





Machine Learning for Cyber Security (CS-602)

L#06

**Introduction – Generative vs Discriminative
Classifiers**

By
Dr Sunita Dhavale

Syllabus

- Data Analytics Foundations: R programming, Python Basics -Expressions and Variables, String Operations, Lists and Tuples, Sets, Dictionaries Conditions and Branching, Loops, Functions, Objects and Classes, Reading/Writing files, Handling data with Pandas, Scikit Library, Numpy Library, Matplotlib, scikit programming for data analysis, setting up lab environment, study of standard datasets. Introduction to Machine Learning- Applications of Machine Learning, Supervised, unsupervised classification and regression analysis
- Python libraries suitable for Machine Learning Feature Extraction. Data pre-processing, feature analysis etc., Dimensionality Reduction & Feature Selection Methods, Linear Discriminant Analysis and Principal Component Analysis, tackle data class imbalance problem

Syllabus

- Supervised and regression analysis, Regression, Linear Regression, Non-linear Regression, Model evaluation methods, Classification, K-Nearest Neighbor, Naïve Bayes, Decision Trees, Logistic Regression, Support Vector Machines, Artificial Neural Networks, Model Evaluation. Ensemble Learning, Convolutional Neural Networks, Spectral Embedding, Manifold detection and Anomaly Detection
- Unsupervised classification K-Means Clustering, Hierarchical Clustering, Density-Based Clustering, Recommender Systems-Content-based recommender systems, Collaborative Filtering, machine learning techniques for standard dataset, ML applications, Case studies on Cyber Security problems that can be solved using Machine learning like Malware Analysis, Intrusion Detection, Spam detection, Phishing detection, Financial Fraud detection, Denial of Service Detection.

Text/Reference Books

1. Building Machine Learning Systems with Python – Willi Richert, Luis Pedro Coelho
 2. Alessandro Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies
Publication date :Aug 2, 2019, Packt, ISBN-13, 9781789804027
 3. Machine Learning: An Algorithmic Perspective – Stephen Marsland
 4. Sunita Vikrant Dhavale, “Advanced Image-based Spam Detection and Filtering Techniques”, IGI Global, 2017
 5. Soma Halder , Sinan Ozdemir, Hands-On Machine Learning for Cybersecurity: Safeguard your system by making your machines intelligent using the Python ecosystem, By
Publication date : Dec 31, 2018, Packt, ISBN-13 :9781788992282
-
1. Stuart Russell, Peter Norvig (2009), “Artificial Intelligence – A Modern Approach”, Pearson Elaine Rich & Kevin Knight (1999), “Artificial Intelligence”, TMH, 2nd Edition
 2. NP Padhy (2010), “Artificial Intelligence & Intelligent System”, Oxford
 3. ZM Zurada (1992), “Introduction to Artificial Neural Systems”, West Publishing Company
 4. Research paper for study (if any) – White papers on multimedia from IEEE/ACM/Elsevier/Spinger/ Nvidia sources.

Lab assignments

1	Python Programming part-1
2	Python Programming part-2
3	Study and Implement Linear Regression Algorithm for any standard dataset like in cyber security domain
4	Study and Implement KMeans Algorithm for any standard dataset in cyber security domain
5	Study and Implement KNN for any standard dataset in cyber security domain
6	Study and Implement ANN for any standard dataset in cyber security domain
7	Study and Implement PCA for any standard dataset in cyber security domain
8	Case Study: Use of ML along with Fuzzy Logic/GA to solve real world Problem in cyber security domain
9	Mini assignment: Apply ML along with PSO/ACO to solve any real world problem in cyber security domain
10	ML Practice Test – 1 Quiz

Defence Institute of Advanced Technology

School of Computer Engineering & Mathematical Sciences

SEMESTER-I TIME TABLE (AUTUMN 2024)[§]

PROGRAMMES: (I) CS [M.TECH IN CYBER SECURITY] (II) AI [M.TECH CSE (ARTIFICIAL INTELLIGENCE)]

