E2 236
Foundations of Machine Learning (FoML)

Sundeep Prabhakar Chepuri

Office hours: by appointment (SPC: MP 128)

Indian Institute of Science
भारतीय विज्ञान संस्थान

Course information

- Instructor:
  Sundeep Prabhakar Chepuri

- Class schedule for Jan-April 2026:

  T/Th 8.30-10 am, MP 20, ECE
  Last class on 14/4/2026

- Teaching assistants:

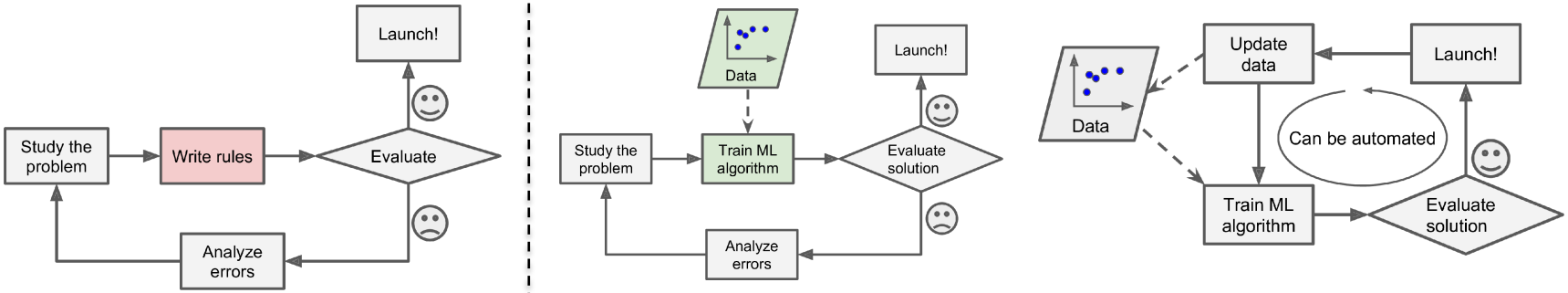
  - Suvam Dey, Indranil Patra, Anatap Mitra, Sonakshi Dua, and hopefully a few more…

Course objective

Introduce *theory*, *methods*, and *concepts* essential for developing programs that ***learn from data***

What is machine learning?

- Machine learning is the science (and **art**) of programming computers so they can learn from data.
- Your *spam filter* is a machine learning program that, given examples of spam emails (e.g., flagged by users) and examples of regular (nonspam, also called “ham”) emails, can learn to flag spam.



Traditional approach

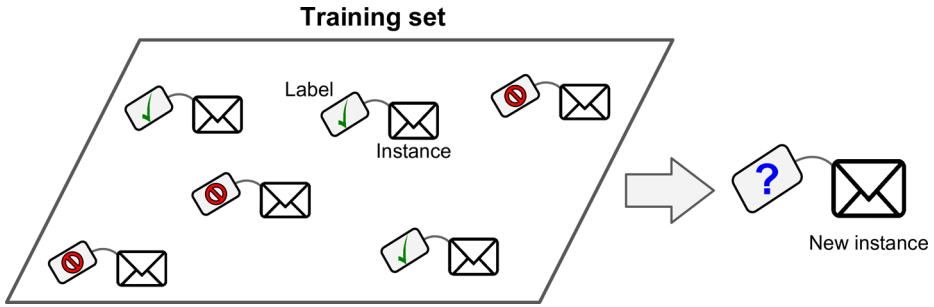
Machine learning approach

Images from HML

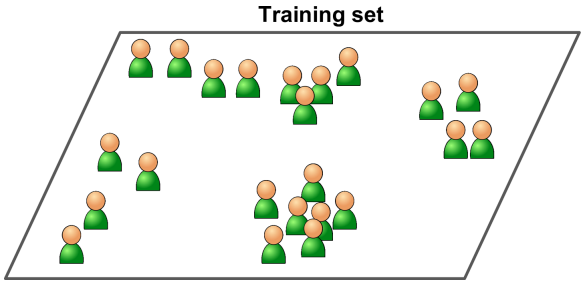
Schedule

Lecture number	Topic	Lab and Project
0	Introduction and logistics	
1	Machine Learning landscape	
2	Classification	
3	Classification	Lab 1
4	Classification	
5	Regression	
6	Regression	Lab 2
7	Regression	
8	Optimization	Lab 3
9	Optimization	
10	PAC learning framework	**Assignment 1**
11	PAC learning framework	
12	Kernel methods	
13	Gaussian Processes	Lab 3
14	SVMs	Lab 4
15	PCA and CCA	Lab 5
16	Clustering	Lab 6
17	Clustering	**Assignment 2**
18	EM method	**Project**
19	Ensemble learning	
20	Multilayer perceptron	
21	Multilayer perceptron	Lab 7
22	RNN and its variants	Lab 8
23	CNNs	Lab 9
24	Autoencoders	**Assignment 3**/ Lab 10
25	GANs	Lab 11
26	Transformers	Lab 12
27	Graph ML	
28	Graph ML	Lab 13

