МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Механико-математический факультет
Кафедра «Теоретическая кибернетика»
Направление подготовки «Прикладная математика и информатика»,
магистратура

КУРСОВАЯ РАБОТА МАГИСТРА

Кочанов Максим Александрович

Разработка метода визуализации многомерных данных

Заведующий кафедрой

д.ф.-м.н., профессор

Ерзин А.И. /_____

«____»_____20____г.

Научный руководитель

д.т.н., профессор

Постовалов С.Н. /_____

«____»_____20____г.

Новосибирск
2022

Реферат

Название работы: Разработка метода визуализации многомерных данных

Количество страниц: 28

Количество рисунков: 33

Количество использованных источников: 9

Ключевые слова: понижение размерности многомерных данных, визуализация данных, анализ данных, tSNE, PCA, UMAP

КРАТКОЕ ОПИСАНИЕ

В курсовой работе рассматривается подход к решению проблемы потери информации о глобальной структуре данных в существующих алгоритмах понижения размернсоти данных (tSNE, UMAP) с помощью разработки нового алгоритма gSNE.

Содержание

1. ВВЕДЕНИЕ

Задача понижения размерности является одной из важных задач разведочного анализа данных для выявления структуры многомерных данных. Существует достаточно большое число методов понижения размерности.

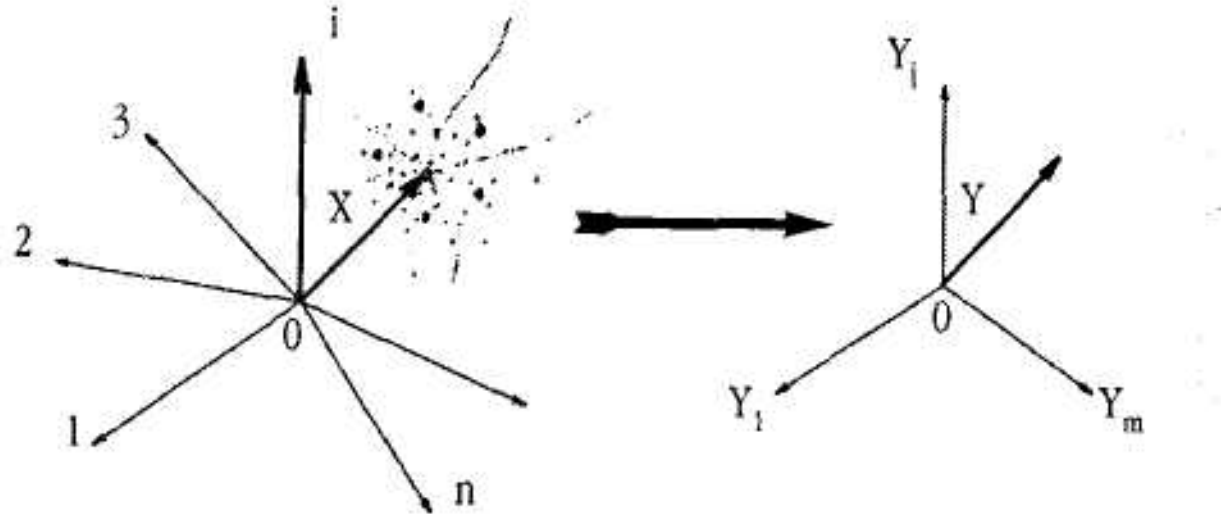
Наиболее известным методом является метод главных компонент (PCA), который линейно проецирует точки из пространства большой размерности в пространство меньшей размерности. В настоящее время более наглядными считаются методы, которые проецируют точки с помощью нелинейного отображения, такие как tSNE и UMAP, в которых используется принцип, что близкие точки в исходном пространстве должны быть близкими в пространстве меньшей размерности.

Недостатком этих методов является излишнее внимание к локальной структуре и потеря информации о глобальной структуре многомерных данных. В tSNE эта проблема является следствием проблемы скученности. Проблема скученности состоит в следующем: при снижении размерности данных с помощью tSNE далекие точки в пространстве высокой размерности могут оказаться близкими в новом пространстве низкой размерности.

Целью работы является разработка нового метода понижения размерности gSNE путем замены t-распределения Стьюдента на обобщенное гауссовское распределение.

1.1. Аналитический обзор

В статистике, машинном обучении и теории информации снижение размерности — это преобразование данных, состоящее в уменьшении числа переменных путём получения главных переменных. Преобразование может быть разделено на отбор признаков и проекцию признаков. В настоящей работе мы подробно рассмотрим методы проекции признаков.

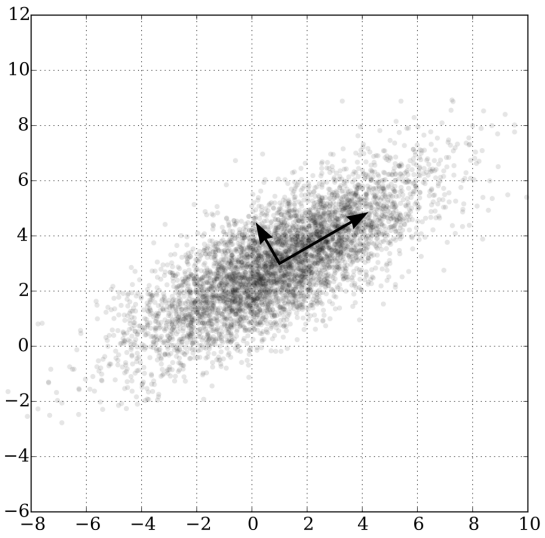


Проекция признаков

1.1.1. Метод главных компонент

Метод главных компонент (Principal Component Analysis) — один из самых интуитивно простых и часто используемых методов для снижения размерности данных и проекции их на ортогональное подпространство признаков.

В совсем общем виде это можно представить как предположение о том, что все наши наблюдения скорее всего выглядят как некий эллипсоид в подпространстве нашего исходного пространства и наш новый базис в этом пространстве совпадает с осями этого эллипсоида. Это предположение позволяет нам одновременно избавиться от сильно скоррелированных признаков, так как вектора базиса пространства, на которое мы проецируем, будут ортогональными.



В общем случае размерность этого эллипсоида будет равна размерности исходного пространства, но наше предположение о том, что данные лежат в подпространстве меньшей размерности, позволяет нам отбросить "лишнее"подпространство в новой проекции, а именно то подпространство, вдоль осей которого эллипсоид будет наименее растянут. Мы будем это делать жадно, выбирая по-очереди в качестве нового элемента базиса нашего нового подпространства последовательно ось эллипсоида из оставшихся, вдоль которой дисперсия будет максимальной.

Рассмотрим как это делается математически: Чтобы снизить размерность наших данных из n в $k, k \leq n$, нам нужно выбрать топ-k осей такого эллипсоида, отсортированные по убыванию по дисперсии вдоль осей.

Начнём с того, что посчитаем дисперсии и ковариации исходных признаков. Это делается просто с помощью матрицы ковариации. По определению ковариации, для двух признаков X_i и X_j их ковариация будет

$$cov(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] = E[X_i X_j] - \mu_i \mu_j \tag{1.1}$$

где μ_i — матожидание i-ого признака. При этом отметим, что ковариация симметрична и ковариация вектора с самим собой будет равна его дисперсии.

Таким образом матрица ковариации представляет собой симметричную матрицу, где на диагонали лежат дисперсии соответствующих признаков, а вне диагонали — ковариации соответствующих пар признаков. В матричном

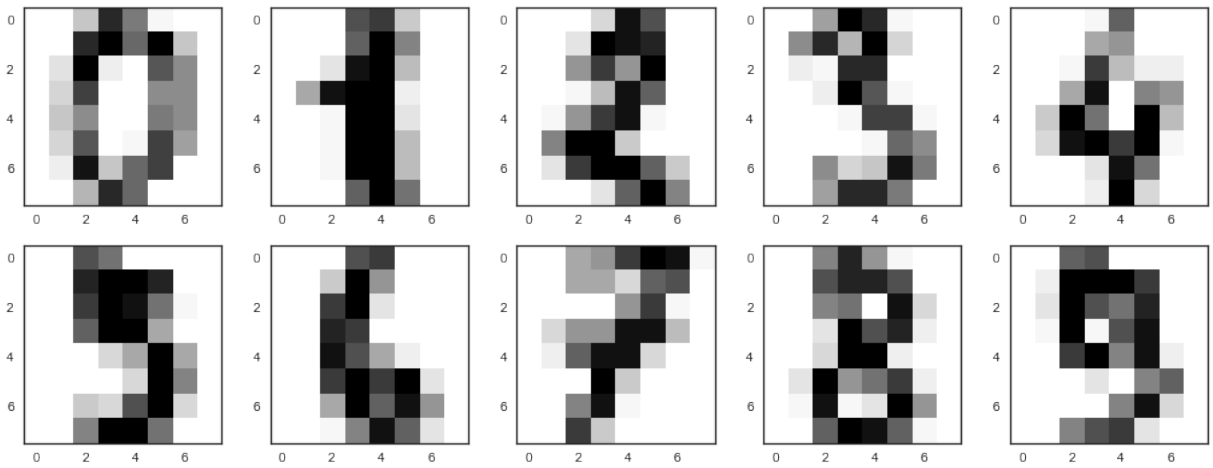
виде, где $\mathbf{X}$ это матрица наблюдений, наша матрица ковариации будет выглядеть как

$$\Sigma = E[(\mathbf{X} - E[\mathbf{X}])(\mathbf{X} - E[\mathbf{X}])^T] \tag{1.2}$$

