Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра ИИС

Иванов Дмитрий Александрович

Синтез трехмерных моделей методами машинного обучения

ВВЕДЕНИЕ В МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

Научный руководитель:

к.ф.м.н.

А.В. Игнатенко

Москва, 2018

Содержание

1 Введение

1.1 Машинное обучение

Машинное обучение (Machine Learning) – обширный подраздел теории искусственного интеллекта, математическая дисциплина, использующая разделы математической статистики, численных методов оптимизации, теории вероятностей, математического анализа и линейной алгебры. Алгоритмы машинного обучения формируют математическую модель данных из множества доступных примеров (обучение на основе обучающей выборки), которая позволяет затем алгоритму делать предсказания или принимать решения.

Один из подразделов машинного обучения является *обучение по признакам (Feature Learning, Represenation Learning)*. Данный набор техник, позволяют системе автоматически обнаружить представления, необходимые для выявления признаков или классификации исходных (сырых) данных. Это заменяет ручное конструирование признаков и позволяет машине как изучать признаки, так и использовать их для решения специфичных задач, что в свою очередь приводит к лучшим результатам и позволяет системам ИИ быстро адаптироваться к новым задачам с минимальным вмешательством человека. В то время как процесс ручного конструирования признаков может затягиваться на года и требует приложения огромного количества усилий со стороны исследователей, процесс автоматического извлечения признаков обычно занимает от нескольких часов до нескольких дней с приложением минимальных усилий со стороны человека [1].

Одной из основных проблем обучения по признакам является сложность отделения факторов изменчивости данных (к примеру, изменение освещения или наклона) от самих данных, так-как подобные факторы обычно влияют на все части данных, которые доступны. Чтобы справиться с такой задачей необходимо выделять высокоуровневые или абстрактные признаки, извлечение которых может быть крайне сложно [1]. Одним из решений данной проблемы является глубинное обучение.

Глубинное обучение (Глубокое обучение, Deep Learning) является подразделом обучения по признакам и характеризуется как класс алгоритмов машинного обучения, [2] который:

- Использует каскад нелинейных фильтров для извлечения признаков и преобразований. Каждый последующий слой получает на вход выходные данные предыдущего слоя.
- Формирует в процессе обучения слои на нескольких уровнях представлений, которые соответствуют различным уровням абстракции; слои образуют иерархию понятий.

Таким образом глубинное обучение решает одну из основных проблем обучения по

признакам, позволяя извлекать высокоуровненвые представления. Тем не менее для успешной работы алгоритмов глубинного обучения необходимы большие вычислительные мощности и большие объемы обучающей выборки. К сожалению, получение подобного объема данных обычно сопряжено с большими трудностями и требует большого числа человеческих, временных и денежных ресурсов. Одним из потенциальных решений данной проблемы является генерация синтетических (искусственных) данных.

1.2 Синтетические данные

Синтетические (искусственные) данные - данные, полученные искусственно (синтезированные), то есть, не прибегая к непосредственному измерению в реальности. При этом подразумевается, что синтетические данные эмулируют реальные данные с сохранением статистических зависимостей, то есть переносят свойства реальных данных на синтезируемые. Конечная цель подобной эмуляции – создание алгоритма, позволяющего генерировать данных на основе реальных. Синтетические данные при их анализе должны приводить к тем же результатам и выводам, что и реальные.

В общем случае, синтетические данные имеют следующие преимущества:

- Если имеется алгоритм для генерации данных, то можно практически без затрат получить необходимое количество данных
- Синтетические данные могут быть безошибочно размечены, в то время как, это может быть крайне затратно или невозможно сделать на реальных данных
- Синтетические данные могут быть использованы в качестве замены для определенных частей реальных данных (к примеру, если реальные данные содержат конфидециальную информацию)

Использование синтетических данных для пополнения обучающей выборке находит применение во многих областях, особенно в задачах компьютерного зрения и в задачах обнаружения объектов. К примеру в работе [3] трехмерное синтетическое окружение на основе CAD моделей позволило дополнить обучающую выборку двухмерных изображений, что привело к увеличению эффективности работы алгоритма распознования.

1.3 Генеративные модели

Одним из методов для генерации синтетических данных являются *генеративные модели.*

Сами генеративные модели как концепция существуют уже не первое десятилетие и различными исследователями было создано большое количество различных моделей [5], [19], :

- *Генеративно-состязательные сети (Generative Adversarial Networks, GANs)*: обучение модели похоже на игру, в которой соревнуются *генеративная (генератор)* и *дискриминативная (дискримантор)* сети, которые и составляют два основных элемента данной модели [6], [8].

- *Вариационные автоэнкодер (Variational Autoencoders, VAEs)* использует архитектуру кодировщик-декодировщик, при этом использует вероятностный подход для семплирований переменных в латентном пространстве.

- *Авторегрессионные модели (Autoregressive models)*: обучают модель определять значения пикселя на основе значений пикселей слева и снизу.

- *Модели основанные на смеси гауссовых распределений (Gaussian mixture model, GMM)*. GMM - вероятностная модель для аппроксимации распределения данных в виде суммы гауссовых распределений [23].

- *Модели основанные на статистическом анализе формы объекта (3D statistical shape spaces)*. Данная модель описывает данные в виде суммы «средней» формы по всем данным и вектора флуктуаций. Получив среднюю форму и зная диапазоны варьирования вектора флуктуаций, можно получать новые объекты, сохраняя статистические свойства исходных объектов [21], [22].

Последние успехи в области построения генеративных моделей, особенно в генеративно-состязательных сетях, естественным образом приводят к мысли о том, что подобные модели способны самостоятельно синтезировать все необходимые данные для последующего обучения. Подобный подход, полностью основанный на синтетических данных, пока не реализован, хотя вышеперечисленные подходы уже существенно улучшили генерацию синтетических данных. К примеру, в работе [4] предлагаются различные методы по генерации высококачественных и близких к реальным изображений.