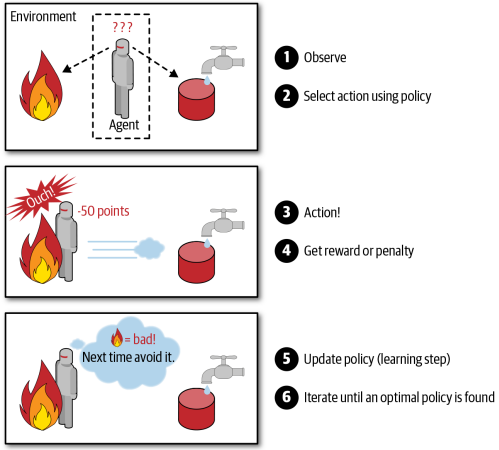
ML landscape



Supervised learning



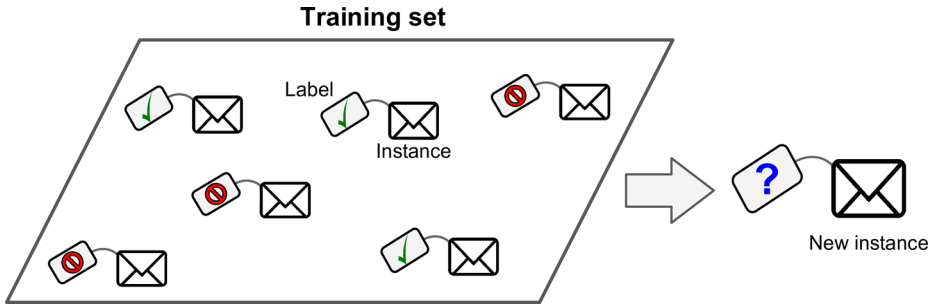
Unsupervised learning



Reinforcement learning

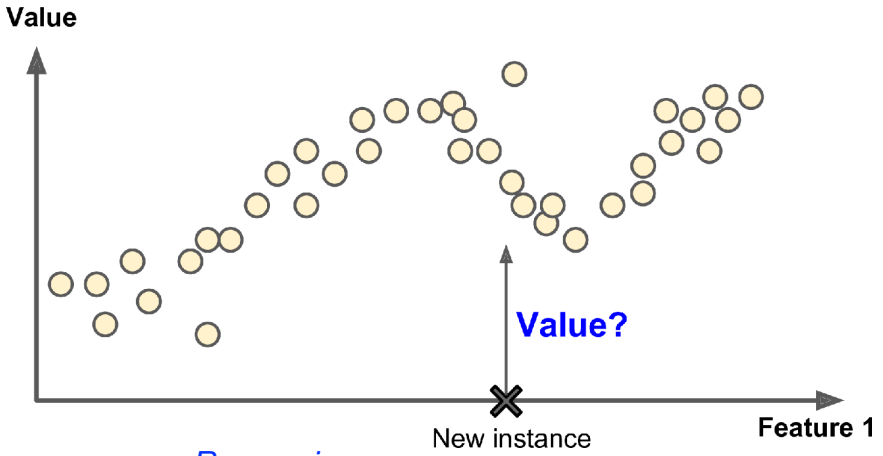
Images from HML

Classification and regression



Classification

- Multiclass and multilabel
- generative and discriminative models
- SVMs
- Neural networks