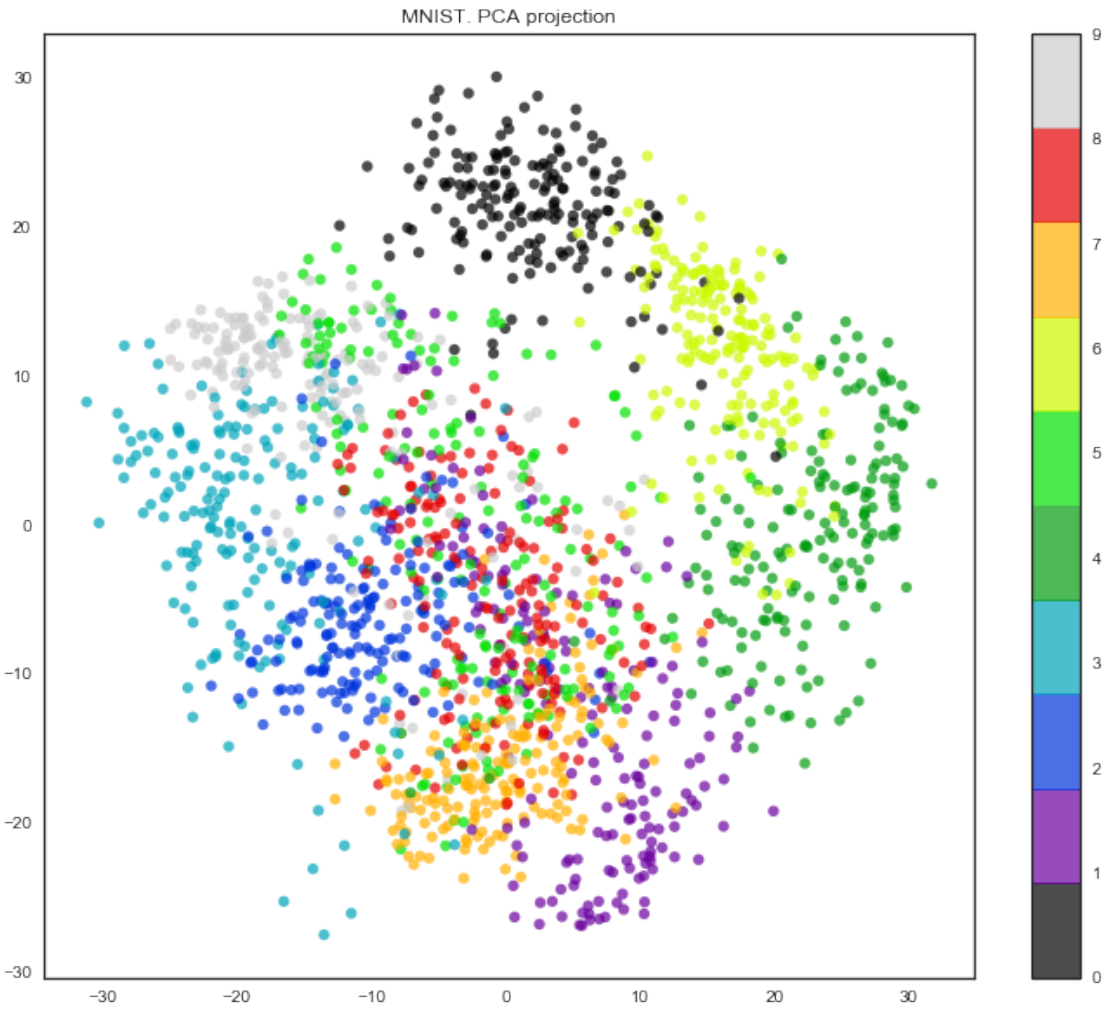
Матрицу ковариации для нашей выборки $\mathbf{X}$ можно представить в виде произведения $\mathbf{X}^T\mathbf{X}$. Из отношения Релея вытекает, что максимальная вариация нашего набора данных будет достигаться вдоль собственного вектора этой матрицы, соответствующего максимальному собственному значению. Таким образом главные компоненты, на которые мы бы хотели спроецировать наши данные, являются просто собственными векторами соответствующих топ-k штук собственных значений этой матрицы.

Далее надо умножить нашу матрицу данных на эти компоненты и мы получим проекцию наших данных в ортогональном базисе этих компонент. Теперь если мы транспонируем нашу матрицу данных и матрицу векторов главных компонент, мы восстановим исходную выборку в том пространстве, из которого мы делали проекцию на компоненты. Если количество компонент было меньше размерности исходного пространства, мы потеряем часть информации при таком преобразовании.

Для примера снижения размерности с помощью PCA и последующей визуализации возьмем набор данных MNIST.



Картинки здесь представляются матрицей 8 x 8 (интенсивности белого цвета для каждого пикселя). Далее эта матрица "разворачивается"в вектор длины 64, получается признаковое описание объекта. Снизим с помощью PCA размерность простанства до 2 и визуализируем.



1.1.2. SNE

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. SNE начинается с преобразования многомерной евклидовой дистанции между точками в условные вероятности, отражающие сходство точек. Математически это выглядит следующим образом:

$$p_{j|i} = \frac{exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{1.3}$$

Где $H(P_i)$ – энтропия Шеннона в битах:

$$H(P_i) = -\sum_j p_{j|i} log_2 p_{j|i} \tag{1.4}$$

Эта формула показывает, насколько точка X_j близка к точке X_i при гауссовом распределении вокруг X_i с заданным отклонением σ. Сигма будет различной для каждой точки. Она выбирается так, чтобы точки в областях с большей плотностью имели меньшую дисперсию. Для этого используется оценка перплексии:

$$Perp(P_i) = 2^{H(P_i)} \tag{1.5}$$

В данном случае перплексия может быть интерпретирована как сглаженная оценка эффективного количества «соседей» для точки X_i. Она задается в качестве параметра метода. Авторы рекомендуют использовать значение в интервале от 5 до 50. Сигма определяется для каждой пары X_i и X_j при помощи алгоритма бинарного поиска.

Для двумерных или трехмерных «коллег» пары X_i и X_j, назовем их для ясности Y_i и Y_j, не представляет труда оценить условную вероятность, используя формулу для $p_{i|j}$. Стандартное отклонение предлагается установить в $1/\sqrt{2}$:

$$q_{j|i} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} exp(-\|y_i - y_k\|^2} \tag{1.6}$$

Если точки отображения Y_i и Y_j корректно моделируют сходство между исходными точками высокой размерности X_i и X_j, то соответствующие условные вероятности $p_{j|i}$ и $q_{j|i}$ будут эквивалентны. В качестве очевидной оценки качества, с которым $q_{j|i}$ отражает $p_{j|i}$, используется дивергенция или расстояние Кульбака-Лейблера. SNE минимизирует сумму таких расстояний для всех точек отображения при помощи градиентного спуска. Функция потерь для данного метода будет определяться формулой:

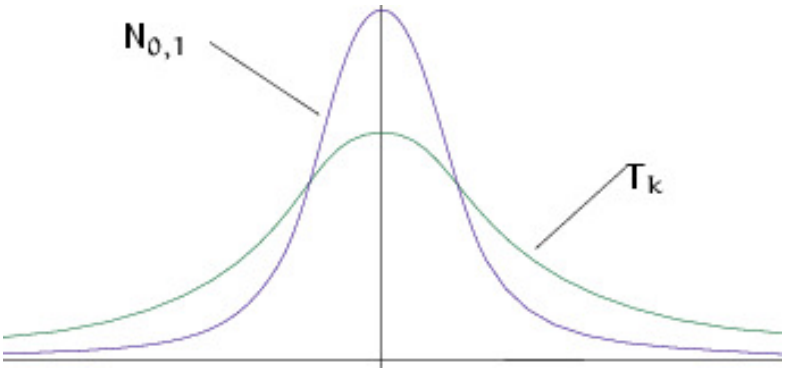
$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \tag{1.7}$$

Соответственно, градиент:

$$\frac{\partial Cost}{\partial y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j) \tag{1.8}$$

1.1.3. tSNE

Использование классического SNE позволяет получить неплохие результаты, но может быть связано с трудностями в оптимизации функции потерь и проблемой скученности (в оригинале – crowding problem). tSNE если и не решает эти проблемы совсем, то существенно облегчает. Функция потерь tSNE имеет два принципиальных отличая. Во-первых, у tSNE симметричная форма сходства в многомерном пространстве и более простой вариант градиента. Во-вторых, вместо гауссова распределения для точек из пространства отображения используется t-распределение (Стьюдента), «тяжелые» хвосты которого облегчают оптимизацию и решают проблему скученности.



