**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙣 🙣 🙣 🕮 🙡 🙡 🙡

Logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**MATH FOR COMPUTER SCIENCE**

***Đề tài:* GENERATIVE ADVERSARIAL NETWORKS**

***LỚP:*** CS115.M11 – Khoa học Máy tính

***GVHD:*** TS. Lương Ngọc Hoàng

***THÀNH VIÊN:***

1. Nguyễn Hoài Nam - 20520075

2. Bùi Nguyễn Anh Trung - 20520332

3. Đoàn Phương Khanh - 20521443

4. Lê Nguyễn Minh Huy - 20521394

5. Vũ Thị Phương Linh - 20521541

*TP. Hồ Chí Minh - 12/2021*

***MỤC LỤC***

[**1. ĐẶT VẤN ĐỀ 3**](#_heading=h.30j0zll)

[**2. TỔNG QUAN ĐỀ TÀI 3**](#_heading=h.1fob9te)

[**3. MINIMAX GAME 3**](#_heading=h.3znysh7)

[**4. GENERATIVE ADVERSARIAL NETWORKS 4**](#_heading=h.2et92p0)

[**4.1. Ý tưởng hoạt động của GAN 4**](#_heading=h.tyjcwt)

[**4.2. Latent Space 4**](#_heading=h.3dy6vkm)

[**4.3. Generator and Discriminator 5**](#_heading=h.1t3h5sf)

[4.3.1. Mặt toán học 5](#_heading=h.4d34og8)

[4.3.2. Loss function 5](#_heading=h.2s8eyo1)

[**4.4. Định lý quan trọng nhất của thuật toán GAN 7**](#_heading=h.17dp8vu)

[**4.5. Thuật toán của GAN 8**](#_heading=h.3rdcrjn)

[**4.6. Mệnh đề về sự hội tụ 9**](#_heading=h.26in1rg)

[**4.7. Một số vấn đề của GAN 9**](#_heading=h.lnxbz9)

[4.7.1. Failure to Converge 9](#_heading=h.35nkun2)

[4.7.2. Vanishing Gradient 9](#_heading=h.1ksv4uv)

[4.7.3. Mode collapse 10](#_heading=h.44sinio)

[**5. TƯƠNG TRỢ THÔNG TIN 10**](#_heading=h.2jxsxqh)

[**5.1. Khái niệm 10**](#_heading=h.z337ya)

[**5.2. Tính chất 10**](#_heading=h.3j2qqm3)

[**6. INCEPTION SCORE 12**](#_heading=h.1y810tw)

[**7. THỰC NGHIỆM VÀ KẾT QUẢ 14**](#_heading=h.4i7ojhp)

[**7.1. Thực nghiệm 14**](#_heading=h.2xcytpi)

[7.1.1. Dữ liệu 14](#_heading=h.1ci93xb)

[7.1.2. Chi tiết cài đặt 14](#_heading=h.3whwml4)

[**7.2. Kết quả 14**](#_heading=h.2bn6wsx)

[7.2.1. Kết quả thực nghiệm 14](#_heading=h.qsh70q)

[7.2.2. Kết quả đánh giá 16](#_heading=h.3as4poj)

[**8. KẾT LUẬN 16**](#_heading=h.1pxezwc)

[**9. TÀI LIỆU THAM KHẢO 16**](#_heading=h.49x2ik5)

# **ĐẶT VẤN ĐỀ**

Hiện nay, Deep Learning đã tạo ra một bước tiến vượt bậc, cung cấp khả năng nhận diện hình ảnh và âm thanh, hiểu biết ngôn ngữ tự nhiên cùng nhiều tác vụ khác có thể thay thế con người.

Tuy nhiên, để huấn luyện được một mô hình tốt cùng với độ chính xác cao thì nguồn dữ liệu chính là phần đóng vai trò quan trọng và ảnh hưởng lớn nhất. Chúng ta không thể sử dụng làm dữ liệu huấn luyện và đánh giá nếu không có sự quan sát, chọn lọc và gán nhãn tỉ mỉ của con người. Điều này rất tốn thời gian và công sức.

