COSC343: Artificial Intelligence

Lecture 3: Machine Learning: introduction

Lech Szymanski

Dept. of Computer Science, University of Otago

In today's lecture

- Mathematical framework for machine learning
- Types of machine learning
- Consistency and generalisation
- Training and testing
- Noise

Hand-built vs. learning agents

Two different approaches to building an AI agent.

- Designing the whole agent by hand
- Designing an AI agent which can learn from the sensory data it receives

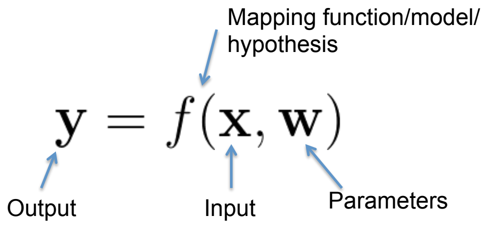
What are the benefits of building a learning agent?

-
-
-

Machine Learning (ML)

- Machine Learning = Pattern Recognition
- Inference – a computational model (a hypothesis) created based on observation of sample data
- Probability/Optimisation

ML: Model



$$\mathbf{y} = f(\mathbf{x}, \mathbf{w})$$

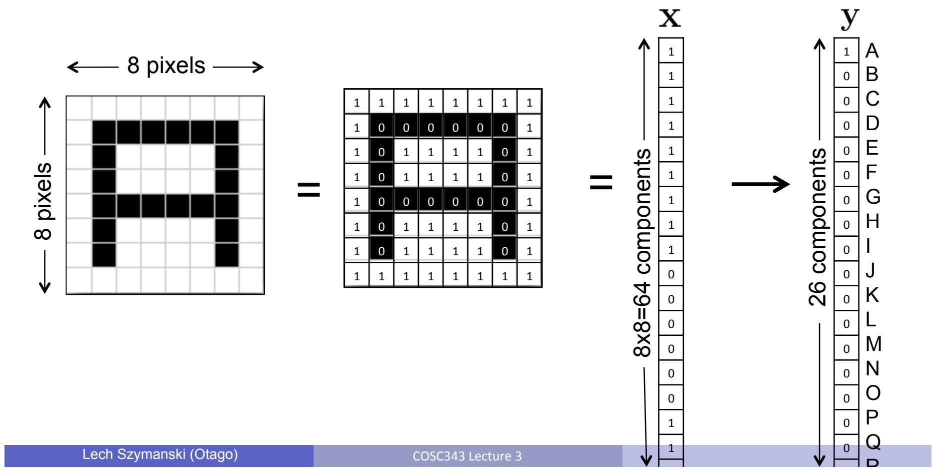
ML: Supervised learning

In **supervised learning**, the agent learns a function from inputs to outputs:

- During the training period, the learning algorithm receives a set of *sample* inputs, with their associated outputs.
- It uses these to work out what the function is.

ML: Supervised learning

$$\mathbf{y} = f(\mathbf{x}, \mathbf{w})$$



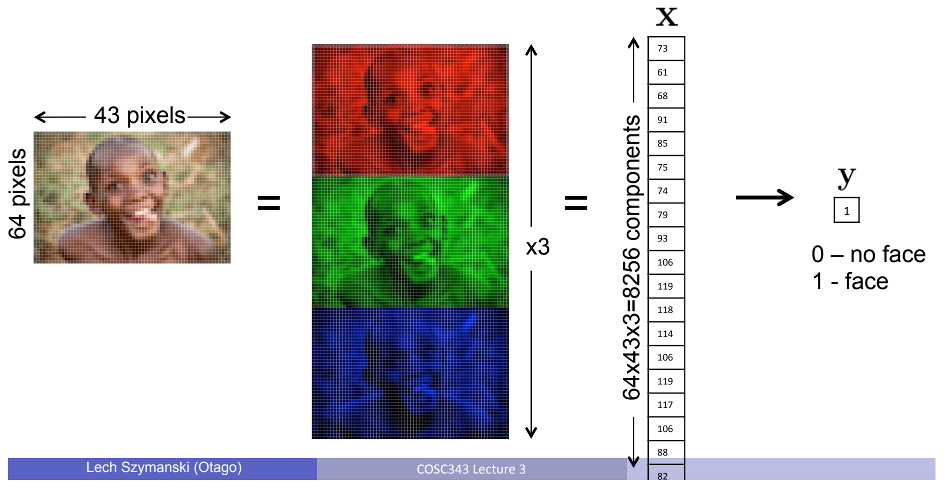
1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1
1	0	1	1	1	1	0	1
1	0	1	1	1	1	0	1
1	0	0	0	0	0	0	1
1	0	1	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	1	1	1	1	1	1

