Learning from Demonstration

Oana Cocarascu & Helen Yannakoudakis

Department of Informatics
King's College London

Today

References

[Argall] Brenna D. Argall, Sonia Chernova, Manuela Veloso and Brett Browning (2009), A survey of robot learning from demonstration, *Robotics and Autonomous Systems*.

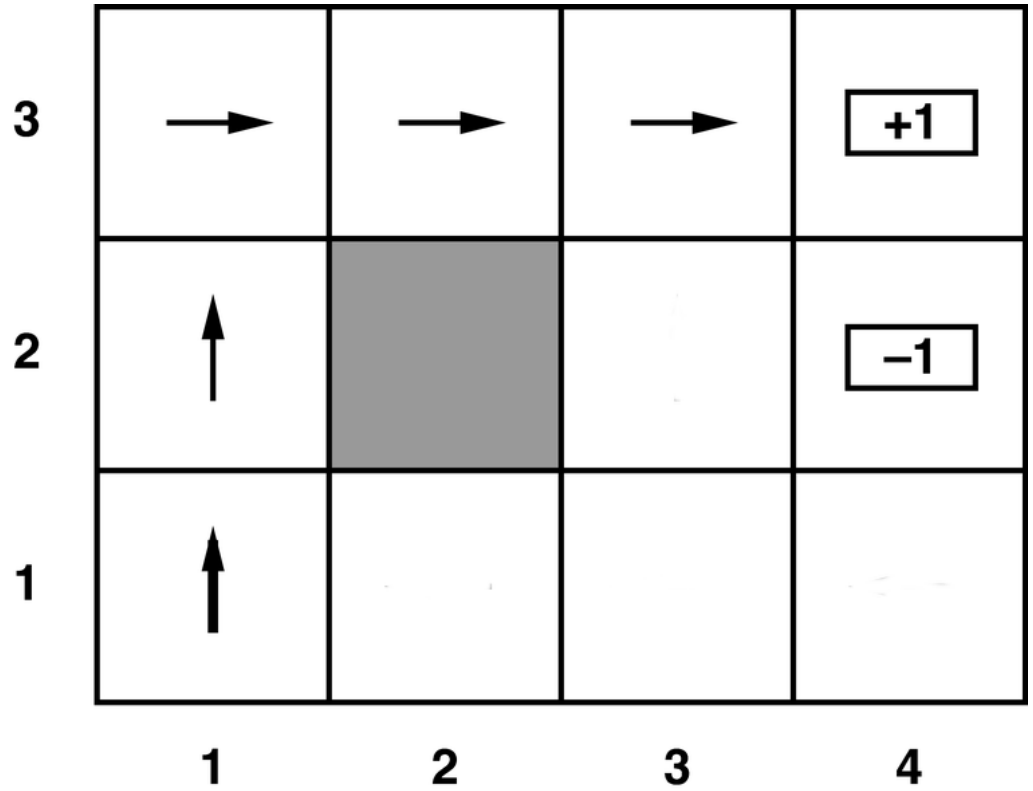
[Chernova] Sonia Chernova and Manuela Veloso (2009), Interactive Policy Learning through Confidence-Based Autonomy, *Journal of Artificial Intelligence Research*, 34, pp1-25.

Learning from Demonstration

- Problem: learn a mapping from (world) state to actions.
- This mapping is called a policy.
- Same as in reinforcement learning (RL).
- In RL, learning is from experience.
- In Learning from Demonstration (LfD), learning is from *examples*.
- LfD is a *supervised* learning of policies.
- An example in LfD is a sequence of *state-action* pairs.
  - Model-free learning.
- The policy is derived only from those states encountered during learning.
- So the policy is *partial*.

KING'S College LONDON

Example



- Given these states, an example of demonstration is to be guided on this good path.