BATCH: 2024-2026

Lecture Day	L1 0900-1000	L2 1000-1100	L3 1100-1200	L4 1200-1300		L4 1400-1500	L4 1500-1600	L4 1600-1700	L4 1700-1800
Monday	CE-602 (AI) CS-602 (CS)	CE-604 (AI) CS-603 (CS)	CE-601 (AI) CS-604 (CS)	CE-601 (AI) LAB CS-603 (CS)	Lunch Break 1300-1400	LAB CE-601 (AI) LAB CS-602 (CS)		AM607	
Tuesday	CE-603 (AI) LAB CS-603 (CS)	CE-602 (AI) CS-602 (CS)	CE-601 (AI) CS-605 (CS)	CE-604 (AI) CS-604 (CS)		PGC 601		AM607	
Wednesday	CS-605 (CS)	CE-603 (AI) CS-602 (CS)	CE-602 (AI) CS-603 (CS)	CE-604 (AI) CS-604 (CS)		CE-605(AI) LAB CS-605 (CS)	LAB CS-605 (CS)	AM607	
Thursday	LAB CE-604 (AI) CS-603 (CS)	LAB CE-604 (AI) CS-605 (CS)	LAB CE-602 (AI) CS-601 (CS)	CE-603 (AI) CS-601 (CS)		PGC 601		AM607	
Friday	LAB CE-603 (AI) LAB CS-601 (CS)		LAB CE-602 (AI) CS-601 (CS)	LAB CS-604 (CS)		CE-605(AI) LAB CS-604 (CS)	CE-605(AI)	LAB CE-605(AI)	

COURSE CODE & COURSE NAME		FACULTY
Programme: CS [M.Tech in Cyber Security] Classroom: Arjun	Programme: AI [M.Tech CSE (Artificial Intelligence)] Classroom: Kaveri	
CS-601 Data Security & Privacy	CE-601 Responsible Artificial Intelligence;	MJN: Dr. Manisha J. Nene
CS-602 ML for Cyber Security	CE-604 Practical Machine Learning;	SVD: Dr. Sunita V. Dhavale
CS-605 Network and Cloud Security	CE-602 Intelligent Algorithms	CRS: Prof. CRS Kumar
CS-604 Advanced System Security	-----	DVV: Dr. Deepti V. Vidyarthi
CS-603 Applied Cryptography	-----	AM: Dr. Arun Mishra
-----	CE-603 Deep Neural Network;	US: Dr. Upasna Singh
-----	CE-605 Mathematics for ML;	Unit-2: Dr Upasna, Unit 4: Dr Sunita, Unit3:MJM, Unit 1: Faculty To be Nominated
AM-607 Mathematics for Engineers	AM-607 Mathematics for Engineers	OO/DS/DP: Dr Odellu O., Dr Dasari S., Dr. Debasis P.
PGC-601 Research Methodology	PGC-601 Research Methodology	Common Subject for All

§ TENTATIVE T.T. SUBJECT TO CHANGE

Program Coordinator,
M.Tech (CS & AI), Batch 2024-26

Director, SoCE&MS

Story

- Two students: A and B
- Student A has a special character whereas he can learn everything in depth and understands every little detail about a subject. Once he's grasped it, he never forgets it. But, this is cumbersome, if there's a lot to learn under said topic.
- Student B can only learn the differences between what he saw. i.e. learns by learning the differences.

Story

- One fine day, they visit small zoo that has only two kinds of animals - lion and an elephant.
- After they came out of the zoo, Teacher asked them to identify one animal if it “**is a lion or an elephant?**”
- Student A draw the image of lion and elephant and compared both the images with the animal standing before. Based on the **closest match** of image & animal, he answered: “The animal is Lion”.
- Student B knows only the differences, based on **different properties learned**, he answered: “The animal is a Lion”.
- Here, we can see **both of them is finding the kind of animal, but the way of learning and the way of finding answer is entirely different.**

Story

- Here, we can see
- both of them is finding the kind of animal,
- but the way of learning and the way of finding answer is entirely different.