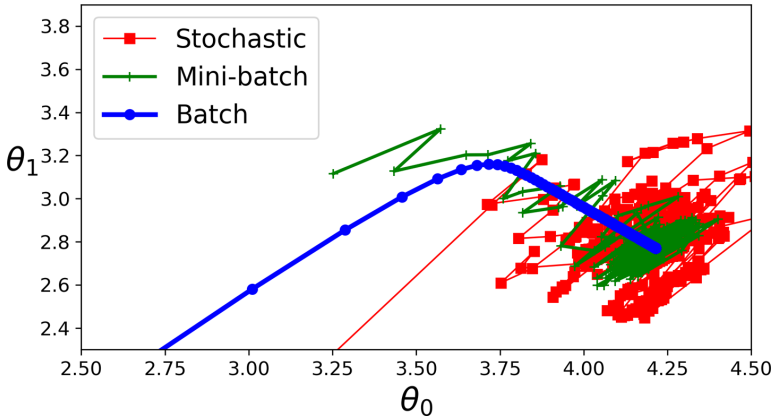


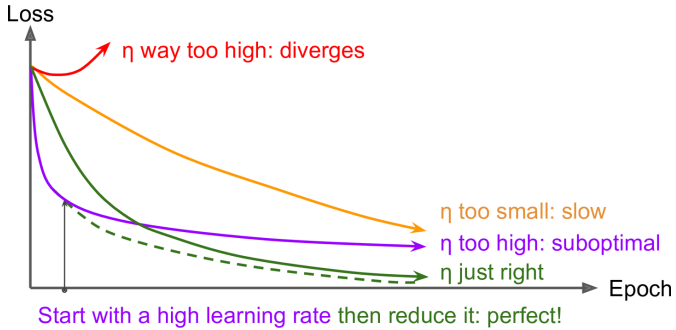
Regression

- Linear regression
- Bayesian linear regression
- Regularization (Lasso, ridge, elastic-net)

Images from HML

Optimization



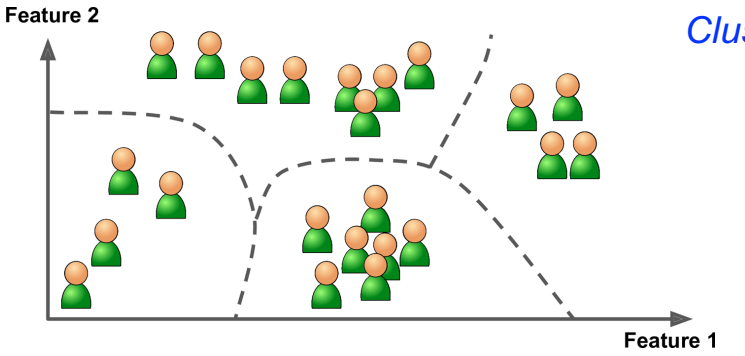


- *Convex and nonconvex functions*
- *Stochastic gradient descent, autodiff, backprop*
- *Adam optimizer*

Class	Convergence speed	Convergence quality
`SGD`	*	***
`SGD(momentum=...)`	**	***
`SGD(momentum=..., nesterov=True)`	**	***
`Adagrad`	***	* (stops too early)
`RMSprop`	***	** or ***
`Adam`	***	** or ***
`Nadam`	***	** or ***
`AdaMax`	***	** or ***

