Spot the Bot

Cristiana Pacheco, Charlie Ringer
& Diego Pérez-Liébana

Overview

- Introduction
  - Overview
  - Who are we?
  - Why does this matter?
- Machine Learning 101
  - What is Machine Learning?
  - Supervised Learning
  - Unsupervised Learning
- The Data
  - What's in the data?
  - Intro to GVGAI
  - Data Set details
- Challenges
  - Code Repository
  - Supervised Challenge
  - Unsupervised Challenge

Who are we?

Cristiana Pacheco

- From Porto, Portugal.
- BSc Computer Games from The University of Essex.
- Doing a PhD at Queen Mary University of London.
- Topics include: believable NPCs and assessment of believability.

Who are we?

Charlie Ringer

- From Isle of Wight, England.
- Worked for several years in the games industry.
- Doing a PhD at Goldsmiths.
- Topics include: affective computing and machine learning.

Who are we?

Dr Diego Pérez-Liébana

- From Madrid, Spain.
- PhD in Computer Science at University of Essex.
- Lecturer in Computer Games and AI at QMUL.
- Topics include: Reinforcement Learning, Tree Search and Evolutionary Computation.

Why does this matter?

- It’s important to understand your players.
- By collecting data on how people play we can determine what type of players you have.
- Help create proper leveled AI to play both against and with your players.
- Tailor experiences to them.
- Distinguish between AI and people as well...

Why does this matter?

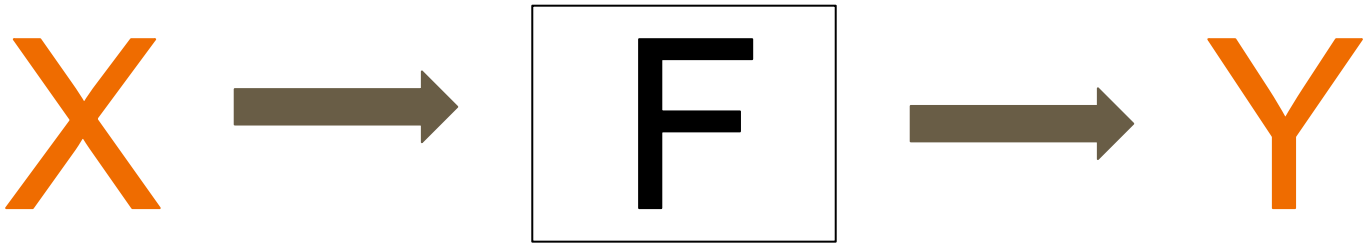
- Botting is still a problem in the industry.
- Players create bots to avoid doing tasks themselves - farming.
- Many disadvantages:
  - This gives them an advantage over honest players.
  - Flooded economy with items.
  - Honest players will have to work twice as hard if not more to achieve the same.
  - Makes players leave the game.
  - ...

Machine Learning 101

Overview

- Introduction
  - Overview
  - Who are we?
  - Why does this matter?
- Machine Learning 101
  - What is Machine Learning?
  - Supervised Learning
  - Unsupervised Learning
- The Data
  - What's in the data?
  - Intro to GVGAI
  - Data Set details
- Challenges
  - Code Repository
  - Supervised Challenge
  - Unsupervised Challenge

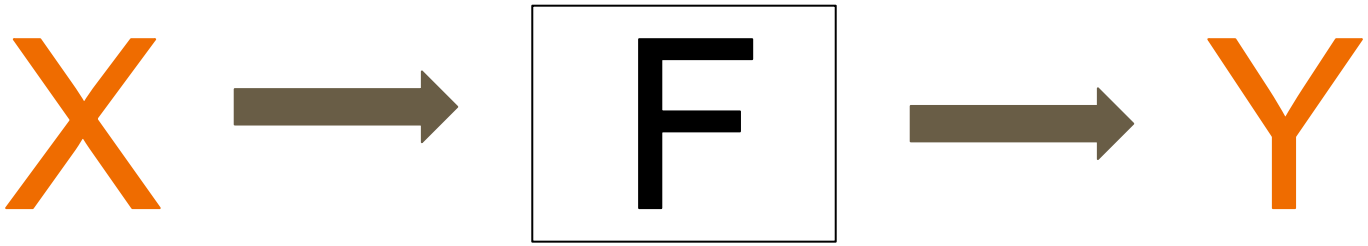
What is Machine Learning?