Example : classification problem to decide if an **email is spam/not spam based on words present** in email

- Labels: $Y=y$, and
- Features: $X=\{x_1, x_2, \dots, x_n\}$
- Therefore, the joint distribution of the model can be represented as
- $p(Y,X) = P(y,x_1,x_2,\dots,x_n)$
- Goal : to estimate the probability of spam email i.e., $P(Y=1 | X)$

Approach 1 like student A

- focus on the distribution of individual classes in a dataset
- Learn $p(x,y)$ and makes prediction to get $p(y|x)$ and picks most likely label y
- Assume some functional form for the probabilities such as $\mathbf{P(Y)}$, $\mathbf{P(X|Y)}$
- With the help of training data, we estimate the parameters of $\mathbf{P(X|Y)}$, $\mathbf{P(Y)}$
- E.g. Use the Bayes theorem to calculate the posterior probability $\mathbf{P(Y|X)}$
- first estimate the prior probability $\mathbf{P(Y)}$ and likelihood probability $\mathbf{P(X|Y)}$ with the help of the training data and then use

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \Rightarrow P(Y|X) = \frac{P(Y) \cdot P(X|Y)}{P(X)}$$

Approach 2 like student B

- estimating a function $f: X \rightarrow Y$, or model posterior probability $P(Y|X)$ directly
- Assume some functional form for $P(Y|X)$
- With the help of training data, we estimate the parameters of $P(Y|X)$
- separate classes instead of modeling the conditional probability and don't make any assumptions about the data points.
- focus on modeling the decision boundary between classes in a classification problem.

Story conclusion

- Approach-1:
- A Generative Model explicitly models the **actual distribution of each class**.
- to determine the language of a text document - We can learn each language and then determine the language. This is how generative models work.

Story conclusion

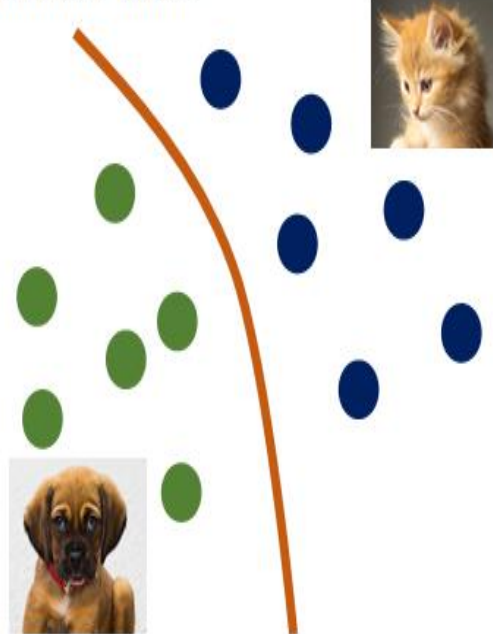
- Approach-2:
- we can learn just the linguistic differences and common patterns of languages without actually learning the language. This is the discriminative approach. In this case, we don't speak any language.
- A Discriminative model models the **decision boundary between the classes.**