Vì nhu cầu đó, bài toán Generator ra đời với mục đích sinh dữ liệu giả có độ tin cậy tương đối để phục vụ cho các mô hình huấn luyện. Generative Adversarial Networks (GAN)**[1]** là một phương pháp sinh dữ liệu ra đời vào năm 2014, nhưng cho đến hiện nay vẫn chứng minh được sự hiệu quả bên cạnh các phương pháp phức tạp và hiện đại hơn. Nó chính là một trong những kỹ thuật thể hiện sự sáng tạo trong trí tuệ nhân tạo.

# **TỔNG QUAN ĐỀ TÀI**

Generative Adversarial Networks lấy ý tưởng từ Minimax Game **[8],** được hiểu như là trò chơi cạnh tranh giữa hai người. Tất cả đều mong muốn cải thiện điểm số của mình để dành chiến thắng.

Trong bài báo cáo này, chúng tôi tập trung vào việc tìm hiểu, phân tích đầy đủ mặt toán học của thuật toán GAN và phương pháp đánh giá Inception Score **[3]** dưới lý thuyết tương trợ thông tin (Mutual information)**[9].** Đây là điều chưa được thực hiện đồng thời ở các bài báo liên quan.

Bên cạnh đó, chúng tôi cũng thực hiện đánh giá so sánh giữa hai phương pháp GAN và Deep Convolutional GAN (DCGAN)**[2]** trên bộ dữ liệu Fashion MNIST**[4**] và ANIME**[10]** để đưa kết luận về độ hiệu quả cùng với những vấn đề chưa được khắc phục của GAN.

# **MINIMAX GAME**

Trong lý thuyết trò chơi, giả sử người chơi chọn chiến lược và những người còn lại là chiến lược . Gọi là hàm lợi ích của người chơi trong chiến lược , Minimax của trò chơi được định nghĩa

Tương tự ta cũng có Maximin được định nghĩa bởi:

**Định lý Minimax của Von Neumman**

Tiếp theo, ta sẽ chứng minh Minimax theo Von Neumman**[11]** được định nghĩa như sau:

Cho là ma trận đại diện cho ma trận chi phí của hai người chơi. Khi đó trò chơi sẽ luôn tồn tại một giá trị và một cặp chiến lược hỗn hợp tối ưu cho hai người chơi. Với là các vector xác xuất tương ứng cho mỗi người chơi khi ra quyết định, ta có các cặp chiến lược hỗn hợp

Với mỗi cặp chiến lược ta định nghĩa:

Trong đó là chi phí tại vị trí trong .

Ta gọi một cặp chiến lược hỗn hợp là một cặp cân bằng (equilibrium point) cho một trò chơi Minimax game hai người thoả mãn:

với mọi , và với mọi

Điều này tương đương với:

Khi cân bằng xảy ra, xác suất thắng của hai người chơi là như nhau. Do đó ta cần chứng minh sẽ luôn tồn tại cặp cân bằng .

# **GENERATIVE ADVERSARIAL NETWORKS**

## **Ý tưởng hoạt động của GAN**

Ý tưởng cơ bản của GAN là thiết lập một trò chơi hai người. Một trong số đó là Generator (module G), Generator tạo ra các mẫu (samples) từ không gian ngầm (latent space) có cùng phân phối với dữ liệu huấn luyện. Người chơi còn lại là Discriminator (module D), người này đánh giá các samples và xác định xem các samples là thật hay giả.

## **Latent Space**

Trong thống kê, các biến tiềm ẩn (latent variables) là các biến không được quan sát (thông qua phép đo) trực tiếp nhưng được suy ra (thông qua mô hình toán học) từ các biến khác đã được quan sát.

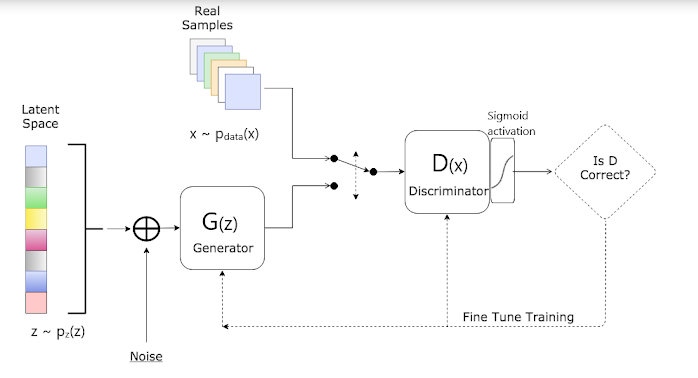
Khi đó, latent space **[12]** là không gian của latent variables, đề cập đến một không gian đa chiều trừu tượng chứa các đặc trưng mà chúng ta không thể giải thích trực tiếp.

