получить гауссово распределение)
- Проще с вычислительной точки зрения (нет экспоненты)

# t-распределение Стьюдента и градиент

- Градиент:

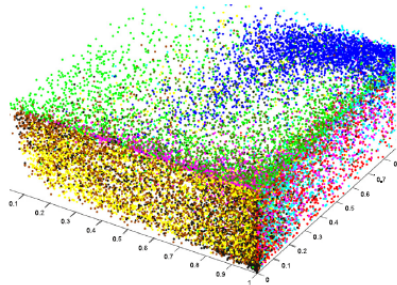
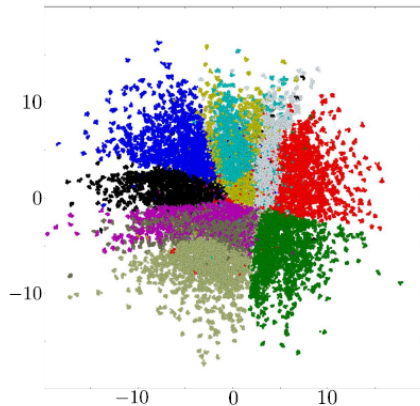
$$\frac{\partial \text{Cost}}{\partial y_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij}) \frac{y_i - y_j}{1 + \|y_i - y_j\|^2}$$

- Затем в цикле  $t = 1, \dots, T$ :

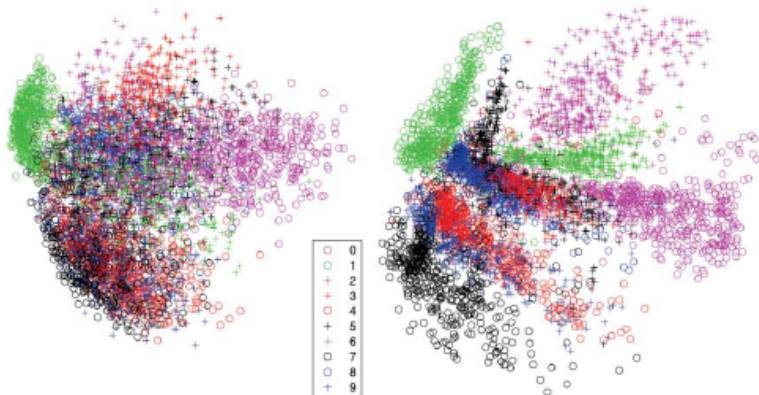
$$Y(t) = Y(t-1) + \eta \frac{\partial \text{Cost}}{\partial Y} + \alpha(t) (Y(t-1) - Y(t-2))$$

- $\eta$  – параметр, определяющий скорость обучения,  $\alpha$  – коэффициент инерции.

# Примеры того, что получается



# Еще примеры того, что получается



# Как эффективно использовать t-SNE

Wattenberg, et al. "How to Use t-SNE Effectively", Distill, 2016.

(<http://distill.pub/2016/misread-tsne/>)

# t-SNE в R

- R-пакет **tsne** (Package 'tsne')
- `tsne(X, initial_config = NULL, k = 2, initial_dims = 30, perplexity = 30, max_iter = 1000, min_cost = 0, epoch_callback = NULL, whiten = TRUE, epoch = 100)`
- Интересное описание с реализацией:  
<https://habrahabr.ru/post/267041/>

# Ресурсы

- Visualizing MNIST: An Exploration of Dimensionality Reduction <http://colah.github.io/posts/2014-10-Visualizing-MNIST/>



# Вопросы

?