В качестве альтернативы минимизации суммы дивергенций Кульбака-Лейблера между условными вероятностями $p_{i|j}$ и $qi_|j$ предлагается минимизировать одиночную дивергенцию между совместной вероятностью P в многомерном пространстве и совместной вероятностью Q в пространстве отображения:

$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{1.9}$$

где p_{ii} и $q_{ii} = 0$, $p_{ij} = p_{ji}$, $q_{ij} = q_{ji}$ для любых i и j, а p_{ij} определяется по формуле:

$$p_{ij} = \frac{p_{ji} + p_{ij}}{2n} \tag{1.10}$$

где n — количество точек в наборе данных.

Проблема скученности заключается в том, что расстояние между двумя точками в пространстве отображения, соответствующими двум среднеудаленным точкам в многомерном пространстве, должно быть существенно больше, нежели расстояние, которое позволяет получить гауссово распределение.

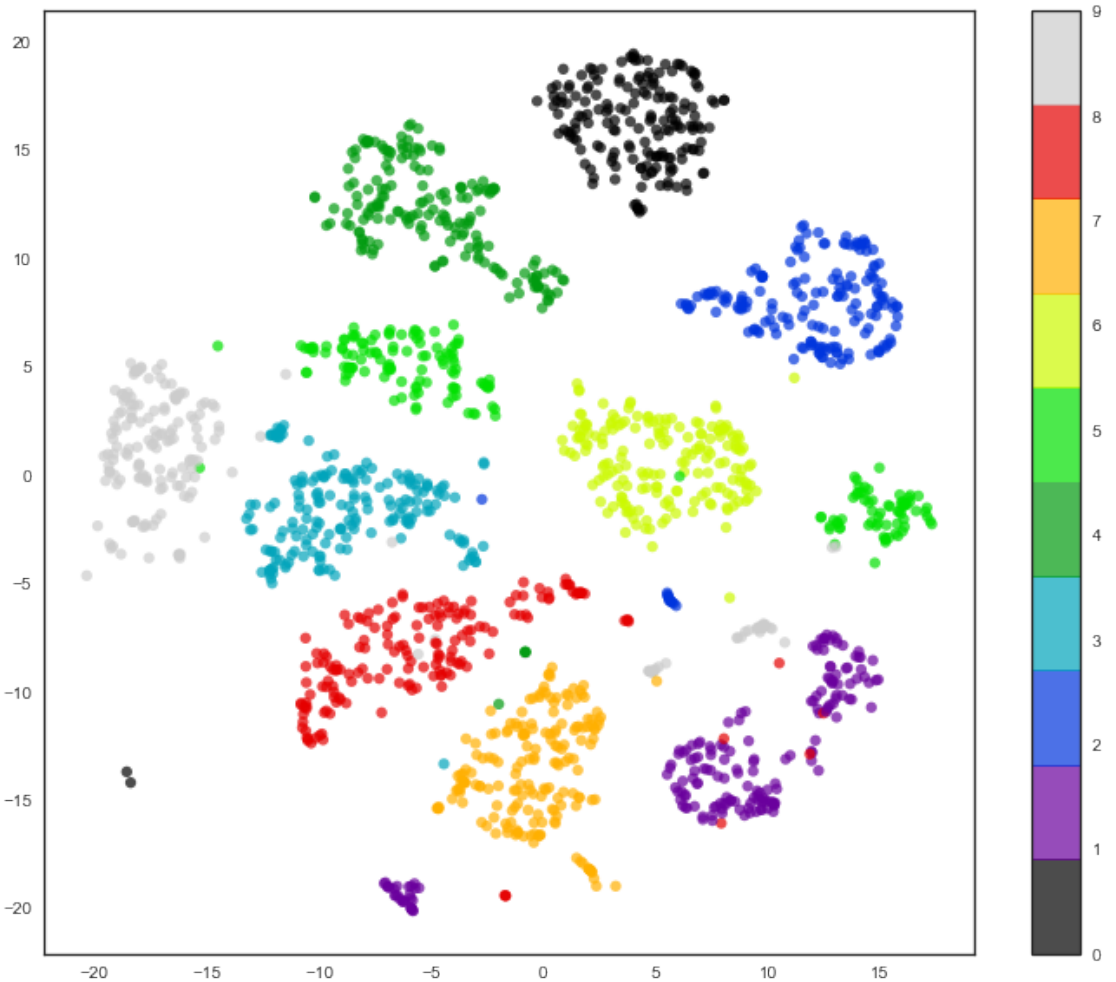
Проблему решают хвосты Стьюдента. В tSNE используется t-распределение с одной степенью свободы. Совместная вероятность для пространства отображения в этом случае будет определяться формулой:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}} \tag{1.11}$$

И соответствующий градиент:

$$\frac{\partial Cost}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \tag{1.12}$$

Для показательного примера возьмем тот же набор данных MNIST и снизим размерность с помощью tSNE:

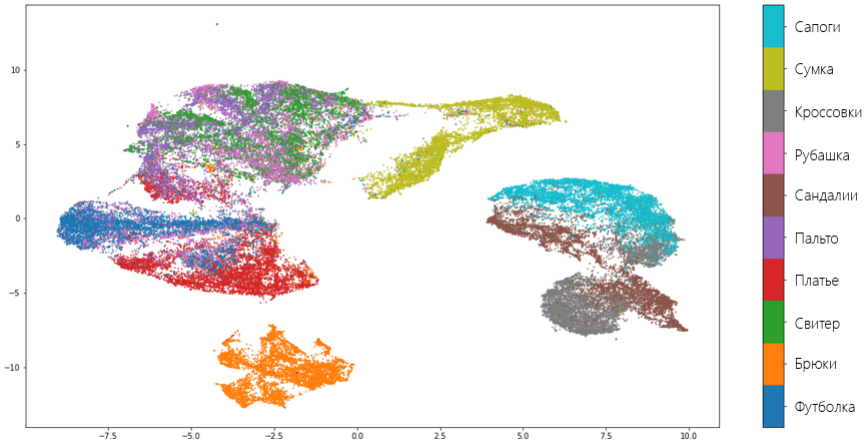


Невооруженным взглядом видно, что, по сравнению с кластерами PCA, кластеры tSNE намного более различимы. Это заслуга нейлинейности tSNE.

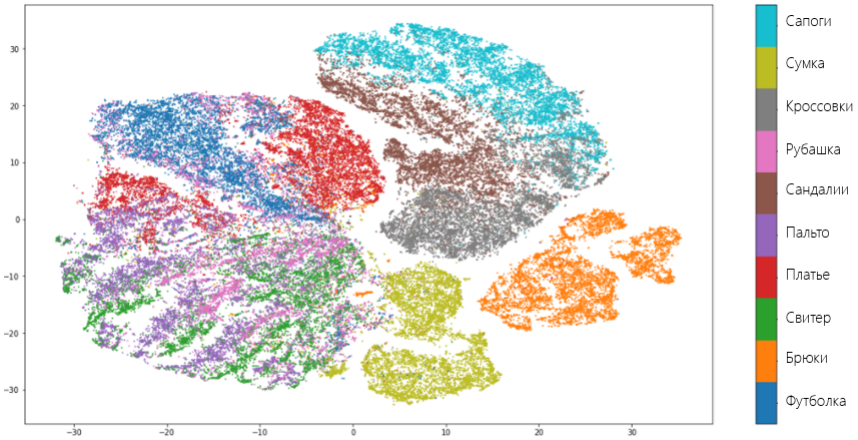
1.1.4. UMAP

Поскольку для глубокого понимания принципов работы алгоритма требуется знание римановой геометрии и топологии, мы опишем лишь общую идею модели UMAP, которую можно разделить на два основных шага. 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 размерности.

Посмотрим на результат работы UMAP на наборе данных FASHION MNIST и сравним его с результатом работы tSNE:



UMAP



tSNE

tSNE показывает схожие с UMAP результаты и допускает те же ошибки. Однако, в отличии от UMAP, tSNE не так очевидно объединяет виды

одежды в отдельные группы: брюки, вещи для туловища и для ног находятся близко друг к другу. Однако в целом можно сказать, что оба алгоритма одинаково хорошо справились с задачей и исследователь волен на свой вкус делать выбор в пользу одного или другого. Можно было, если бы не одно «но». Это «но» заключается в скорости обучения. UMAP значительно более вычислительно эффективен, что дает ему огромное преимущество перед другими алгоритмами, в том числе и перед tSNE.