e.g. {0,1,2,3,4} *f(x) = ?* *e.g. {0,2,4,6,8}*

"Machine Learning is learning an approximate mapping (f) of data (x) to labels (y). Put another way it is learning the function f where f(x) ≈ y." - Charlie Ringer 2018

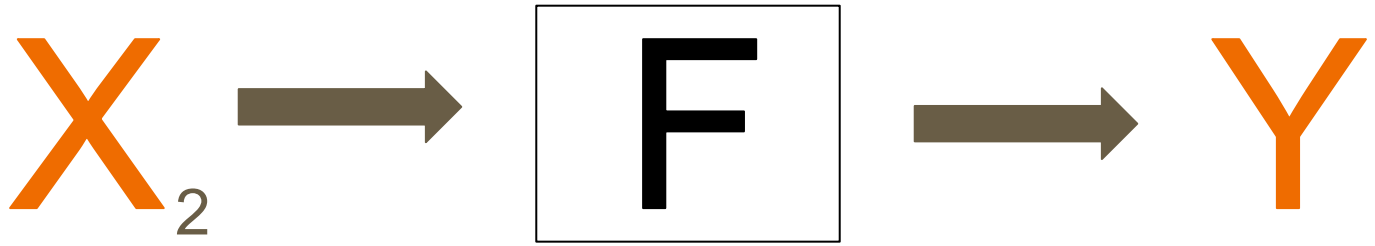
What is Machine Learning?



e.g. {0,1,2,3,4} *f() = x*2* *e.g. {0,2,4,6,8}*

"Machine Learning is learning an approximate mapping (f) of data (x) to labels (y). Put another way it is learning the function f where f(x) ≈ y." - Charlie Ringer 2018

Why is this useful?



e.g. {5,6,7,8,10}

*f() = x*2*

e.g. {?,?,?,?,?}

Now we have learnt f(x) = x*2 we know the labels y for new the data.
X = 5 - Y = 10, X = 6 - Y = 12 etc.

What is data?

- Data can be … anything!
- Data describes something, e.g. data about different type of animals
- Data often has more than one value, are called ‘features’ e.g. Animal data could have number of legs, has tail, has fur, number of eyes etc.
- A collection of data is a dataset e.g. data of 500 cats and 500 dogs is a dataset

What are the type of data?

- Data comes in 3 flavours: Binary, Categorical, Numeric or Ordinal
  - Binary: This animal has fur
  - Categorical: This animal is a mammal
  - Numeric: This animal has 2 eyes
  - Ordinal: This animal is the 4th largest
- Most algorithms cannot handle categorical data so encoding is required.
  - One-Hot Encoding: Turn categories into vectors with mostly 0s and a single 1. E.g.
    - Categories: Mammal, Reptile, Bird
    - Encoding:
    - [1,0,0] - Mammal
    - [0,1,0] - Reptile
    - [0,0,1] - Bird