1.4 Машинное обучение для обработки трехмерных объектов

Одной из возможных областей применения различных нейронных сетей является обработка тремерных изображений. В последнее время это область особенно активно развивается в связи с полученим больших объемов трехмерных данных с лидаров, которые активно используются в автономных траснпортных средствах, с трехмерных сканеров, которые все чаще используются в различных областях человеческой деятельности (к примеру, в медицине), CAD моделей, сцен из компьютерных игр и симуляторов.

Растет и количество предложенных моделей и методов для задач распознавания и сегментации трехмерных объектов [11], [18], [12], [13], [14].

Одной из основных проблем, с которыми сталкиваются, при работе с трехмерными объектами, представленными *полигональными сетки (polygon meshes)* является их

чрезмерная разреженность и неравномерность [9]. Альтернативой этому выступает использование регулярных сеток - *вокселей (voxels, трехмерных пикселей, occupancy grid)*, которые позволяют переносить методы, аппробированные на двумерных данных в эту область. Но при таком подходе, в случае низкого уровня дискретизации, теряется информация, а в случае повышение уровня дискртизации – приводит к огромному потреблению памяти.

1.5 Проблематика

В задачах обработки трехмерных объектов методами глубинного обучения, как и во всех задачах глубинного обучения, очень часто возникает проблема недостатка данных для обучения. Причины возникновения такой проблемы могут быть разными: от невозможности получить и разметить эти данные из-за недостатка ресурсов, до ограничений из-за конфидециальности данных.

Все эти факторы естественным образом приводят к идее использовать различные генеративные модели для генерации синтетических данных. В последние несколько лет исследователями были предложены различные попрождающие модели как на основе генеративно-состязательных моделей [15], так и на основе вариационных энкодировщиков [16]. Стоит отметить, что входные данные для данных моделей были представлены в виде вокселей. В новых работах уже предложены методы, которые позволяют работать с облаками точек [17], [19].

Целью данной работы является построение новых алгоритмов генерации синтетических 3D данных на основе методов машинного обучения. Обобщая вышесказанное, можно с уверенностью сказать, что задача генерации трехмерных синтетических данных нова и актуальна.

2 Постановка задачи и описание данных

2.1 Неформальная постановка задачи

Пусть имеется выборка трехмерных обектов, представленных в виде облаков точек (point clouds). Необходимо создать алгоритм, который на основе данной выборки будет генерировать *новые* объекты подобные объектам в выборке, то есть будет генерировать новые объекты с сохранением статистических и геометрических свойств исходных объектов.

2.2 Формальная постановка задачи

Облако точек - структура данных для представления геометрической формы некоторого объекта в виде неупорядоченного множества точек. Для трехмерного случая облаком точек будет называться неупорядоченное множество $P = \{(x_i, y_i, z_i)\}_{i=1}^{N}$, где N – число точек в облаке.

Пусть у нас имеется выборка облаков точек подчиненных некоторому вероятностному распределению $p(X)$: $X_1, \ldots, X_L \sim p(X)$. Пусть распределение $q(X|z, \theta)$ аппроксимирует (оценивает, estimate) распределение p. Требуется построить алгоритм для семлирования объектов, подчиняющийся распределению q.

2.3 Описание данных

В данной работе, поставленная задача, изучается и решается на основе выборки зубов человека.

Опишем некоторые термины и понятия из зубного дела.

2.3.1 Зубная анатомия

Зубы классифицируются на резцы (incisors), клыки (canines), премоляры (premolars) и моляры (molars). Вместе зубы объединяются в верхний и нижний зубные ряды. Каждая зубная дуга (dental arch), или по-другому ряд зубов, разделяется на левую и правую часть. С каждой стороны у человека имеется два резца, один клык, два премоляра и три моляра [7] см. рис. 1.

2.3.2 Размер выборки

В выборке присутствуют 28 типов зубов (все, кроме третьих моляров - зубов мудрости), для каждого типа зуба имеется от 124 до 150 экземпляров. Все экземпляры представлены в виде трехмерных сеток (перед началом работы основного алгоритма

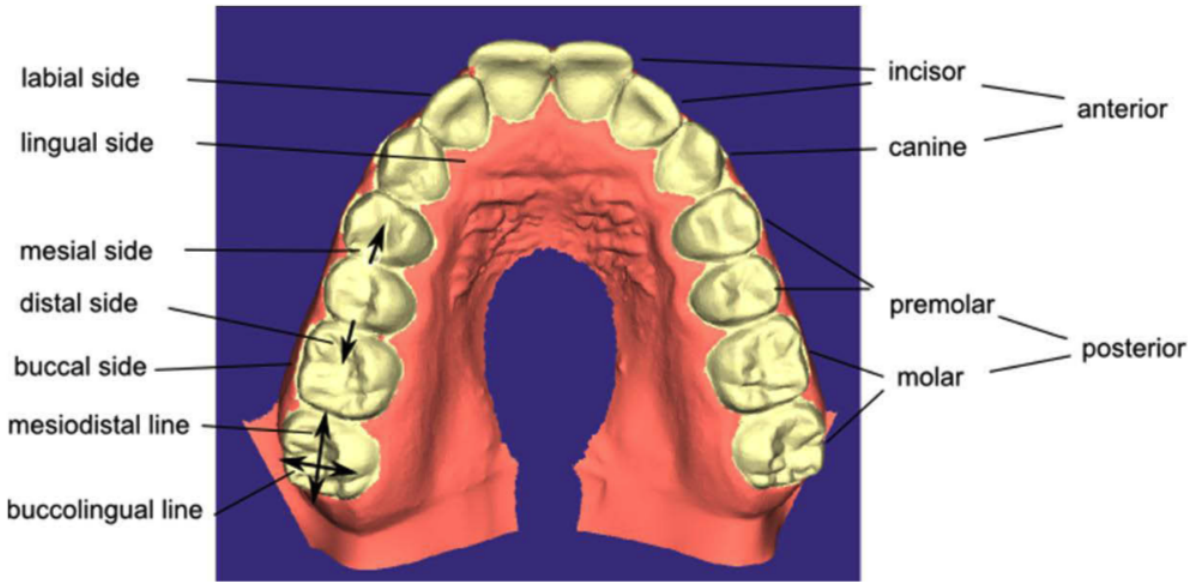

Рис. 1: Анатомия зубов

мы переводим их в облака точек, путем удаления ребер). Примеры экземпляров из выборки можно увидеть на изображениях 2 - 6.

