SUPSI

Machine Learning

Introduction

Dario Piga
SUPSI - Scuola Universitaria Professionale della Svizzera Italiana
IDSIA – Dalle Molle Institute for Artificial Intelligence
dario.piga@supsi.ch

Artificial Intelligence

- Artificial Intelligence (AI) is "the science and engineering of making intelligent machines" (John McCarthy, 1956)

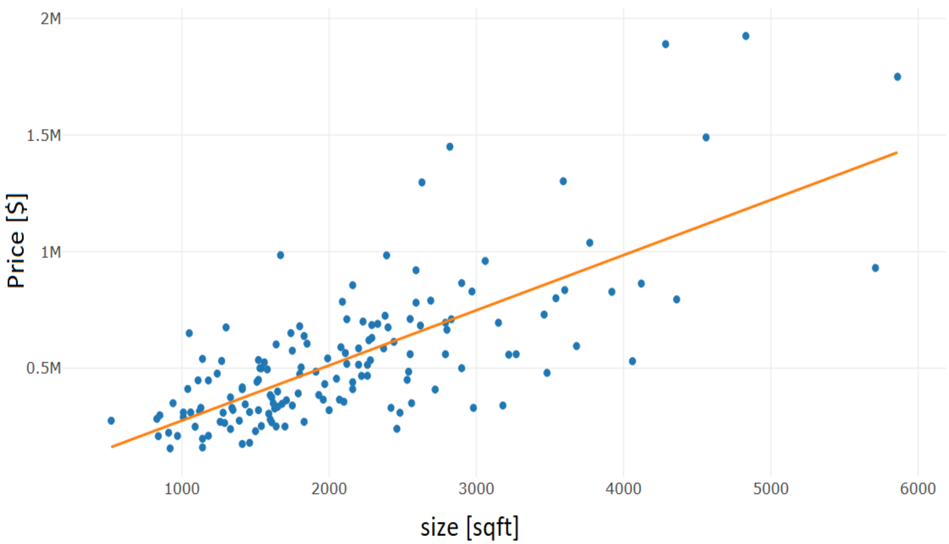
- Development of computer systems able to perform tasks that normally require human intelligence (e.g., game playing, driving a car, walking, recognize a face, a digit or a sound, detect an anomalous behaviour, etc.)

- Two main approaches for AI:
  - program the machine to perform a specific task
  - let the machine learn from experience (machine learning)

Supervised learning

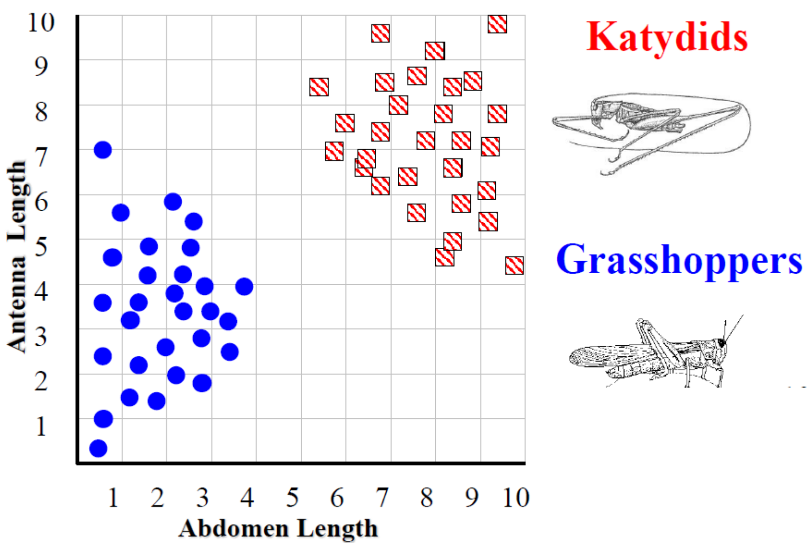
- Information: set of input-output data is given
- Goal: discover relation between input and output
- Utility: given a new input, predict the output

Regression



Continuous-value output

Classification



Discrete-value output

Examples of classification

- Image recognition

- Predicting tumor cells as benign or malignant (extract features from cells images)

- Detecting faults (unbalanced problem)

- Predicting level of affinity (low/high) between a protein and a ligand

Classification: notation

- We are given *N* training input-output data $\{x_i, y_i\}_{i=1}^{N}$:
  - $x_i \in \mathbb{R}^D$
  - $y_i \in \{True, False\}, \quad \{0,1,2,\dots\}, \quad \{Cat, Dog, Rat\}, \dots.$

x: image, y:rat

D=64x64x3=12'288

Classification: parametric approach

x: image, y:rat

D=64x64x3=12'288

Unknown weights

$f(x,W)$

Score for each possible class

$$\boxed{s=f(x,W)} = W\boxed{x} + b$$

3x1

12'288x1

Assume 3 classes

Classification: parametric approach

x: image, y:rat

D=64x64x3=12’288

$\longrightarrow$ f(x,W) $\longrightarrow$

-10	cat
9.1	dog
4.2	rat

Are the chosen values of the weights W good or bad?

