CSC 461: Machine Learning

Fall 2024

Supervised learning

Prof. Marco Alvarez, Computer Science
University of Rhode Island

Definition

- Paradigm in ML where a model learns from labeled data
  - mapping inputs to corresponding outputs by minimizing a predefined "loss function"
- Key components
  - data instance: $(x, y), x \in \mathcal{X}, y \in \mathcal{Y}$
    - input space $\mathcal{X}$
    - output space $\mathcal{Y}$
  - training data: $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$
  - model (hypothesis): $h : \mathcal{X} \mapsto \mathcal{Y}, h \in \mathcal{H}$
    - hypothesis space $\mathcal{H}$

Dataset

- Observations are independently drawn from a **joint distribution** of inputs and outputs

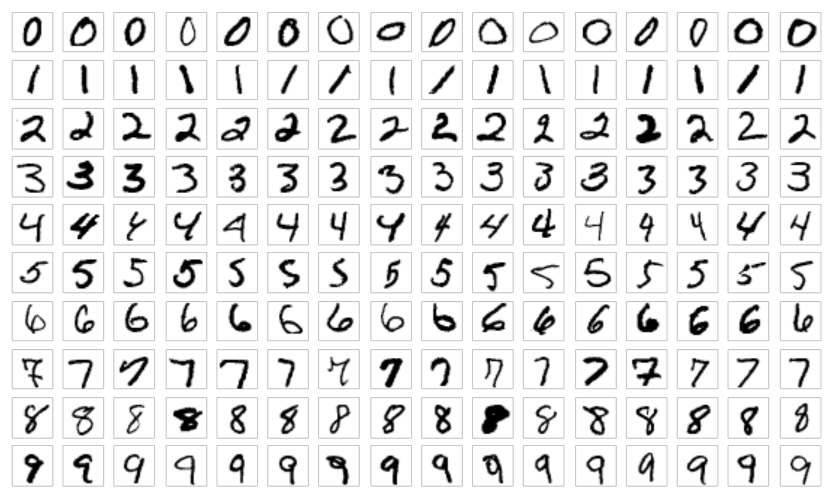
$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

$$(x_i, y_i) \sim P$$

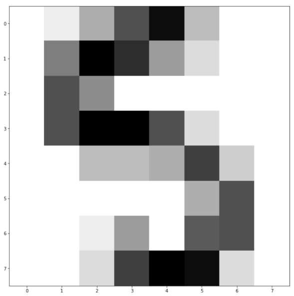
unknown

- Input space $\mathcal{X}$
  - numerical — continuous (age, income, …)
  - categorical (gender, product type, …)
  - text (customer reviews, documents, …)
  - images (photos, medical images, …)
  - audio (speech, music, …)

Example: MNIST dataset



Example: MNIST dataset



Image

```
[[ 0.   1.   5. 11. 15.   4.   0.   0.]
 [ 0.   8. 16. 13.   6.   2.   0.   0.]
 [ 0. 11.   7.   0.   0.   0.   0.   0.]
 [ 0. 11. 16. 16. 11.   2.   0.   0.]
 [ 0.   0.   4.   4.   5. 12.   3.   0.]
 [ 0.   0.   0.   0.   0.   5. 11.   0.]
 [ 0.   0.   1.   6.   0. 10. 11.   0.]
 [ 0.   0.   2. 12. 16. 15.   2.   0.]]
```

Matrix representation

```
[ 0.  1.  5. 11. 15.  4.  0.  0.  0.  8. 16. … 11.  0.  0.  0.  2. 12. 16. 15.  2.  0.]
```

