Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра _теории функций и стохастического анализа_

Исследование методов понижения размерности: алгоритмы и их применение

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ (МАГИСТЕРСКОЙ, ДИПЛОМНОЙ) РАБОТЫ

Студента__2__ курса __248___группы
направления (специальности)_____09.04.03 – Прикладная информатика_____
код и наименование направления (специальности)
_____механико-математического факультета_____
наименование факультета, института, колледжа
_____Айрапетяна Оганеса Владимировича_____
фамилия, имя, отчество

Научный руководитель
доцент, к.ф.-м.н.
Д.В. Мельничук_____ _____ _____
должность, уч. степень, уч. звание дата, подпись
инициалы, фамилия

Заведующий кафедрой
д. ф.-м.н., доцент
С.П. Сидоров_____ _____ _____
должность, уч. степень, уч. звание дата, подпись
инициалы, фамилия

Саратов_2025 год

Введение. По мере совершенствования технологий объем информации о мире, которая накапливается и, со временем увеличивается до больших размеров, что во многих приложениях в области компьютерного зрения, интеллектуального анализа данных и распознавания образов данные характеризуются десятками или сотнями тысяч переменных или функций. Сенсорные сети собирают информацию о дорожном движении или погоде в режиме реального времени, документы и новостные статьи распространяются и ищутся в режиме онлайн, информация в медицинских записях собирается и хранится в электронной форме. Всю эту информацию можно извлечь, чтобы лучше понять отношения между компонентами базовых систем и построить их модели. Однако, высокая размерность существенно увеличивает требования ко времени и пространству для обработки данных. Более того, некоторые функции неактуальны и избыточны. Существование этих функций может привести к низкой эффективности, переобучению и плохой эффективности прогнозирования в задачах обучения. Следовательно, снижение размерности стало важным этапом предварительной обработки данных в таких приложениях. Уменьшение размерности может быть достигнуто либо путем выбора функций, либо путем преобразования в пространство низкой размерности. Выбор признаков, также известный как выбор переменных, представляет собой проблему выбора подмножества исходных признаков. В отличие от методов, основанных на преобразовании, которые позволяют модифицировать входные объекты в новое пространство объектов; при выборе признаков исходное представление переменных не меняется. Выбор признаков обычно предпочтительнее преобразования, если нужно сохранить исходное значение признаков и определить, какие из этих признаков важны. Так же, алгоритмы сокращения размерности можно поделить на две категории: те, которые стремятся сохранить структуру парных расстояний среди всех выборок данных, и те, которые способствуют сохранению локальных расстояний на глобальном расстоянии.

Цель ВКР:

— изучение методов отбора признаков для понижения размерности и обзор их современного состояния;

— проведение эксперимента, используя стандартные datasets и анализ ре-

зультатов;

— разработка и тестирование алгоритма предложенного автором данной работы Algoritm_X.

Объект исследования – задача понижения размерности и способы ее решений.

Предмет исследования – разрабатываемый алгоритм Algoritm_X для решения задач понижения размерности.

Для достижения поставленных целей в работе необходимо решить следующие **задачи:**

— сформулировать задачу понижения размерности;

— провести обзор соответствующий литературы;

— исследовать существующие решения для задачи понижения размерности;

— рассмотреть такие алгоритмы как: PCA; LDA; ICA; t-SNE и UMAP;

— найти различные datasets для тестирования Algoritm_X;

— применить алгоритмы понижения размерности на datasets;

— описать стратегию Algoritm_X;

— описать Algoritm_X математическим языком;

— реализация Algoritm_X;

— тестирование Algoritm_X;

— применения Algoritm_X на datasets и сравнение результатов.

Положения, выносимые на защиту – разработанный алгоритм работает в соответствии с требованиями.

Структура и содержание ВКР. Работа состоит из введения, трех разделов, заключения, списка используемых источников, и двух приложений. Общий объем работы составляет более 60 страниц.

Основное содержание работы:

Во **введение** обосновывается актуальность темы работы, формируется цели и задачи, отмечается практическая значимость полученных результатов.

В **первом** разделе формулируется задача понижения размерности, проводится обзор соответствующий литературы, исследуются существующие решения для задачи понижения размерности, в частности:

— методы отбора признаков;

— метод фильтраций;

— метод-оболочки;

— встроенные методы.