2. gSNE

2.1. Теоретическая часть

Одним из главных различий между SNE и $tSNE$ было применение вероятностного распределения с более "тяжелыми"хвостами в пространстве низкой размернсости. Мы решили пойти еще дальше, посчитав, что распределения Стьюдента, не хватает для решения проблемы скученности. В пространстве низкой размерности мы будем применять обобщенное нормальное распределение, плотность которого равна:

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)}\, e^{-(|x-\mu|/\alpha)^{\beta}} \tag{2.1}$$

где Γ - гамма-функция, α - мат. ожидание, μ - дисперсия. Возьмем $\alpha = 0$, а $\mu = 1$.

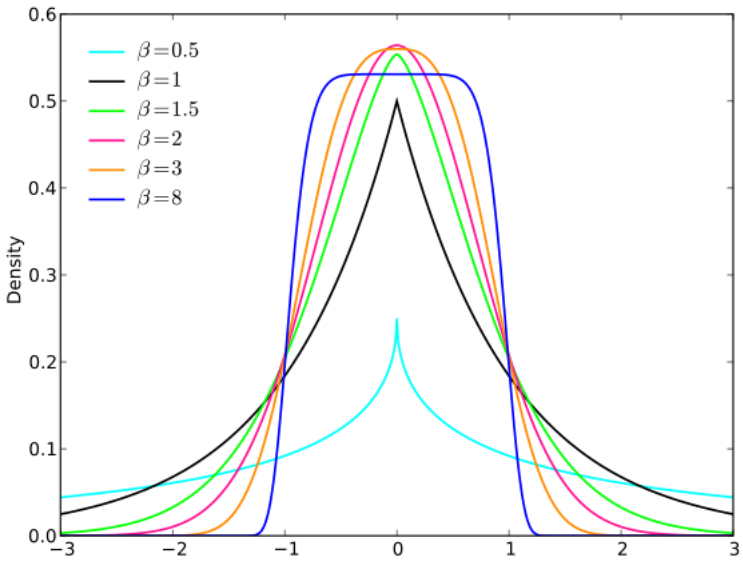
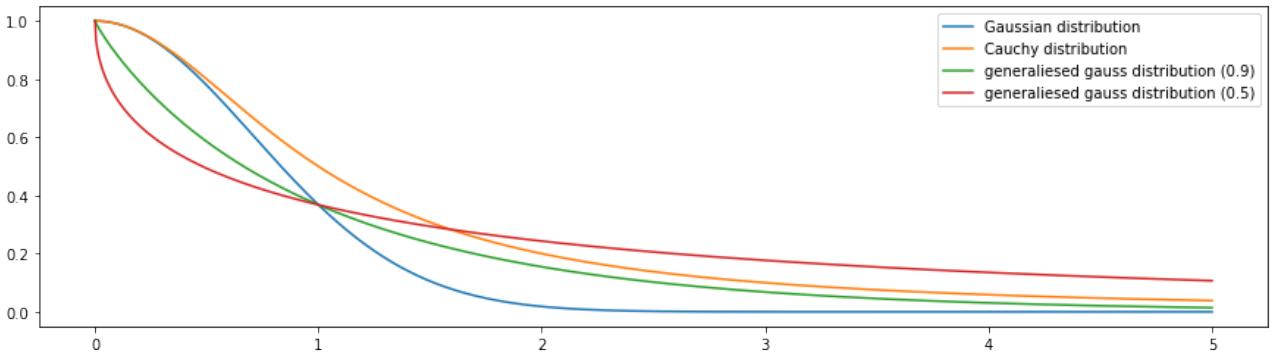


График плотности в зависимости от β

Нужно понять, какой β будет оптимальным в нашей задаче. Для начала поймем, в каком интервале нужно искать, чтобы добиться более «тяжелых» хвостов: .

На графике мы можем видеть, что хвост обобщенного нормального распределения с параметром $\beta = 0.9$ уже "легче чем хвост распределения Коши



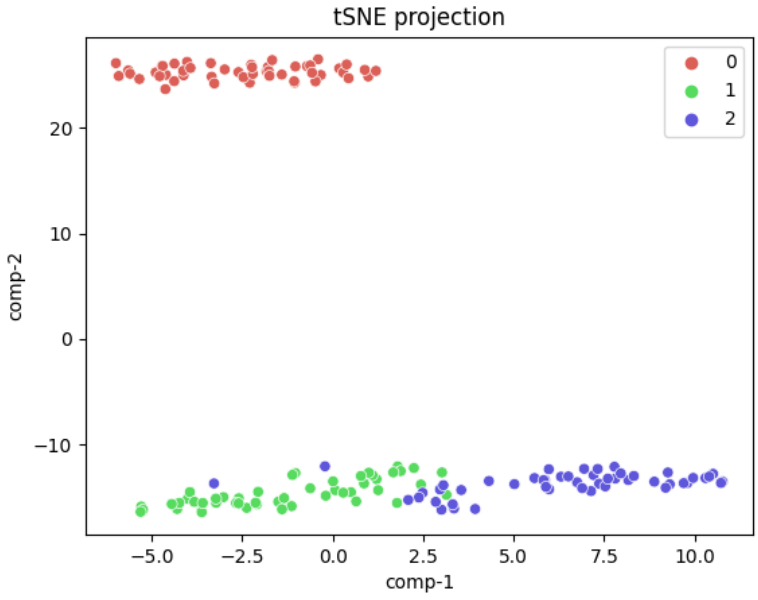
Хвосты разных распределений β

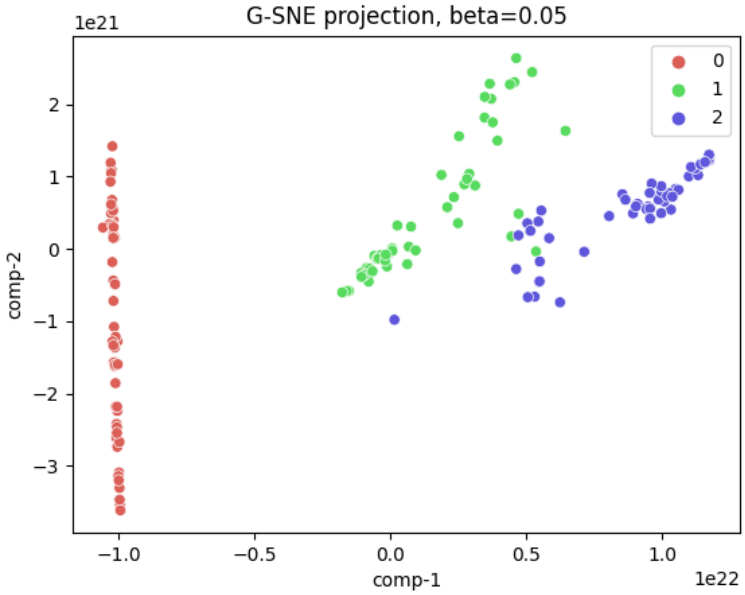
(Стьюдента со степенью свободы 1). Поэтому отрезок, в которым мы будем искать оптимальный β, будет находиться между 0 и 1.