Với Deep Learning, ý tưởng của Latent Space là quan trọng vì nó giúp học các đặc trưng của dữ liệu và đơn giản hoá các biểu diễn của dữ liệu cho mục đích tìm kiếm các mẫu.

## **Generator and Discriminator**

**Một số ký hiệu sử dụng:**

* lần lượt là không gian miền dữ liệu thật, dữ liệu giả và dữ liệu nhiễu
* : Generator
* : Discriminator
* : Tham số Generator
* : Tham số của Discriminator
* : phân phối của dữ liệu thật
* : phân phối của dữ liệu nhiễu
* : phân phối của dữ liệu giả tạo ra bởi G



***Hình 1:*** Kiến trúc mô hình GAN

### **Mặt toán học**

* là một hàm khả vi đi từ Latent Space vào , được đại diện bởi một multilayer perceptron với các tham số
* là một hàm xác xuất đi từ không gian hoặc , đại diện bởi một multilayer perception với các tham số

### **Loss function**

Loss function trong GAN

Rõ ràng mục tiêu của Discriminator là maximize giá trị đối với những điểm dữ liệu và minimize đối với những điểu dữ liệu hay maximize

Ta thấy rằng, .

Khi đó, mục tiêu của là maximize tổng:

Ký hiệu tối ưu là , với cho trước.

Giả sữ đã có , do mục tiêu của và là ngược nhau, ta tìm để minimize , lúc này đã tối ưu được ký hiệu là

Ta đi chứng minh bài toán tối ưu có nghiệm duy nhất thoả điều kiện .

Theo định lý Radon-Nikodym **[13],** ta có:

Ta có , để tìm maximun , ta thực hiện như sau:

Thay biểu thức của vào ta được:

(luôn đúng)

Vậy là maximum của tích phân trên, hay

Khi đó, nếu gọi là mục tiêu huấn luyện dữ liệu giả (virtual training criterion)

## **Định lý quan trọng nhất của thuật toán GAN**

Global minimum của đạt được khi và chỉ khi .

Khi đó

Trước khi chứng minh, ta nhắc lại định nghĩa độ đo Kullback-Leinler divergence (KL) **[14]** và Jensen-Shannon divergence (JSD) **[15]**

Với là các phân phối xác suất trên trên cùng một không gian xác xuất . Độ đo Kullback-Leibler divergence từ vào được định nghĩa:

Độ đo Jensen-Shannon divergence được xác định là:

Đây là độ đo không âm nên .

***Chứng minh:***

Ta biến đổi biểu thức như sau:

Dấu xảy ra khi và chỉ khi .

Khi đó

## **Thuật toán của GAN**

**for** *number of training iterations* **do**

**for** *k steps* **do**

* Sử dụng *m* mẫu dữ liệu được tạo ra bởi phân phối
* Sử dụng *m* mẫu dữ liệu được tạo ra bởi phân phối
* Cập nhập tham số discriminator tăng theo stochastic gradient

**end for**

* Sử dụng *m* mẫu dữ liệu được tạo ra bởi phân phối
* Cập nhập tham số generator giảm theo stochastic gradient

**end for**

## **Mệnh đề về sự hội tụ**

Nếu và đủ khả năng, và ở mỗi bước của thuật toán trên, Discriminator được tối ưu với cho trước, và được cập nhập để cải tiến

Khi đó

## **Một số vấn đề của GAN**

### **Failure to Converge**

Một trong những vấn đề đầu tiên của GAN đó chính là khó khăn trong việc hội tụ.

Công việc chính của GAN chính là đi tối ưu cả hai module và , đây là bài toán tối ưu hai biến nên rất khó đạt được hội tụ toàn cục.