Во **втором** разделе рассмотрены пять алгоритмов для решения задач понижения размерности:

— PCA;

— LDA;

— ICA;

— t-SNE;

— UMAP.

Вводятся следующие обозначения, для математического описания PCA:

— $E = A^T A$ – оценка ковариационной матрицы объектов, где A – матрица «объект-признак»;

— y – вектор главных компонент;

— W – весовая матрица (матрица преобразования вращения). Она является ортогональной;

— q – интегральный индикатор.

— $Z = WA$ – матрица, состоящая из столбцов $(z_1, \ldots, z_p)$. Для нахождения первой главной компоненты необходимо найти такие линейные комбинации строк матрицы A, что векторы-столбцы матрицы Z обладали бы наибольшей дисперсией;

— σ^2 – остаточная дисперсия.

Приводится математическое описание алгоритма LDA.

Пусть $x_i \in R^d, i = 1, 2, \ldots$ обозначаем наблюдаемые данные, которые принадлежат ровно одному из доступных K-классов, $\omega_1, \omega_2, \ldots, \omega_K$, и пусть $P(\omega_i)$, $i = 1, 2, \ldots, K$ обозначают априорную вероятность i-го класса ω_i. Пусть

$m_i, i = 1, 2, \ldots, K$ обозначают средний вектор для класса ω_i, т.е.

$$m_i = E(x|x \in \omega_i),$$

и пусть Σ_i обозначает ковариационную матрицу i-го класса, т.е.

$$\Sigma_i = E[(x - m_i)(x - m_i)^t | x \in \omega_i], i = 1, 2, \ldots, K.$$

Для достижения максимальной разделимости классов, помимо уменьшения размерности, определяются следующие три матрицы:

— матрица рассеяния внутри класса Σ_W:

$$\Sigma_W = \sum_{i=1}^{K} P(\omega_i) E[(x - m_i)(x - m_i)^t | x \in \omega_i] = \sum_{i=1}^{K} P(\omega_i)\Sigma_i;$$

— матрица разброса между классами Σ_B:

$$\Sigma_B = \sum_{i=1}^{K} P(\omega_i)(m - m_i)(m - m_i)^t.$$

Приводится пример использования алгоритма t-SNE на языке Python.

```
from sklearn.manifold import TSNE
```

Для того, что бы использовать алгоритм t-SNE нужно вызвать функцию «TSNE», указав новую размерность (обычно 2 или 3), после использовать метод «fit_transform» на вход которого подается dataset.

```
t_sne = TSNE(n_components=n)
result = t_sne.fit_transform(dataset)
```

Описывается наиболее современный алгоритм для понижения размерности UMAP. Сначала UMAP извлекает высокоразмерные сходства между точками данных. Для этого вычисляется граф k ближайших соседей. Пусть $i_1, \ldots, i_k$ обозначают индексы k ближайших соседей x_i в порядке увеличения расстояния до x_i. Затем, используя свое предположение о однородности, UMAP подбирает локальное понятие сходства для каждой точки данных i, выбирая масштаб σ_i таким образом, чтобы общее сходство каждой точки с ее k ближайшими соседями было нормализовано, т.е. находит σ_i таким образом, что

$$\sum_{k=1}^{K} \exp(-\frac{d(x_i,x_{i_k})-d(x_i,x_{i_1})}{\sigma_i}) = \log_2(K).$$

Это определяет направленные высокоразмерные сходства

$$\mu_{i\longrightarrow j} = \begin{cases} \exp(-\frac{d(x_i,x_j)-d(x_i,x_{i_1})}{\sigma_i}) & where \quad j \in (i_1,\ldots,i_k) \\ 0 & other \end{cases}.$$

Наконец, они симметризуются для получения ненаправленных высокоразмерных сходств или входных сходств между элементами i и j

$$\mu_{ij} = \mu_{i\longrightarrow j} + \mu_{j\longrightarrow i} - \mu_{i\longrightarrow j}\mu_{j\longrightarrow i} \in [0,1].$$

Хотя каждый узел имеет ровно k ненулевых направленных сходств $\mu_{i\longrightarrow j}$ с другими узлами, которые в сумме дают $\log_2(k)$, это не выполняется точно после симметризации. Тем не менее, обычно u_{ij} сильно разрежены, каждый узел имеет положительное сходство примерно с k другими узлами, а степень каждого узла $d_i = \sum_{j=1}^{n} u_{ij}$ приблизительно постоянна и близка к $log_2(k)$. Для удобства записи целесообразно устанавливать $u_{ii} = 0$ и определять общее сходство как $u_{tot} = \frac{1}{2}\sum_{i=1}^{n} d_i$.