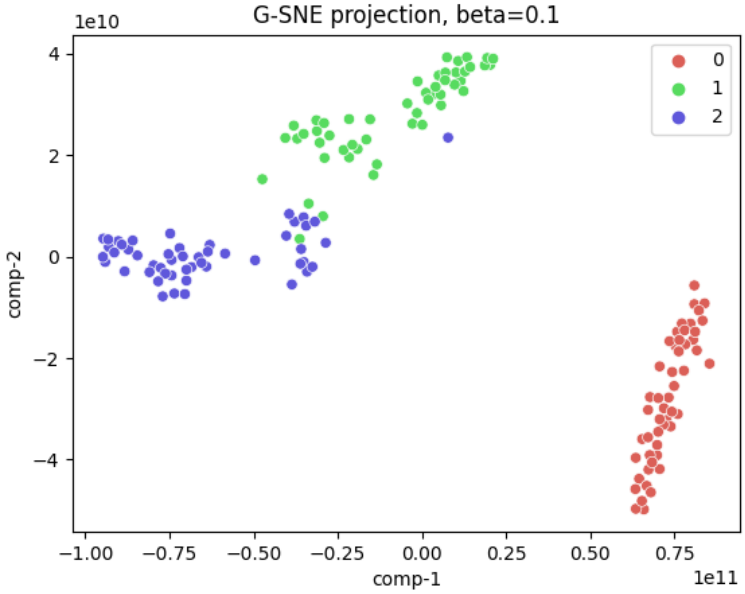
2.2. Практическая часть

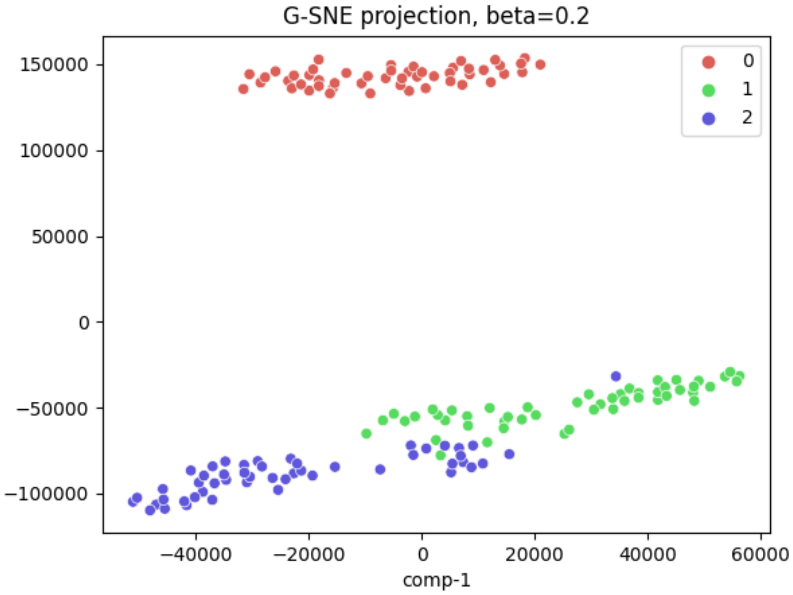
Выберем несколько датасетов, на которых мы будем тестировать gSNE с разными β. Предлагается взять Iris, MNIST, Fashion MNIST и снизить размерность прзнаковых пространств до 2. Мы визуализируем результаты и используем расстояние Кульбака-Лейблера в качестве метрики качества. Также мы визуализируем результаты tSNE на этих датасетах.

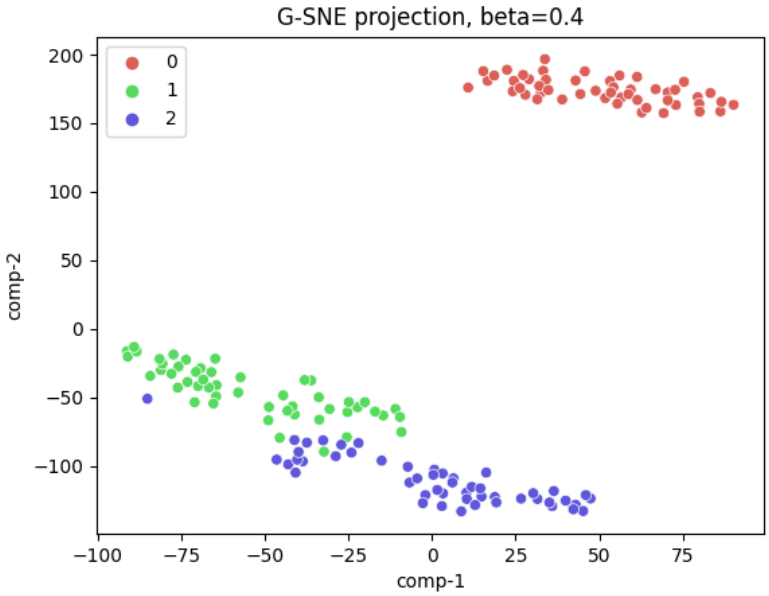
2.2.1. Iris dataset

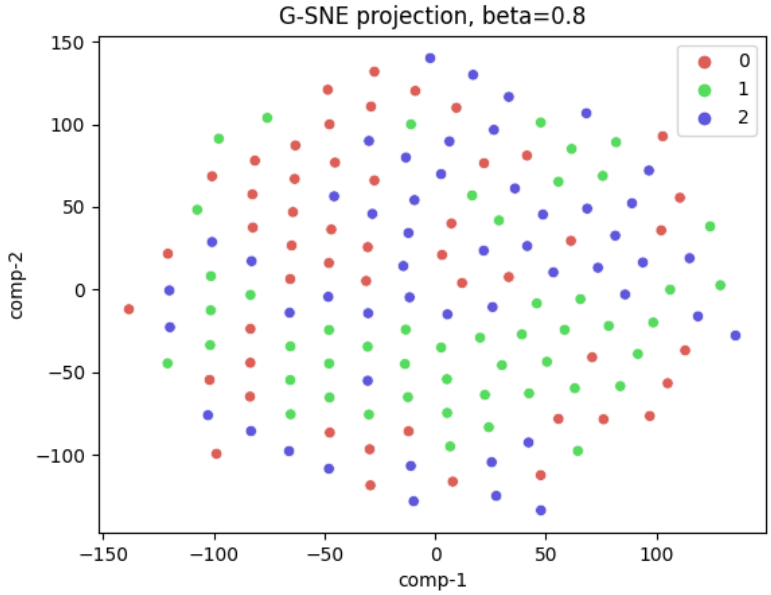


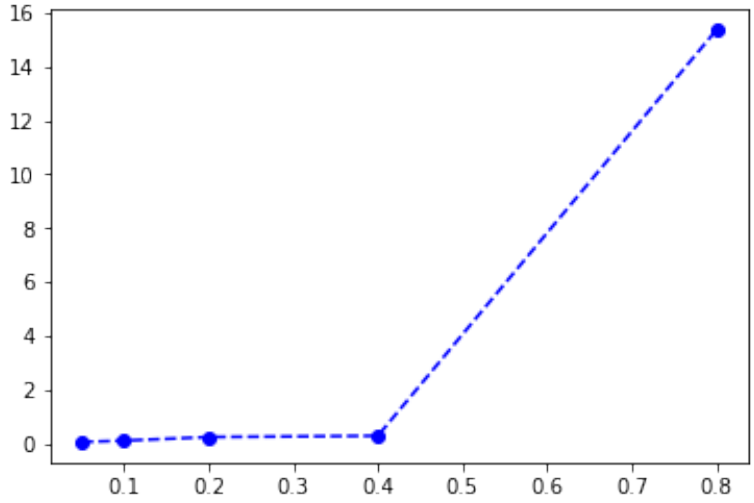

G-SNE projection, beta=0.05
1e21
2
1
0
−1
−2
−3
comp-2
−1.0
−0.5
0.0
0.5
1.0
comp-1
1e22
0
1
2


G-SNE projection, beta=0.1
1e10
4
2
0
−2
−4
comp-2
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
comp-1
1e11
0
1
2


G-SNE projection, beta=0.2
150000
100000
50000
0
−50000
−100000
−40000
−20000
0
20000
40000
60000
comp-1
0
1
2

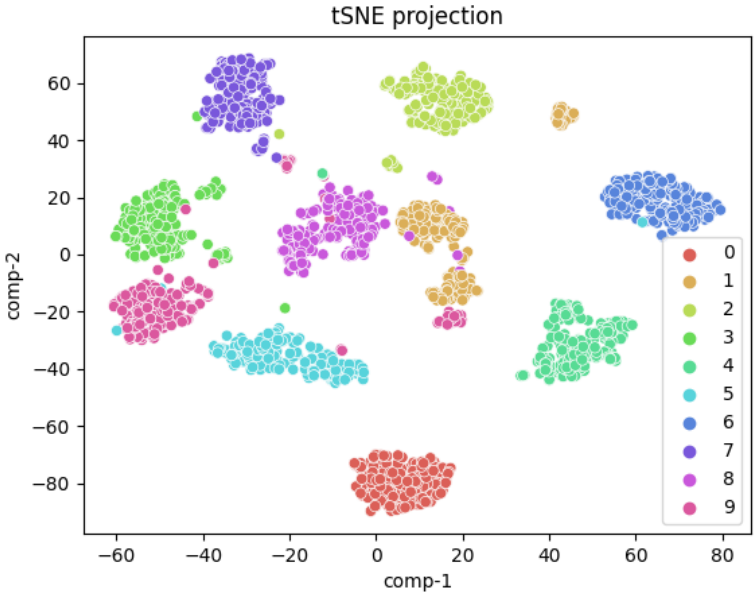

G-SNE projection, beta=0.4
200
150
100
50
0
−50
−100
−100
−75
−50
−25
0
25
50
75
comp-1
comp-2
0
1
2