Giả sử xét và thì điểm cân bằng sẽ là như hình vẽ bên dưới

Chart, line chart

Description automatically generated

***Hình 2***: Sự biến thiên của các hàm số

### **Vanishing Gradient**

Nếu module Discriminator hoạt động quá tốt, thì điều nảy dẫn đến việc module Generator xảy ra hiện tượng vanishing gradient

Cụ thể tại những step đầu tiên thì Generator hoạt động chưa tốt, khi đó sẽ tạo ra sự khác biệt lớn giữa dữ liệu fake (giả) và real (thật). Điều này làm cho Discriminator hoạt động tốt và dẫn đến sự hội tụ nhanh chóng. Tuy nhiên Generator sẽ gặp thất bại trong quá trình huấn luyện, vì nó không thể học thêm được gì khi xảy ra hiện tượng vanishing gradient.

### **Mode collapse**

Hiện tượng Mode collapse xảy khi Generator phát hiện điểm mù của Discriminator. Đây là điểm mà Discriminator không thể phân biệt được thật/giả, vì thế Generator sẽ luôn cố học và tạo ra dữ liệu dựa vào điểm mù đó. Gây ra hiện tượng mode collapse, làm cho tập dữ liệu đầu ra giống hệt nhau.

# **TƯƠNG TRỢ THÔNG TIN**

## **Khái niệm**

Tương trợ thông tin (Mutual information) **[9]** là khái niệm cơ bản trong lý thuyết thông tin định lượng "lượng thông tin" dự kiến ​​có trong một biến ngẫu nhiên.

Cụ thể hơn, nó định lượng "lượng thông tin" (theo các đơn vị như shannons (bit), nats hoặc hartleys) thu được về một biến ngẫu nhiên bằng cách quan sát biến ngẫu nhiên kia.

Cho (X,Y) là cặp biến ngẫu nhiên các giá trị thuộc không gian . Phân phối đồng thời của chúng là và phân phối lề lần lượt là . Khi đó, tương hỗ thông tin được định nghĩa là:

## **Tính chất**

***Tính chất 1:*** Khi hai biến X,Y độc lập thì .

Do đó .

Tức là việc biết X không hỗ trợ gì cho việc tìm hiểu Y và ngược lại. Do đó hỗ trợ thông tin của chúng bằng 0

***Tính chất 2:*** Tính không âm :

Sử dụng bất đẳng thức có thể có thể chứng minh rằng không âm

Chứng minh: Theo bổ đề bất đẳng thức Jensen , với bất kỳ hàm lồi f(x) ta đều có:

Ta chỉ cần chứng minh với hai hàm phân phối và được định nghĩa là không âm thì mutual information cũng không âm

(do hàm log là hàm lõm)

***Tính chất 3:*** Tính đối xứng

Thế nên

***Tính chất 4:*** Biểu diễn theo phân rã Kullback-Leibler

]

***Tính chất 5:*** Biểu diễn qua Entropy điều kiện và Entropy đồng thời

=

(\*)

***Trong đó*** : là entropy

***Theo trực quan:***

+ Entropy được coi là thước đo mức độ đảm bảo của một biến ngẫu nhiên.

+ là thước đo của lượng thông tin không nói về . Đây là "mức độ đảm bảo về sau khi được biết"

+ Và do đó, (\*) có thể được đọc là "mức độ chắc chắn trong , trừ đi mức độ chắc chắn trong khi biết "

+ Tương đương với "độ chắc chắn trong khi được loại bỏ thông tin về ".

+ Điều này chứng thực ý nghĩa trực quan của tương trợ là lượng thông tin (nghĩa là giảm độ không chắc chắn) mà việc biết một trong hai biến cung cấp về biến kia.

# **INCEPTION SCORE**

Inception Score (IS) **[3]** là một thang đo để tự động đánh giá chất lượng của các mô hình tạo hình ảnh.

IS sử dụng một mạng học sâu được huấn luyện trước trên tập dữ liệu cho trước với kích thước***N***và tính toán thống kê kết quả đầu ra của mạng khi áp dụng cho các hình ảnh được sinh ra bởi Generator. Công thức IS được cho bởi:

***Trong đó:***

* nghĩa là x là một mẫu dữ liệu ảnh được từ
* là độ đo Kullback-Leibler divergence **[14]** từ vào
* là phân phối có điều kiện
* = là phân phối cận biên ( phân phối lề )

Hàm exp nhằm giúp hỗ trợ tính toán dễ dàng hơn. Chúng ta có thể bỏ qua bằng cách sử dụng .