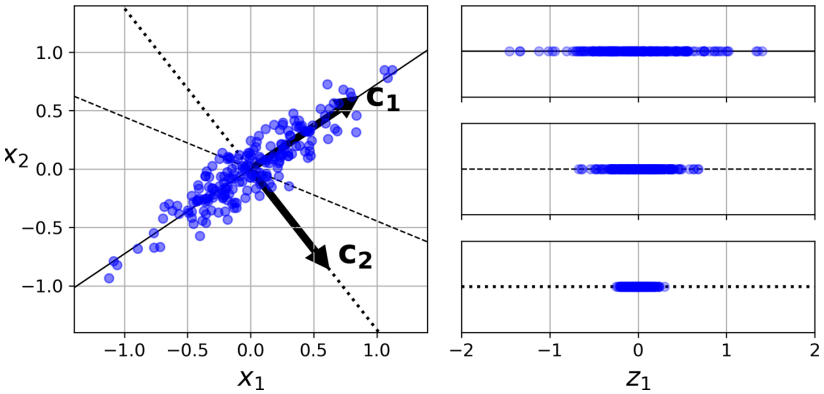
Images from HML

Unsupervised learning

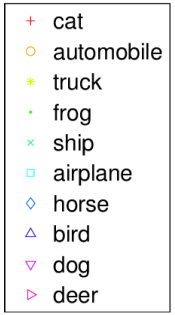


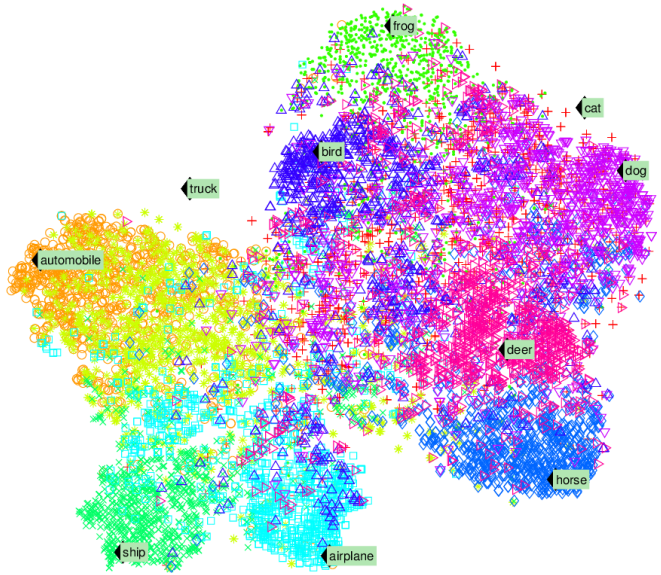
Clustering

- *K-means, spectral clustering*



Principal component analysis, and CCA



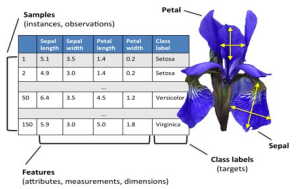


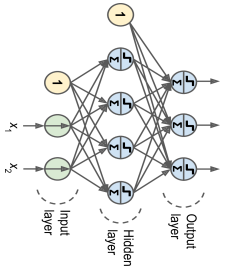
t-sne

Images from HML

Neural networks (MLP, CNN, RNNs, transformers, GNNs)

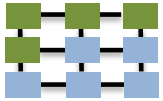
unstructured

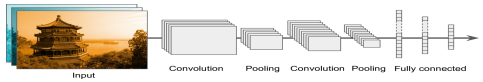





```
model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=[28, 28]),
    keras.layers.BatchNormalization(),
    keras.layers.Dense(300, activation="elu", kernel_initializer="he_normal"),
    keras.layers.BatchNormalization(),
    keras.layers.Dense(100, activation="elu", kernel_initializer="he_normal"),
    keras.layers.BatchNormalization(),
    keras.layers.Dense(10, activation="softmax")
])
```

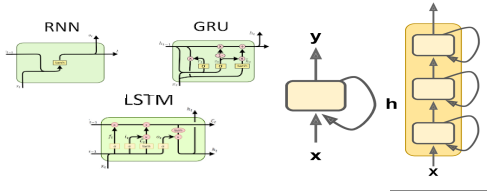
structured



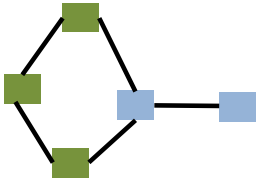



```
model = keras.models.Sequential([
    keras.layers.Conv2D(64, 7, activation="relu", padding="same",
                        input_shape=[28, 28, 1]),
    keras.layers.MaxPooling2D(2),
    keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
    keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
    keras.layers.MaxPooling2D(2),
    keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
    keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
    keras.layers.MaxPooling2D(2),
    keras.layers.Flatten(),
    keras.layers.Dense(128, activation="relu"),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(64, activation="relu"),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(10, activation="softmax")
])
```

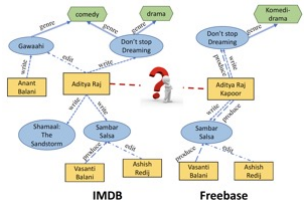
sequences

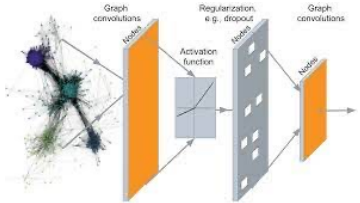



```
model = keras.models.Sequential([
    keras.layers.LSTM(20, return_sequences=True, input_shape=[None, 1]),
    keras.layers.LSTM(20, return_sequences=True),
    keras.layers.TimeDistributed(keras.layers.Dense(10))
])
```