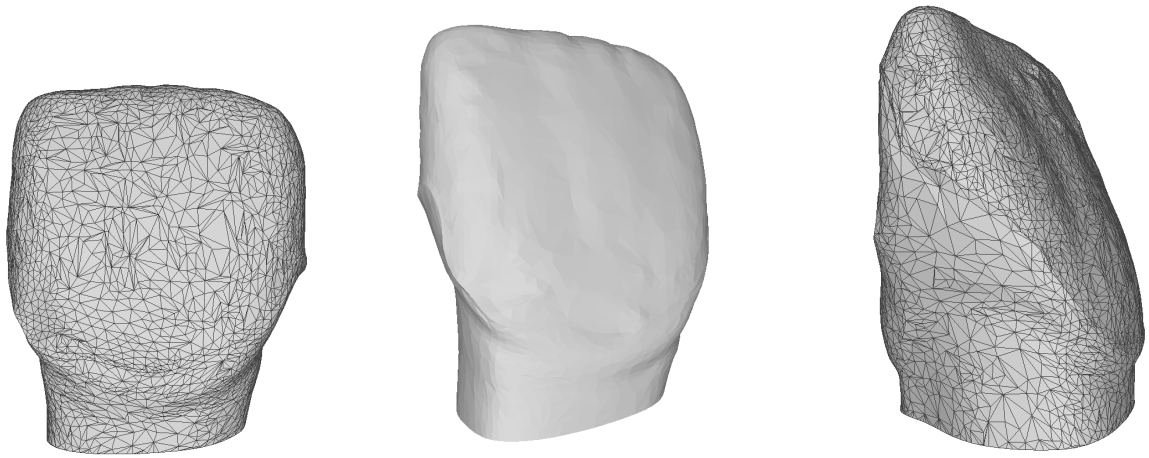

Рис. 2: Примеры данных из выборки в виде трехмерных сеток

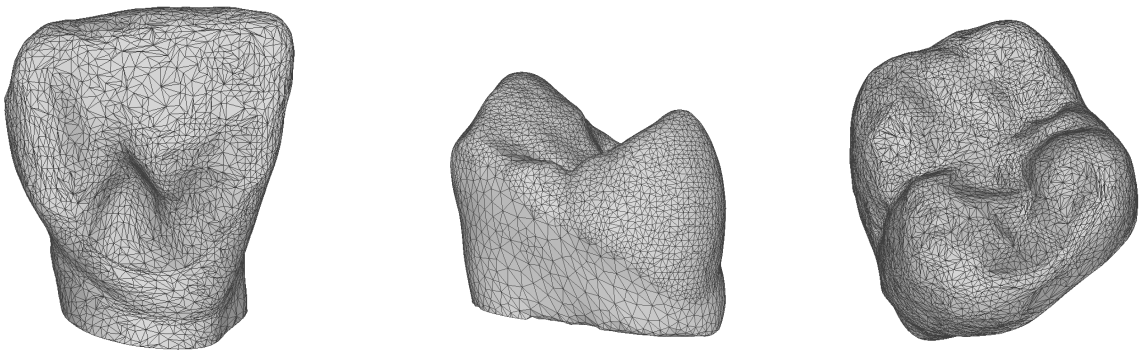

Рис. 3: Примеры данных из выборки в виде трехмерных сеток

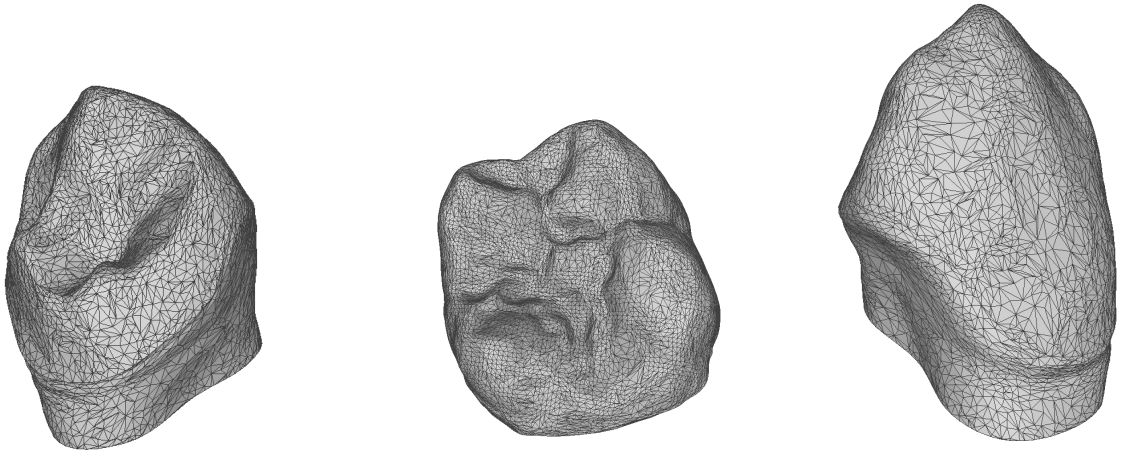

Рис. 4: Примеры данных из выборки в виде трехмерных сеток

Рис. 5: Примеры данных из выборки в виде облаков точек

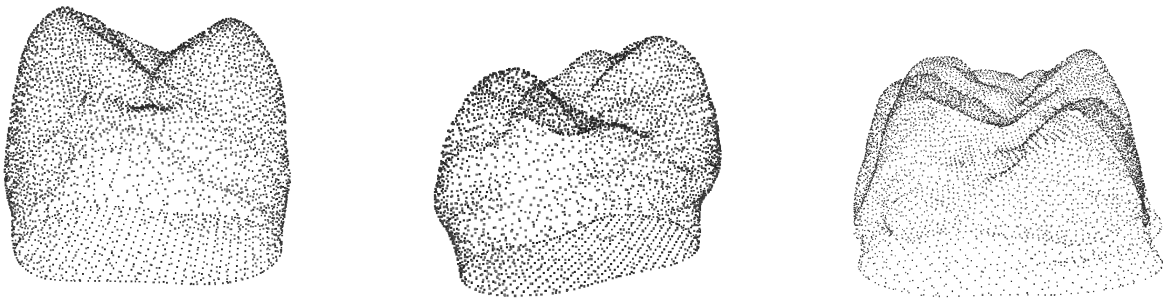

Рис. 6: Примеры данных из выборки в виде облаков точек

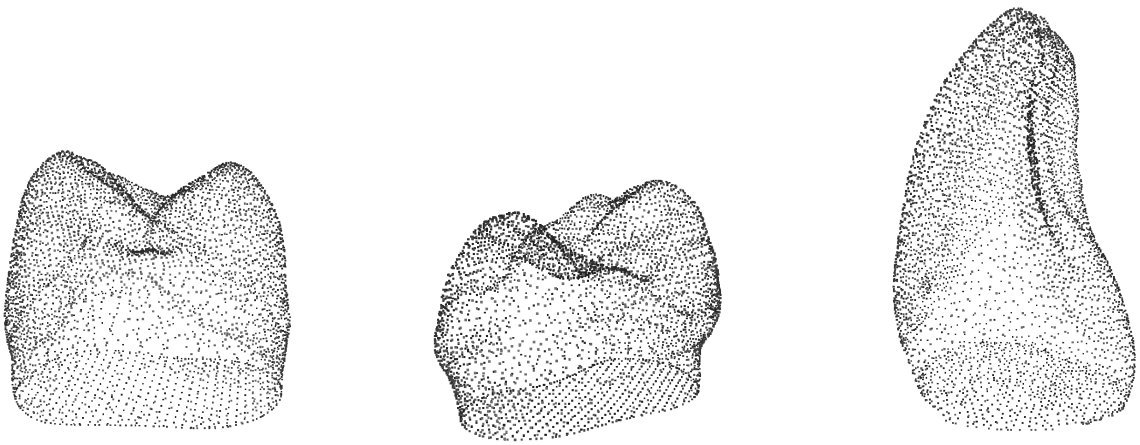

Рис. 7: Примеры данных из выборки в виде облаков точек

3 Метрики сравнения облаков точек

3.1 Проблема сравнения облаков точек

Работа с облаками точек привносит ряд проблем в алгоритмы для их обработки и принципиально отличается от работы с двухмерными изображнениями. К примеру, в отличиии от двухмерных изображений, облака точек не имеют жесткой *решетчатой структуры (grid-like structure)*, что приводит к невозможности использования многих алгоритмов из области обработки двухмерных ихображений при решении данной проблемы. Другой, не менее важной проблемой, является проблема упорядочивания самих точек в облаке. В общем случае, никакого порядка точек не существует, что создает инвариант перстановки (permutation invariant). То есть, в случае задания порядка точек в облаке, два облака с разными упорядочиваниями будут представлять абсолютно один и тот же трехмерный объект.

Вышеперечисленные проблемы сильно усложняют процесс сравнения двух облаков точек, который необходим для вычисления потерь алгоритма при восстановлении формы объектов, и приводят к необходимости выработки метрики, инвариантной к перестановкам порядка точек в облаке.