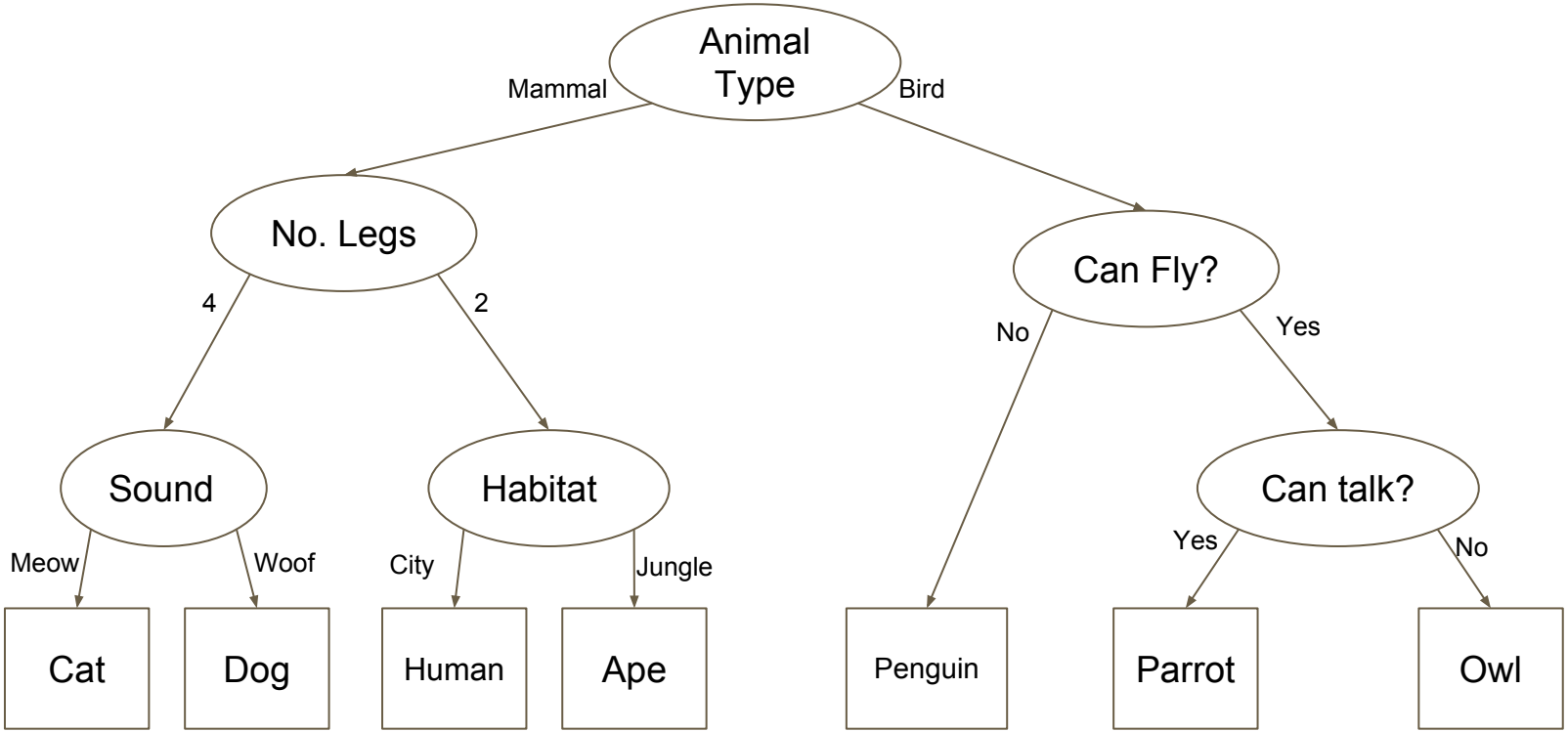
What are labels

- Labels describe the data and are what we want to learn
- Often come in two type: Numeric and Categorical
- Categorial:
  - Often known as classes and require a classification algorithm
  - E.g. for animals the classes could be: Cat, Owl etc. if we want to learn the relationship between having fur, number of legs etc. and what type of animal it is
  - Note: Binary data is a subset of categorical data
- Numeric:
  - What is we want to learn a value? This requires a regression algorithm
  - E.g. We have data on houses (number of rooms, has a garage etc.) and we want to learn the relationship between these features and the value of the house