Vector representation

$\mathcal{X} =$

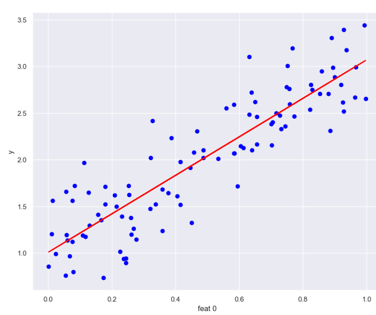
$\mathcal{Y} =$

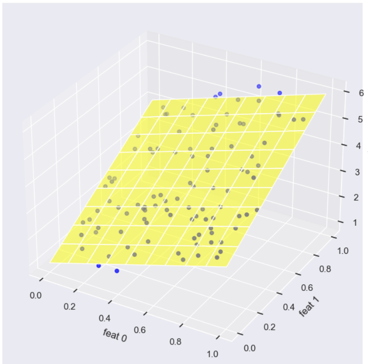
Major tasks

Types of supervised learning

- Regression
  - continuous output: $\mathcal{Y} \subseteq \mathbb{R}$
  - examples: predicting house prices, forecasting stock prices
- Classification
  - discrete output: $\mathcal{Y} = \{1,\ldots,K\}$
    - binary classification: $K = 2$
    - multi-class classification: $K > 2$
  - examples, spam vs not spam (binary), disease present (binary), handwritten digits recognition (multi-class)

Regression





$\mathcal{X} =$

$\mathcal{Y} =$

$\mathcal{X} =$

$\mathcal{Y} =$

Binary classification

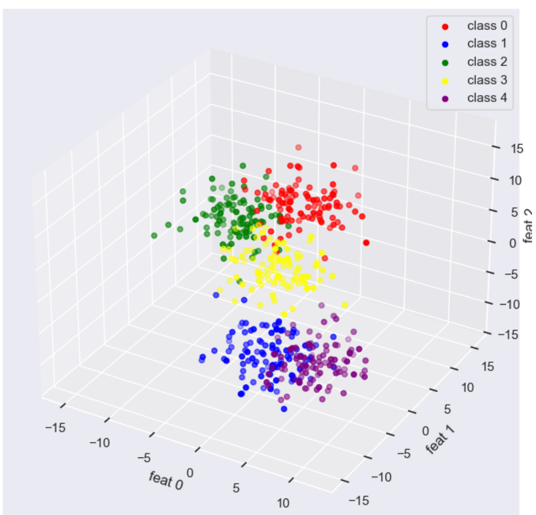
```
array([[ 0.24277092,  0.89098144],      array([[0],
       [-0.57961074,  0.50618765],             [1],
       [ 0.24259841,  0.12209649],             [1],
       [ 1.68348295, -0.10059047],             [1],
       [ 2.00696736, -0.79306007],             [1],
       [ 1.56891881,  0.30515286],             [0],
       [ 0.1314049 , -0.35704446],             [1],
       [ 2.14017386,  0.33933491],             [1],
       [-1.03087047,  1.52609949],             [0],
       [-0.38504321,  1.24209655],             [0],
       [-1.20252537,  0.56167652],             [0],
       [ 0.08590311,  0.68265315],             [1],
       [ 0.88074085, -0.11759523],             [1],
       [ 0.32558238,  0.4181143 ],             [1],
       [-0.74202798,  0.68847344]])            [0]])
```