How to choose W?

Loss function

How to quantify goodness of W?

Suppose 4 training examples $\{x_i, y_i\}_{i=1}^{N=4}$ and 3 classes (cat, dog, rat)

				
cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

Loss for each sample

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i)$$

Loss

SVM loss

$$L_i(f(x_i, W), y_i) = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{o.w.} \end{cases}$$




cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$L_1(f(x_1, W), y_1) = 0 + (9.1 - 4.2 + 1) = 5.9$$

$$L_2(f(x_2, W), y_2) = 0 + 0 = 0$$

$$L_3(f(x_3, W), y_3) = 0 + (2.9 - 3 + 1) = 0.9$$

$$L_4(f(x_4, W), y_4) = (2 - 1 + 1) + (10 - 1 + 1) = 12$$

Softmax classifier

Use scores to define probabilities of the classes

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$P(Y = cat|X = x_1) = \frac{e^{s_{cat}}}{\sum_j e^{s_j}} = \frac{e^{-10}}{e^{-10} + e^{9.1} + e^{4.2}} \cong 0$$

Softmax classifier

Use scores to define probabilities of the classes

$$P(Y = k | X = {\color{red}x_i}) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

				
cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$P(Y = dog | X = {\color{red}x_1}) = \frac{e^{s_{dog}}}{\sum_j e^{s_j}} = \frac{e^{9.1}}{e^{-10} + e^{9.1} + e^{4.2}} \cong 0.9926$$

Softmax classifier

Use scores to define probabilities of the classes

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

				
cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$P(Y = rat | X = x_1) = \frac{e^{s_{rat}}}{\sum_j e^{s_j}} = \frac{e^{4.2}}{e^{-10} + e^{9.1} + e^{4.2}} \cong 0.0074$$

Softmax classifier

Use scores to define probabilities of the classes

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$




cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$P(Y = cat|X = x_1) = \frac{e^{s_{cat}}}{\sum_j e^{s_j}} = \frac{e^{-0.2}}{e^{-0.2} + e^{3} + e^{2.9}} \cong 0.021$$

Softmax classifier

Use scores to define probabilities of the classes

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$




cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$P(Y = dog|X = x_1) = \frac{e^{s_{dog}}}{\sum_j e^{s_j}} = \frac{e^3}{e^{-0.2} + e^3 + e^{2.9}} \cong 0.514$$

Softmax classifier

Use scores to define probabilities of the classes

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$P(Y = rat | X = x_1) = \frac{e^{s_{rat}}}{\sum_j e^{s_j}} = \frac{e^{2.9}}{e^{-0.2} + e^{3} + e^{2.9}} \cong 0.4651$$

Softmax classifier: loss function

$$L_i(f(x_i, W), y_i) = -\log\big(P(Y = y_i | X = x_i)\big) = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

cat	-10	1	-0.2	**1**
dog	9.1	**15**	**3**	2
rat	**4.2**	3	2.9	10

$$L_1(f(x_1, W), y_1) = -\log\left(\frac{e^{4.2}}{e^{-10} + e^{9.1} + e^{4.2}}\right) = 4.9074$$

$$L_2(f(x_2, W), y_2) = -\log\left(\frac{e^{15}}{e^{1} + e^{15} + e^{3}}\right) \cong 0$$

$$L_3(f(x_3, W), y_3) = -\log\left(\frac{e^{3}}{e^{-0.2} + e^{3} + e^{2.9}}\right) = 0.6656$$

$$L_4(f(x_4, W), y_4) = -\log\left(\frac{e^{1}}{e^{1} + e^{2} + e^{10}}\right) = 9.0005$$