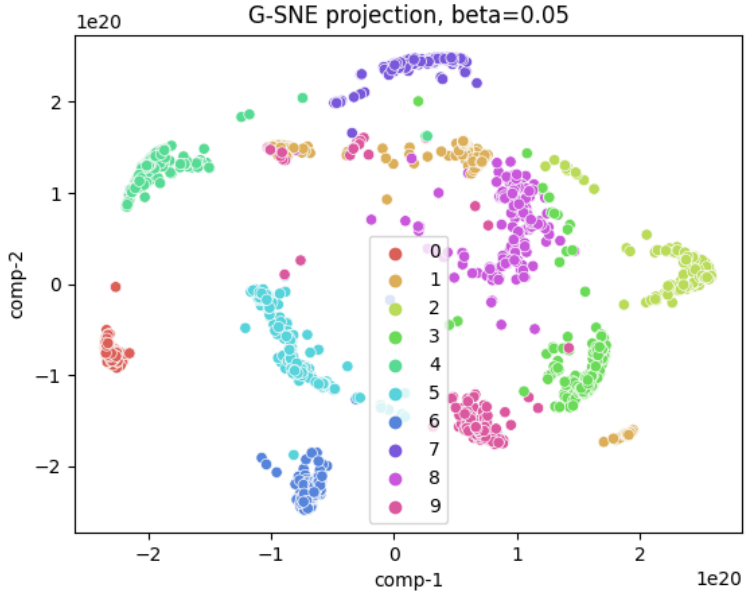


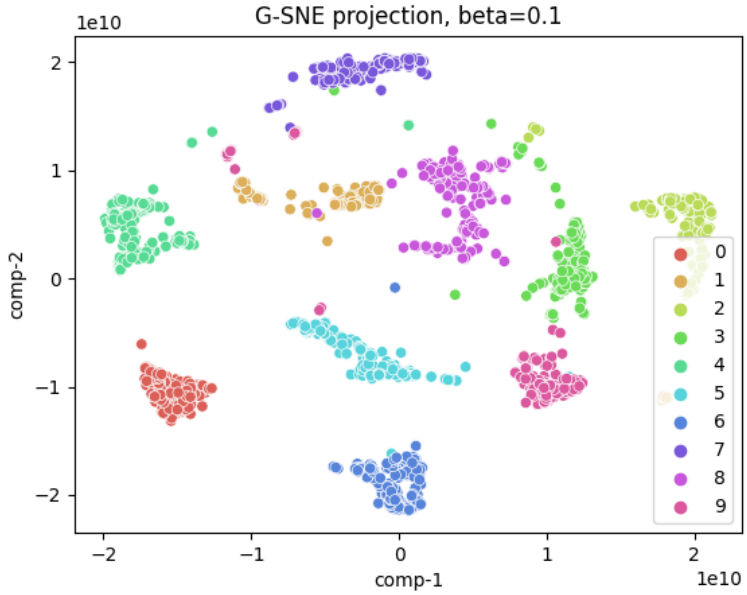


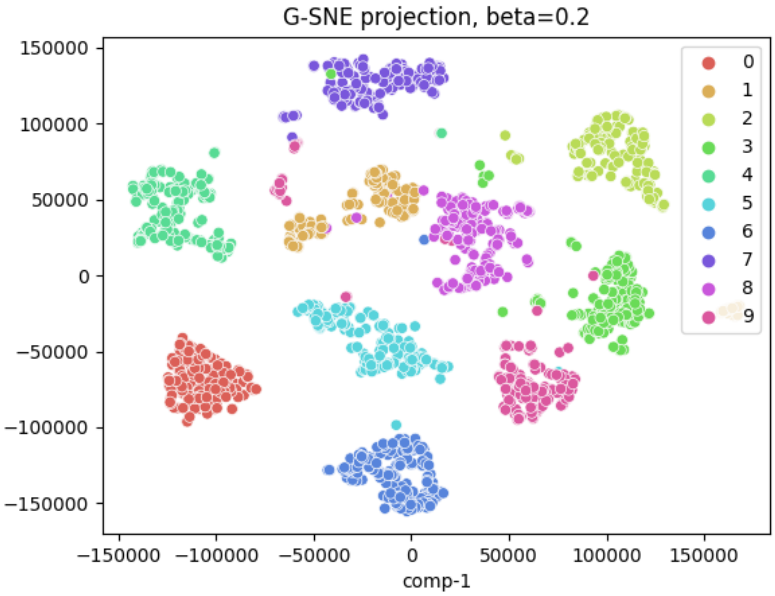
Зависимость расстояния КЛ от β

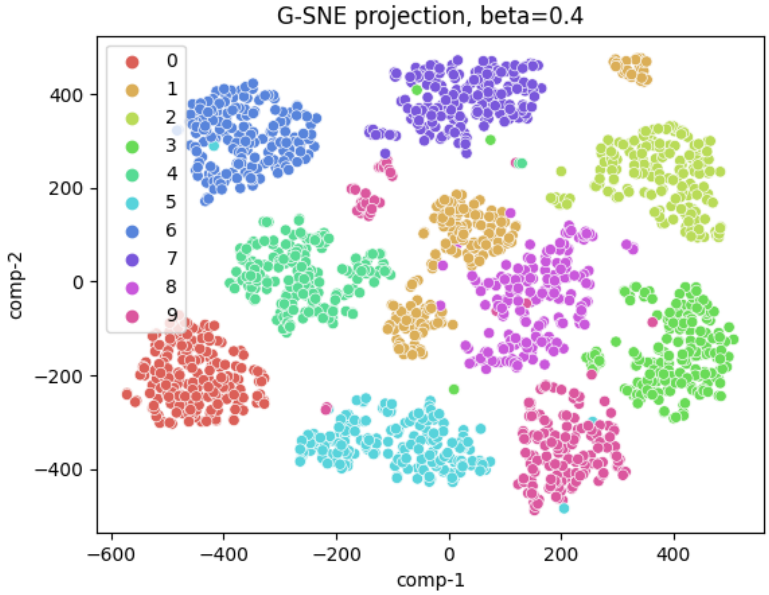
2.2.2. MNIST dataset

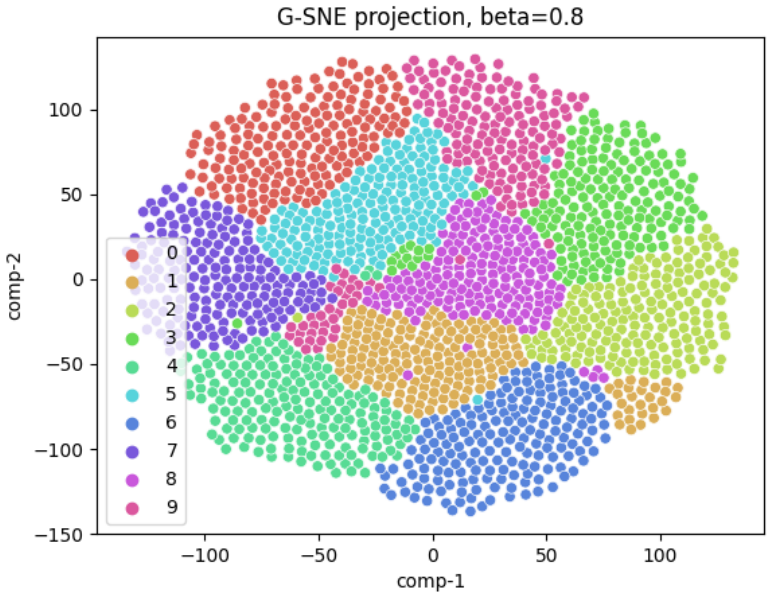


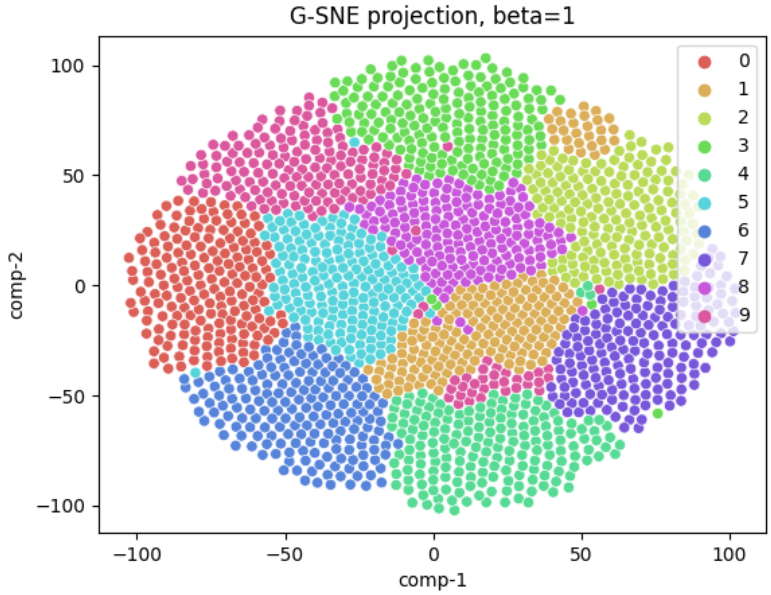


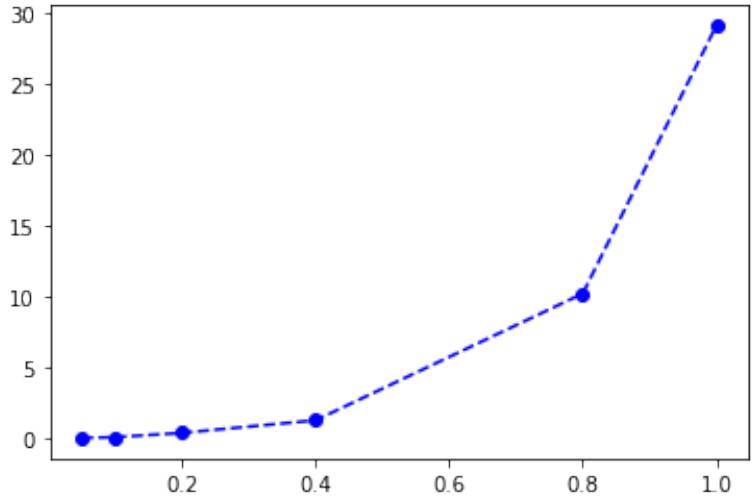

G-SNE projection, beta=0.1
1e10
2
1
0
−1
−2
comp-2
−2
−1
0
1
2
1e10
comp-1
0
1
2
3
4
5
6
7
8
9