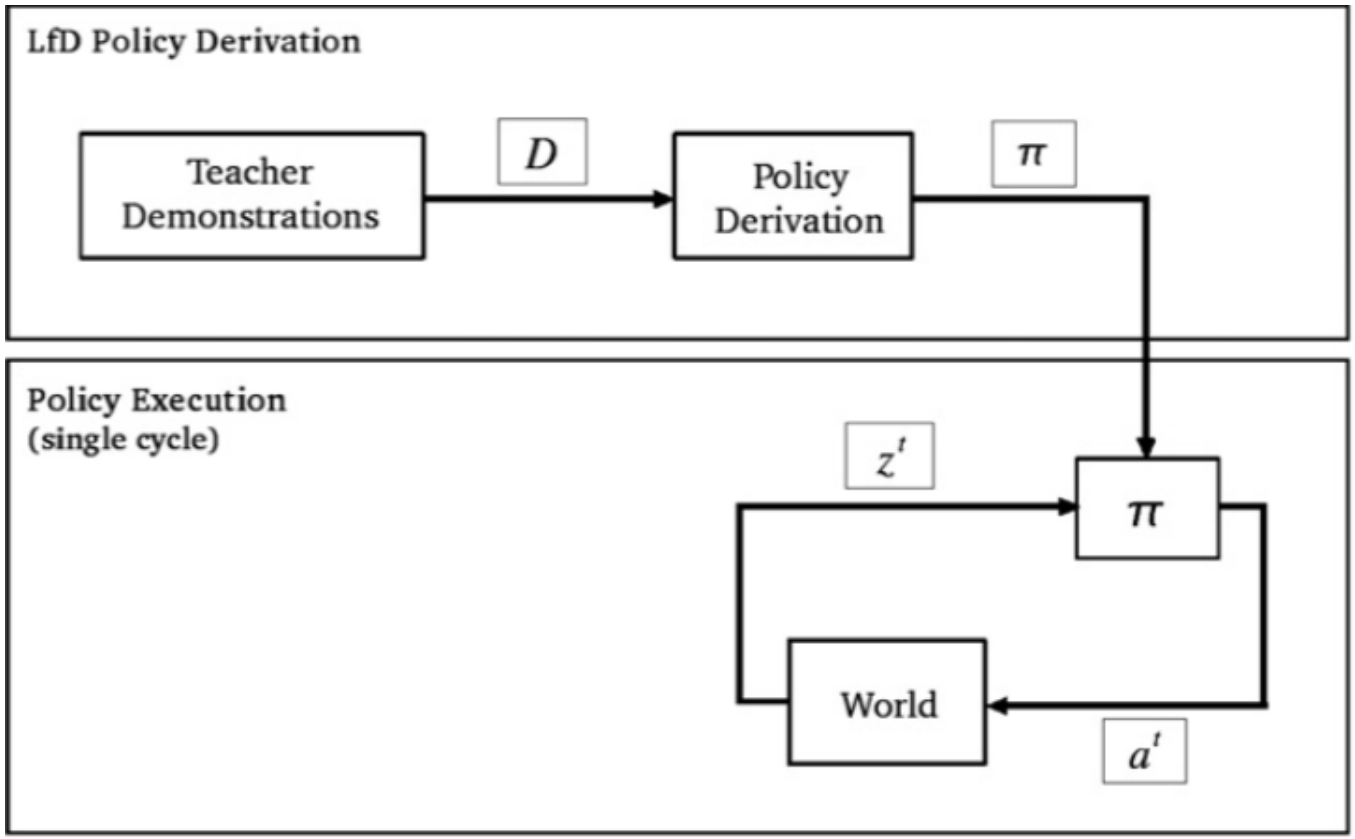
Example

https://vimeo.com/13387420

- A demonstration of how to flip pancakes.

Learning from Demonstration

- LfD is a form of supervised learning.



[Argall Fig 1]

Formal definition

- S = set of states
- A = set of actions
- Transition function:

$$T(s'|s,a) : S \times A \times S \rightarrow [0,1]$$

- Z = observed state, accessed through mapping:

$$M : S \rightarrow Z$$

- If state is fully observable, then $M = I$ (identity)
- Policy selects actions:

$$\pi : Z \rightarrow A$$

KING'S College LONDON

Formal definition

- Rather like an MDP.
- But with the extra indirection of:

$$\pi : Z \rightarrow A$$

rather than:

$$\pi : S \rightarrow A$$

- Partially observable.

Learning from Demonstration

- In LfD, there is an explicit teacher who provides demonstrations that the learner trains on.
- Demonstration $d_j \in D$, such that:

$$d_j = \{(z_j^i, a_j^i)\}$$

where $z_j^i \in Z$ and $a_j^i \in A$ and:

$$i = 0 \ldots k_j$$

- This is the sequence of state/action pairs discussed above.
- The set of demonstrations, D, is what differentiates LfD from other learning methods.

Varieties

- Demonstration-based learning methods include:
  - Learning from Demonstration (LfD)
  - Learning by Demonstration (LbD)
  - Programming by Demonstration (PbD)
  - Learning from Observations
  - ...
- Fundamentally the same ideas.

Process

- Teacher demonstrates execution of some behaviour.
- Learner receives these demonstrations and derives a policy able to reproduce the demonstrated behaviour.

1. Make design choices
    1. Demonstration approach
    2. Problem space representation
    3. Policy derivation
2. Execution
    1. Gathering examples
    2. Deriving policy

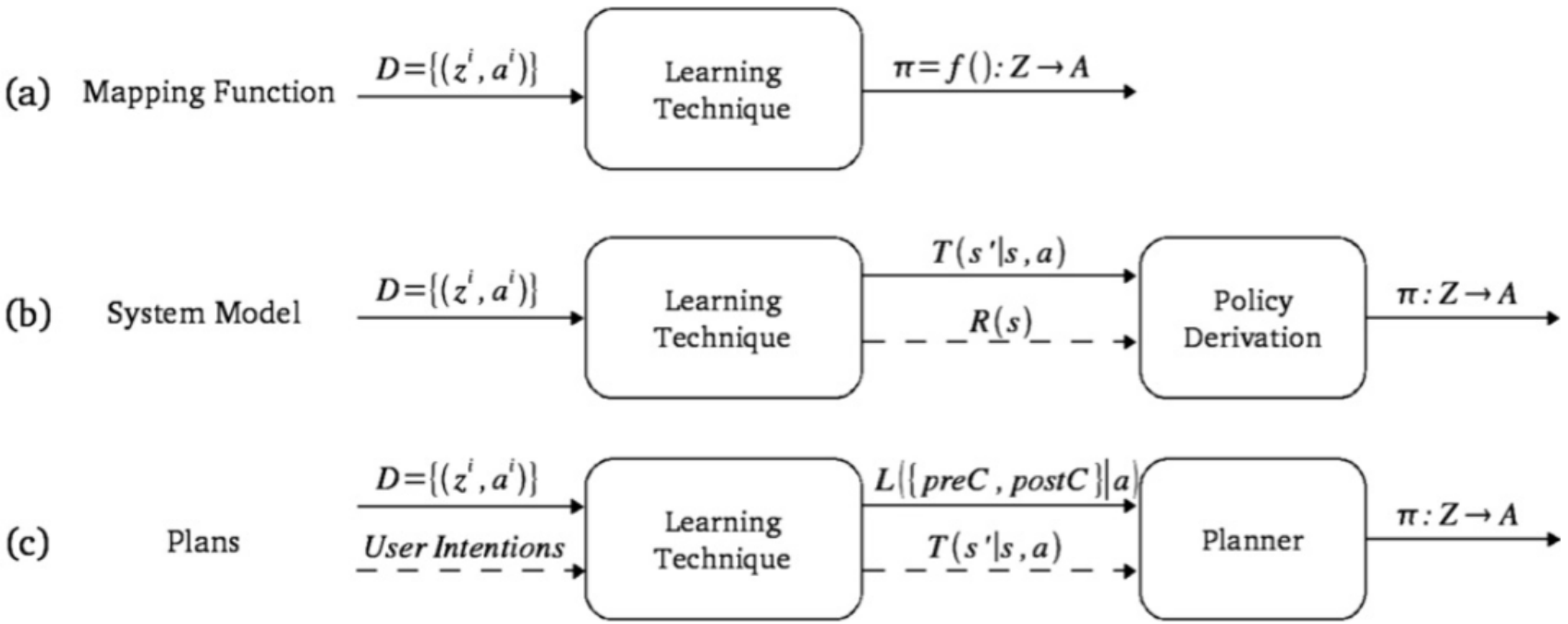
Design choices - Demonstration approach

- Choice of demonstrator
  - Who demonstrates
  - Usually a human
  - Human performs the task themselves
    (Robot observes)
  - Human teleoperates the robot to perform the task
    (Robot observes)
- Demonstration technique
  - How are the demonstrations gathered
  - Batch learning: Policy learnt once, after all data is gathered.
  - Interactive learning: Policy learnt incrementally, as data is gathered.