3.2 Метрики сравнения облаков точек

Одной из наиболее известных метрик для вычисления расстояния между компактными подмножествами метрического пространства является *метрика Хаусдорфа*:

$$d_H(X,\ Y) = \max\left\{\sup_{x\in X}\inf_{y\in Y}|xy|,\ \sup_{y\in Y}\inf_{x\in X}|xy|\right\}$$

Однако она является крайне неусточивой к небольшим выбросам во множествах.

В работах [20], [19] были предложены две метрики: *CD расстояние, (Расстояние Чамфера, Chamfer distance, CD)* и *EMD расстояние (Earth Mover's distance, EMD).*

3.2.1 CD расстояние

Пусть $S_1, S_2 \subset \mathbb{R}^3$, тогда CD расстояние между ними определяется как:

$$d_{CD}(S_1, S_2) = \sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2 + \sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2$$

Строго говоря CD расстояние не является метрикой, так-как не выполняется неравенство треугольника. Вычисление данной метрики для каждой пары точек проихсодит независимо, в связи с чем данная задача может быть эффективно распараллелена и скорость вычисления данной функции кратно увеличится. Также операции поиска

ближайшего соседа могут быть существенно ускорены путем использования различных пространственных структура данных, таких как *KD-деревья (KD-tree)*.

3.2.2 EMD расстояние

Рассмотрим множества $S_1, S_2 \subset \mathbb{R}^3$ одинакового размера $s = |S_1| = |S_2|$. Тогда EMD расстояние между ними определяется как:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \mapsto S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

где отображение $\phi : S_1 \mapsto S_2$ является биекцией.