Story conclusion

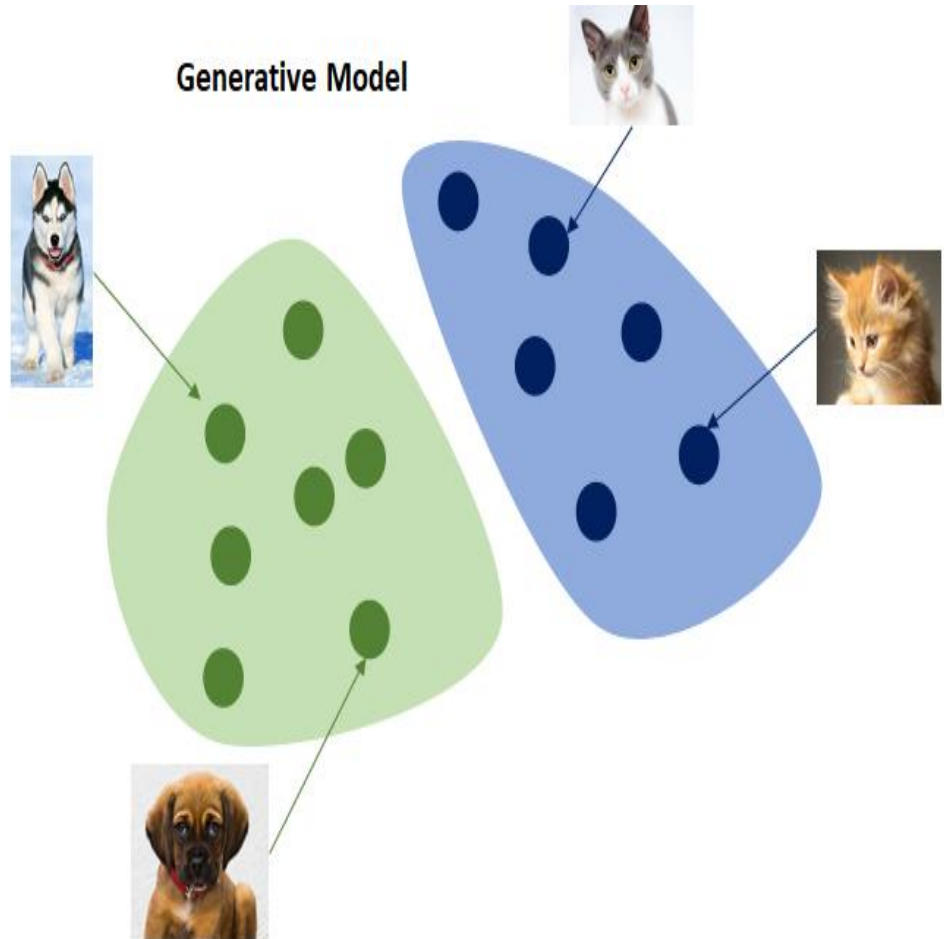
- Machine learning models can be classified into two types of models
 - **Generative (Approach-1)** models
 - **Discriminative (Approach-2)** models
- In final, both of them is predicting the conditional probability $P(\text{Animal} \mid \text{Features})$.
- But Both models learn different probabilities.
- generative model focuses on the distribution of a dataset to return a probability for a given example.
- Discriminative model makes predictions on the unseen data by learning the boundaries between classes or labels in a dataset.