What are models

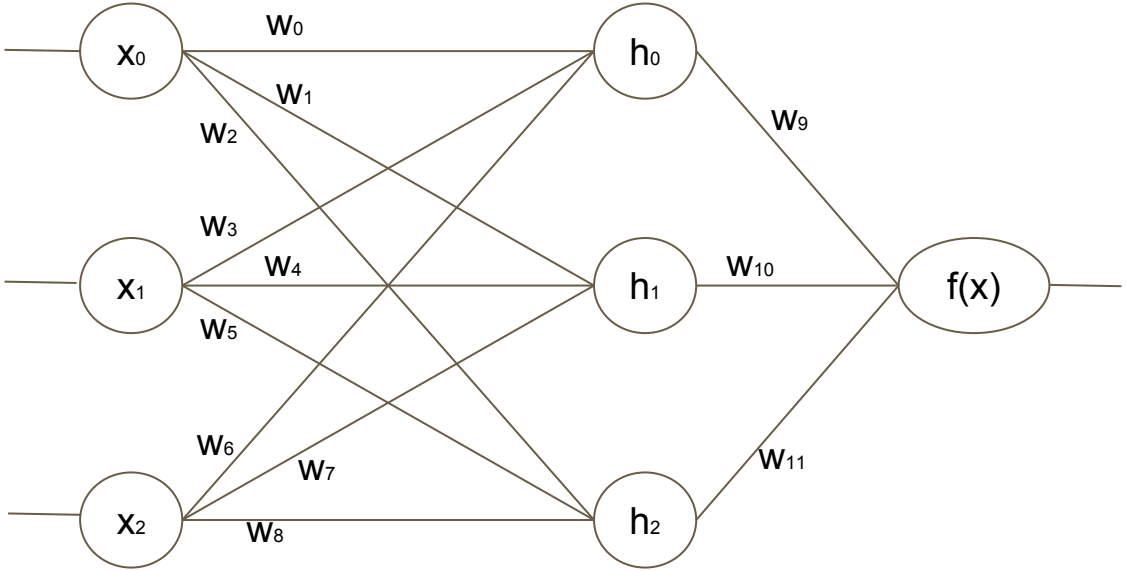
- f() is our “model” (because it models the real world, but probably not perfectly).
- Models are constructed from parameters and an algorithm which takes the data and parameters and produces predicted labels
- Training is the process of learning the best parameters to the model
- There are numerous different models/algorithms e.g. Neural Networks, Decision trees, Linear Regression etc.

Example Model 1: Decision Tree



Example Model 2: Neural Network

Input:
Data (X)



Output:
Predicted
Label ($\bar{Y}$)

$f(x) = g(h_0*w_9 + h_1*w_{10} + h_2*w_{11})$

g = activation function

$h_0 = g(x_0*w_1 + x_1*w_4 + x_2*w_7)$
$h_1 = g(x_0*w_2 + x_1*w_5 + x_2*w_8)$
$h_2 = g(x_0*w_0 + x_1*w_3 + x_2*w_6)$

How to evaluate a model?

- Confusion Matrix:

	Actually AI	Actually Human
Predicted AI	True Positive (TP)	False Positive (FP)
Predicted Human	False Positive (FP)	True Negative (TN)

How to evaluate a model?

- Confusion Matrix:

	Actually AI	Actually Human
Predicted AI	True Positive (TP)	False Positive (FP)
Predicted Human	False Negative (FN)	True Negative (TN)

- Model Accuracy: TP+TN/TP+TN+FP+FN

How to evaluate a model?

- Confusion Matrix:

	Actually AI	Actually Human
Predicted AI	True Positive (TP)	False Positive (FP)
Predicted Human	False Negative (FN)	True Negative (TN)

- Model Accuracy: TP+TN/TP+TN+FP+FN
- Model Precision: TP/TP+FP (w/respect to AI)

How to evaluate a model?

- Confusion Matrix:

	Actually AI	Actually Human
Predicted AI	True Positive (TP)	False Positive (FP)
Predicted Human	False Negative (FN)	True Negative (TN)

- Model Accuracy: TP+TN/TP+TN+FP+FN
- Model Precision: TP/TP+FP (w/respect to AI)
- Model Recall: TP/TP+FN (w/respect to AI)

How to evaluate a model?

- Confusion Matrix:

	Actually AI	Actually Human
Predicted AI	True Positive (TP)	False Positive (FP)
Predicted Human	False Negative (FN)	True Negative (TN)

- Model Accuracy: TP+TN/TP+TN+FP+FN
- Model Precision: TP/TP+FP (w/respect to AI)
- Model Recall: TP/TP+FN (w/respect to AI)
- Model Specificity: TN/TN+FP (w/respect to AI)

Supervised Learning

- Everything we have spoken about so far is called Supervised Learning.
- Supervised Learning is the most common form of Machine Learning and requires you to have both data X and labels Y
- Pros:
  - Clean training data, very easy to learn
  - Popular, meaning a lot of research is done into it
- Cons:
  - Requires labeled data
  - Only gives us insight into the labels

Unsupervised Learning

- What if we have no labels? Then we must use unsupervised learning
- Since we can't learn f(x) = y we must learn something else.
- Auto-encoders:
  - Learns f(x) = x
  - Why? Because doing so can sometimes give us insights into our data
- Clustering algorithms:
  - Learns patterns in the data by finding common groups of similar data
  - Examples: k-means, birch
- Pros:
  - Trainable on any data, even unlabeled
- Cons:
  - It can be hard to know the meaning of that we learn