Стоит отметить, что нахождение EMD расстояния аналогично решению задачи о назначениях [24], [25], которая решается *Венгерским алгоритмом*[26] за полиномиальное время (в худшем случае $O(n^4)$). Разумеется, на практике, точное вычисление EMD будет крайне ресурсозатратным, особенно для таких объектов как трехмерные изображения. В связи с чем, авторы статьи [20] предлагают использовать аппроксимацию, которая позволяет более быстро вычислять ответ, пусть и с небольшими неточностями. Необходимо отметить, что данный метод работает для облаков точек с одинаковыми количествами точек.

В результате, данные метрики обладают следующими свойствами:

1. Диффернцируемость относительно координат точек сравниваемых множеств
2. Устойчивость к малым выборсам
3. Относительная вычислительная эффективность для CD расстояния и для EMD аппроксимации

В данной работе, при сравнении облаков точек, мы будем использовать CD расстояние, как более простое, более эффективное в реализации, интуитивно понятное и умеющее работать с облаками точек, состоящих из разных количеств точек.

4 Метрики для оценки генеративных моделей для трехмерных данных

Имея метод для сравненения двух облаков точек (см гл. 3) можно перейти к созданию метрики, оценивающей потери алгоритма при воспроизведении форм исходных объектов.

Пусть у нас имеется набор облаков точек A и набор облаков точек B. Требуется оценить насколько хорошо набор A воспроизводит форму исходных данных, представленных набором B.

В работе [19] авторы предлагают ряд метрик для данной задачи. Опишем две из них: *Покрытие (Coverage, COV)* и *Минимальное сопоставимое расстояние (Minimum matching distance, MMD).*

4.1 COV метрика

Для каждого объекта из набора A, находим ближайшего соседа к нему из набора B. Покрытие (COV метрика) считается как отношение количества объектов из набора B, которые были сопоставлены объектам из набора A, к числу всех объектов в наборе B. Чем больше значение данной метрики, тем лучше набор A покрывает набор B [19].

4.2 MMD метрика

Для вычисления данной метрики, мы каждому объекту из набора B сопоставляем ближайший к нему объект из набора A и затем высчитываем среднюю велечину расстояний. Чем меньше значение данной метрики, тем ближе набор A лежат к набору B [19].

4.3 Обсуждение

COV метрика показывает насколько хорошо происходит покрытие набора B объектами из набора A, а MMD метрика показывает близость (точность, fidelity) набора A к набору B.

Несложно придумать случай, когда COV метрика будет иметь большое значение и при это MMD также будет иметь большое значение. К примеру, пусть оба набора находятся внутри двух параллелипипедов, параллельных друг-другу и сильно удаленных друг от друга, причем объекты внутри парллелипиедов выстроены вдоль линий параллельности (см рис. 8). Тогда для каждого объекта из первого набора (красные элементы) ближайшим к нему из второго набора (зеленые элементы) будет объект лежащий напротив него (соединены пунктирной линией). То есть, будет выполнено полное по-

крытие зеленого набора красным набором. Но при этом, среднее значение MMD может быть сколь угодно большим (на рисунке она равняетя 80).

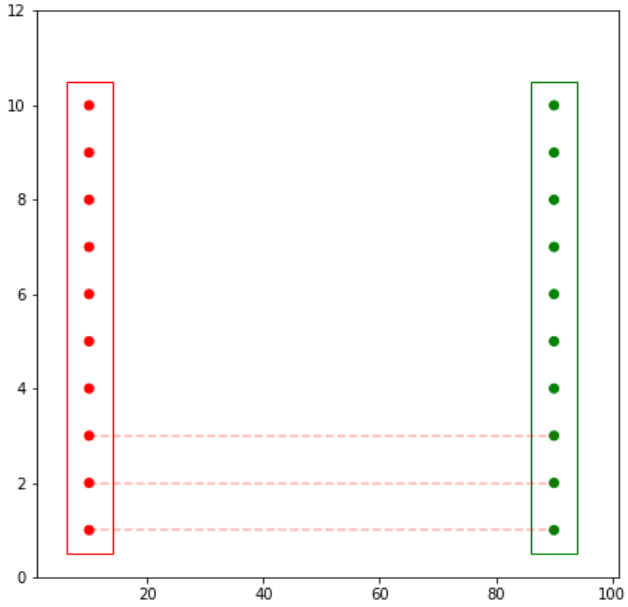

Рис. 8: Случай когда значения MMD и COV метрик велики

Аналогично, можно придумать пример, когда одновременно COV метрика и MMD метрика будут иметь маленькие значения. К примеру, объкты из набора A сосредточены внутри некоторого шара S_A. И пусть почти все объекты набора B также сосредоточены внутри шара S_B. Причем шары не пересекаются: $S_A \cap S_B = \emptyset$, но при этом лежат довольно близко. И пусть один из объектов b из набора B не лежащих внутри шара лежит на отрезке соединяющим центры шаров S_A и S_B (см рис. 8). Тогда получится, что COV метрика будет крайне малой (покрывается только объект b), но значение метрики MMD будет мало, ибо расстояния между группами объектов небольшие (на рисунке оно не больше 10).

В связи с этим, мы приходим к выводу о том, что COV и MMD метрики друг-друга отлично дополняют и их следует использовать в совокупности. И можно говорить о том, что набор сгенерированных объектов A воспроизводит исходные формы объектов, представленные набором B, тогда, когда COV метрика велика, а MMD, наоборот, мала. Стоит отметить, что это является лишь необходимым, но не достаточным условием.