G-SNE projection, beta=0.2
150000
100000
50000
0
−50000
−100000
−150000
−150000
−100000
−50000
0
50000
100000
150000
comp-1
0
1
2
3
4
5
6
7
8
9


G-SNE projection, beta=0.4
0
1
2
3
4
5
6
7
8
9
400
200
0
−200
−400
comp-2
−600
−400
−200
0
200
400
comp-1

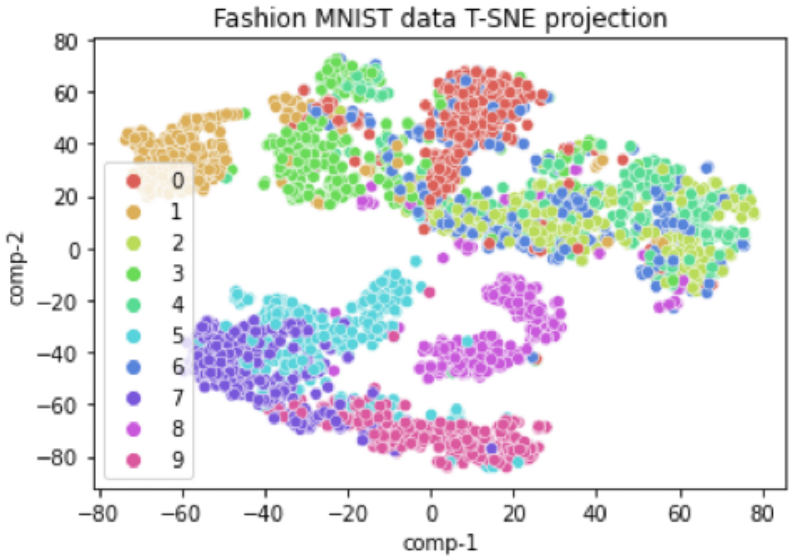

G-SNE projection, beta=0.8
0
1
2
3
4
5
6
7
8
9
100
50
0
−50
−100
−150
comp-2
−100
−50
0
50
100
comp-1