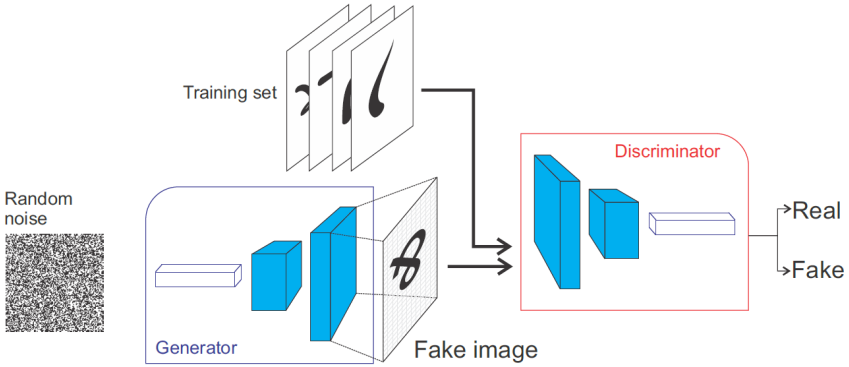
Relational data



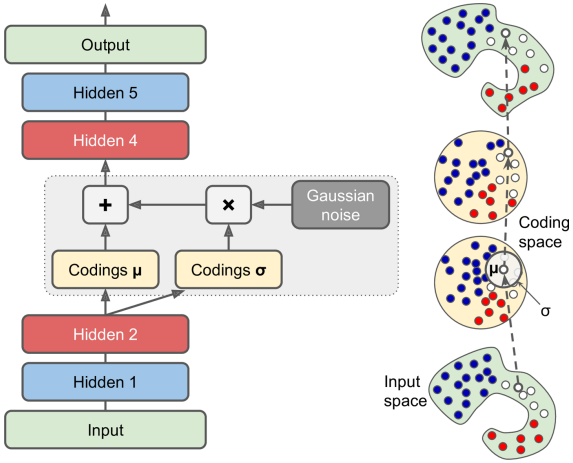


Images from HML/internet

Generative AI



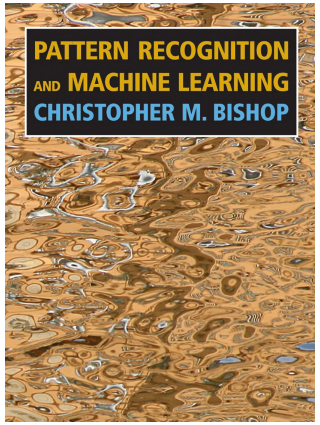
GANs



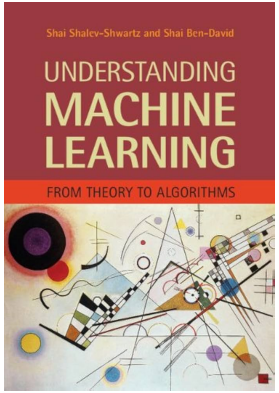
Variational autoencoders

Images from HML/internet

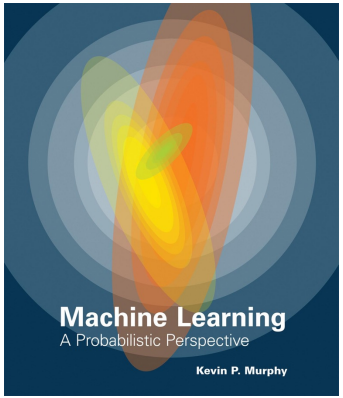
Books



PRML

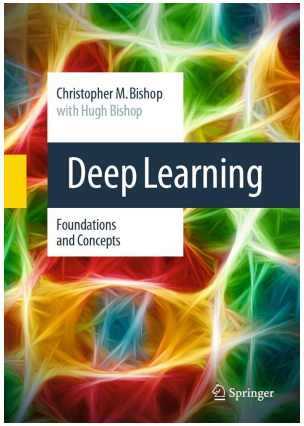


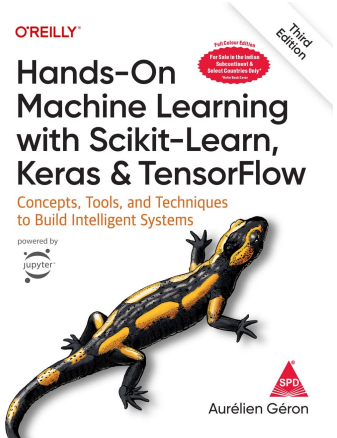
UML



MLPP

DL





HML

Grading and course requirements

- **Prerequistite:** Basic linear algebra and Python (Google CoLab) for assignments and project.
- This is a **3:1** course
- Final Exams will be CLOSED book exam
- Projects can be done in a group of N (to be defined) members
- Assignments to be done individually
- Course page: https://ece.iisc.ac.in/~spchepuri/e2236.html

Teams code:
1a77qjo

Grading	%
Three assignments (10 pts each)	30
Project	20
Final Exam	50
Total	100