



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	3
1.1	Машинное обучение	3
1.2	Синтетические данные	4
1.3	Генеративные модели	4
1.4	Машинное обучение для обработки трехмерных объектов	5
1.5	Проблематика	6
2	Постановка задачи и описание данных	7
2.1	Неформальная постановка задачи	7
2.2	Формальная постановка задачи	7
2.3	Описание данных	7
2.3.1	Зубная анатомия	7
2.3.2	Размер выборки	8
3	Метрики сравнения облаков точек	10
3.1	Проблема сравнения облаков точек	10
3.2	Метрики сравнения облаков точек	10
3.2.1	CD расстояние	10
3.2.2	EMD расстояние	11
4	Метрики для оценки генеративных моделей для трехмерных данных	12
4.1	COV метрика	12
4.2	MMD метрика	12
4.3	Обсуждение	12
5	Описание алгоритма	15
6	Результаты	16
7	Заключение	17

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, Representation 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 высокоуровневые 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 конфиденциальную 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 двумерных изображений, что привело к увеличению эффективности работы алгоритма распознавания.

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-сопоставительные сети (Generative Adversarial Networks, GANs)*: обучение модели похоже на игру, в которой соревнуются *генеративная (генератор)* и *дискриминативная (дискриминатор)* 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 семплирования 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-сопоставительных 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 трехмерных изображений. В последнее время эта область особенно активно развивается в связи с получением больших объемов трехмерных данных с лидаров, которые активно используются в автономных транспортных 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 сетками (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, апробированные на двумерных данных в эту область. Но при таком подходе, в случае низкого уровня дискретизации, теряется информация, а в случае повышения уровня дискретизации – 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 конфиденциальности 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 объектов, 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

2.3 Описание данных

В данной работе, поставленная задача 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].

