Lecture 01: Introduction

Introduction to Machine Learning [25737]

Sajjad Amini

Sharif University of Technology

Contents

Section 1

Course Introduction

Subsection 1

State of Machine Learning

State of Machine Learning

Andrew Ng

Electricity transforms countless industries: transportation, manufacturing, healthcare, communications and more. AI (machine learning) will bring about an equally big transformation

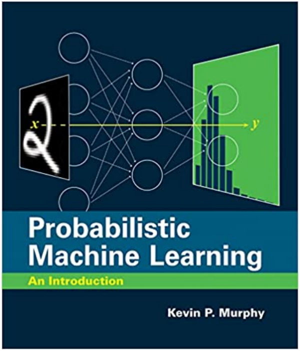
Subsection 2

Course Logistics

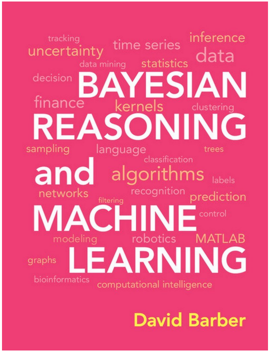
Head Assistant

Figure: Mohammad Reza Rahmani

Main Textbook



Textbook


tracking
uncertainty
time series
inference
data mining
statistics
data
decision
BAYESIAN
finance
kernels
clustering
REASONING
sampling
language
trees
classification
and
algorithms
labels
networks
recognition
filtering
prediction
MACHINE
control
modeling
robotics
MATLAB
graphs
LEARNING
bioinformatics
computational intelligence
David Barber

Outline

- Foundation
  - Introduction
  - Univariate Probability
  - Multivariate Probability
  - Statistics
  - Decision Theory
  - Optimization [Exercise Set 1]
- Supervised Learning
  - Linear Discriminant Analysis
  - Logistic Regression
  - Linear Regression [Exercise Set 2]
  - Neural Networks [Exercise Set 3] [Midterm]
  - Exemplar-Based Methods
  - Kernel Methods [Exercise Set 4]
  - Trees
  - Bagging
  - Forest
  - Boosting [Exercise Set 5]
- Unsupervised Learning
  - Dimensionality Reduction
  - Clustering [Exercise Set 6]

Grading Policy

Your grade

Activity	Grade
Exercises	6
Midterm	4
Final Exam	4
Final Project	6
Class Activity (Bonus)	1
Summation	21

Regarding your Activities

- One exercise will be neglected.
- Exercises will be precisely inspected for probable similarities (Similar ones would be graded as 0)
- The best communication line is my email: *s_amini@sharif.edu*
- If you email TAs regarding the course, please CC me for future followups.
- Extra class time

References

The material in the slides except cited are inspired from the following reference:

- Murphy, K. P. (2022). *Probabilistic machine learning: an introduction.* MIT press.

Color Guided Blocks

Definition Block

Result Block

Note Block

Example Block

Remember Block

What is Machine Learning

Machine Learning [1]

Consider the following three items:

- Experience E
- Class of tasks T
- Performance measure P

Machine learning is to improve the performance measured by P of a computer program on T using E

Machine Learning

Based on *Machine Learning*, definition we have the main following major types of machine learning:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Section 2

Supervised Learning

Supervised Learning

Supervised Learning

Supervised Learning is:

- Task T: Finding mapping $f: \boldsymbol{x} \mapsto \boldsymbol{y}$ ($\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^D$ and $\boldsymbol{y} \in \mathcal{Y}$)
  - $\boldsymbol{x}$: Features, Covariates or Predictors
  - $\boldsymbol{y}$: Label, Target or Response
- Experience E: Set of N input-output pairs $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$
  - $\mathcal{D}$: Dataset
  - N: Sample size
- Performance measure P: Dependent on the task

Subsection 1

Classification

Supervised Learning - Classification

Classification

General Features:

- Task T: Finding mapping $f : \boldsymbol{x} \mapsto y$ ($\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^D$ and $y \in \mathcal{Y}$)
- Experience E: Set of N input-output pairs $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$