Design choices - Problem space representation

- Continuity.
- States: continuous or discrete.
- Actions: continuous or discrete.
- Often defined by domain, but some scope for choice.

Design choices - Policy derivation



[Argall Fig 2]

Policy derivation - Mapping function

- D is a dataset of state/action pairs.
- Directly approximate mapping from observed state to actions.

$$f() : Z \rightarrow A$$

- Classification problem.
- From state, measured by robot, pick action.

Policy derivation - System model

- D is used to determine model of the world $T(s'|s,a)$ and a reward function $R(s)$.
- These are then used to derive policy π.
- Reinforcement learning.
  But a bit different from what we have seen already.

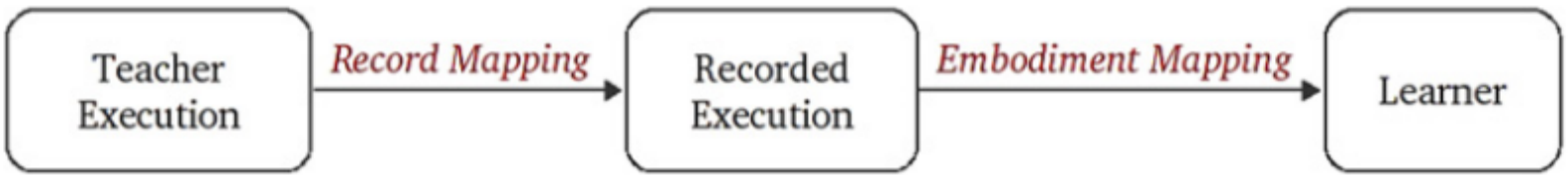
Policy derivation - Plans

- Apply AI planning methods.
- D, and optionally model of users' intentions, used to learn rules.
- Rules associate pre-conditions and post-conditions with each action:

$$L(\{preC, postC\}|a)$$

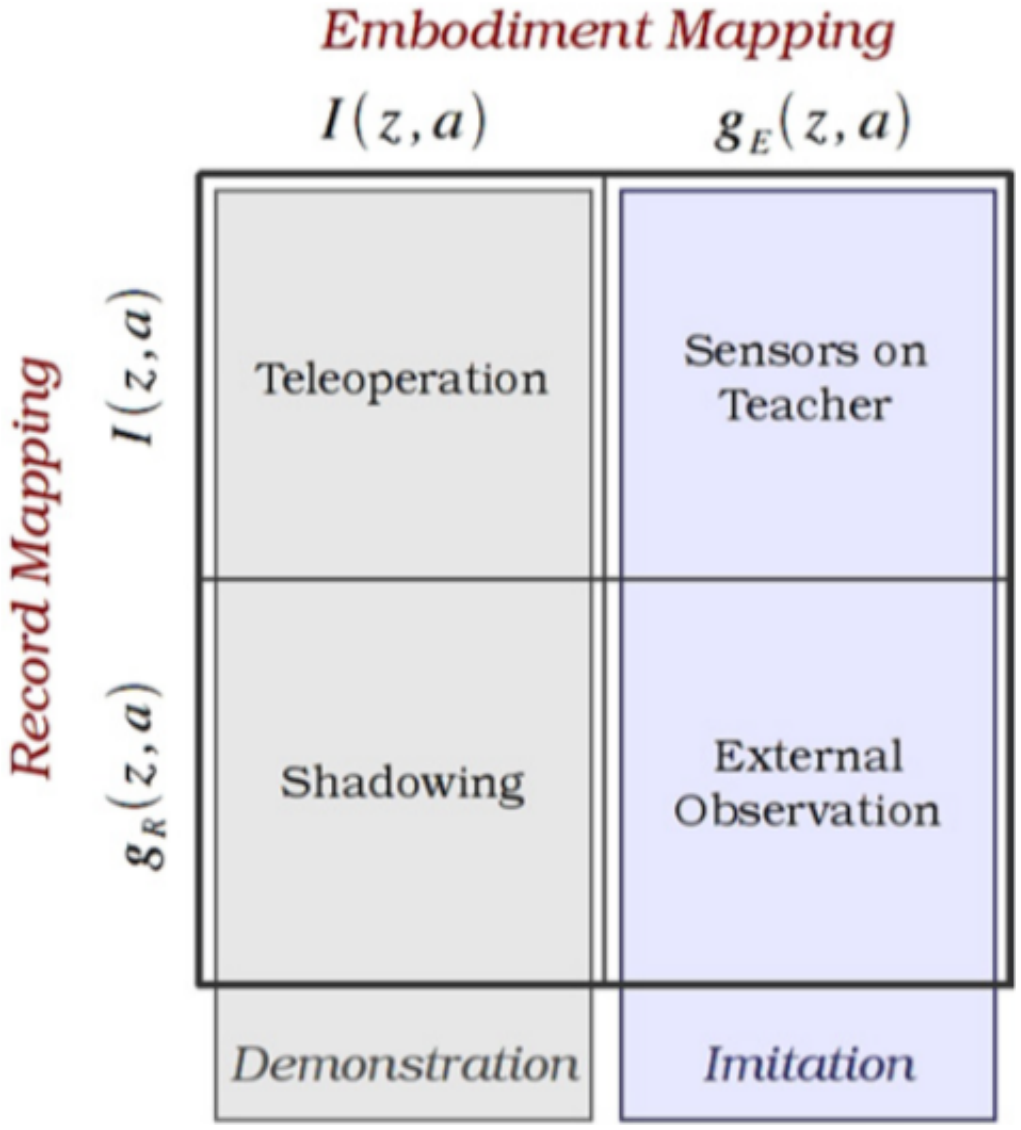
  and optionally world model $T(s'|s, a)$.
- Planner then creates policy π.