Classifier: regularization

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

Data loss

Regularization

$$R(W) = \sum_{i,j} W_{i,j}^2$$

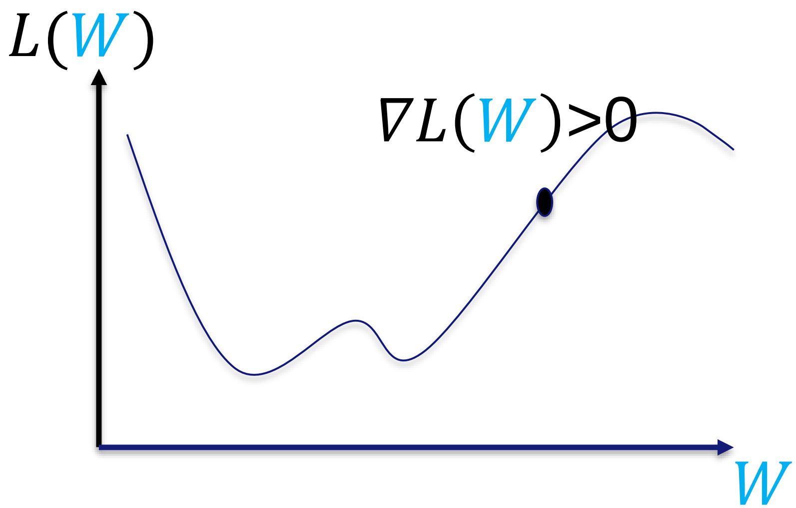
$$R(W) = \sum_{i,j} |W_{i,j}|$$

Optimization

How to find the parameters W minimizing the loss $L(W)$?

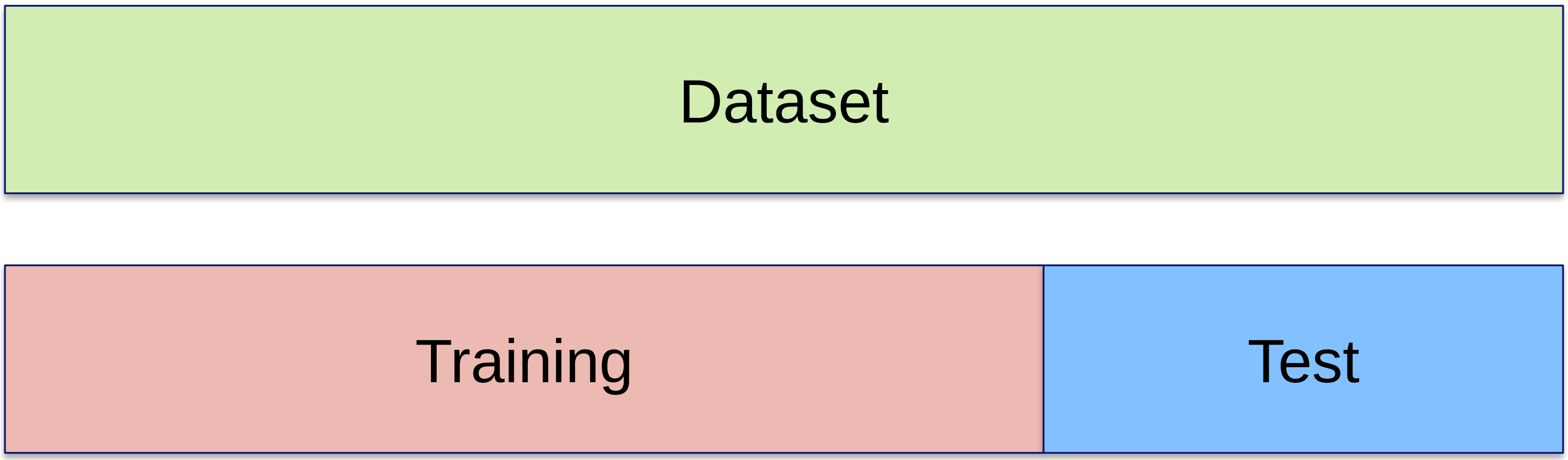
$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

Gradient descent algorithm: go in the opposite direction of the gradient



1. Start with initial value of W
2. Iterate until convergence:
   2.1 $W = W - \gamma \nabla L(W)$

Assessing performance



Use training data to find optimal parameters

Test performance on fresh data

Assessing performance: accuracy

$$\text{accuracy: } \frac{\#\text{correctly classified samples}}{\#\text{ samples}}$$

- If in the test set you have 50 images with dogs and 50 images with cats, are you satisfied if your classifier gives you an accuracy of 95%?
- If in the test set you have 95 images with dogs and 5 images with cats, are you satisfied if your classifier gives you an accuracy of 95%?
- You are training a classifier to detect if a patient is affected or not by COVID 19? In your test set there are 5 COVID-positive patients and 95 COVID-negative patients. Are you satisfied if your classifier gives you an accuracy of 86%?

Assessing performance: confusion matrix

		Actual class	
		Positive	Negative
Predicted class	Positive	TP	FP
	Negative	FN	TN

		Actual class	
		Positive	Negative
Predicted class	Positive	98	25
	Negative	2	75

acc: 86%

sens: 98%

		Actual class	
		Positive	Negative
Predicted class	Positive	90	10
	Negative	10	90

acc: 90%

sens: 90%

Classification pipeline

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)

my_classifier = my_SMV_classifier()  # create an object of class my_SMV_classifier

my_classifier.fit(X_train, y_train) #train classifier

y_test_pred = my_classifier.predict(X_test) #apply trained classifier to test data

accuracy = accuracy_score(y_test, y_test_pred) # compute accuracy

CM = confusion_matrix(y_test, y_test_pred) # compute confusion matrix
```