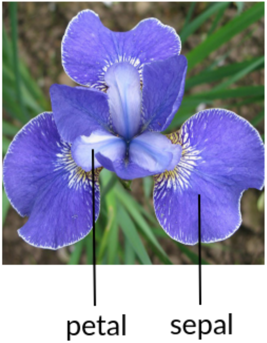
Specific Features:

- $\mathcal{Y} = \{1, 2, \ldots, C\}$ (Unordered and mutually exclusive labels)
- $P = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(y_n \neq f(\boldsymbol{x}_n))$ where:

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

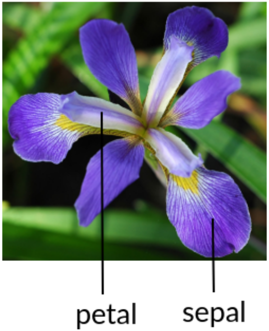
  - P is known as *Misclassification Error*

Example - Iris Flower Classification



(a) Setosa ($y = 1$)

(b) Versicolor ($y = 2$)



(c) Virginica ($y = 3$)

Figure: Different types of Iris flower

Example - Iris Flower Classification

$f_1\Big($  $\Big) = \begin{cases} 1 \text{ (Setosa)} \\ 2 \text{ (Versicolor)} \\ 3 \text{ (Virginica)} \end{cases}$

(a) Approach 1

$$\overbrace{\qquad\qquad\qquad}^{\begin{bmatrix}\text{Sepal length}\\ \text{Sepal width}\\ \text{Petal length}\\ \text{Petal width}\end{bmatrix}}$$

$f_2\Big(g\Big($  $\Big)\Big) = \begin{cases} 1 \text{ (Setosa)} \\ 2 \text{ (Versicolor)} \\ 3 \text{ (Virginica)} \end{cases}$

(b) Approach 2

Figure: Different types of Iris flower

Example - Iris Flower Classification

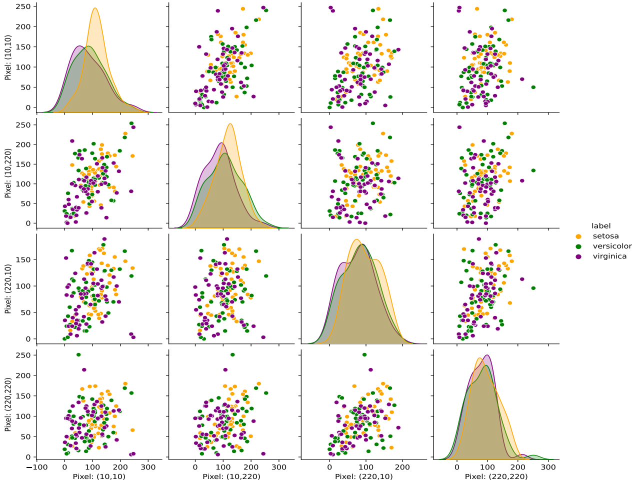


Figure: Exploratory data analysis - Approach 1

Example - Iris Flower Classification

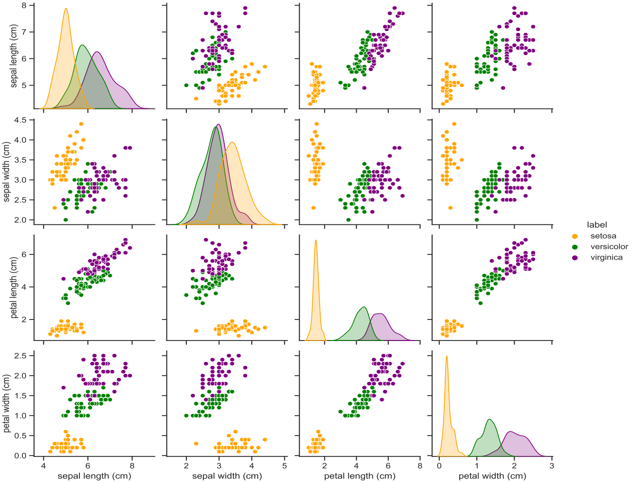
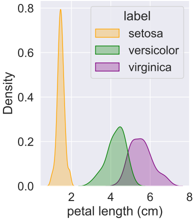
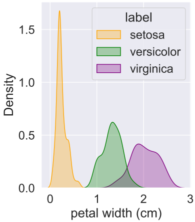