Gathering examples - Collecting Data



[Argall Fig 3]

Gathering examples - Correspondence



[Argall Fig 4]

Gathering examples - Correspondence

- Record mapping: Teacher execution → recorded execution.
  - Are the states/actions experienced by the teacher during demonstration execution recorded in the data set?
    - If yes, then we have an identity record mapping $I(z, a)$.
    - Otherwise, we have some *record mapping* function $g_R(z, a)$.
  - Does the teacher teleoperate the robot, or does the robot observe the teacher?
- Embodiment mapping: Recorded execution → learner.
  - Are the states/actions recorded in the data set those that the learner will observe/execute?
    - If yes, then we have an identity embodiment mapping $I(z, a)$.
    - Otherwise, we have some *embodiment mapping* function $g_E(z, a)$.
  - Does the teacher teleoperate the robot, or is the robot given the movements made by the teacher?

KING'S College LONDON

Gathering examples

- How is the data set built?
- Demonstration.
- Imitation.

Gathering examples - Demonstration

- Identity/no embodiment mapping

$$g_E(Z, a) \equiv I(Z, a)$$

- Demonstration is performed on actual robot.
- Teleoperation: human teacher operates the robot and robot's sensors record data

$$g_R(Z, a) \equiv I(Z, a)$$

  - Joystick operation.
  - Kinesthetic teaching.
- Shadowing: robot operates itself but attempts to mimic human

$$g_R(Z, a) \not\equiv I(Z, a)$$

  - Mimic motions, but record own motions.
  - Follow a teacher in a navigation task.

KING'S College LONDON

Gathering examples - Imitation

- Non-identity embodiment mapping

$$g_E(Z, a) \not\equiv I(Z, a)$$

- Demonstration is performed by human and observed on actual robot.
- Sensors on teacher

$$g_R(Z, a) \equiv I(Z, a)$$

so get direct data from demonstrator, but not the robot.

- External observation

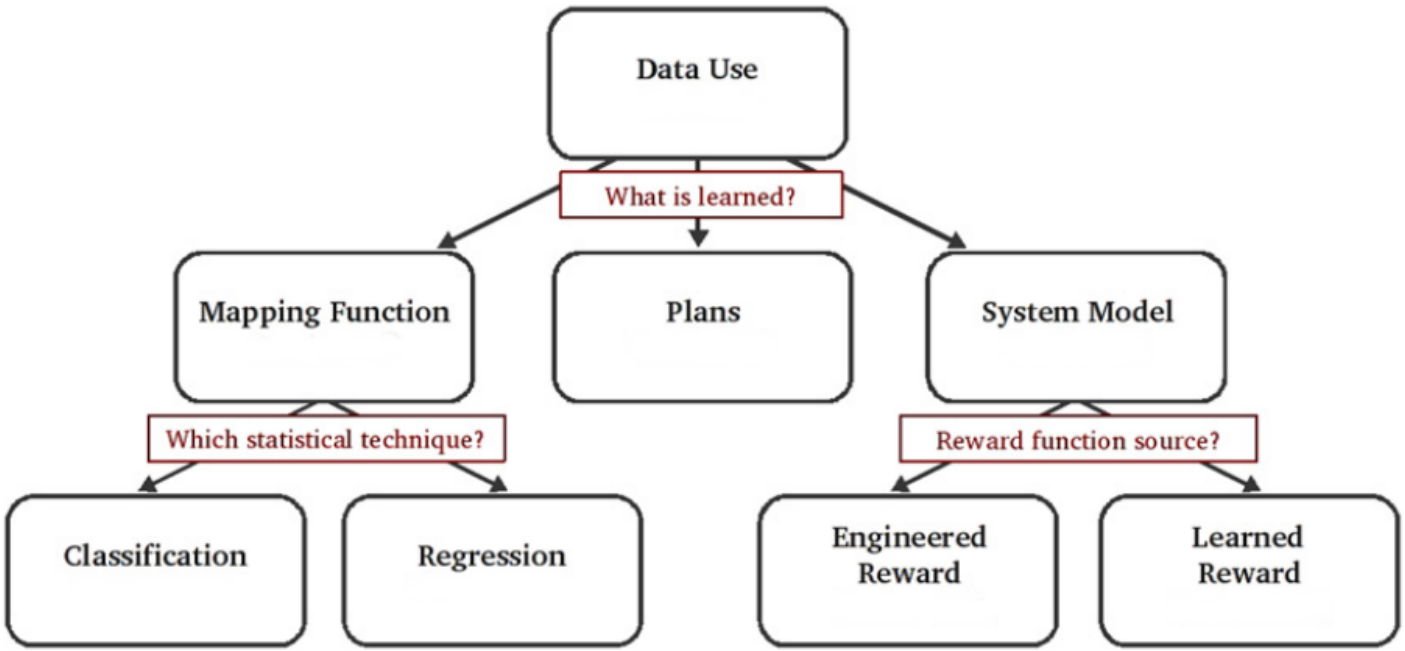
$$g_R(Z, a) \not\equiv I(Z, a)$$

so sensor data is external to demonstrator and robot.

KING'S College LONDON

Deriving policy

- Given the data, how do we decide what to do?
- Depends on what kind of policy we want.



[Argall Fig 6]

Deriving policy - Mapping

- Picking the right action directly from the data.
- Classification
  - Discrete actions
- Regression
  - Continuous actions

Classification

- Simplest idea is to pick an individual low-level action.
  - Do this in many different ways:
    - Gaussian Mixture Models (GMMs)
    - k-Nearest Neighbours (kNN)
    - Decision trees
    - Bayesian networks
  - Low-level actions are then put together in a sequence.
    - Hidden Markov Model (HMM)
    - Dynamic Bayesian Network (DBN)
- More complex idea starts with higher level actions.
  - High level actions are sequences.
  - Classify between sequences.

Regression

- Use any model that maps to continuous values.
- Could just use kNN.
- Locally Weighted Regression.
- Neural Networks.

Deriving policy - Plans

- Represent desired behaviour (i.e., policy choice) as a plan.
- Sequence of actions that lead from initial state to goal state.
- Plan has:
  - pre-conditions
  - post-conditions
  - world state $T(s'|s,a)$
  - action sequence
- Of course, does not handle non-determinism very well.