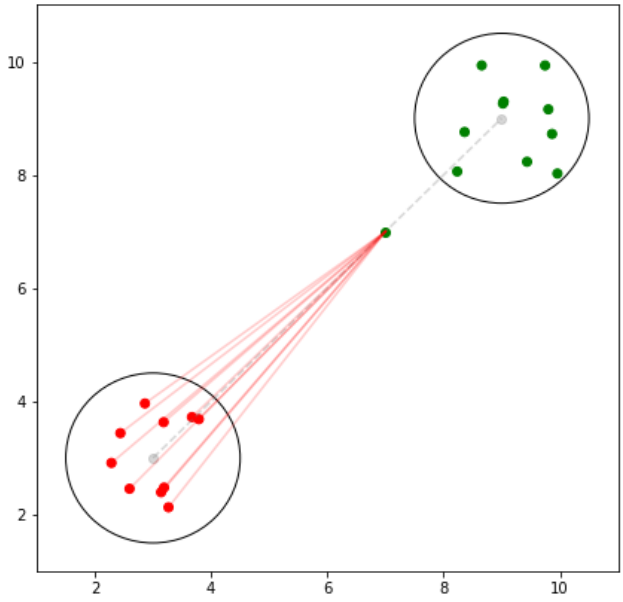

Рис. 9: Случай когда значения MMD и COV метрик малы

5 Описание алгоритма

Существует большое количество методов, позволяющих решать данную задачу. Методы, основанные на GAN (генеративно-состязательных) сетях [6],[15] и VAE (вариацонном автоэнкодировщике) [17], [16],[19], не могут быть применены для решения данной задачи, по причине крайне малого (для задачи глубинного обучения) размера выборки. К примеру, в работе [19] при обучении на одну категорию используется более 1000 объектов. Более того, далеко не все архитектуры нейронных сетей позволяют на вход передавать облака точек с разными количествами вершин. В работе [19] авторы принудительно ограничивают вход сети определенным количеством точек. Стоит отметить, что в данном случае, можно было бы использовать специальные свертки , чтобы преодолеть подобные ограничения.

В связи с вышеперечисленными фактами в данной работе мы будем использовать алгоритм, частично основанный на идеях *статистического анализа форм объектов (Statisitical shape spaces analysis)*.

Пусть у нас имеется выборка облаков точек размерности M.

5.1 Выравнивание объектов (Alignment)

Вначале все объекты в выборке необходимо выравнить. То есть, сделать так, чтобы ориентация всех зубов была одинаковой. Задача оптимального совместного положения геометрических объектов широко известна и хорошо изучена. В общем случае задачу можно сформулировать следующим образом: пусть у нас имеется два облака точек A и B. Необходимо найти матрицу поворота R, вектор сдвига t и коэффициент размера s, такие что при применении их к объекту B значение метрики расстояния между объектами стало минимальным. Обычно говорят токлько о нахождении матрицы поворота. В общем случае мы не обговариваем о какой именно метрике идет речь.

В данной задаче возможны два случая: случай, когда имеется имеется взаимно однозначное соответствие точек двух объектов, и случай, когда его нет.

5.1.1 Случай наличия взаимно однозначного соответствия точек двух объектов

Пусть у нас имеются два объекта, причем для каждой точки a_i первого объекта $A = \{a_1, \dots, a_N\}$, существует соответствующая точка b_i второго объекта $B = \{b_1, \dots, b_N\}$. Наличие данного соответствия отражается в функции расстояния между объектами (из соответствующей точке первого объекта вычисляется соответствующая точка второго объекта): $D(A, B) = \sum_{k=1}^{N} \|a_k - b_k\|_2^2$.

Тогда задача поиска матрицы поворота в точности является *задачей Вахба (Wahba's*

problem) [29], которая заключается в нахождении матрицы поворота R, минимизирующей функция ошибки

$$I(\mathbf{R}) = \sum_{k=1}^{N} \|a_k - \mathbf{R}\, b_k\|_2^2$$

. Также существует *ортогональная задача Прокрустеса (Orthogonal Procrustes problem)*, являющаяся аналогичной поставленной задаче.

5.1.2 Случай отсутствия взаимно однозначного соответствия точек двух объектов

В случае отсутствия взаимно однозначного соответствия задача выглядит следующим образом: пусть у нас имеются два объекта $A = \{a_1, \ldots, a_N\}$ и $B = \{b_1, \ldots, b_K\}$. Требуется найти такую матрицу R, чтобы функция ошибки

$$I(\mathbf{R}) = \sum_{a \in A} \|a - \mathbf{R}\, closest(a, B)\|$$

, где $closest(a, B) = \underset{b_1,\ldots,b_K}{\operatorname{argmin}} \|a - b\|$ принимала минимальное значение.

Данная задача в точности является задачей выравнивания множеств точек (point set registration, point). Данная задача хорошо изучена и существуют различные алгоритмы для ее решения.

Одним из таких алгоритмов является ICP алгоритм [30], [31]. Он имеет две версии: rigid ICP и non-rigid ICP. Первая форма использует только жесткие преобразования (поворот, масштабирование и перенос), вторая же форма использует весь спектр аффинных преобразований. Идея алгоритма заключается в итеративной оценке и итеративном преобразовании (поворот, масштабирование и т.д.) исходных объектов.

Так-как функции ошибки зависит от взаимного положения объектов, то она модет меняться по ходу алгоритма. В связи с этим сложно доказать достижения локального оптимума алгоритмом. На самом деле, ICP в большинстве случаев не достигает локальных минимум функции ошибки [32]. Тем не менее, благодаря своей простоте, данный алгоритм остается самым распространенным алгоритмом для задачи выравнивания объектов.

Существуют также и другие алгоритмы: алгоритм надежного сопоставления точек (Robust point matching), алгоритм корреляции ядер (Kernel correlation).

5.1.3 Выбор алгоритма

Так-как в выборке нету однозначного соответствия между точками (более того, объекты выборки имеют разные количества вершин), то мы будем использовать ICP алгоритм. Самой простой версии алгоритма - rigid ICP оказалось достаточно и в дальнейшем мы ее будем называть ICP.

5.2 Нахождение общего порядка точек

В дальнейших шагах алгоритма нам понадобится иметь общий порядок точек для всех облаков, то есть требуется найти взаимнооднозначное соответствие между вершинами облаков точек. Предыдущий шаг, во многом был нужен именно для того, чтобы процедура поиска соответствующий точек выполнялась как можно точнее.