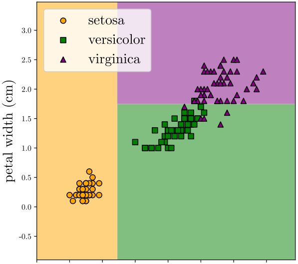
Figure: Exploratory data analysis - Approach 2



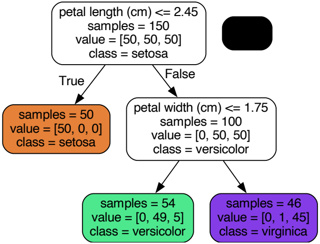
(a) Petal length



(b) Petal width



(c) Decision tree surface



(d) Decision tree

Automate Classification

Steps to Automate Classification

- Parameterizing mapping: $f(\cdot) \Rightarrow f(\cdot; \boldsymbol{\theta})$
- Finding suitable $\boldsymbol{\theta}$ ($\widehat{\boldsymbol{\theta}}$) using Experiment E (Model Fitting)
  - Enhancing performance measure $P \Rightarrow$ decreasing misclassification error

Note: Limitation with Misclassification Error

Inability to distinguish different errors

		$f(\boldsymbol{x}; \boldsymbol{\theta})$		
		Setosa	Versicolor	Virginica (Poisonous)
y	Setosa	0 (0)	1 (1)	1 (1)
	Versicolor	1 (1)	0 (0)	1 (1)
	Virginica (Poisonous)	1 (10)	1 (10)	0 (0)

Automate Classification

Empirical Risk

Empirical Risk, the generalization of misclassification error is:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} l(y_n \neq f(\boldsymbol{x}_n; \boldsymbol{\theta}))$$

Using above definition, model fitting can be done via *Empirical Risk Minimization* (ERM) as:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{N}\sum_{n=1}^{N} l(y_n \neq f(\boldsymbol{x}_n; \boldsymbol{\theta}))$$

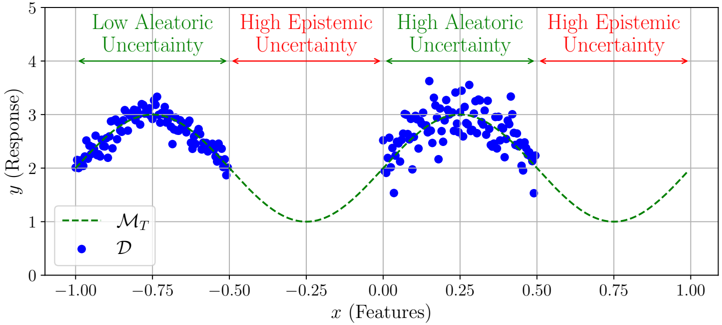
Uncertainty [2]

Epistemic Uncertainty (Model Uncertainty)

Uncertainty originated from lack of knowledge about true input-output mapping

Aleatoric Uncertainty (Data Uncertainty)

Uncertainty originated from inherent randomness in experiment E



Uncertainty in Classification

Capturing Uncertainty

To capture uncertainty, we can define *Conditional Probability Density* (CPD) as:

$$p(y = c|\boldsymbol{x};\boldsymbol{\theta}) = f_c(\boldsymbol{x};\boldsymbol{\theta}), \quad \begin{cases} 0 \leq f_c \leq 1 \\ \sum_{c=1}^{C} f_c = 1 \end{cases}$$

Softmax Function

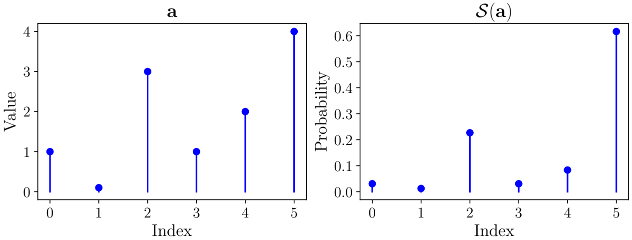
Softmax Function

Consider the *logits* vector defined as:

$$\boldsymbol{a} \triangleq [a_1, \ldots, a_C] = [f_1(\boldsymbol{x};\boldsymbol{\theta}), \ldots, f_C(\boldsymbol{x};\boldsymbol{\theta})] = \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})$$