Расстояние в пространстве вложений преобразуется в маломерное сходство путем плавного приближения к многомерной функции сходства, $\phi(d;a,b) = (1+ad^{2b})-1$, для всех пар точек. Параметры формы a, b по сути являются гиперпараметрами UMAP. Перегрузим нотацию и запишем

$$v_{ij} = \Phi(\|e_i - e_j\|) = \Phi(e_i, e_j)$$

для маломерных или вложенных сходств и обычно подавим их зависимость от a и b.

Предположительно, UMAP приблизительно оптимизирует следующую целевую функцию относительно вложений $e_1,\ldots,e_n$:

$$\iota(e_i|\mu_{ij}) = -2\sum_{1\le i\le j\le n}(u_{ij}\log(v_{ij})) + (1-u_{ij})\log(1-v_{ij}) =$$

$$= -2\sum_{1\le i\le j\le n}(u_{ij}\log(\Phi(e_i,e_j))) + (1-u_{ij})\log(1-\Phi(e_i,e_j)).$$

Хотя многомерные сходства u_{ij} симметричны, реализация UMAP учитывает их направление во время оптимизации. Рассматривая через призму модели, направленной на силу, производная первого члена в каждом слагаемом функции потерь, $-\frac{\delta \iota_{ij}^a}{\delta e_i}$, отражает притяжение e_i к e_j из-за высокоразмерного сходства u_{ij}, а производная второго члена, $-\frac{\delta \iota_{ij}^a}{\delta e_i}$, представляет отталкивание, которое e_j оказывает на e_i из-за отсутствия сходства в высокоразмерном, $1 - u_{ij}$. В качестве альтернативы потери можно рассматривать как сумму потерь бинарной перекрестной энтропии для каждого попарного сходства. Таким образом, они минимизируются, если низкоразмерные сходства v_{ij} точно соответствуют своим высокоразмерным аналогам u_{ij}, то есть если UMAP удается воспроизвести входные сходства в пространстве встраивания.

Третий раздел посвящен главной задачи ВКР, она состоит в разработке собственного алгоритма для решения задач понижения размерности, его тестировании и сравнение результатов от применения алгоритма на различных datasets.

Дается математическое описание Algoritm_X.

Пусть $X_1, X_2, \ldots, X_n \in R^D$ данные в высокомерном пространстве, где $X_i = \{x_1, x_2, \ldots, x_D\}$, где x_j – j-ая координата наблюдения. Необходимо получить $Y_1, Y_2, \ldots, Y_n \in R^2$, где $Y_i = \{y_1, y_2\}$ – данные в двумерном пространстве. Причем, разница между соответствующими ненулевыми элементами матриц смежность построенных на X и Y была минимальной.

Матрица смежности построенная на X будет выглядеть следующим образом:

$$d = \begin{pmatrix} 0 & \rho(X_1, X_2) & \cdots & \rho(X_1, X_n) \\ \rho(X_2, X_1) & 0 & \cdots & \rho(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(X_n, X_1) & \rho(X_n, X_2) & \cdots & \rho(X_n, X_n) \end{pmatrix},$$

где $\rho(X_i, X_j) = \sqrt{\sum_{k=1}^{D} (x_{i_k} - x_{j_k})^2}$ – евклидово расстояние между X_i и X_j. Матрица d будет служить входными данными для решения задачи, однако ее нужно было разрядить. Поэтому, для каждой строки в матрице d находятся собственное число K минимальных ненулевых значений, эти значе-

ния сохраняются. Значение K может подбираться различными способами, об этом рассказано в подразделе. Также находятся несколько максимальных значений в матрицы d, они понадобятся для того, чтобы избежать наложения наблюдений друг на друга, об этом рассказывается подробнее в подразделе. Остальные значения в строках обнуляются.