Для этого мы из выборки выбираем облако точек с наименьшим количеством вершин (все равно больше соответствий мы найти не сможем). Если такие объектов будет несколько, то выбираем среди них один случайным образом. Будем называть найденное облако точек *референсным*.

Теперь опишем процедуру поиска соответствия между вершинами для двух облаков точек. Пусть на вход передаются два облака точек, причем второе имеет меньше либо равное количество вершин чем первое облако. Для каждой вершины во втором облаке мы ищем ближайшую к нему вершину в первом облаке, и ставим их в однозначное соответствие. Разумеется, что часть точек первого облака мы потеряем, более того, возможно, что нескольким разным вершинам из второго облака будут соответствовать одинаковые вершины из первого.

Выполним описанную выше процедуру для каждого объекта из выборки и референсного облака. Теперь, у нас есть взаимно однозначные соответствия вершин для всех облаков в выборке. Вершинами, которым не были найдены соответствующие вершины референсного облака - отбрасываются.

В итоге, мы задаем некоторый случайный порядок вершин на референсном облаке, и, имея взаимно однозначное соответствие вершин для всех объектов из выборки, мы получаем порядок для вершин облаков из выборки.

Несмотря на то, что данный подход весьма прост, на имеющейся выборке он показывает хорошие результаты (в данном случае, большую роль играет выровненность объектов и их однородность).

5.3 Снижение размерности и анализ главных компонент

Имея порядок вершин и их одинаковое количество для всех облака, мы превращаем облака точек в вектора, согласно порядку и получаем матрицы рамзерности $(N \times 3)$. Затем мы конкатенируем строчки данных матриц и получаем вектора размерности $(3N)$.

Сформируем матрицу из этих векторов. В итоге, у нас получится матрица $V \in^{(} M \times 3N)$.

Теперь, к этой матрице можно применить алгоритм *PCA (Principal Component Analysis* [23] и получить в результате значения главных компонент.

Стоит отметить,

Получив значения главных компонент, можно проанализировать насколько каждая компонента важна, для этого все главные компоненты упорядочиваются в порядке убывания: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$. Необходимо выбрать первые K компонент, которые будут сильно больше, чем последующие $M - K$ компонент. Пример распределения главных компонент представлен на рис. 10, 11. Как можно будет увидеть дальше, для выборки зубов оптимальной областью значений K является диапазон от 6 до 9 включительно.

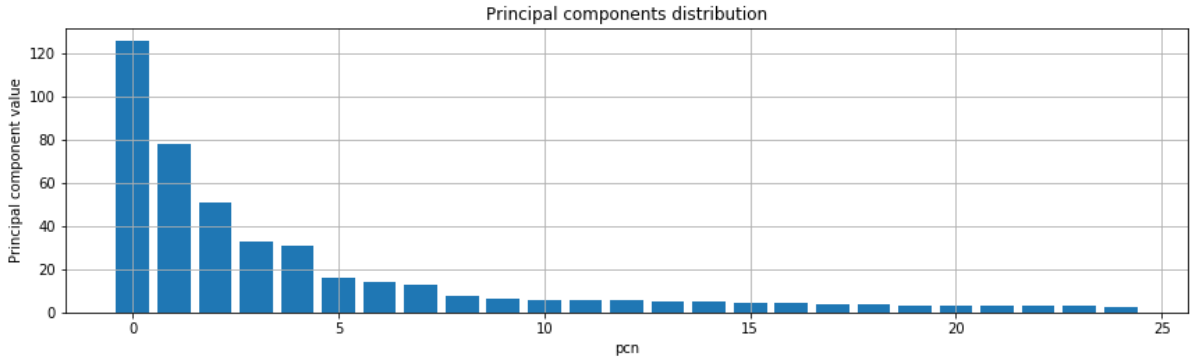

Рис. 10: Пример распределения значений главных компонент

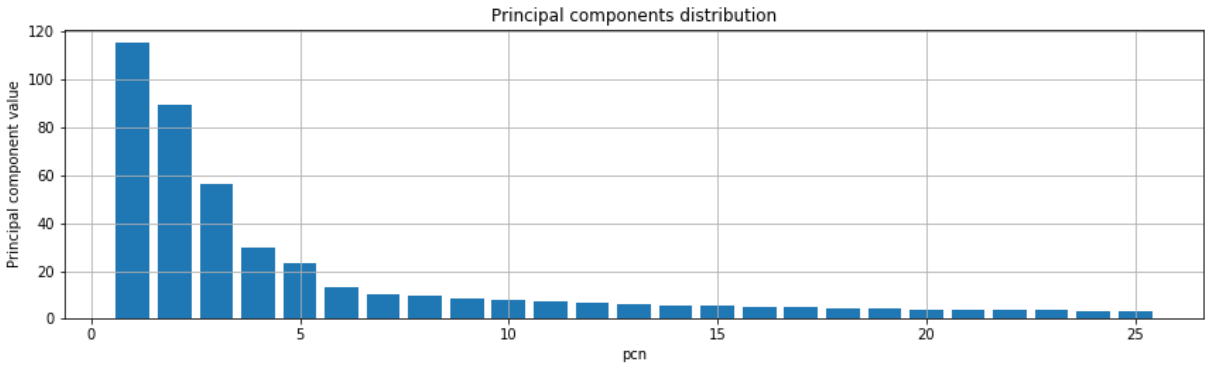

Рис. 11: Пример распределения значений главных компонент

5.4 Генерация новых объектов

На предыдущем шаге мы оставили K главных компонент. Каждой главная i-ой компоненте соответствует стандартное отклонение σ_i. То есть, из них можно сформировать вектор стандартных отклонений $\overline{\sigma} = \{\sigma_1, \ldots, \sigma_K\}$.