The softmax function for this vector is defined as:

$$\boldsymbol{\mathcal{S}} \triangleq \left[\frac{e^{a_1}}{\sum_{c'=1}^{C} e^{a_{c'}}}, \ldots, \frac{e^{a_C}}{\sum_{c'=1}^{C} e^{a_{c'}}} \right]$$



Application of Softmax Function in Classification

Capturing Uncertainty Using Softmax Function

Previously we define:

$$p(y = c|\boldsymbol{x}; \boldsymbol{\theta}) = f_c(\boldsymbol{x}; \boldsymbol{\theta}), \;\; \begin{cases} 0 \leq f_c \leq 1 \\ \sum_{c=1}^{C} f_c = 1 \end{cases}$$

Now using softmax we have:

$$p(y = c|\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\mathcal{S}}_c(\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta}))$$

where the following constraints are met:

$$0 \leq \boldsymbol{\mathcal{S}}_c(\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta})) \leq 1, \;\; c = 1, 2, \ldots, C$$

$$\sum_{c=1}^{C} \boldsymbol{\mathcal{S}}_c(\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta})) = 1$$

Fitting Probabilistic Model

Maximum Likelihood Estimation

One approach to fit probabilistic models is *Maximum Likelihood Estimation* (MLE). We can equivalently define loss function as:

$$l(y, f(\boldsymbol{x};\boldsymbol{\theta})) = -\log p(y|f(\boldsymbol{x};\boldsymbol{\theta}))$$

Using above loss, the *Negative Log Likelihood* (NLL) over training set is:

$$\mathrm{NLL}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} \log p(y_n|f(\boldsymbol{x}_n;\boldsymbol{\theta}))$$

Then the MLE for model parameters is:

$$\widehat{\boldsymbol{\theta}}_{mle} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \mathrm{NLL}(\boldsymbol{\theta})$$

Subsection 2

Regression

Supervised Learning - Regression

Regression

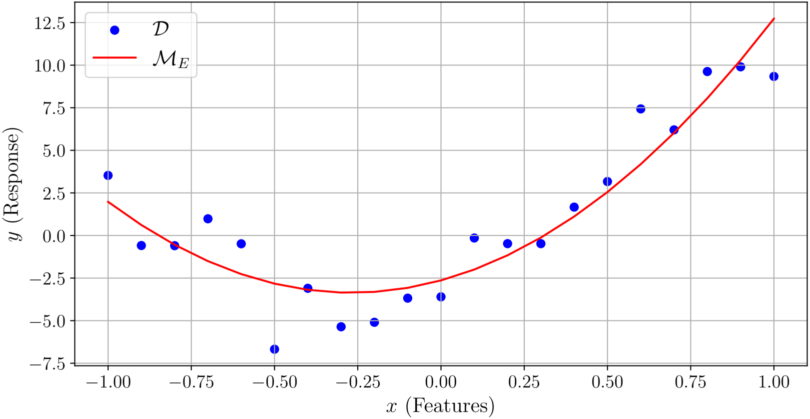
General Features:

- Task T: Finding mapping $f : \boldsymbol{x} \mapsto \boldsymbol{y}$ ($\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^D$ and $y \in \mathcal{Y}$)
- Experience E: Set of N input-output pairs $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$

Specific Features:

- $\mathcal{Y} = \mathbb{R}$
- $P = \frac{1}{N} \sum_{n=1}^{N} (y_n - f(\boldsymbol{x}_n; \boldsymbol{\theta}))^2$
  - P is known as *Mean Square Error* (MSE)

Example - One Dimensional Curve Fitting



Model Fitting in Regression

Model Fitting via ERM

Similar to classification, model parameters for regression problem can be found via ERM as:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \overbrace{\frac{1}{N}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n;\boldsymbol{\theta}))^2}^{\mathrm{MSE}(\boldsymbol{\theta})}$$