Binary classification

$\mathcal{X} =$

$\mathcal{Y} =$

Multi-class classification



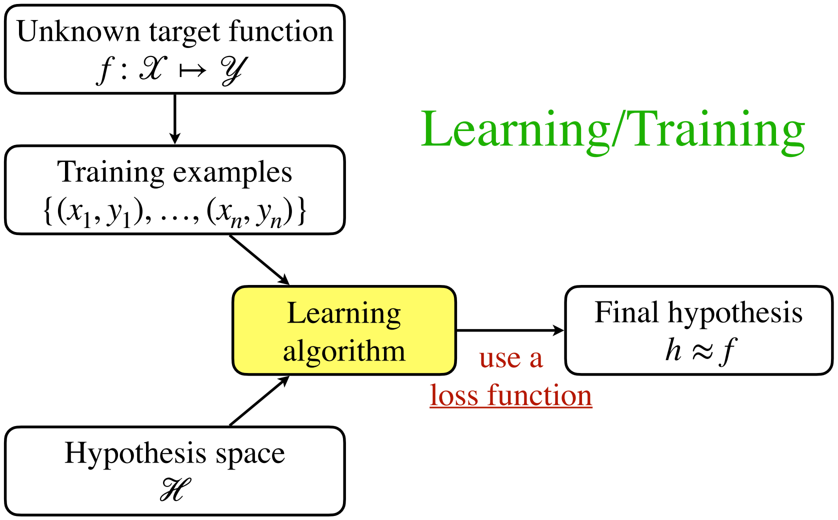
$\mathcal{X} =$

$\mathcal{Y} =$

Learning

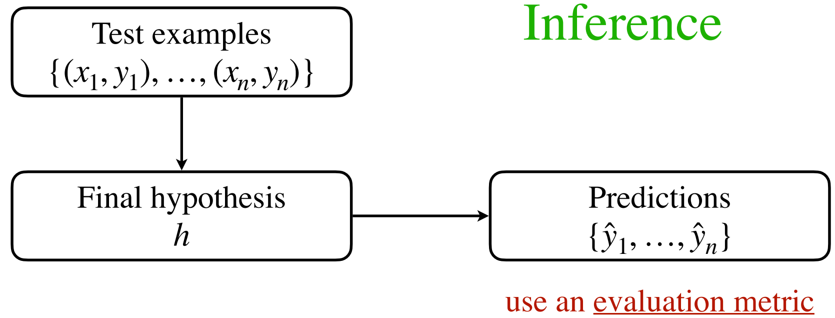
Supervised learning setup

Learning/Training



Supervised learning setup

Inference



Loss functions and evaluation

Loss functions

- Purpose
  - guide the learning process during training/learning
- Characteristics
  - differentiable (in most cases — neural networks)
  - optimized during training/learning
  - reflect model performance on individual examples or batches of examples

0/1 loss

$$l_{0/1}(h, x_i, y_i) = I\left(h(x_i) \neq y_i\right)$$

indicator function

Prediction	Target
5	5
1	9
2	2
7	7
8	0
0	0
0	8
3	3
6	6
4	4

Squared loss (L2 loss)

$$l_{sq}(h, x_i, y_i) = \left(h(x_i) - y_i\right)^2$$

positive loss,
penalizes
big mistakes

Prediction	Target
1.2	1.4
2.3	2.3
1.1	1.2
3.4	4.1
2.3	2.5
1.1	1.1
2.5	2.6
3.1	3.2
1.7	1.8
2.3	2.3

Absolute loss

$$l_{abs}(h, x_i, y_i) = \left|h(x_i) - y_i\right|$$

Prediction	Target
1.2	1.4
2.3	2.3
1.1	1.2
3.4	4.1
2.3	2.5
1.1	1.1
2.5	2.6
3.1	3.2
1.7	1.8
2.3	2.3

Evaluation metrics

- Purpose
  - assess model performance after training
- Characteristics
  - may not be differentiable
  - used to compare different models/hypotheses or report final performance
  - often more interpretable
  - reflect performance on an entire dataset
- Examples
  - accuracy, f1-score, precision, recall, mean absolute error, R-squared

More formally …

From statistical learning theory

- Expected Loss
  - theoretical average loss over **<u>all possible data points</u>**
  - including those not in our training set

$$\mathbb{E}\left[l\left(h, x_i, y_i\right)\right]_{(x_i, y_i)\sim P}$$

We cannot calculate this term, but we can approximate it

- Empirical Loss
  - the average loss calculated on the **<u>training data</u>**
  - what we can actually measure during training/learning

$$\frac{1}{n}\sum_{i=1}^{n} l\left(h, x_i, y_i\right)$$

Learning formulation

- Given:
  - training set: $\{(x_1, y_1), \ldots, (x_n, y_n)\}, x_i \in \mathcal{X}, y_i \in \mathcal{Y},$
- Objective:
  - find a function $h : \mathcal{X} \to \mathcal{Y}, h \in \mathcal{H}$ that minimizes the <u>empirical loss</u> L

$$L = \frac{1}{n}\sum_{i=1}^{n} l(h, x_i, y_i)$$

As the size of our training dataset increases, the **<u>empirical loss</u>** tends to approach the **<u>expected loss</u>**

Challenges

- Overfitting
  - model performs well on training data but poorly on unseen data
  - solutions include: regularization, data augmentation
- Underfitting
  - model is too simple to capture underlying patterns
  - solutions include: increase model complexity
- Imbalanced Datasets
  - one class may be overrepresented.
  - solutions include: resampling, class weighting
- Curse of Dimensionality
  - too many features can cause models to perform poorly
  - solutions include: dimensionality reduction, using more data