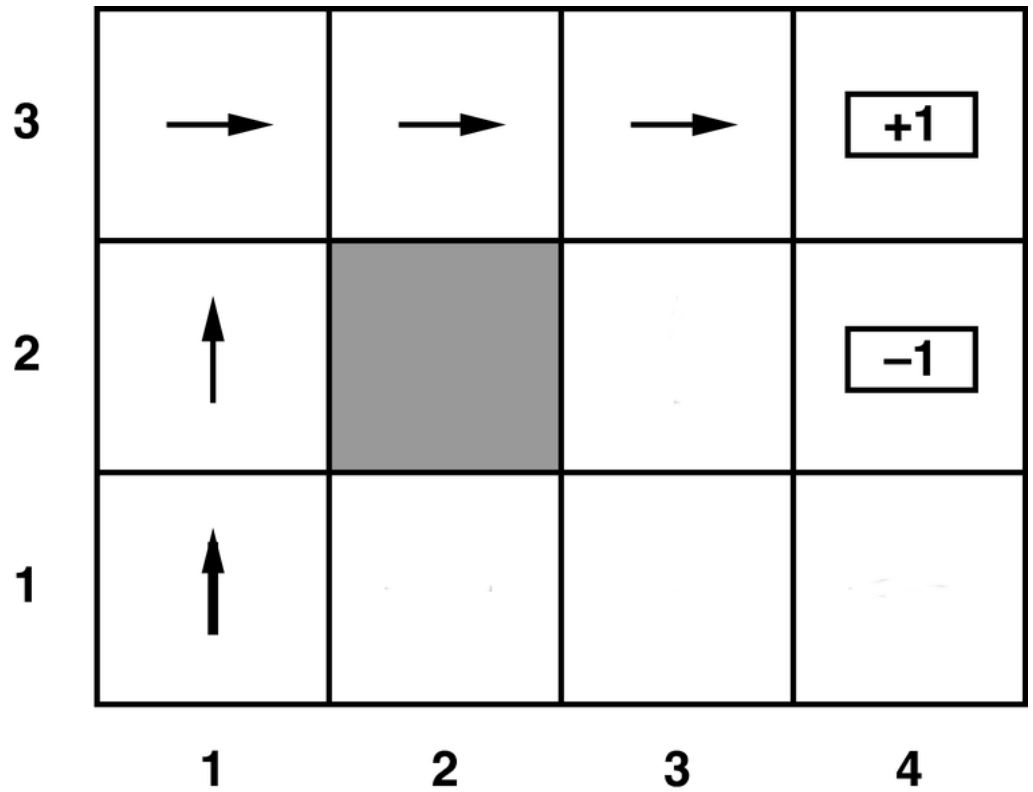
Deriving policy - System model

- Derives policy

$$\pi : Z \rightarrow A$$

  from model of the world $T(s'|s, a)$ and reward $R(s)$.
- Reinforcement learning.
- Aim is to maximise cumulative reward over time.
- Reward function could be engineered or learn.

Deriving policy - System model



- Start with more policy than in active learning, less policy than in passive learning.
- Start with the partial policy and exploration.

Rewards

- Engineered reward
  - Corresponds to the usual case.
    - Sparse rewards for particular states.
  - LfD has concentrated on helping to locate rewards, reducing blind exploration.
  - Demonstration can highlight rewards.
  - “Ask for Help” requests advice when all actions are approximately equally valued.
- Learned reward.
  - Try to learn a reward function that generalises from individual rewards.
  - Function approximation in RL.

KING'S College LONDON

Confidence-Based Autonomy

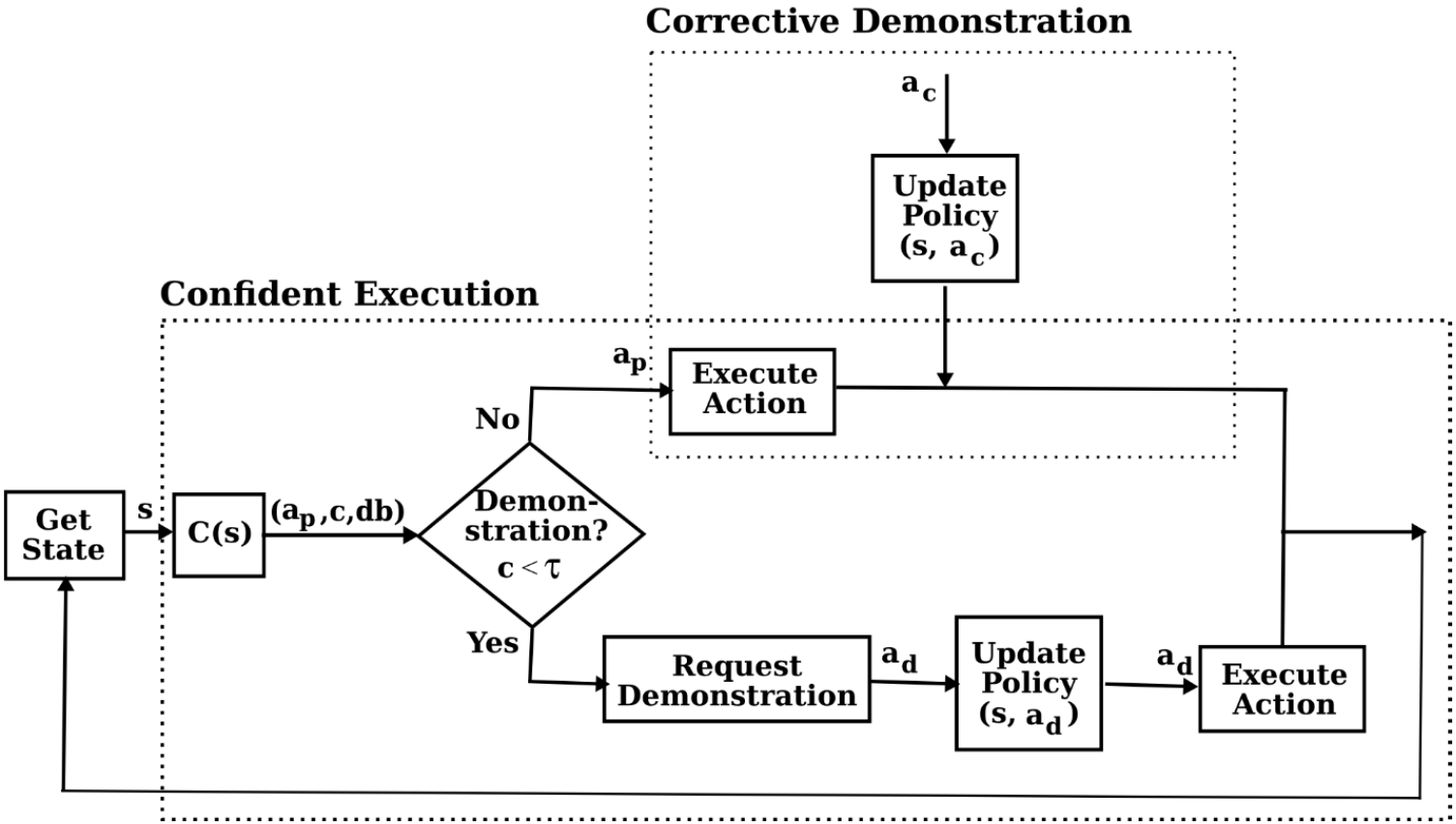
- Confidence-Based Autonomy [Chernova].
- Interactive algorithm for LfD policy learning.
- Initially, learner has no knowledge.
- Learner acquires policy through observing demonstrations.
- Learner practices the task.
- If learner encounters unfamiliar state, then learner can ask for more demonstrations.