Capturing Uncertainty in Regression

To capture uncertainty, we assume the output distribution to be Gaussian (Normal) as:

$$\mathcal{N}(y|\mu, \sigma^2) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-\mu)^2}$$

We make the mean depend on the inputs by defining $\mu \triangleq f(\boldsymbol{x}_n, \boldsymbol{\theta})$. Then we have the following CPD:

$$p(y|\boldsymbol{x}; \boldsymbol{\theta}) = \mathcal{N}(y|f(\boldsymbol{x}; \boldsymbol{\theta}), \sigma^2)$$

Subsection 3

Generalization

Overfitting and Generalization

Population Risk

Consider $p^\star(\boldsymbol{x}, y)$ to be the true generating distribution of training set. Then population risk is defined as:

$$\mathcal{L}(\boldsymbol{\theta}; p^\star) \triangleq \mathbb{E}_{p^\star(\boldsymbol{x},y)}[l(y, f(\boldsymbol{x}; \boldsymbol{\theta}))]$$

Generalization Gap

The difference $\mathcal{L}(\boldsymbol{\theta}; p^\star) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{train})$ is called generalization gap where $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{train})$ is ERM defiend as:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{train}) = \frac{1}{|\mathcal{D}_{train}|} \sum_{(\boldsymbol{x}_n, y_n) \in \mathcal{D}_{train}} l(y_n, f(\boldsymbol{x}_n; \boldsymbol{\theta}))$$

Overfitting

Overfitting occure when the generalization gap is large.

Population Risk

Population Risk Estimation

- In practice, we don't know $p^\star(\boldsymbol{x}, y)$.
- We partition the data into two subsets, known as the training set and test set.
- We use test set to estimate population risk as:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{test}) = \frac{1}{|\mathcal{D}_{test}|} \sum_{(\boldsymbol{x}_n, y_n) \in \mathcal{D}_{test}} l(y_n, f(\boldsymbol{x}_n; \boldsymbol{\theta}))$$

Subsection 4

Real World Applications

Real World Classification: Super resolution [3]

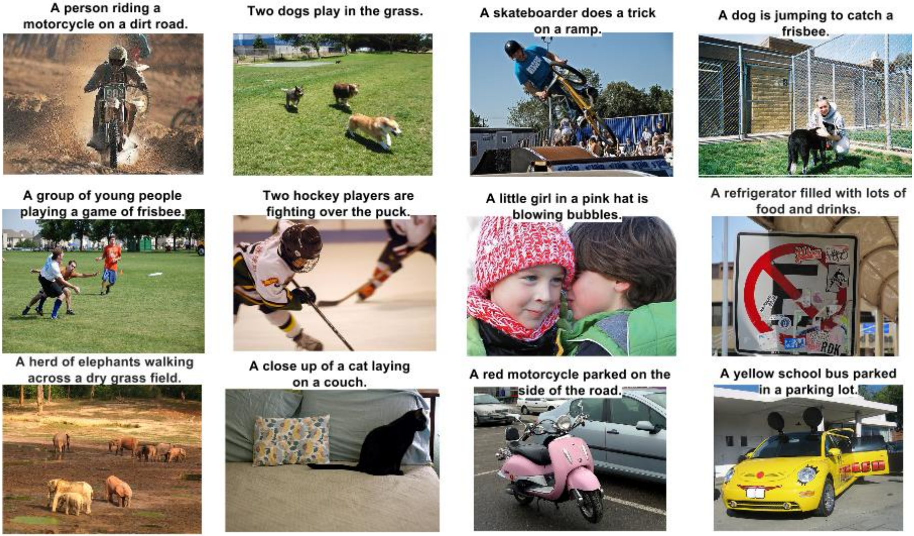
Real World Classification: Image Generation [4]

(a) a tapir made of accordion. a tapir with the texture of an accordion.