The Data

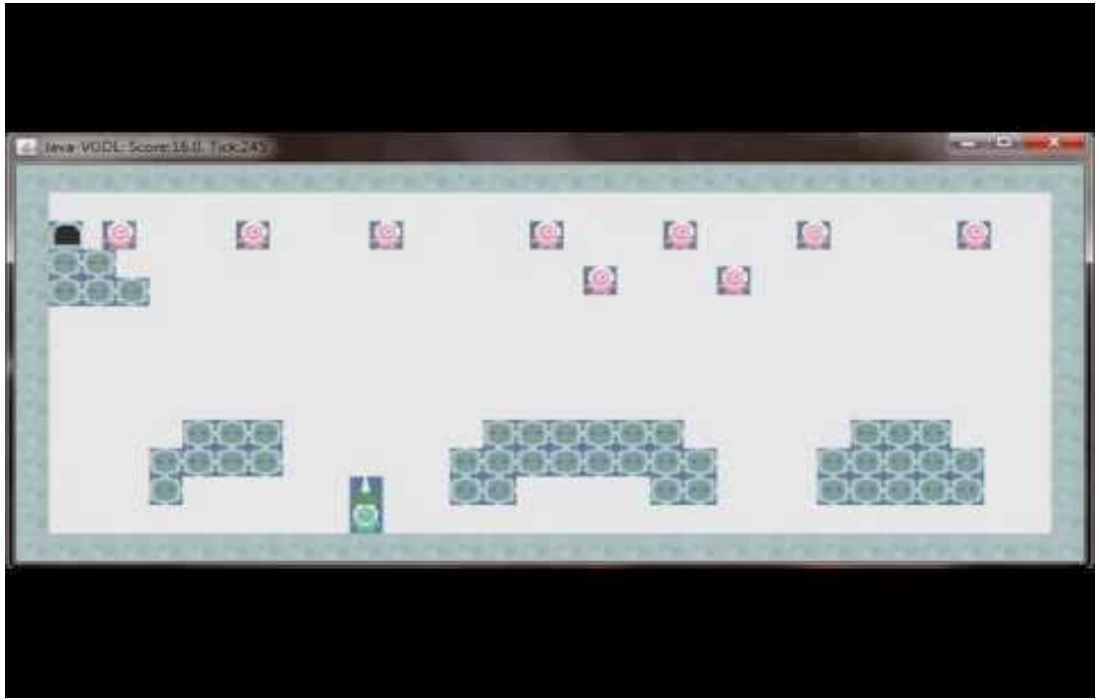
Overview

- Introduction
  - Overview
  - Who are we?
  - Why does this matter?
- Machine Learning 101
  - What is Machine Learning?
  - Supervised Learning
  - Unsupervised Learning
- The Data
  - What's in the data?
  - Intro to GVGAI
  - Data Set details
- Challenges
  - Code Repository
  - Supervised Challenge
  - Unsupervised Challenge

GVGAI

- Framework tackling General Game Playing (GGP).
- Uses a Video Game Description Language (VGDL).
- Provides a forward model - which allows game simulation.
- 2D Games.
- 80 publicly available single-player games (with another 20 not).
- Many competitions run.

GVGAI



Data Set Details

- Data gathered for 3 games - Aliens, Frogs and SeaQuest.
- For each game we gathered data from each player/AI playing 5 levels
- Each player/AI therefore provides 15 logs
- Data gathered using the GVGAI logging tool

Human players

- Data from 6 humans collected
- We have attempted to collect different ranges of expertise - not an expert, some experience and experienced
- Players were given a test level to familiarise themselves with the game before they were recorded

AI Agents

- YOLOBot - High Skill - Ensemble method which selects an algorithm
- YBCriber - High Skill - Iterative Width with dynamic lookahead
- SampleMCTS - Medium Skill - Vanilla MCTS
- CatLinux - Medium Skill - Used evolutionary computing
- OneStepLookAhead - Low Skill - Just looks for the greedy best next move
- sampleGA - Low Skill - Used evolutionary computing

What's in the Data?

