An Informal Introduction to Machine Learning for Geometry

Ed Hirst
Queen Mary, University of London

BRIDGES Workshop

June 23

e.hirst@qmul.ac.uk

Contents:

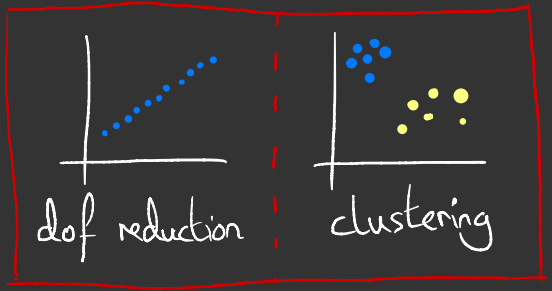
① Machine Learning

Supervised

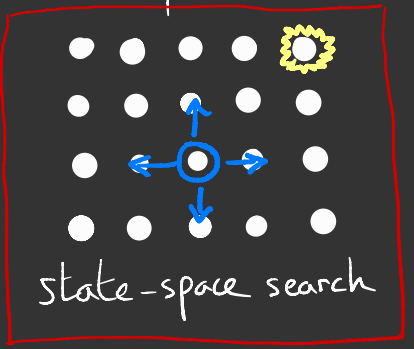
f: inputs $\longmapsto$ outputs

function fitting

Unsupervised



Réinforcement



Semi-supervised

algebraic ⟷ algébrique

geometry ⟷ géométrie

algebraic geometry ⟶ ???

generative models

Data format :

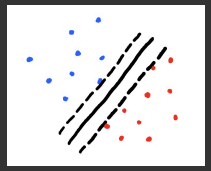
Supervised	⟶	input & output
Unsupervised	⟶	input only
Semi-supervised	⟶	large input & small output
Réinforcement	⟶	input only : state & action

② Supervised Learning

- **Classification** — output ⇒ finite set — Loss ~ cross-entropy
- **Regression** — output ⇒ continuous range — Loss ~ mean squared error

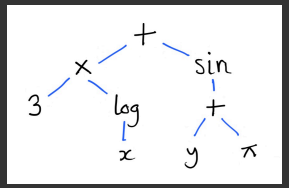
Techniques:

* support vector machine (binary classification)



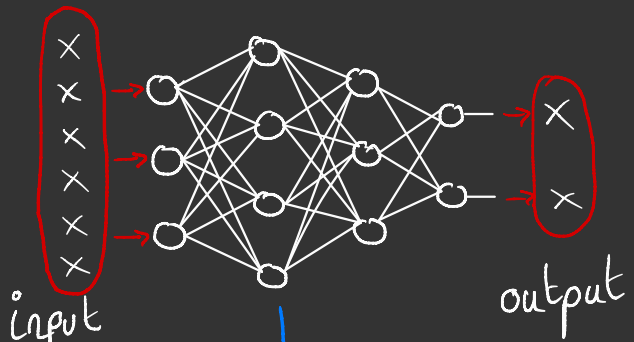
* linear regression (regression)

* symbolic regression (regression)



⇒ $3\log x + \sin(y+\pi)$

* neural network (both)



neuron

$$\phi\left[\sum_i (w_i \cdot x_i) + b\right]$$

non linear — linear

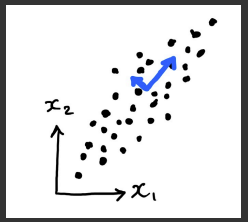
③ Unsupervised Learning

Degree of Freedom Reduction

Clustering

Techniques:

* principal component analysis

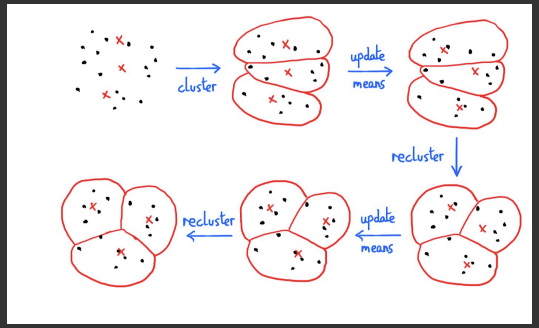


(x_1, x_2)

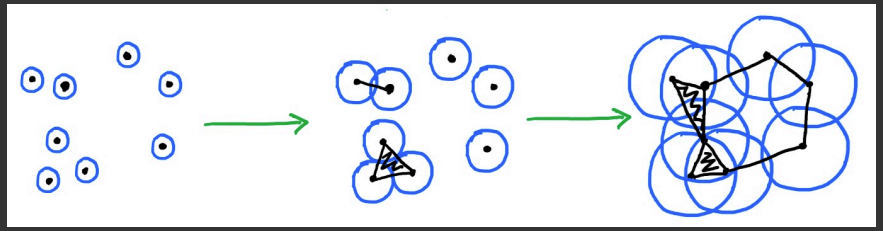
↓

$(x_1 + x_2, 0)$

* K-means clustering



* topological data analysis



$S_i = \{\underline{x}^1, \underline{x}^2, \ldots\}$

↓

$\{\beta_0, \beta_1, \beta_2, \ldots\}$

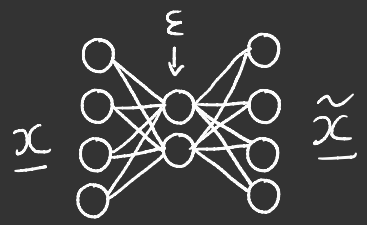
④ Semi-supervised Learning

Generative Models ⇒

Natural Language
Image Interpolation
Data Diffusion

Techniques:

* variational autoencoders



$\mathbb{R}^4 \to M^2 \to \mathbb{R}^4$

generator

* transformers (ie GPT)

...infers syntactical rules & "knowledge" from context of text data

x_1: "I Love mathematics"

x_2: "She loves mathematics"

word 1 correlates with conjugation of word 2 in the sentence set $\{x_i\}$

⑤ Reinforcement Learning

Agent

train a policy

$P(a^t = a_i \mid s^t = s_j)$

State space

states $\{s_i\}$

Action space

actions $\{a_i\}$

Reward

$R_{a_i^t}(s_j^t, s_k^{t+1})$

Transition

$P(s^{t+1} = s_i \mid s^t = s_j, a^t = a_k)$

Techniques:

* **Genetic Algorithms**
  - ~ set of agents, each at a state with a score (reward)
  - ~ select best agents to "breed" their states, and randomly mutate (action) updating agents to next states

* **Deep-Q Learning**
  - ~ use a NN to predict rewards for each action for the states
  - ~ policy selects action with ↑Q
  - ~ update Q NN over agent runs

⑥ Examples

Supervised:

predicting hodge #s : CY → $h^{i,j}$

... from configuration matrices — 2007.15706, 2202.02164

... from weight systems — 1706.02714, 2112.06350

predicting CY metrics — 2205.13408, 2210.12520

predicting Hilbert series parameters from expansion coefficients — 2103.13436

Unsupervised:

separate phases of NN functions learning CY → hodge #s — 1809.02547

Reinforcement:

searching string vacua for standard models ... with RL / GA resp. — 1903.11616, 2306.03147

finding new 5d reflexive polytopes — 2306.06159