KING'S College LONDON

Confidence-Based Autonomy

- Learner's actions: $a \in \mathcal{A}$.
- Underlying Markov Decision Process (MDP).
- Demonstrations: (s_i, a_i), where $a_i \in \mathcal{A}$, selected by teacher.
- Two components:
  - Confident execution — demonstrations triggered by learner's confidence.
  - Corrective demonstration — demonstrations triggered by teacher correcting errors.
- Goal is for learner to generalise from demonstrations and learn a policy.

Confidence-Based Autonomy



[Chernova Fig 1]

Confident execution

- Robot's behaviour policy is represented by a classifier:

$$C : s \rightarrow (a, c, db)$$

where

  - s = state
  - $a \in \mathcal{A}$ (action)
  - c = action selection confidence
  - db = decision boundary

- Policy is trained on state vectors s_i (inputs) and actions a_i (labels).

KING'S College LONDON

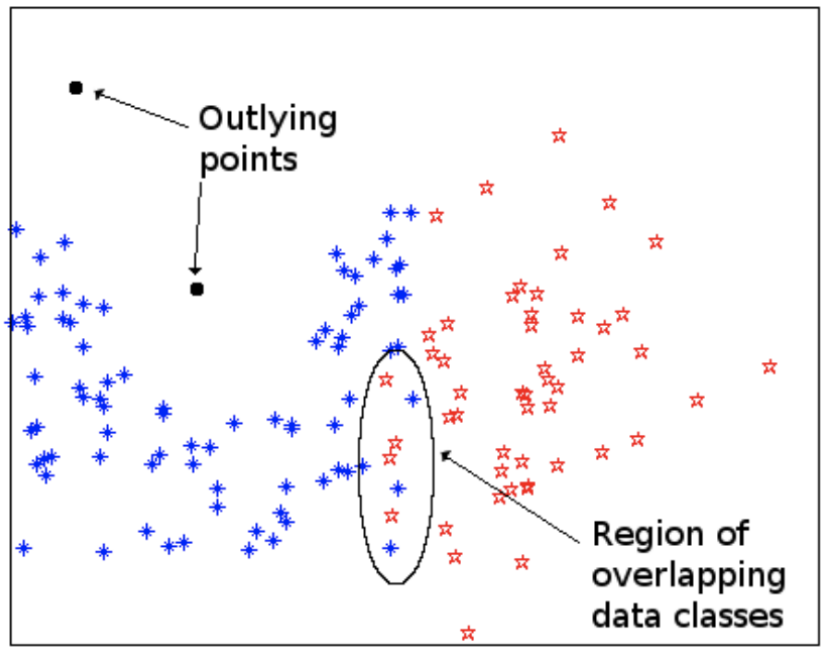
Confident execution & Corrective demonstration

- Confident execution
  - if $c > \tau$, learner executes a_p
  - otherwise, learner receives a demonstration
    Demonstrated action: a_d
- Corrective demonstration
  - Teacher may decide to provide a corrective demonstration
    Corrective action: a_c

KING'S College LONDON

Confident execution

- State space divided into two regions:
  - High confidence → autonomous execution
  - Low confidence → demonstration
- Two situations where confidence may be low:
  - Unfamiliar (outlying points)
  - Ambiguous (overlapping regions)



[Chernova Fig 2]

Confident execution

- Two evaluation criteria are used to decide between autonomous execution and demonstration:

$$\begin{array}{llll} \textit{if } (c > \tau_{conf}) \wedge (d < \tau_{dist}) & \textit{then} & \textit{autonomous} \\ & \textit{else} & \textit{demonstration} \end{array}$$

- Nearest neighbour distance
  - Euclidean distance from current state to most similar training data point.
  - Threshold $\tau_{dist} = 3\times$ average nearest-neighbour distance across data set of demonstrations.
- Classification confidence
  - Threshold τ_{conf} more complicated to compute.
  - Different threshold computed for each decision boundary.
- Initially $\tau_{dist} = 0$ and $\tau_{conf} = \infty$.