Discriminative and Generative models

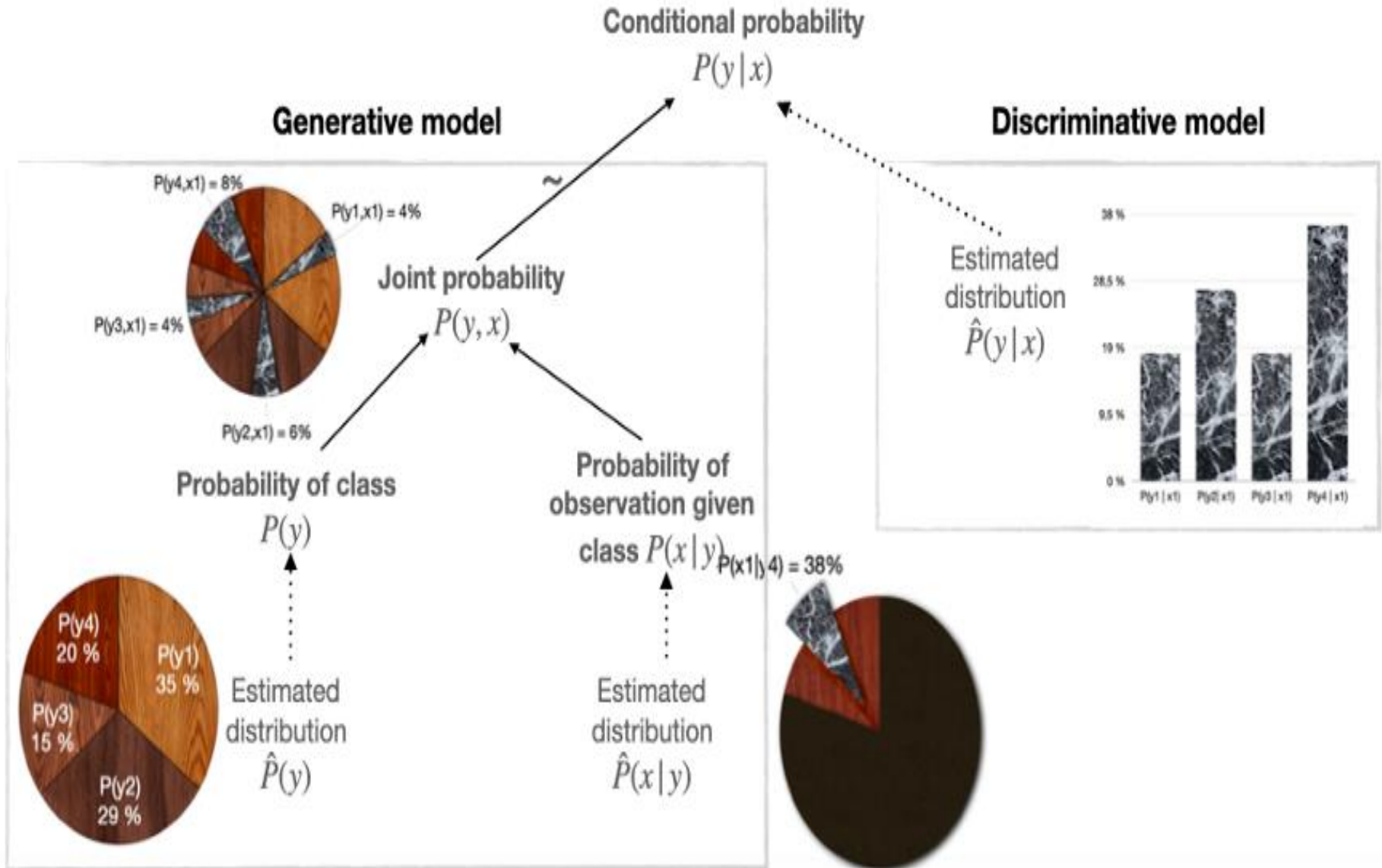
Discriminant Model



Generative Model



Discriminative and Generative models



Generative classifiers

- samples of training data are a class assumed to come from a probability density function that is the class conditional pdf.
- can generate new data instances.
- Can be used in unsupervised machine learning
- E.g. used in Probability and Likelihood estimation, Modeling data points, describe the phenomenon in data, distinguish between classes based on these probabilities
- presence of outliers in the dataset->affects performance

Generative classifiers

- Examples:
- Naïve Bayes, Bayesian networks, Markov random fields, Hidden Markov Models (HMMs), Latent Dirichlet Allocation (LDA), Generative Adversarial Networks (GANs), Autoregressive Model

Discriminative classifiers

- no such assumption of data being drawn from an underlying pdf.
- Maximum likelihood estimation is often used to estimate the parameters of the discriminative model, such as the coefficients of a logistic regression model or the weights of a neural network.
- focus on modeling a direct solution. For example, the logistic regression algorithm models a decision boundary. Then it decides on the outcome of an observation based on where it stands relative to the decision boundary.
- these models are not capable of generating new data points.
- Therefore, the ultimate objective of discriminative models is to separate one class from another.
- it models the decision boundary by adopting the gradient descent like techniques
- discriminative models are more robust to outliers.
- start with initial weights that the define the decision surface. Then update the weights based on some optimization criterion