$\mathbf{x}$
1
1
1
1
1
1
1
1
1
0
0
0
0
0
0
1
1

$\mathbf{y}$	
1	A
0	B
0	C
0	D
0	E
0	F
0	G
0	H
0	I
0	J
0	K
0	L
0	M
0	N
0	O
0	P
0	Q

ML: Supervised learning

$$\mathbf{y} = f(\mathbf{x}, \mathbf{w})$$



$\mathbf{x}$
73
61
68
91
85
75
74
79
93
106
119
118
114
106
119
117
106
88
82

$\mathbf{y}$
1

0 – no face
1 - face

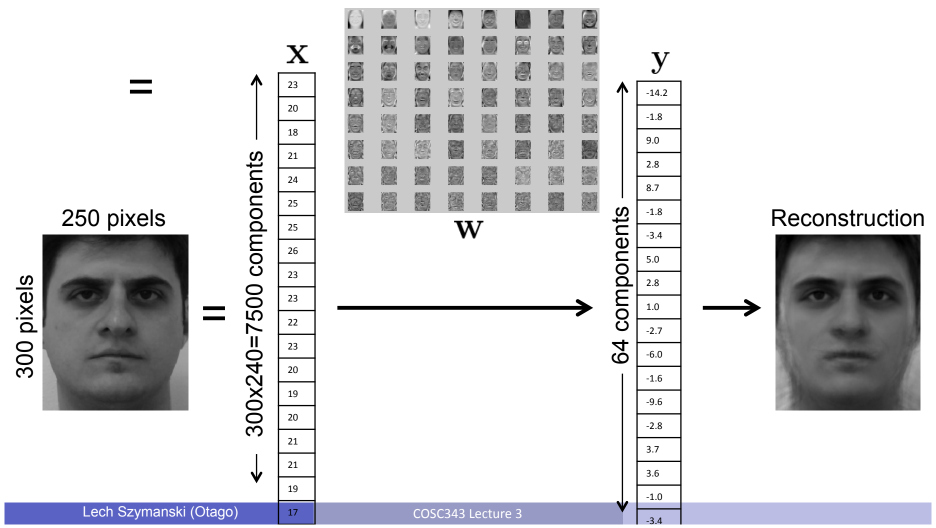
ML: Unsupervised learning

In **unsupervised learning**, the learning algorithm receives a set of training data, and has to work out what regularities it contains:

- E.g. say we have a set of data, where there is a lot of redundant information; unsupervised algorithm can spot the most relevant and common patterns to retain as the compressed representation.

ML: Unsupervised learning

$$\mathbf{y} = f(\mathbf{x}, \mathbf{w})$$



ML: Reinforcement learning

In **reinforcement learning**, the agent receives data and generates actions in response:

- The agent also receives a reinforcement signal ('good'/'bad') which depends on the effects of its actions