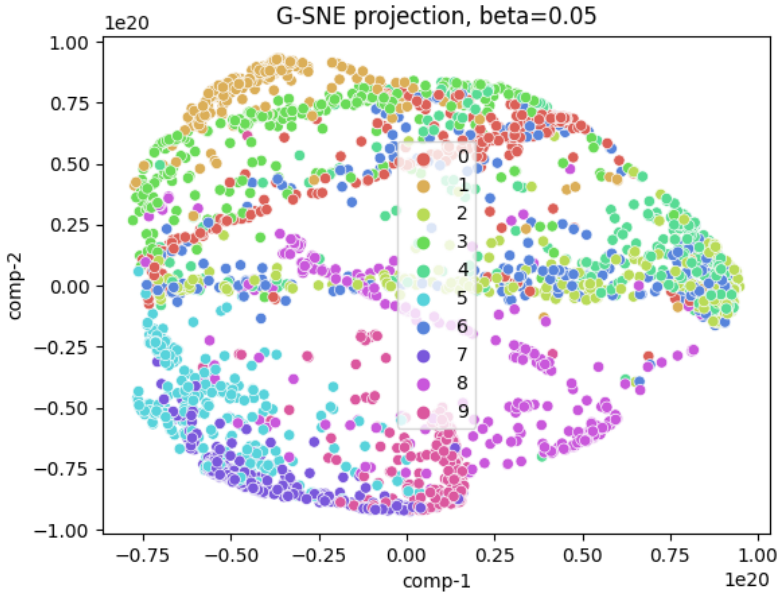


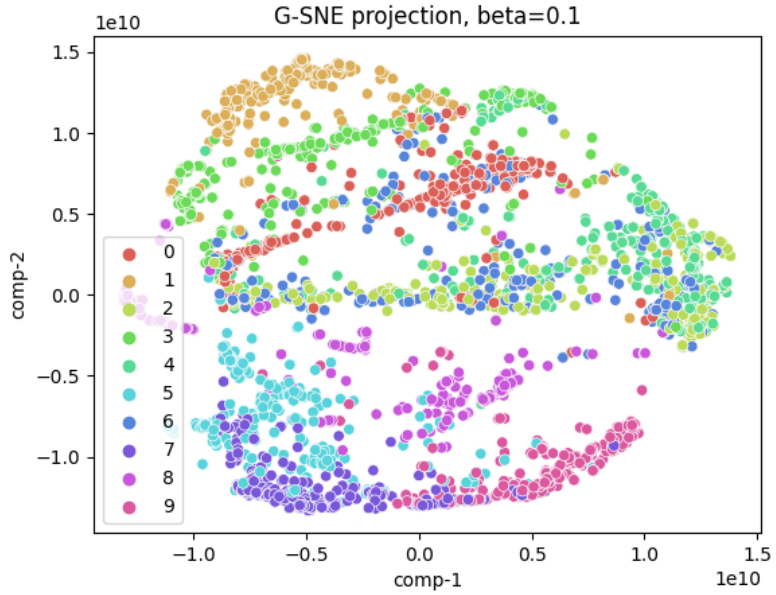


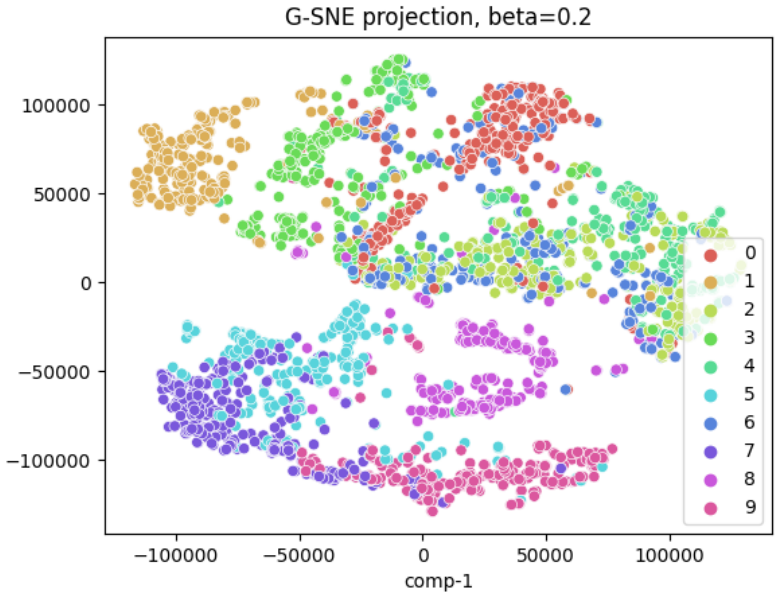
Зависимость расстояния КЛ от β

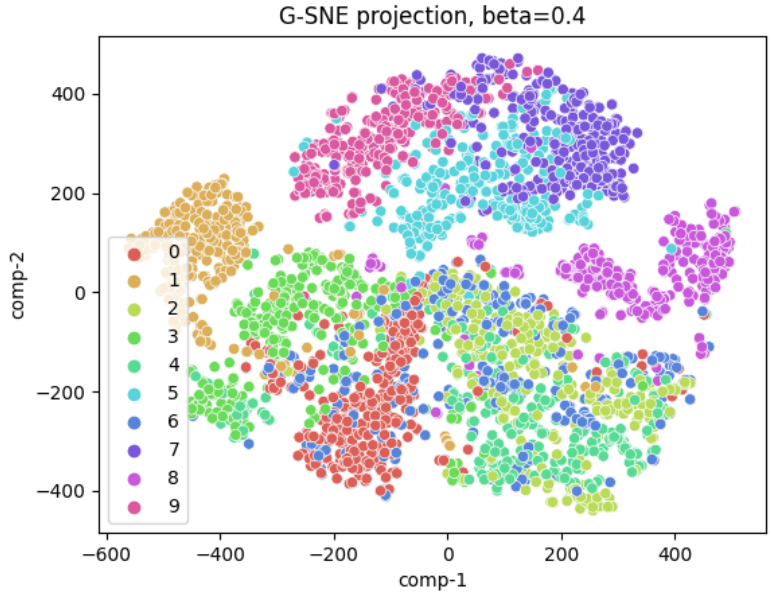
2.2.3. Fashion MNIST dataset

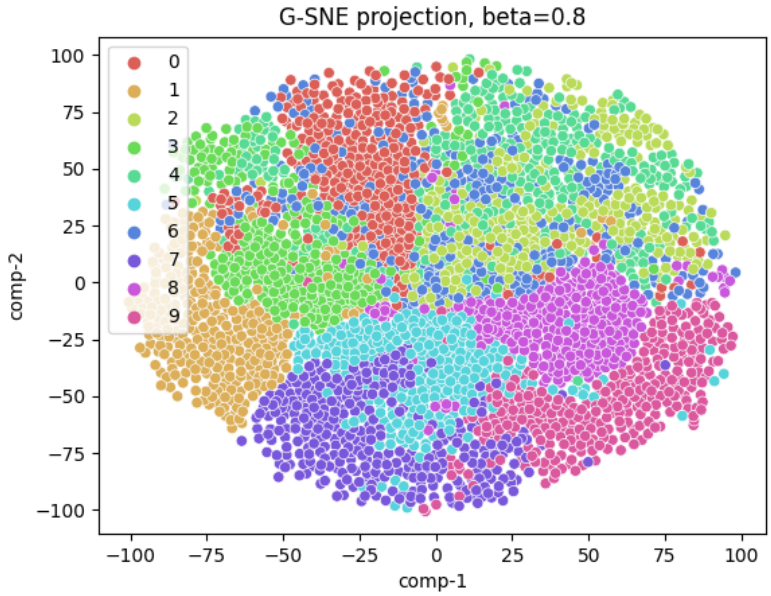


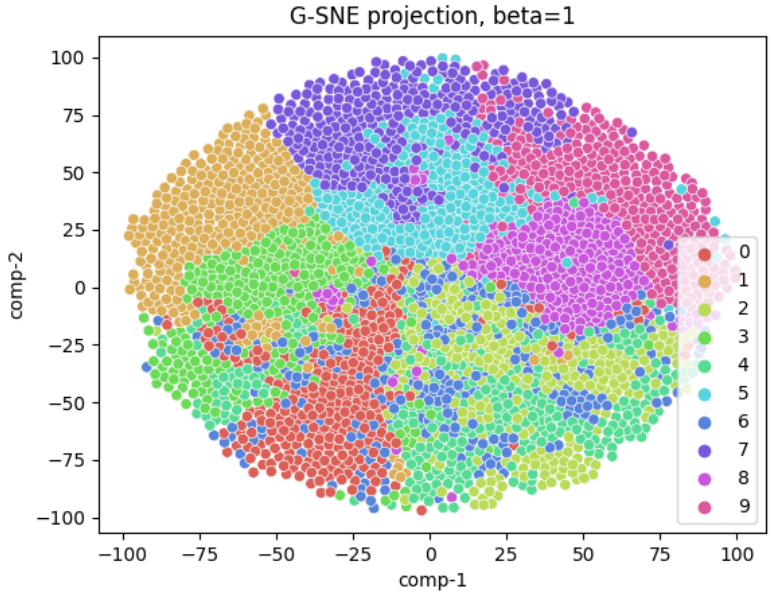


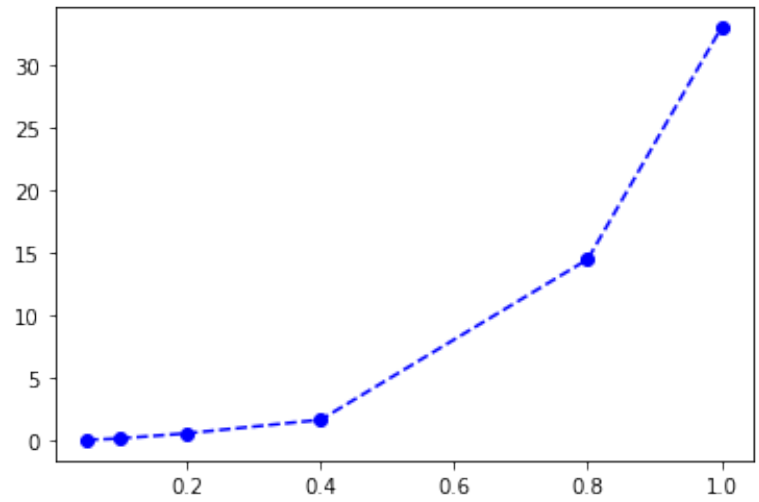

G-SNE projection, beta=0.1
1e10
1.5
1.0
0.5
0.0
−0.5
−1.0
comp-2
0
1
2
3
4
5
6
7
8
9
−1.0
−0.5
0.0
0.5
1.0
1.5
comp-1
1e10


G-SNE projection, beta=0.2
100000
50000
0
−50000
−100000
0
1
2
3
4
5
6
7
8
9
−100000
−50000
0
50000
100000
comp-1


G-SNE projection, beta=0.4
comp-2
comp-1
400
200
0
−200
−400
−600
−400
−200
0
200
400
0
1
2
3
4
5
6
7
8
9


G-SNE projection, beta=0.8
comp-2
comp-1
100
75
50
25
0
−25
−50
−75
−100
−100
−75
−50
−25
0
25
50
75
100
0
1
2
3
4
5
6
7
8
9





Зависимость расстояния КЛ от β

2.3. Выводы

Во всех экспериментах мы видим тенденцию к ухудшению качества визуализации с увеличением параметра β. Это доказывает как графики зависимостей расстояния Кульбака-Лейблера от β, так и тот факт, что с увеличением β кластеры становятся менее собранными и более близкими.

Мы также видим, что структура проекции данных с помощью gSNE отличается от tSNE. Это указывает на тот факт, что наша гипотеза о том, что распределения Коши не достаточно для решения проблемы скученности, верна. gSNE сохраняет глобальную структуру данных лучше, чем tSNE.

3. ЗАКЛЮЧЕНИЕ

В работе были проанализированы существующие методы понижения размерности данных и представлен новый алгоритм, который явялется обобщением tSNE. Благодаря использованию обобщенного нормального распределния и подбору оптимального β, мы решили проблему скученности, которая была вызвана «легкими» хвостами распределения Коши, используемого в tSNE.

Работа заняла 1 место на международной научной студенческой конференции 2022 в секции «Машинное обучение и нейронные сети».

Дальнейшие планы заключаются в добавлении параметра β в набор обучаемых (с помощью градиентного спуска) параметров и выбор походящего вероятностного распределения для пространства высокой размерности. Изменение задачи оптимизации за счет добавления в нее параметра β сделает gSNE гибким для работы с любыми датасетами: в каждом конкретном случае будет выбираться свой конкретный β. Выбор нового вероятностного распределения для пространства высокой размерности, возможно, еще сильнее увеличит внимание к глобальной структуре данных, сохранив внимание в локальной структуре. Также планируется более подробно изучить теоретическое обоснование UMAP, чтобы иметь возможность переиспользовать методы и приемы авторов этого алгоритма.

Список литературы

1. Laurens van der Maaten, Geoffrey Hinton. Visualizing Data using t-SNE. URL: https://www.jmlr.org/papers/volume9/vandermaaten08a/
2. Geoffrey Hinton and Sam Roweis. Stochastic Neighbor Embedding. URL: https://www.cs.toronto.edu/ hinton/absps/sne.pdf
3. Leland McInnes, John Healy, James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. URL: https://arxiv.org/abs/1802.03426
4. Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. URL: https://doi.org/10.1098/rsta.2015.0202
5. Minh-Hien Tran. Comparing UMAP vs t-SNE in Single-cell RNA-Seq Data Visualization, Simply Explained. URL: https://blog.bioturing.com/2022/01/14/umap-vs-t-sne-single-cell-rna-seq-data-visualization/
6. Mathias Gruber. Why you should not rely on t-SNE, UMAP or TriMAP. URL: https://towardsdatascience.com/why-you-should-not-rely-on-t-sne-umap-or-trimap-f8f5dc333e59
7. Yingfan Wang, Haiyang Huang, Cynthia Rudin, Yaron Shaposhnik. Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization. Journal of Machine Learning Research 22 (2021) 1-73
8. Sivakar Sivarajah. Dimensionality Reduction for Data Visualization: PCA vs TSNE vs UMAP vs LDA. URL: https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29
9. Simon Andrews. DDimension Reduction PCA, tSNE, UMAP. URL: https://www.bioinformatics.babraham.ac.uk/training/10XRNASeq/Dimension