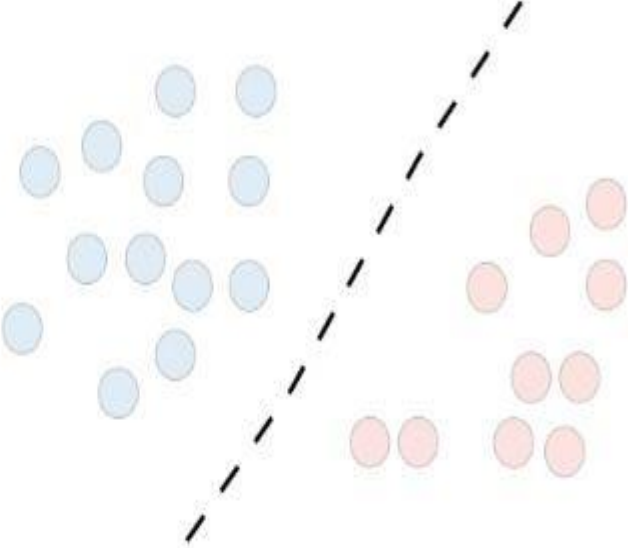
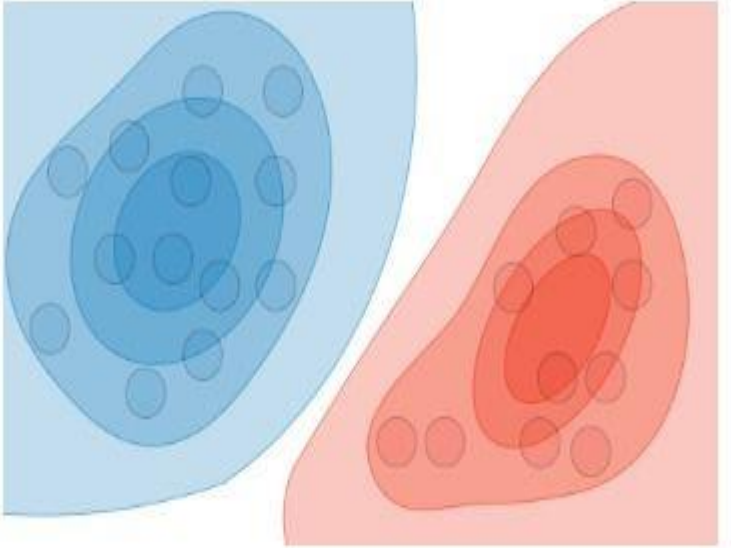
Discriminative classifiers

- E.g. ANN,
- SVM,
- Nearest neighbor,
- Conditional Random Fields (CRFs),
- Decision Trees
- Random Forest

Questions??

- Which type of ML models draw boundaries in the data space?
- Which type of ML models try to model how data is placed throughout the space?
- Which type of ML models focuses on explaining how the data was generated?
- Which type of ML models focuses on predicting the labels of the data?