В итоге можно семлировать случайные вектора с диапазоном $\pm c \cdot \overline{\sigma}$, где c - некоторая константа. В дальнейшем будет показано, что значения $c \geq 3$ не дают особого прироста к точности (что хорошо коррелирует с правилом трех сигм).

6 Результаты

В целом, алгоритм показал себя хорошо: на некоторах типов зубов показатели COV метрики достигали значений 0.9. Что при столь малом размере выборки является очень хорошим результатом. Для избеганий зависимости от разбиений выборки на тестовую и обучающую части была использована кросс-валиадация, с коэффициентом разбиения 10.

6.1 Время работы

Сам алгоритм на компьютере MacBook Pro(2015) с процессором 2,7 GHz Intel Core i5 при размере выборки в 150 элементов работает меньше чем за секунду. Во многом, такая скорость была достигнута за счет использования KD-деревьев для расчета CD-расстояния (см. 3).

Тем не менее, расчет метрик COV и MMD для оценивания качества сгенерированных объектов ...

6.2 Метрики

6.3 Примеры данных

На рисунках представлены результаты генерации новых зубов.

7 Заключение

Список литературы

[1] Ian Goodfellow and Yoshua Bengio and Aaron Courville *Deep Learning* MIT Press, 2016

[2] Deng, L.; Yu, D. *Deep Learning: Methods and Applications* // Foundations and Trends in Signal Processing, 2014 1-199

[3] Peng, Xingchao; Sun, Baochen; Ali, Karim; Saenko, Kate Saenko *Learning Deep Object Detectors from 3D Models* University of Massachusetts Lowell, 2015

[4] Shrivastava, Ashish; Pfister, Tomas; Tuzel, Oncel; Susskind, Josh; Wang, Wenda; Webb, Russ *Learning from Simulated and Unsupervised Images through Adversarial Training* arxiv.org, 2017

[5] Sanchez, Cassie *At a Glance: Generative Models & Synthetic Data* September 2017.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio *Generative Adversarial Networks* 2014

[7] Y. Kumar, R. Janardan, and B. Larson, *Automatic feature identification in dental meshes* Computer-Aided Design and Applications, vol. 9, no. 6, pp. 747–769, 2012.

[8] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network* 2017

[9] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, Pheng-Ann Heng *PU-Net: Point Cloud Upsampling Network* The Chinese University of Hong Kong, Tel Aviv University, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 2018

[10] Truc Le, Ye Duan *PointGrid: A Deep Network for 3D Shape Understanding* University of Missouri, 2018

[11] Charles R. Qi Li Yi Hao Su Leonidas J. Guibas *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space* Stanford University, 2017

[12] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, Xin Tong *O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis* 2017

[13] Huan Lei, Naveed Akhtar, Ajmal Mian *Spherical Convolutional Neural Network for 3D Point Clouds* University of Western Australia, 2018

[14] Taco S. Cohen, Mario Geiger, Jonas Köhler, Max Welling *SPHERICAL CNNS* University of Amsterdam, 2018

[15] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, Joshua B. Tenenbaum *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling* 2017

[16] Andrew Brock, Theodore Lim, J.M. Ritchie, Nick Weston *Generative and Discriminative Voxel Modeling with Convolutional Neural Networks* 2016

[17] Maciej Zamorski, Maciej Zieba, Rafal Nowak, Wojciech Stokowiec and Tomasz Trzcinski *Adversarial Autoencoders for Generating 3D Point Cloud* 2018

[18] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, Silvio Savarese *SEGCloud: Semantic Segmentation of 3D Point Clouds* 2017

[19] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J Guibas. *Learning Representations and Generative Models For 3D Point Clouds* Proceedings of the 35th International Conference on Machine Learning 2018

[20] Haoqiang Fan, Hao Su, Leonidas Guibas *A Point Set Generation Network for 3D Object Reconstruction from a Single Image* arXiv e-prints 2016

[21] Alan Brunton, Augusto Salazar, Timo Bolkart, Stefanie Wuhrer *Statistical Shape Spaces for 3D Data: A Review* Chi Hau Chen. Handbook of Pattern Recognition and Computer Vision, 5th Edition, 2016, 978-981-4656-52-8.

[22] Alan Brunton, Augusto Salazar, Timo Bolkart, Stefanie Wuhrer *Review of Statistical Shape Spaces for 3D Data with Comparative Analysis for Human Faces*

[23] Christopher M. Bishop *Pattern Recognition and Machine Learning* Information Science and Statistics, 2006

[24] Хемди А. Таха. *Введение в исследование операций* 7-е издание. Пер. с англ. — М.: Издательский дом «Вильямс», 2005.

[25] Вагнер Г. *Основы исследования операций* Пер. с англ. — М.: Издательство «Мир», 1972.. — Т. 1.

[26] Harold W. Kuhn *Variants of the Hungarian method for assignment problems* Naval Research Logistics Quarterly, 3: 253—258, 1956

[27] Python v3.5 documentation [HTML] (http://docs.python.org/3.5).

[28] Лутц М. *Изучаем Python, 4-е издание.* – Пер. с англ. – СПб.: Символ-Плюс, 2011.

[29] Wahba, G. *Problem 65–1: A Least Squares Estimate of Spacecraft Attitude*, SIAM Review, 1965, 7(3), 409

[30] P.J. Besl, N.D. McKay. *A method for registration of 3D shapes* IEEE Transactions on Pattern Analysis and Machine Intellinegce 14 (1992) 239– 254.

[31] B. K. P. Horn *Closed-form solution of absolute orientation using unit quaternions* J. Opt. Soc. Amer. Avol. 4, no. 4, pp. 629-642, Apr. 1987.

[32] Tsin, Yanghai; Kanade, Takeo. *A Correlation-Based Approach to Robust Point Set Registration* Computer Vision ECCV. Lecture Notes in Computer Science. 3023. Springer Berlin Heidelberg