На основе полученной матрицы d строим систему нелинейных уравнений. Каждое уравнение имеет следующий вид:

$$(y_{i_1} - y_{j_1})^2 + (y_{i_2} - y_{j_2})^2 \approx \rho(X_i, X_j)^2 \neq 0,$$

где y_{i_1} и y_{i_2} координаты i-го наблюдения в двумерном пространстве который нужно найти, решив систему состоящий из уравнений выше.

Из системы удаляются дублирующийся уравнения, однако, индекс удаляемого уравнения сохраняется в массив arr_index_duble. Это нужно будет, чтобы при поисках решения, учитывать, что расстояние некоторых пары наблюдений более приоритетное. Также, индексы уравнение, основанные на максимальных расстояниях, сохраняются в массиве arr_index_max. Так как уравнение основанные на максимальном расстоянии нужны только для того, чтобы избежать наложения наблюдений (групп наблюдений) друг на друга, то значении ошибки в этих уравнениях имеют не столь значительный весь по сравнения с расстояниями между близкими наблюдениями.

Система уравнений решается с помощью МНК. На каждой итерации соответствующих значения вектора ошибок увеличиваются в w_1 и уменьшаются в w_2 раза. Под соответствующими элементами, понимается элементы индексы которых есть в массиве arr_index_duble и arr_index_max соответственно. w_1 и w_2 – параметры МНК. Реализация МНК будет представлена в соответствующем подразделе. Решение полученной системы, и есть наблюдения с координатами в двумерном пространстве.

Далее проводится тестирования Algoritm_X на искусственно с генерированных данных. Результаты тестирования анализируются и формулируются выводы по разработанному алгоритму. Для подкрепления сформулированных выводов. Сравниваются результаты полученные алгоритмом Algoritm_X и остальными алгоритмами на различных datasets.

В **заключении** приведены основные результаты ВКР.

Основные результаты:

1. Сформулирована задача понижения размерности, проведен обзор соответствующий литературы и проведено исследование существующих решений для задач понижения размерности.

2. Рассмотрены пять алгоритмов для решения задач понижения размерности:

— PCA;
— LDA;
— ICA;
— t-SNE;
— UMAP.

Проанализированы результаты работ алгоритмов на различных datasets. Результаты приведены в приложении Б.

3. Разработан собственный алгоритм для решения задач понижения размерности. Проведено его тестирование и сравнение результатов от применения алгоритма на различных datasets. Результаты тестирования приведены в приложении А.

4. Сделан вывод по работоспособности разработанного алгоритма и по ВКР в целом.

Заключение. В данной ВКР была рассмотрена задача понижения размерности и различные методы для решения подобных задач.

Были рассмотрены 5 алгоритмов для понижения размерности PCA, LDA, ICA, t-SNE и UMAP. Каждый из алгоритмов обозревался в соответствующих подразделах. Каждый из алгоритмов был рассмотрен с точки зрения математики и реализовывался на языке python с помощью таких библиотек как sklearn.decomposition, sklearn.manifold, sklearn.discriminant_analysis и umap.

Были описаны 10 наборов данных в многомерном. Каждый из пяти алгоритмов решал задачу понижения размерности для каждого из 10 datasets.

Был разработан алгоритм Algoritm_X, приводится его математическое описание. Было проведено тестирование Algoritm_X. В ходе тестирование были выделены некоторые особенности алгоритма и его сильные и слабые стороны. Например: невосприимчивость к неравномерному распределению.

Были приведены результаты Algoritm_X на 10-и наборах данных и было проведено сравнение полученных результатов с результатами других алгоритмов. По результатам алгоритмов и поверхностному анализу данных, были выявлены слабые места алгоритмов, в частность алгоритма Algoritm_X. Главной слабостью Algoritm_X стало его производительность. Однако, как помечалось в подразделах «Тестирование алгоритма Algoritm_X» и «Применение алгоритма Algoritm_X на наборах данных», Algoritm_X успешно справился с поставленными задачами. В следствии чего было установлено, что все поставленные цели и задачи в данной ВКР были выполнены.

Результаты тестирование алгоритма Algoritm_X были приведены в приложении А. Результаты всех алгоритмов, включая Algoritm_X на 10-и наборах данных были приведены в приложении Б.