Алгоритмы снижения размерности

В статистике, машинном обучении и анализе данных снижение размерности – это преобразование данных, состоящее в уменьшении числа переменных путём получения главных переменных. Например, если характеристика представлена недостаточно полно (много пропусков, неоднородность ее показателей). Снижение размерности осуществляется для:

1. Уменьшения времени и памяти для обучения модели
2. Повышение качества модели
3. Проще представить данные визуально, если свести к очень низким размерностям, таким как 2D или 3D.

Алгоритмы понижения размерности можно разделить на 2 основные группы: они пытаются сохранить либо глобальную структуру данных, либо локальные расстояния между точками. К первым относятся такие алгоритмы как Метод главных компонент (PCA) и MDS (Multidimensional Scaling), а ко вторым — t-SNE, ISOMAP, LargeVis, UMAP и другие.

PCA (principal component analysis)

В переводе – метод главных компонент. Один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации. Зная зависимости и их силу, мы можем выразить несколько признаков через один. Конечно, избежать потерь информации, скорее всего не удастся, но минимизировать ее нам поможет как раз метод PCA.

Для начала несколько понятий. **Дисперсия** – величина показывающая, как размазаны наши данные. Математически, это средний квадрат отклонений от среднего значения по выборке, где E(t) - математическое ожидание какой-то величины t, а - значение x в i – ом измерении.

$$var(x) = \frac{\sum(x_i - \bar{x})^2}{N} \quad cov(x,y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N}$$

Ковариация – мера линейной зависимости двух выборок. Положительная ковариация означает то, что с увеличением X также увеличивается и У. Отрицательная ковариация изображает в точности противоположную взаимосвязь. Нулевая ковариация означает то, что X и У не связаны вообще.

Мы хотим, чтобы данные были достаточно распределены в пространстве, т.е. имели высокую дисперсию по каждому измерению. Также хотелось бы убрать коррелируемые измерения, т.е. ковариация между измерениями должна быть нулевой (они должны быть линейно независимы). Т.е. матрица ковариаций должна быть приближенной к диагональной.

PCA находит множество новых измерений (сторон, с которых можно смотреть на данные), которые ортогональны (таким образом линейно независимы) и отранжированы по величине дисперсии, которую они объясняют. Это значит, что более значимые principal оси будут первыми (более значимые = больше дисперсия/ больше разброс в данных).

1. Стандартизация, нормирование признаков
2. Вычисление ковариационной матрицы по всем признакам, которая описывает разброс случайной величины (новой характеристики)
3. Вычисление собственных значений и векторов
4. Сортировка собственных векторов по значениям их собственных значений в убывающем порядке
5. Выбрать k первых собственных векторов и это будет новое k-мерное пространство
6. Преобразовать исходные точки из n-мерного пространства в k-мерное

Ошибка проекции - сумма ошибок проекций по всем точкам. Для проекции из двумерного случая в одномерный - сумма длин перпендикуляров к проекции

PCA - это метод линейного уменьшения размеров, который стремится максимизировать дисперсию и сохраняет большие попарные расстояния. Другими словами, разные вещи оказываются далеко друг от друга. Это может привести к плохой визуализации, особенно при работе с нелинейными структурами многообразия.

Ссылки:
http://data4.ru/pca
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
https://habr.com/ru/post/304214/

T - SNE

t-SNE отличается от PCA тем, что сохраняет только небольшие попарные расстояния или локальные сходства, тогда как PCA занимается сохранением больших попарных расстояний для максимизации дисперсии. Очень часто t-SNE используется поверх PCA.

Алгоритм t-SNE вычисляет меру сходства между парами экземпляров в пространстве с высокой размерностью и в пространстве с низкой размерностью. Затем он пытается оптимизировать эти два показателя сходства, используя функцию стоимости.

Сначала t-SNE создаёт распределение вероятностей по парам объектов высокой размерности таким образом, что похожие объекты будут выбраны с большой вероятностью, в то время как вероятность выбора непохожих точек будет мала. Затем t-SNE определяет похожее распределение вероятностей по точкам в пространстве малой размерности и минимизирует расстояние Кульбака — Лейблера между двумя распределениями с учётом положения точек.

алгоритм t-SNE часто способен обнаружить хорошо отделённые друг от друга кластеры

1. Измерить сходства между точками (признаками) в данном многомерном пространстве. Распределение Стьюдента (t-распределение)
2. Измерить сходства между точками (признаками) в пространстве меньше размерности

Ссылки:
https://www.machinelearningmastery.ru/an-introduction-to-t-sne-with-python-example-5a3a293108d1/
https://habr.com/ru/post/267041/

UMAP

UMAP (Uniform Manifold Approximation and Projection) — это новый алгоритм уменьшения размерности, библиотека с реализацией которого вышла совсем недавно. У UMAP нет ограничений на размерность исходного пространства признаков, которое необходимо уменьшить, он намного быстрее и более вычислительно эффективен, чем t-SNE, а также лучше справляется с задачей переноса глобальной структуры данных в новое, уменьшенное пространство. К тому же UMAP использует не только линейные функции для перевода.

Интуитивно: на первом этапе оцениваем пространство, в котором по нашему предположению и находятся данные. Его можно задать априори (как просто R^n), либо оценить на основе данных. А на втором шаге пытаемся создать отображение из оцененного на первом этапе пространства в новое, меньшей.

Число соседей — n_neighbors. Варьируя этот параметр, можно выбирать, что важнее сохранить в новом пространственном представлении данных: глобальную или локальную структуру данных. Маленькие значения параметра означают, что, пытаясь оценить пространство, в котором распределены данные, алгоритм ограничивается малой окрестностью вокруг каждой точки, то есть пытается уловить локальную структуру данных (возможно в ущерб общей картине). С другой стороны большие значения *n_neighbors* заставляют UMAP учитывать точки в большей окрестности, сохраняя глобальную структуру данных, но упуская детали.

Минимальное расстояние — min_dist. Данный параметр стоит понимать буквально: он определяет минимальное расстояние, на котором могут находиться точки в новом пространстве. Низкие значения стоит применять в случае, если вас интересует, на какие кластеры разделяются ваши данные, а высокие — если вам важнее взглянуть на структуру данных, как единого целого.

Метрика расстояния — metric. Определяет, каким образом будут рассчитаны расстояния в пространстве исходных данных

Размерность конечного пространства — n_components

1. Расчет всех расстояний между объектами, k ближайших соседей и расстояние до ближайшего
2. Построение взвешенного неориентированного графа. Вес ребра рассчитывается следующим образом: $w(x_i \to t_j) = \exp\left(-\frac{d(x_i, t_j) - \rho_i}{\sigma_i}\right)$

$$w(x_i, x_j) = w(x_i \to x_j) + w(x_j \to x_i) - w(x_i \to x_j) \cdot w(x_j \to x_i).$$

3. Алгоритм создает новый граф в низкоразмерном пространстве и приближает множество его ребер к исходному. Для этого он минимизирует сумму дивергенций Кульбака–Лейблера для каждого ребра из исходного и нового множеств

UMAP решает задачу минимизации с помощью SGD и полученное множество из ребер определяет новое расположение объектов и, соответственно, низкоразмерное отображение исходного пространства.

Ссылки:
https://m.habr.com/ru/company/newprolab/blog/350584/
https://ru.wikipedia.org/wiki/UMAP
https://umap-learn.readthedocs.io/en/latest/
https://github.com/elizacc/CW_2019/blob/master/Makhneva%202019.pdf