Phương pháp IS nhằm hệ thống hóa hai tính chất mong muốn của một mô hình chung thành một thước đo:

* Hình ảnh được tạo ra phải chứa các vật thể rõ ràng (tức là hình ảnh sắc nét chứ không phải mờ) hoặc phải có entropy thấp. Nói cách khác, mạng khởi động nên tin chắc rằng có một đối tượng duy nhất trong hình ảnh.
* Thuật toán tổng hợp phải tạo ra độ đa dạng cao của hình ảnh từ tất cả các lớp khác nhau trong mạng học sâu được huấn luyện ban đầu, tức là phải có entropy cao.

Nếu cả hai đặc điểm này đều được thỏa mãn bởi một mô hình tổng hợp, thì chúng ta mong đợi Kullback-Leibler divergence lớn giữa các phân phối p (y) và p (y | ), dẫn đến IS lớn.

Ta thực hiện biến đổi và đánh giá như sau:

1. ***Biến đổi trên biểu thức đánh giá :***

( Theo tính chất 3 của tương trợ thông tin )

Điều này có nghĩa là :

Ta có : :

Do đó : và để IS(G) đạt được giá trị cực đại thì :

+ khi và chỉ khi là một phân phối chuẩn

+ xảy ra khi với duy nhất một i và các thì

1. ***Thực hiện phương pháp đánh giá:***

Với các mẫu . Ta trước tiên tính phân phối lề :

# **THỰC NGHIỆM VÀ KẾT QUẢ**

## **Thực nghiệm**

### **Dữ liệu**

Về phần dữ liệu, chúng tôi thực hiện huấn luyến trên tập dữ liệu Fashion MNIST**[4]** với 10 lớp đối tượng, mỗi lớp với 6000 ảnh trong tập huấn luyện và 1000 ảnh trong tập kiểm tra. Ở thực nghiệm với GAN, chúng tôi chỉ sử dụng tập kiểm tra với 60000 ảnh với kích thước mỗi ảnh là . Bên cạnh đó chúng tôi của thực hiện thử nghiệm trên tập dữ liệu ANIME **[10]** để xem xét sự khác biệt khi thực hiện huấn luyện ảnh màu.

### **Chi tiết cài đặt**

Chi tiết phần cài đặt cho cả mô hình GAN và DCGAN, chúng tôi huấn luyện với 200 epochs và tốc độ học (learning rate) là . Thuật toán tối ưu sử dụng là Adam và đánh giá chi phí mỗi bước huấn luyện thông qua Binary Cross Entropy Loss.

Với kích cỡ chiều khởi tạo của không gian nhiễu, chúng tôi mặc định là 100. Kích thước ảnh giả được tạo ra bởi module Generator sẽ là , với là số kênh màu tương ứng khi sử dụng GAN và khi sử dụng DCGAN.

## **Kết quả**

### **Kết quả thực nghiệm**

*Với mô hình GAN*, sau quá trình huấn luyện chúng tôi sinh được một tập dữ liệu giả, với sự quan sát định tính, chúng tôi đánh giá rằng dữ liệu được sinh ra còn kém chất lượng và chưa có sự đa dạng nhất định.

*Với mô hình DCGAN,* với lần huấn luyện đầu tiên chúng tôi nghĩ rằng mô hình của mình đã gặp phải hiện tượng mode collapse khi trong 10 epochs đầu, module Generator chỉ sinh ra toàn ảnh cùng một đối tượng (cụ thể là hình ảnh giày) (Hình vẽ). Vì thế, chúng tôi đã phải thực hiện huấn luyện lại lần hai và thu được dữ liệu sinh ra đa dạng và có chất lượng tốt hơn.

Bên cạnh đó, chúng tôi cũng thực hiện huấn luyện mô hình DCGAN trên tập dữ liệu ANIME để quan sát kết quả.

|  |  | Ảnh có chứa văn bản, thiết bị điện tử  Mô tả được tự động tạo |
| --- | --- | --- |
| *a) Hình ảnh gốc* | *b) Hình ảnh khởi tạo của GAN* | *c) Hình ảnh khởi tạo của DCGAN* |
|  |  | A picture containing text, flock, wire, light  Description automatically generated |
| *d) Hình ảnh sinh ra bởi GAN* | *e) Hình ảnh sinh ra bởi DCGAN* | *f) Hình ảnh hiện tượng mode collapse bởi DCGAN* |