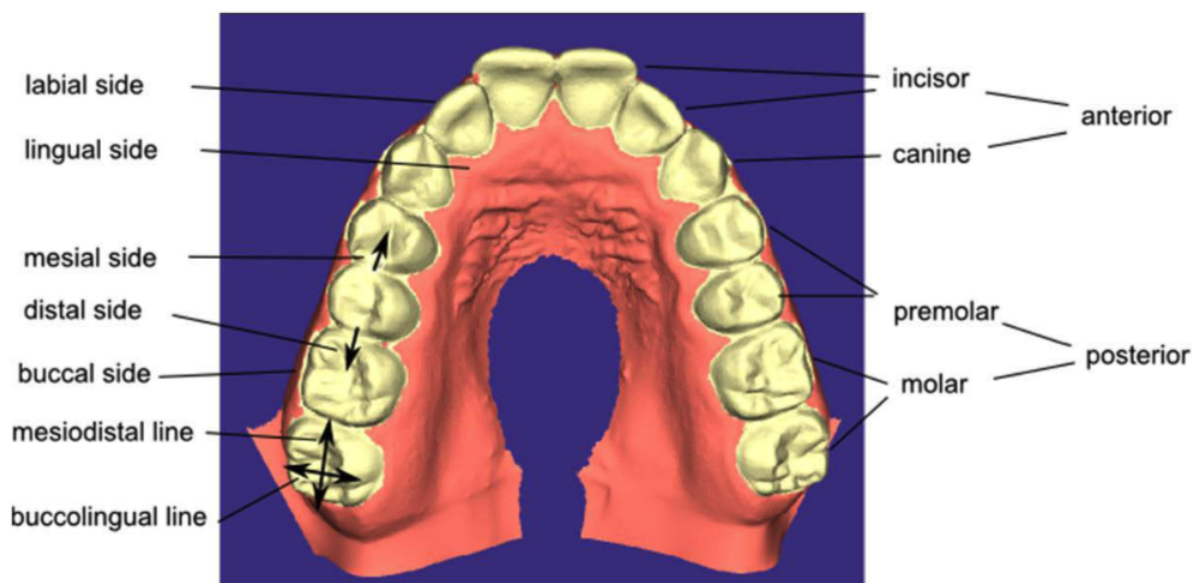
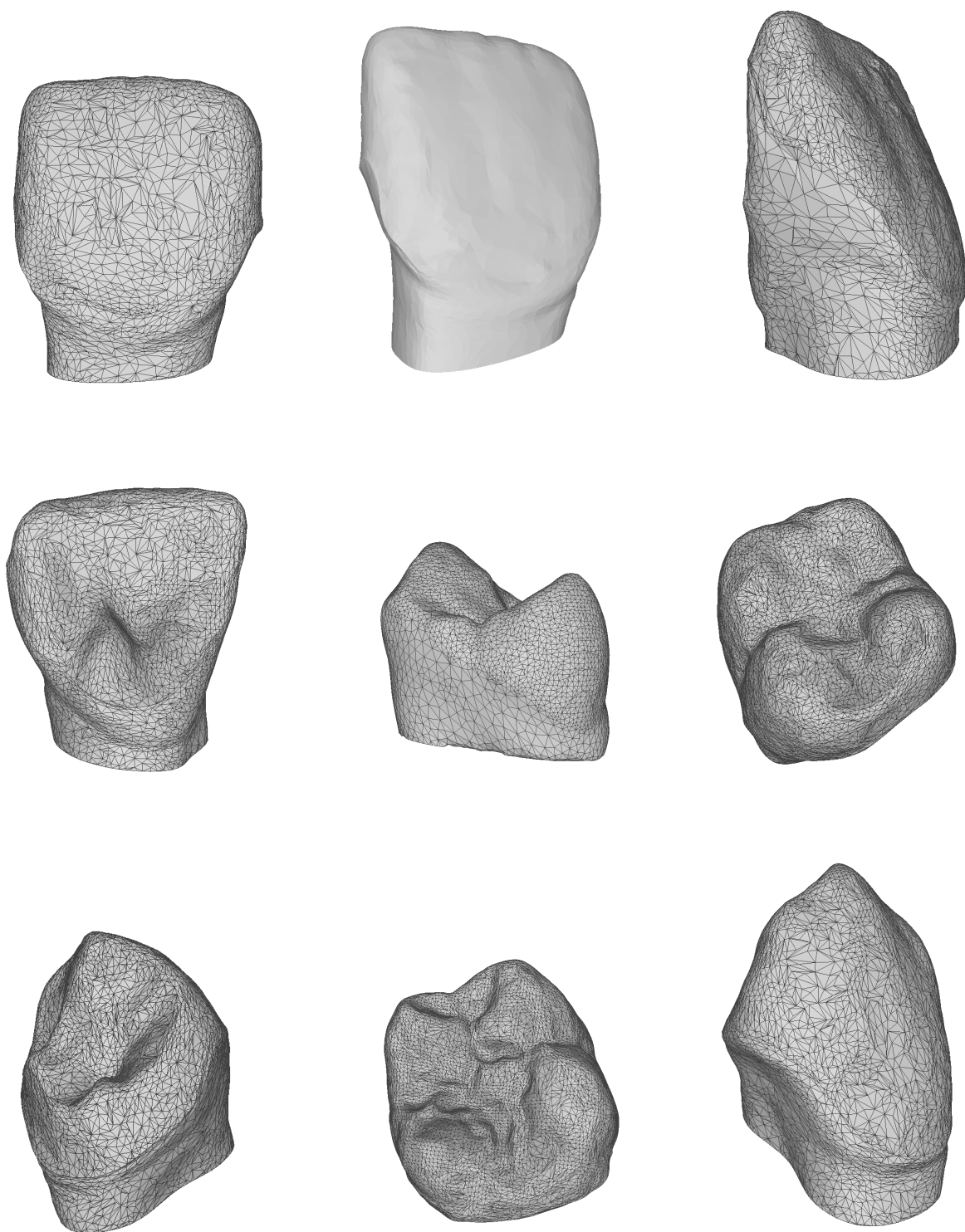
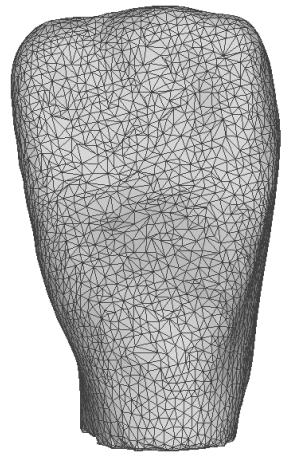
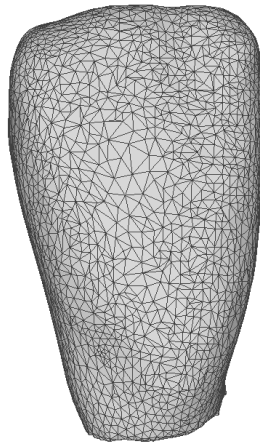
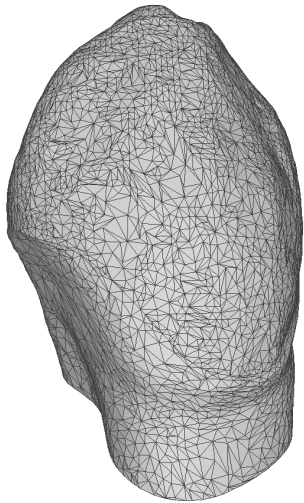