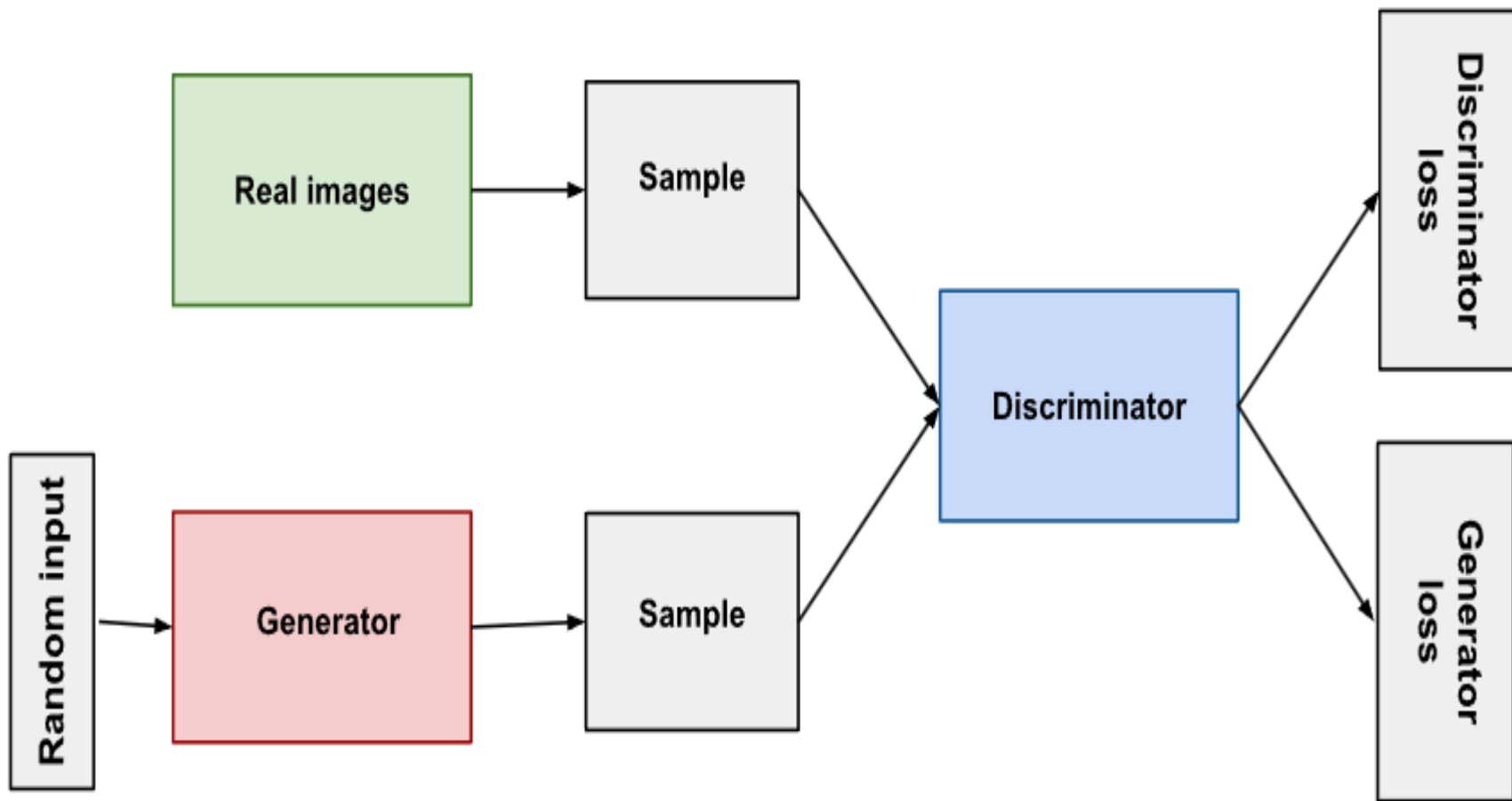
Summary

	Discriminative model	Generative model
Goal	Directly estimate $P(y x)$	Estimate $P(x y)$ to then deduce $P(y x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration	 A scatter plot illustrating a discriminative model. It shows two classes of data points: blue circles on the left and red circles on the right. A dashed black line represents the decision boundary separating the two classes.	 A scatter plot illustrating a generative model. It shows two classes of data points: blue circles on the left and red circles on the right. Each class is enclosed by a shaded region representing its probability distribution. The blue region is on the left and the red region is on the right, with some overlap between them.
Examples	Regressions, SVMs	GDA, Naive Bayes

Generative adversarial network (GAN)

- has two parts:
- The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.
- When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake.
- As training progresses, the generator gets closer to producing output that can fool the discriminator.
- Finally, if generator training goes well, the discriminator gets worse at telling the difference between real and fake. It starts to classify fake data as real, and its accuracy decreases.
- Both the generator and the discriminator are neural networks. The generator output is connected directly to the discriminator input. Through backpropagation, the discriminator's classification provides a signal that the generator uses to update its weights.