KING'S College LONDON

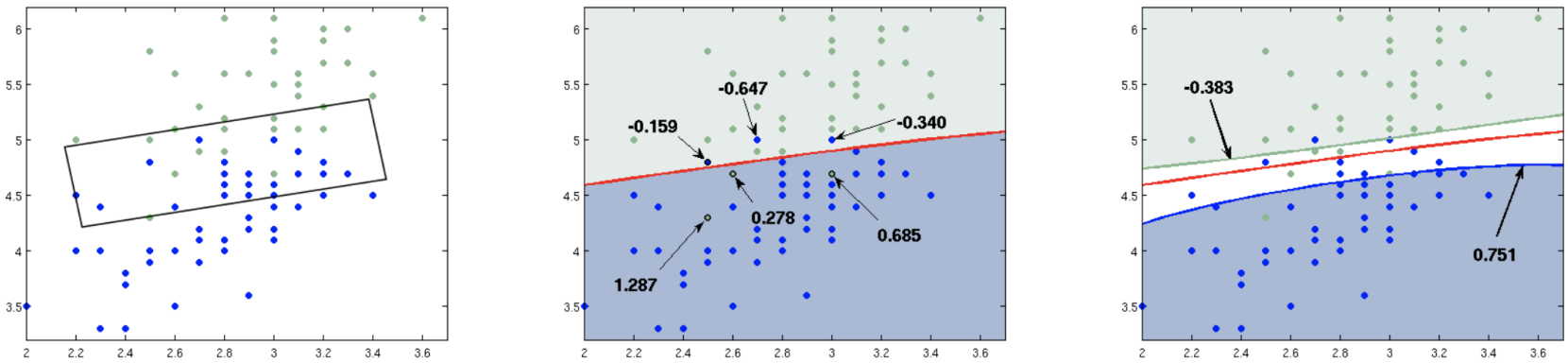
Confident execution

- Classified point = (o, a, a_m, c) where
  - o = original observation
  - a = demonstrated action
  - a_m = model-selected action
  - c = model action confidence (returned by classifier)
- $M_i = \{(o, a_i, a_m, c) | a_m \neq a_i\}$
  Set of all points mistakenly classified by decision boundary i.
- Confidence threshold is average confidence of misclassified points:

$$\tau_{conf_i} = \frac{\sum_{j=1}^{M_i} c}{|M_i|}$$

- If $M_i = 0$, then there were no classification mistakes.

Classification



(a) (b) (c)

[Chernova Fig 4]
Confidence threshold computation:
(a) overlapping region; (b) misclassified points; (c) learnt thresholds

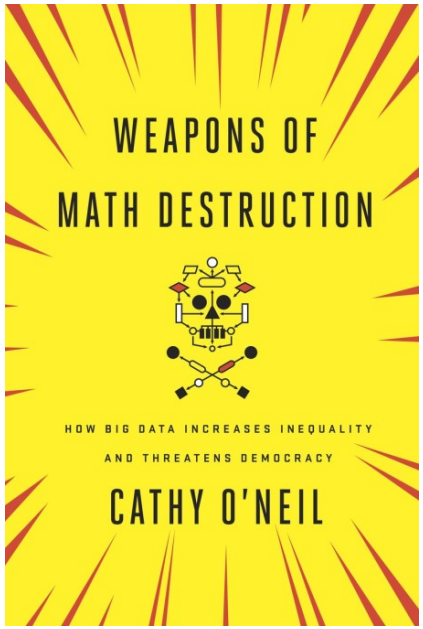
Ethics of Machine Learning

- Need to think about more than just the algorithm.
- ML algorithms are used to make decisions.
- First, do no harm.

Example tweet of chatbot

Weapons of Math Destruction

- Term coined by Cathy O'Neil.
- WMD.
- Concept that ML models have the capacity to create great damage to society.



Weapons of Math Destruction

- Opacity
  Back box models that cannot easily be interrogated.
  Especially by the subjects.
- Scale
  The effects of the model can extend beyond the decision that it helps to make.
- Damage
  The model causes damage.
  The model causes harm to people, especially members of vulnerable groups.

Weapons of Math Destruction

- Often start out as ideas with the best of intentions.
- Value-added teacher model.
- Medical school admission.
- Employee profiling.
- Predictive policing.
- Recidivism risk model.