Рис. 1: Анатомия зубов

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 мы переводим их в облака точек, путем отбрасывания ребер). Примеры экземпляров из выборки можно увидеть на изображениях:





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 двумерными изображениями. К примеру, в отличие от двумерных изображений, облака точек не имеют жесткой *решетчатой структуры* (*grid-like structure*), что приводит к невозможности использования многих алгоритмов из области обработки двумерных изображений 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 перестановки (*permutation invariant*): изменение порядка точек в облаке приводит к другому облаку, описывающему абсолютно такой 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 неустойчивой 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:

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

Строго говоря CD расстояние не является метрикой, так-как не выполняется неравенство треугольника. Вычисление данной метрики для каждой пары точек происходит 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. Дифференцируемость относительно координат точек сравниваемых множеств
2. Устойчивость к малым выборкам
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 сравнения 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 величину расстояний. Чем меньше значение данной метрики, тем ближе набор A лежит к набору B [19].

4.3 Обсуждение

COV метрика показывает насколько хорошо происходит покрытие набора B объектами из набора A , а MMD метрика показывает точность (близость, 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 параллелипипедов выстроены вдоль линий параллельности (см рис. 2). Тогда для каждого объекта из первого набора (красные элементы) ближайшим к нему из второго набора (зеленые элементы) будет объект лежащий напротив него (для примера, соединены пунктирной линией). То есть, будет

выполнено полное покрытие зеленого набора красным набором. Но при этом, среднее значение MMD может быть сколь угодно большим (на рисунке она равняется 80).

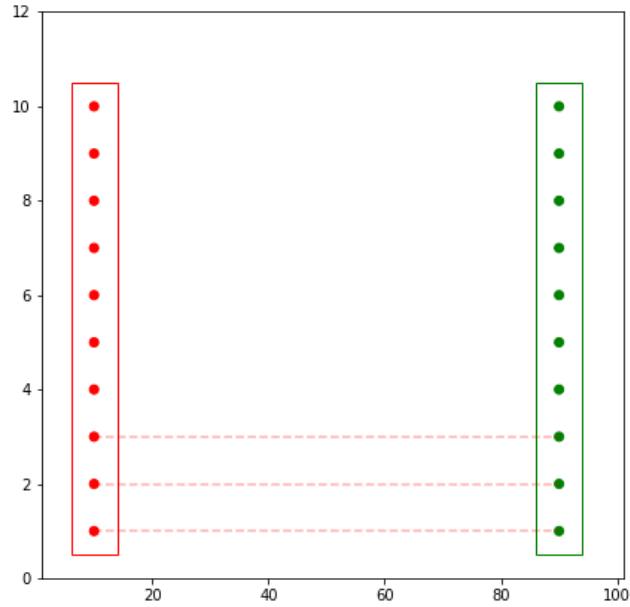


Рис. 2: Случай когда значения MMD и COV метрик велики

Аналогично, можно придумать пример, когда одновременно COV метрика и MMD метрика будут иметь маленькие значения. К примеру, объекты из набора A сосредоточены 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. 2). 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.

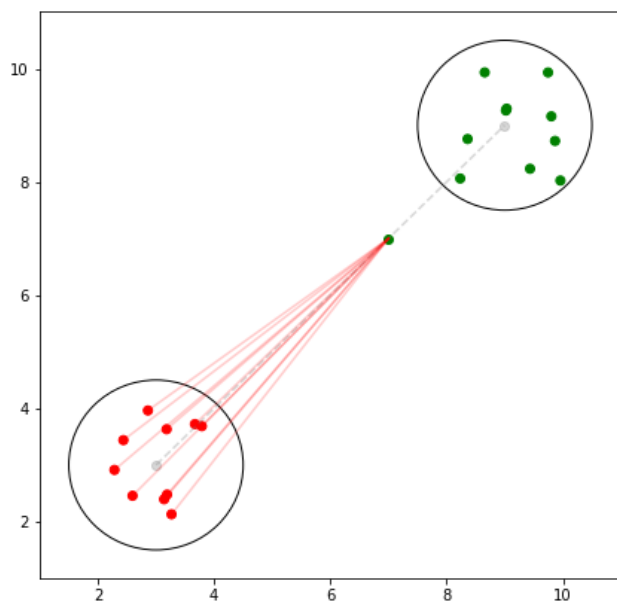


Рис. 3: Случай когда значения MMD и COV метрик малы

5 Описание алгоритма

6 Результаты

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.