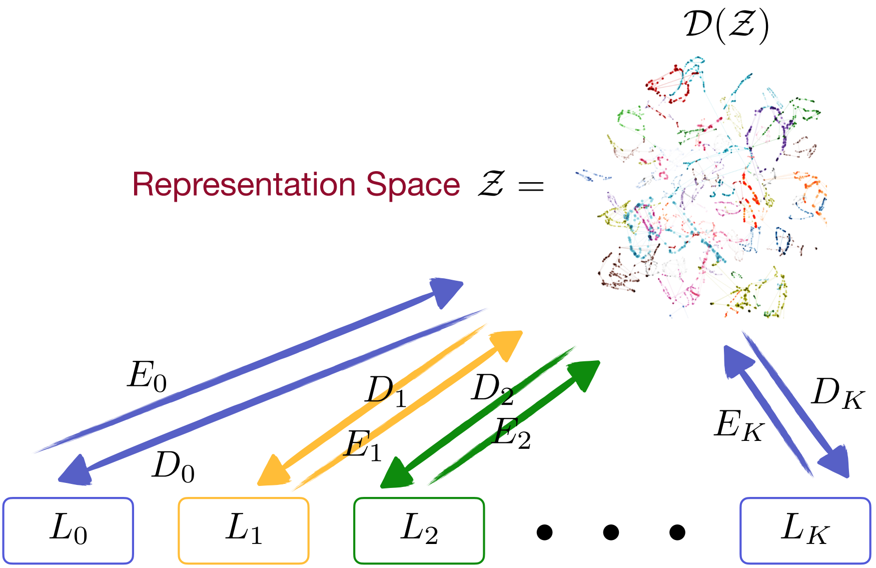
(b) an illustration of a baby hedgehog in a christmas sweater walking a dog

(c) a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

Real World Classification: Image Captioning [5]



Real World Classification: Machine Translation [6]



Section 3

Unsupervised Learning

Unsupervised Learning

Unsupervised Learning

Unsupervised Learning is:

- Task T: Dependent on the task
- Experience E: Set of N samples $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$
- Performance measure P: Dependent on the task

Subsection 1

Clustering

Unsupervised Learning - Clustering

Clustering

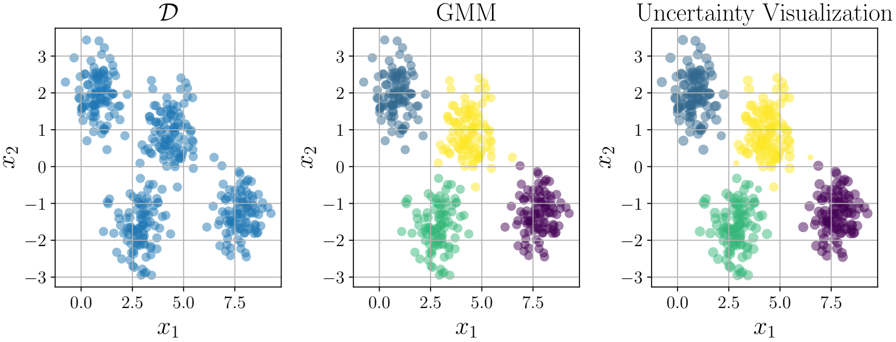
General Features:

- Experience E: Set of N samples $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$

Specific Features:

- Task T: Partition the input into regions that contains *similar* points.
- Performance measure in *Compression*: Compression loss

Clustering: Gaussian Mixture Model [7]



Subsection 2

Factors of Variation

Unsupervised Learning - Factors of Variation

Factors of Variation

General Features:

- Experience E: Set of N samples $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$

Specific Features:

- Task T: Projecting data into low dimensional subspace which captures its main aspects
- Performance measure: Performance of low dimensional data in various downstream tasks

Principle Component Analysis [8]