GAN



GAN Applications

- **Image-to-Image Translation GANs**- take an image as input and map it to a generated output image with different properties. For example, we can take a mask image with blob of color in the shape of a car, and the GAN can fill in the shape with photorealistic car details.

CycleGANs learn to transform images from one set into images that could plausibly belong to another set. For example, a CycleGAN produced the righthand image below when given the lefthand image as input. It took an image of a horse and turned it into an image of a zebra.



GAN Applications

- **Text-to-Image Synthesis** GANs take text as input and produce images that are plausible and described by the text. For example, the flower image below was produced by feeding a text description to a GAN.

Super-resolution GANs increase the resolution of images, adding detail where necessary to fill in blurry areas. For example, the blurry middle image below is a downsampled version of the original image on the left.

GANs have been used for the *semantic image inpainting* task. In the inpainting task, chunks of an image are blacked out, and the system tries to fill in the missing chunks.

GANs can produce speech from text input.

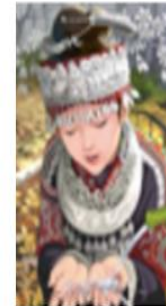
"This flower has petals that are yellow with shades of orange."



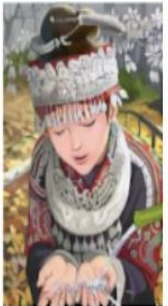
Original



Blurred



Restored with GAN



Summary

- Generative models have more impact on outliers than discriminative models.
- Discriminative models are computationally cheap as compared to generative models.
- Generative models need fewer data to train compared with discriminative models since generative models are more biased as they make stronger assumptions i.e. **assumption of conditional independence**.
- Generative models can work with missing data. discriminative models usually require all the features X to be observed.
- If the assumption of conditional independence violates, then at that time generative models are less accurate than discriminative models.

statistical learning methods - parametric or non-parametric

- **parametric model**

- A learning model that summarizes data with a set of fixed-size parameters (independent on the number of instances of training).
- can predict future values using only the parameters
- deal with discrete values
- able to infer the traditional measurements associated with normal distributions including mean, median, and mode.
- Example – Linear Regression model, Naive Bayes, Simple Neural Networks

- **Non-parametric model**

- do not make specific assumptions about the type of the mapping function. E.g. cannot assume the data comes from a normal distribution.
- They are prepared to choose any functional form from the training data, by not making assumptions.
- parameters are adjustable and can change
- use continuous values.
- We can feed all the data we have to those non-parametric algorithms and the algorithm can ignore unimportant features. It would not cause overfitting.
- slower and require large amounts of data
- E.g. KNN, Decision Trees, SVM

Inference – Important questions

- Which predictors are associated with the response?
 - only a small fraction of the available predictors are substantially associated with Y .
 - Identifying the few important predictors among a large set of possible variables depending on the application.
- What is the relationship between the response and each predictor?
 - Some predictors may have a positive relationship. Other predictors may have the opposite relationship.
 - Depending on the complexity of f , the relationship between the response and a given predictor may also depend on the values of the other predictors.
- Can the relationship between Y and each predictor be adequately summarized using a linear equation?
 - In some situations, such an assumption is reasonable/desirable.
 - But often the true relationship is more complicated, in which case a linear model may not provide an accurate representation of the relationship between the input and output variables

References

- https://courses.washington.edu/css490/2012.Winter/lecture_slides/02_math_essentials.pdf
- Christopher Bishop: "Pattern Recognition and Machine Learning" , 2006
- Kevin Murphy: "Machine Learning: a Probabilistic Perspective"
- David Mackay: "Information Theory, Inference, and Learning Algorithms"
- Ethem Alpaydin: "Introduction to Machine Learning" , 2nd edition, 2010.
- R. Duda, P. Hart & D. Stork, ***Pattern Classification*** (2nd ed.), Wiley T. Mitchell, ***Machine Learning***, McGraw-Hill

Thank You

- ????