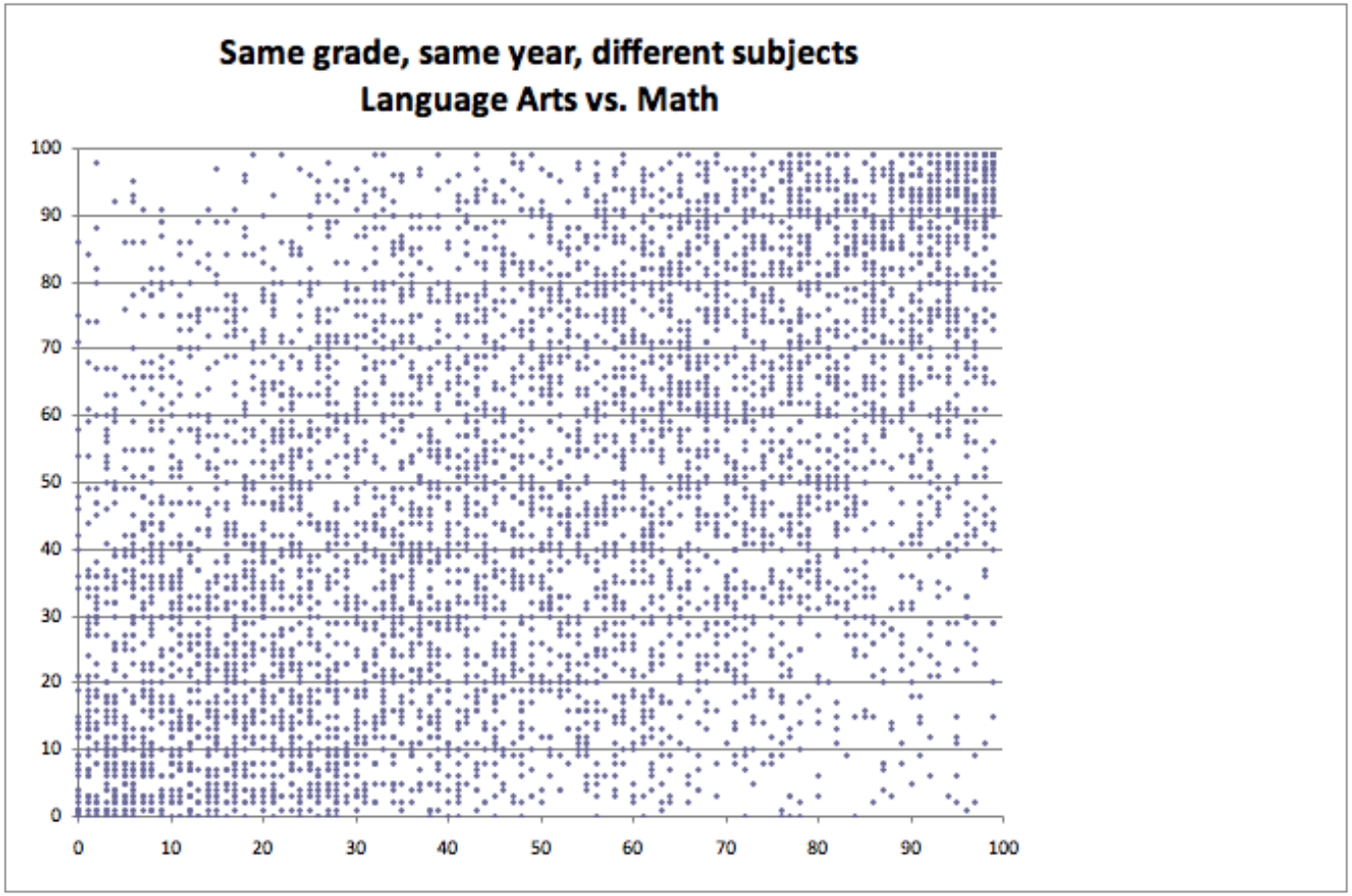
Value-added teacher model

- Problem: some schools do not do a good job of teaching.
- Washington DC, 2007.
- 8% of 8th grade students (13/14) were at grade level in maths.
- Only around 50% of 9th grade students (14/15) continue to graduate.
- Getting rid of poor teachers would improve the system.

Value-added teacher model

- Idea: we could use student results to rate teachers.
- Reality: student results don't tell you how good teachers are.

Value-added teacher model



(Gary Rubinstein)

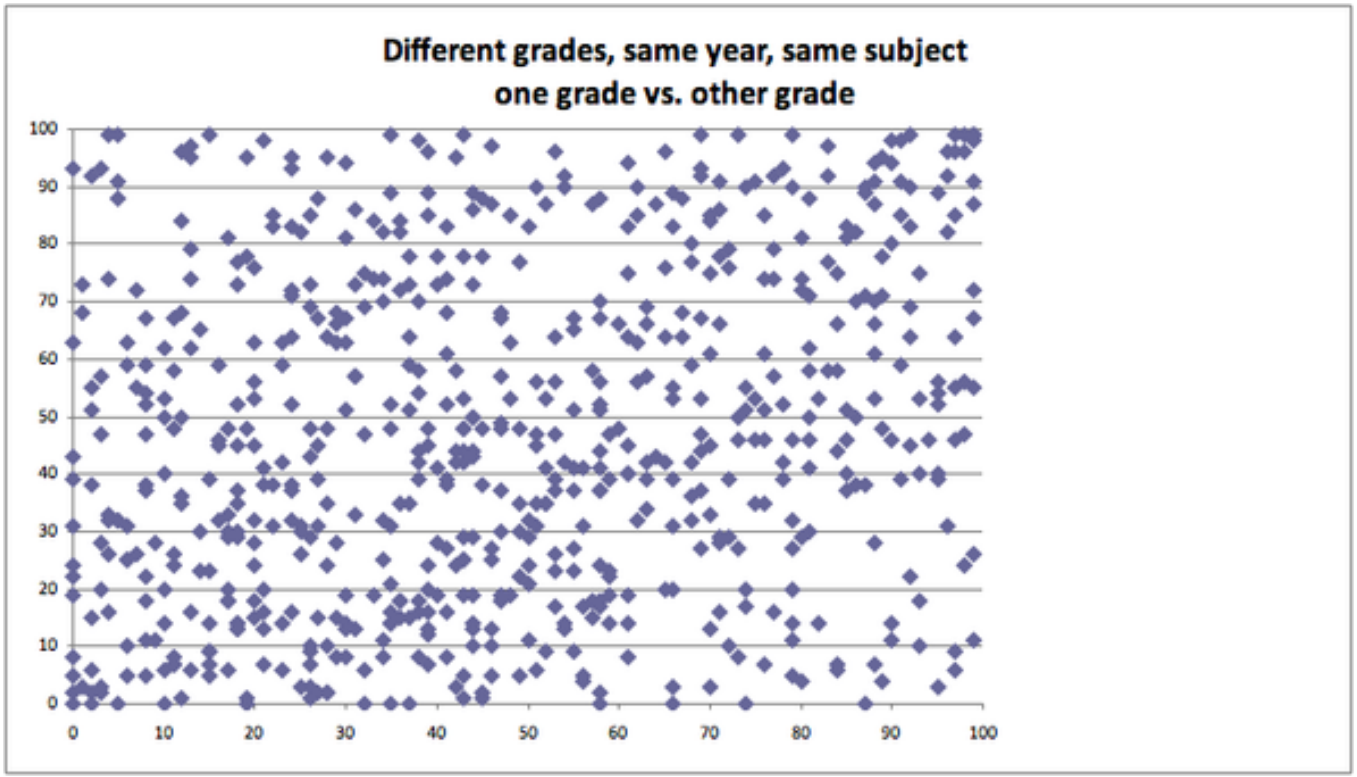
- Grades for maths versus arts.
- Same teacher, same year, same students.

Value-added teacher model

Out of 5,675 elementary school teachers, the average difference between the two scores was a whopping 22 points. One out of six teachers, or approximately 17%, had a difference of 40 or more points. One out of 25 teachers, which was 250 teachers altogether, had a difference of 60 or more points, and, believe it or not, 110 teachers, or about 2% (that's one out of fifty!) had differences of 70 or more points.

(Gary Rubinstein)

Value-added teacher model



(Gary Rubinstein)

- Grades for two different sets of students.
- Same teacher, same subject, same year.

Value-added teacher model

- So the results are essentially random.
- Reasonable as a basis for deciding someone's career?

Medical school admissions

- Problem: St George's Hospital Medical School.
- Many applicants (12 for each place).
- 25% of applicants are interviewed.
- Concerns about consistency in screening candidates for interview, plus a desire to make the process less time consuming.
- Late 1970s.

Medical school admissions

- Idea: Computerised screening of applications.
- Reality: Just implements the biases already in the selection process.

Medical school admissions

The admissions data that was used to define the model's outputs showed bias against females and people with non-European-looking names.

[The Guardian]

Medical school admissions

- Problem was discovered internally.
- “At the time, St George’s actually admitted a higher proportion of ethnic minority students than the average across London” [The Guardian]

Weapons of Math Destruction

- My interpretation.
- Need a good *system*.
  System = algorithm + data
- So need to think about what the data brings.
- Need to audit against the WMD characteristics.

Summary

- Learning from Demonstration.
- Confidence-based Autonomy as a form of LfD.
- Ethical considerations around ML.
- Weapons of Math Destruction.
- Examples and cautionary tales.