- The learning algorithms adapts the agent function to maximise 'good' states

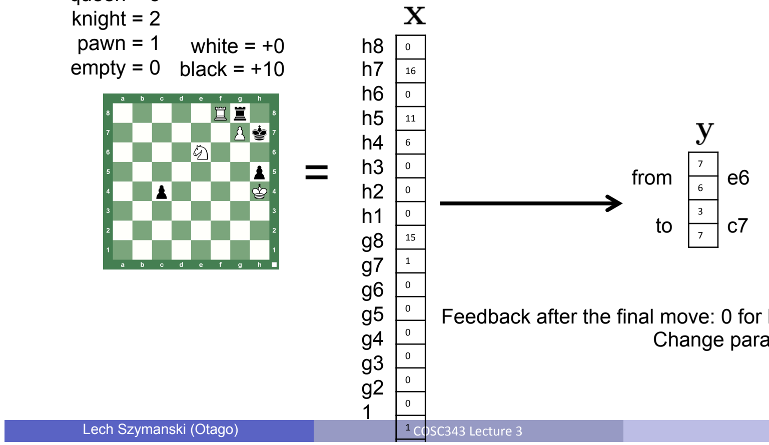
ML: Reinforcement learning

$$\mathbf{y} = f(\mathbf{x}, \mathbf{w})$$

king = 6
rook = 5
bishop = 4
queen = 3
knight = 2
pawn = 1
empty = 0

white = +0
black = +10

a = 1
b = 2
c = 3
d = 4
e = 5
f = 6
g = 7
h = 8



Feedback after the final move: 0 for loss, 1 for win
Change parameters to win

ML: Induction

The basic principle behind supervised learning is **induction**.

We can define an inductive learning procedure as follows:

- Assume there is some "true", unknown function $\tilde{f}$, which takes input $\mathbf{x}$ and returns "true" output $\tilde{\mathbf{y}} = \tilde{f}(\mathbf{x})$
- An indirect evidence of the "true" function is a pair $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$
- Make a hypothesis that function f approximates the "true" function and returns output $\mathbf{y} = f(\mathbf{x}, \mathbf{w})$
- An inductive learning procedure takes a set of examples and modifies $\mathbf{w}$ so that $\mathbf{y}_i \approx \tilde{\mathbf{y}}_i, \ \forall i$ $\forall$ - means "for all"
- Hopefully $f(\mathbf{x}, \mathbf{w}) \approx \tilde{f}(\mathbf{x})$

We want a hypothesis function which **generalises** well to unseen examples of $\tilde{f}$.

ML: Generalisation

"Essentially, all models are wrong, but some are useful"
George Box

- Useful models are the ones that perform well on unseen data, i.e. **they generalise**

ML: Consistency and simplicity

A **consistent** hypothesis is one which agrees with all the training examples.

There are typically many consistent hypotheses for a given training set.

How to choose between them?

- **Occam's razor** tells us to prefer the *simplest* hypothesis. Simpler solutions tend to generalise better to new examples.

There is typically a trade off between consistency and simplicity:

- We can often get a much simpler hypothesis if we're allowed to ignore a few training examples

ML: Consistency and overtraining

A **consistent** hypothesis is one which agrees with all the training examples.

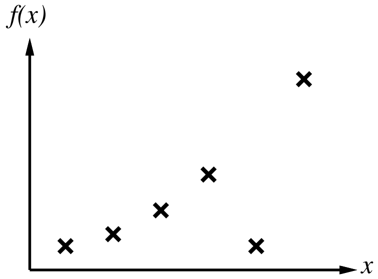
There are typically many consistent hypotheses for a given training set.

It's possible for a consistent model to not perform well on test data – such model is said to have been **overtrained**, and it **overfits** the data.

An example: curve-fitting

Here are some examples

Question: what’s the function?

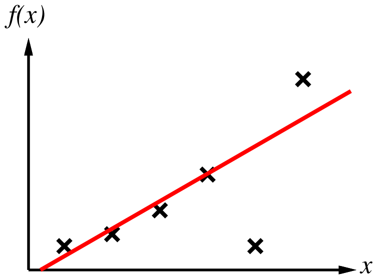


An example: curve-fitting

Here are some examples

Question: what’s the function?