***Hình 3:*** Một số hình ảnh dữ liệu sinh ra trong quá trình huấn luyện

|  | A collage of a person's face  Description automatically generated with medium confidence |
| --- | --- |
| *a) Hình ảnh gốc* | *b) Hình ảnh sinh ra bởi DCGAN* |

***Hình 4:*** *Kết quả mô hình DCGAN trên bộ dữ liệu ANIME*

### **Kết quả đánh giá**

Ở bước đánh giá, chúng tôi chọn ngẫu nhiên 1000 ảnh trên mỗi tập dữ liệu *(bao gồm cả dữ liệu gốc*) và thực hiện đánh giá bằng phương pháp Inception Score. Nhận được kết quả tốt nhất với dữ liệu được sinh ra bởi DCGAN *(không bao gồm dữ liệu gốc).*

| **Dataset**  **(Fashion MNIST)** | **Inception Score** | |
| --- | --- | --- |
| **AVG** | **STD** |
| ***Original data*** | *4.6558* | *0.2116* |
| ***Generative by GAN model*** | *3.1660* | *0.1673* |
| ***Generative by DCGAN model*** | ***3.4590*** | ***0.1284*** |
| ***Generator by DCGAN model (Mode collapse)*** | *2.0260* | *0.0895* |

***Bảng 1:*** Kết quả đánh giá mô hình GAN và DCGAN   
trên tập dữ liệu Fashion MNIST

# **KẾT LUẬN**

Qua bài báo cáo này, chúng tôi đã thực hiện tìm hiểu và phân tích các mặt toán học của thuật toán Generative Adversarial Networks cùng phương pháp đánh giá Inception Score. Chúng tôi cũng đã thực hiện xây dựng thành công hai mô hình GAN và DCGAN và thực hiện đánh giá trên tập dữ liệu Fashion MNIST.

Thông qua việc thử nghiệm, tìm hiểu và nghiên cứu này đã giúp nhóm chúng tôi hiểu sâu hơn các vấn đề gặp phải của mô hình GAN. Trong tương lai, chúng tôi vẫn sẽ dự định tiếp tục nghiên cứu đề tài này sâu hơn bằng cách tìm hiểu các biến thể của mô hình GAN và các phương pháp giải quyết nhược điểm của mô hình GAN.

# **TÀI LIỆU THAM KHẢO**

*[1] Generative Adversarial Networks, Ian J.Goodfelow, arXiv:1406.2661, 2014.*

*[2] Unsupervised Representation Learning with Deep Convolutional Generative. Adversarial Networks, Alec Radford,ICLR , arXiv:1511.06434, 2016.*

*[3] A Note on the Inception Score, Shane Barrat, arXiv:1801.01973, 2018.*

*[4] Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, Han Xiao, arXiv:1708.07747, 2017.*

*[5]*[*Convergence Problems with Generative Adversarial Networks*](https://arxiv.org/pdf/1806.11382.pdf)*, S. A. Barnett.*

*[6]* [*Math behind gans*](https://towardsdatascience.com/the-math-behind-gans-generative-adversarial-networks-3828f3469d9c)*, Mayank Vadsola Mayank Vadsola.*

*[7]* [*Understanding latent space in machine learning*](https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d)*, Ekin Tiu.*

*[8]* [*Minimax – Wikipedia*](https://vi.wikipedia.org/wiki/Minimax)*.*

*[9]* [*Mutual information - Scholarpedia*](http://www.scholarpedia.org/article/Mutual_information#:~:text=Mutual%20information%20is%20one%20of,variable%20given%20knowledge%20of%20another.)

*[10]* [*Anime Face Dataset | Kaggle*](https://www.kaggle.com/splcher/animefacedataset)

*[11] John von Neumann – Wikipedia*

*[12]* [*Latent space - Wikipedia*](https://en.wikipedia.org/wiki/Latent_space)

*[13]* [*Radon–Nikodym theorem - Wikipedia*](https://en.wikipedia.org/wiki/Radon%E2%80%93Nikodym_theorem)

*[14]* [*Kullback–Leibler divergence - Wikipedia*](https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence)

*[15]* [*Jensen–Shannon divergence - Wikipedia*](https://en.wikipedia.org/wiki/Jensen%E2%80%93Shannon_divergence)