- Rows are game logs
- Columns:
  - Human or AI
  - Skill
  - GameID
  - LevelID
  - PlayerID
  - Seed (Random and can be ignored)
  - Result (Win or Loss)
  - Score
  - Game Ticks
  - Moves for each tick

Challenges

Overview

- Introduction
  - Overview
  - Who are we?
  - Why does this matter?
- Machine Learning 101
  - What is Machine Learning?
  - Supervised Learning
  - Unsupervised Learning
- The Data
  - What's in the data?
  - Intro to GVGAI
  - Data Set details
- Challenges
  - Code Repository
  - Supervised Challenge
  - Unsupervised Challenge

Code Repository

- Find at https://github.com/cring002/SpotTheBotWorkshop
- Contains data:
  - game_logs - Raw data, one file per game
  - data.csv - Combined Data in a format used in the code
- Contains (useful) code:
  - data_loader.py - Loads the data. Don't need to use unless you write your own code
  - classifiers.py - Contains supervised classification algorithm
  - clusterers.py - Contains unsupervised clustering algorithm
- Contains other tools:
  - Can be used to replicate the data here, but not needed for this workshop

data_loader.py

Loads the data (and is therefore used by the other two files).

- load_time_series(file, trainPercent):
  - Loads data as a sequence of moves only.
  - file -> data file (data.csv)
  - trainPercent -> Amount to retain as training data (leave at 0.8 for classifiers and 1 for clusterers)
- load_features(file, trainPercent):
  - Loads data as a set of 9 handcrafted features: win/lose, no. ticks, score, % moves = up, % moves = down, % moves = left, % moves = right , % moves = use, % moves = nil
  - file -> data file (data.csv)
  - trainPercent -> Amount to retain as training data (leave at 0.8 for classifiers and 1 for clusterers)

classifiers.py

- 7 different classifiers:
  - k-Nearest Neighbour
  - Logistic Regression
  - Support Vector Classifier
  - Decision Tree
  - Gaussian Naive Bayes
  - Multi-Layer Perceptron
  - Recurrent Neural Network (GRU)
- Script has 3 params: --data -> data file, --data_model -> time series or features, --model -> classifiers of your choice
- Example Command:
  - `python3 classifiers.py --data=data.csv --data_model=features --model=knn`

classifiers.py output

- Descriptive Values:
  - Model Accuracy
  - Model Precision (w/respect to AI)
  - Model Recall (w/respect to AI)
  - Model Specificity (w/respect to AI)
- Challenge Score 1 & 2
- Confusion Matrix:

	Actually AI	Actually Human
Predicted AI		
Predicted Human		

clusterers.py

- 3 different clusterers:
  - K-means
  - Spectral
  - Birch
- Script has 3 params: --data -> data file, --data_model -> time series or features, --model -> clusterer of your choice
- Example Command:
  - `python3 clusterers.py --data=data.csv --data_model=features --model=kmeans`

clusterers.py output

- Challenge Score:
  - Highest accuracy value between clusters and ground truth
- Two 3d graphs:
  - The 3 dimensions are calculated using a feature decomposition called Principal Component Analysis.
  - Ground Truth - represents the labels of the data. Each colour (purple and yellow) represent a different class
  - Clusters - represents the learnt clusters. Each colour (purple and yellow) represent a different cluster
  - Approx comparison - does the split in the leant clusters match the split in the ground truth data

Modifying our code

- Most of our models uses default parameters, you are free to change these!
- Often small tweaks to model parameters can result in large improvements
- Non-neural networks are built using sklearn, check out scikit-learn.org/ to see what parameters can be changed
- Neural networks are build using Keras, check out keras.io to see what parameters can be changed

Supervised Challenges

Develop the best classifier

- Can use our models/features/parameters
- Also free to develop your own in any language you like

Judging Criteria:

- Accuracy + Precision: No humans detected but AI can go undetected
- Accuracy + Recall: No AI undetected but humans can be misclassified

Unsupervised Challenges

Develop the best classifier using unsupervised learning

- A much more difficult challenge
- Can use our models/features/parameters
- Also free to develop your own in any language you like

Judging Criteria:

- As classes are not known we assuming that the clusters would get assigned the right label
- Accuracy only as this is a much harder problem

Good luck with your models!

https://github.com/cring002/SpotTheBotWorkshop