Here’s a simple (but inconsistent) hypothesis:

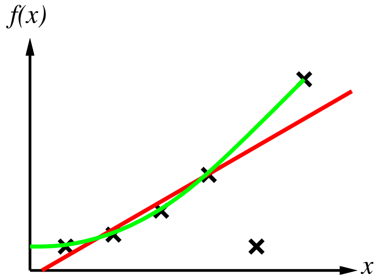


An example: curve-fitting

Here are some examples

Question: what’s the function?

Here’s a hypothesis which is *more* consistent (but also more complex):

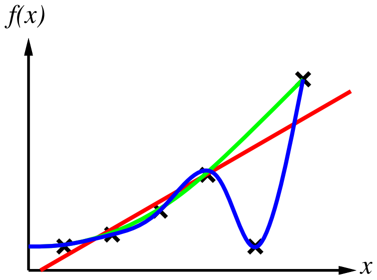


An example: curve-fitting

Here are some examples

Question: what’s the function?

Here’s a *fully* consistent hypothesis:



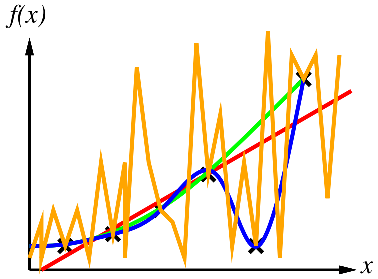
An example: curve-fitting

Here are some examples

Question: what's the function?

Here's a another *fully* consistent hypothesis:

This hypothesis **overfits** the data.



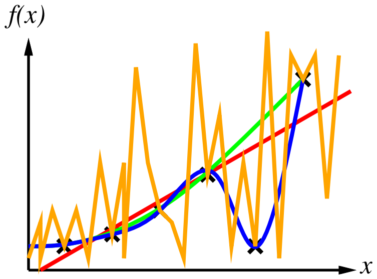
An example: curve-fitting

Here are some examples

Question: what's the function?

Here's a another *fully* consistent hypothesis:

The simplest consistent hypothesis works great, **if there is no noise in the training data**!



Noise

It would be nice if the $\tilde{\mathbf{y}}$ from the set of samples $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$ available for training would correspond to relation:

$$\tilde{\mathbf{y}} = \tilde{f}(\mathbf{x}) \longleftarrow \text{Ideal}$$

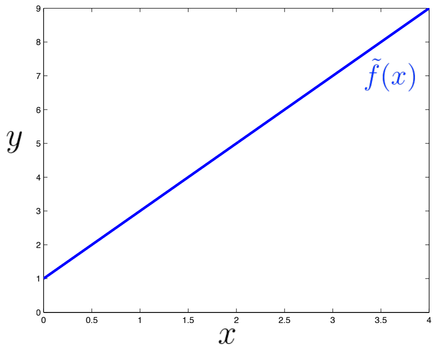
But, in the *real* world nothing is perfect and observations are noisy (and the nature of noise is often unknown):

$$\tilde{\mathbf{y}} = \tilde{f}(\mathbf{x}) + \underset{\nwarrow ?}{\epsilon} \longleftarrow \text{Realistic}$$

Machine learning algorithms must make useful inferences in the *imperfect* world.

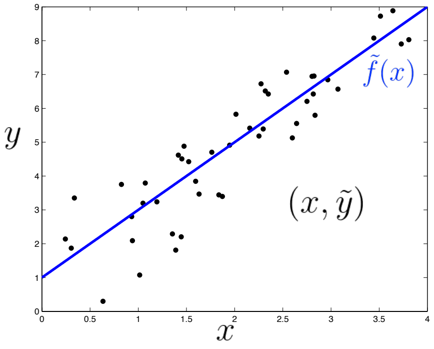
An example: noisy training data

Here's a function: $\tilde{f}(x) = 1 + 2x$



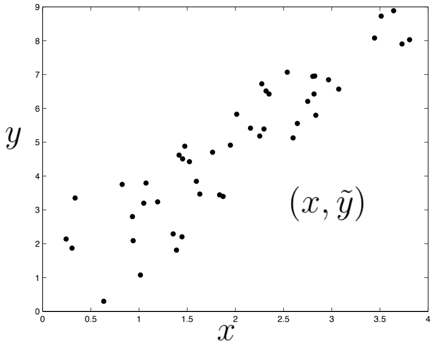
An example: noisy training data

Here are samples with Gaussian noise with variance 1:



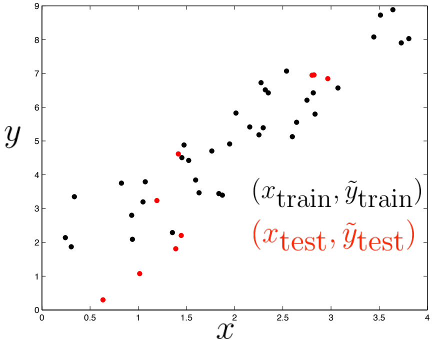
An example: noisy training data

A machine learning algorithm has to use these samples to guess what the function is.



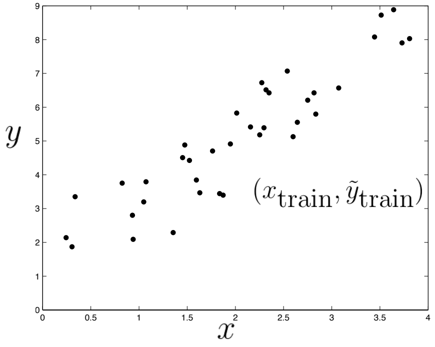
An example: noisy training data

To test for generalisation, available data is typically divided into a train and test set.



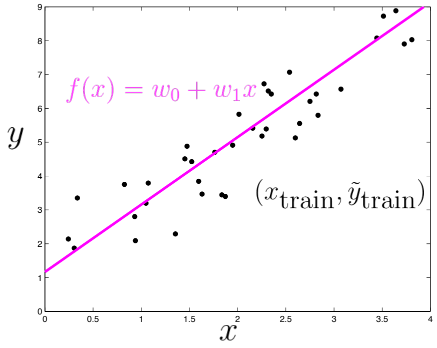
An example: noisy training data

Test set is not used during training.



An example: noisy training data

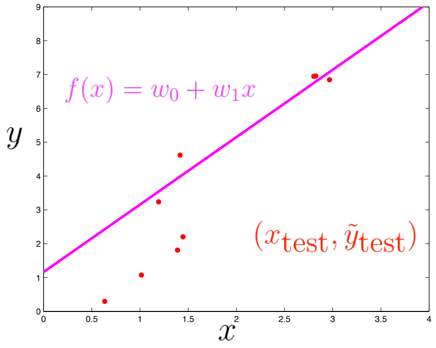
A machine learning algorithm has to make a hypothesis and fit it to training data.



$w_0 = 1.1651, w_1 = 1.9923$

An example: noisy training data

The fit of the resulting function is checked against test data.

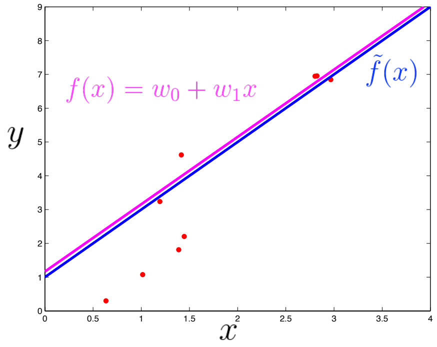


$w_0 = 1.1651, w_1 = 1.9923$

An example: noisy training data

True function: $\tilde{f}(x) = 1 + 2x$

Model: $f(x) = w_0 + w_1 x$, where $w_0 = 1.1651, w_1 = 1.9923$



Summary and reading

- Learning is useful for unknown environment, lazy designers
- Supervised learning: from a finite set of examples of a function, estimate a general hypothesis
- Training data tends to be noisy
- Evaluating learned functions on a withheld test set

Reading for the lecture: AIMA Chapter 18 Sections 1-2

Reading for next lecture: AIMA Chapter 13 Section 1-4