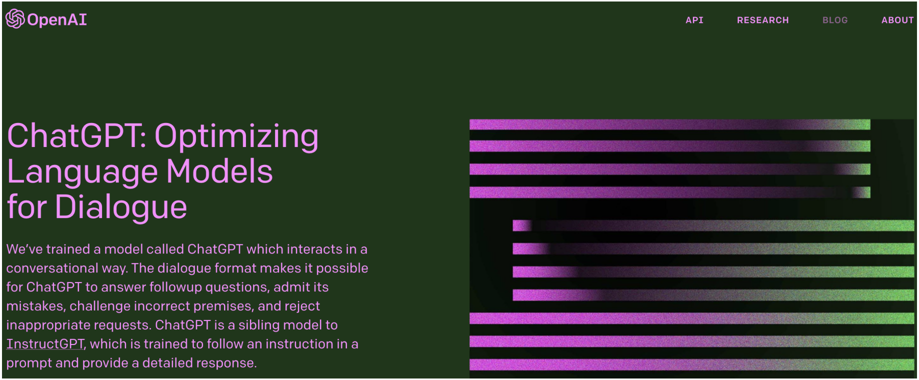
(a) Original 64×64 pixels

(b) Reconstructed from 8×8 pixels

Subsection 3

Real World Applications

ChatGPT [9]


OpenAI
API
RESEARCH
BLOG
ABOUT
ChatGPT: Optimizing Language Models for Dialogue
We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

Language Model

$\{x_i\}$

Language Model

p(x)

p("I am going home") = 0.1
p("I you he blackboard") = 0.00001

Sample p(x): "I show him my drawing."
Conditional Sampling: p(x| "Today is")

Section 4

Reinforcement Learning

Reinforcement Learning

Reinforcement Learning

Reinforcement Learning is:

- Task T: Learning an agent to take action in different environmental conditions.
- Experience E: Set of N condition-action-reward triplet
- Performance measure P: Average reward

Reinforcement Learning

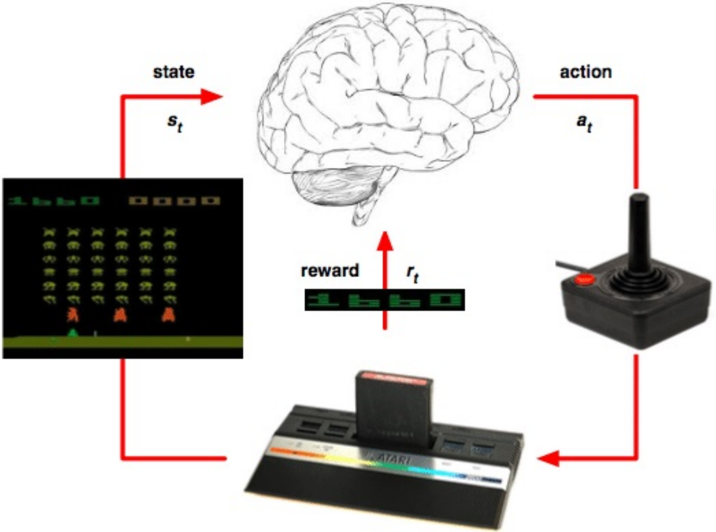
Playing an Atari game

- Task T: learning policy mapping $\boldsymbol{a} = \pi(\boldsymbol{x})$ where:
  - $\boldsymbol{a}$: Action
  - $\boldsymbol{x}$: Environmental conditions
- Experience E: Set of N triplet $\{(\boldsymbol{x}_n, \boldsymbol{a}_n, r_n)\}_{n=1}^{N}$
- Performance measure P: Maximizing the reward

Subsection 1

Real World Applications

Reinforcement Learning



References I

Tom M Mitchell and Tom M Mitchell,
Machine learningg, vol. 1,
McGraw-hill New York, 1997.

"Ensemble distribution distillation,"
`https://campusai.github.io/papers/ensemble-distribution-distillation`,
Accessed: 2022-09-13.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi,
"Image super-resolution via iterative refinement,"
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever,
"Zero-shot text-to-image generation,"
in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan,
"Show and tell: A neural image caption generator,"
in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

Han Zhao, Junjie Hu, and Andrej Risteski,
"On learning language-invariant representations for universal machine translation,"
in *International Conference on Machine Learning*. PMLR, 2020, pp. 11352–11364.

References II

"In depth: Gaussian mixture models," https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html, Accessed: 2022-09-13.

"Facial image compression and reconstruction with pca," https://shankarmsy.github.io/posts/pca-sklearn.html#, Accessed: 2022-09-13.

"Chatgpt: Optimizing language models for dialogue," https://openai.com/blog/chatgpt/, Accessed: 2022-09-13.