Foundational Machine Learning 1.04: Loss Functions and Performance Measures : Is the ML System Working?

Rohit Babbar
rb2608@bath.ac.uk

Terminology

- For binary classification, the accuracy of a classifier f on an input-output pair x, y is measured by 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

Terminology

- For binary classification, the accuracy of a classifier f on an input-output pair x, y is measured by 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

- On a sample of size n, training/test error of f is given by $R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$

UNIVERSITY OF BATH

Terminology

- For binary classification, the accuracy of a classifier f on an input-output pair x, y is measured by 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

- On a sample of size n, training/test error of f is given by $R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$
- However, the Expected (true) loss of f

$$R(f) \triangleq \mathbb{E}_{(x,y)\sim P}(\ell(y, f(x)))$$

The above expectation $\mathbb{E}_{(x,y)\sim P}$ means that it is an expectation (average) that is computed over samples (x, y) drawn from the data distribution P

Terminology

- For binary classification, the accuracy of a classifier f on an input-output pair x, y is measured by 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

- On a sample of size n, training/test error of f is given by $R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$
- However, the Expected (true) loss of f

$$R(f) \triangleq \mathbb{E}_{(x,y)\sim P}(\ell(y, f(x)))$$

The above expectation $\mathbb{E}_{(x,y)\sim P}$ means that it is an expectation (average) that is computed over samples (x, y) drawn from the data distribution P

- Bayes classifier f_{Bayes}, is defined to be the one which has the least classification error, i.e., $f_{Bayes} = \arg\min_f R(f) := \mathbb{E}_P(\ell(y, f(x)))$

UNIVERSITY OF BATH

Loss functions in Machine Learning

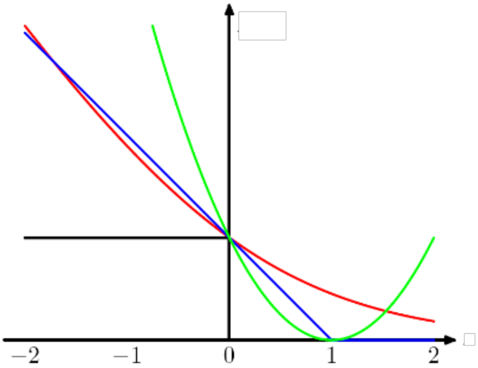


Figure: Horizontal axis is $yf(x)$, vertical is the loss i.e. $\ell(yf(x))$

Convex Upper Bounds on 0-1 loss

- Hinge Loss (in blue) is given by $\max(1 - yf(x), 0)$
- Logistic Loss (in red) is given by $\frac{1}{\log 2} \log(1 + \exp(-yf(x)))$

Bayes Classifier - (1)

Let's say $C_1 = +1$ (Positive class), and $C_2 = -1$ (Negative class) in the figure below. Also $P(.)$ in the text (below) refers to the probablity and $p(.)$ in the picture refers to its density.

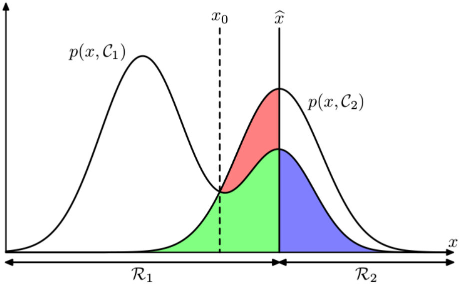
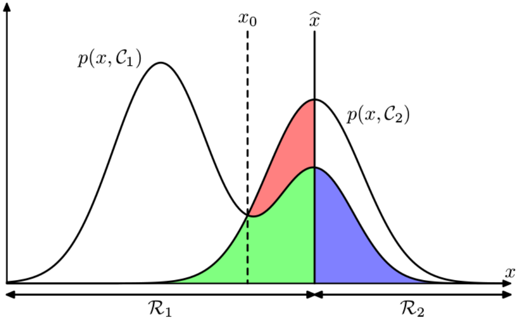


Figure: Depiction of noisy labels (picture from Chris Bishop's book)

- The prediction function of f_{Bayes} is given by

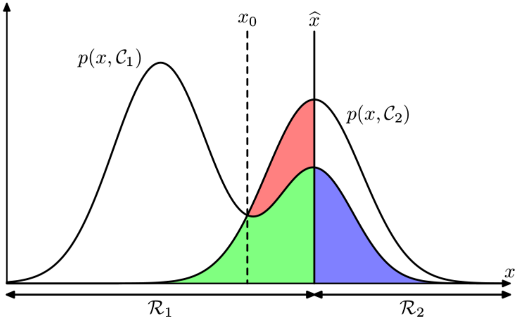
$$f_{Bayes}(x) = \begin{cases} C_1 & \text{if } P(y = C_1 | X = x) \geq 0.5 \\ C_2 & \text{otherwise} \end{cases}$$

Example



- Given a point, say $x = \hat{x}$, how do we compute $P(y = C_1|X = \hat{x})$ or $P(y = C_2|X = \hat{x})$?

Example

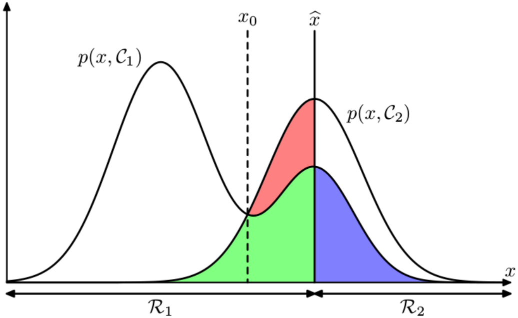


- Given a point, say $x = \hat{x}$, how do we compute $P(y = C_1|X = \hat{x})$ or $P(y = C_2|X = \hat{x})$?

$$P(y = C_1|X = \hat{x}) \stackrel{(1)}{=} \frac{p(X = \hat{x}, Y = C_1)}{p_X(X = \hat{x})} \qquad \text{and } P(y = C_2|X = \hat{x}) \stackrel{(1)}{=} \frac{p(X = \hat{x}, Y = C_2)}{p_X(X = \hat{x})}$$

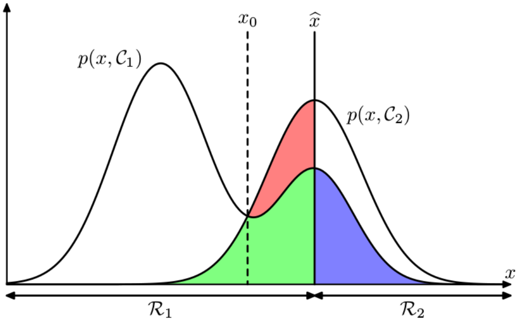
Since the denominator is the same for both, this implies that both probabilities i.e. $P(y = C_i|X = \hat{x})$ for $i = 1, 2$ are proportional to their respective densities at $X = \hat{x}$. Therefore, $P(y = C_1|X = \hat{x}) = \frac{p(X=\hat{x}, Y=C_1)}{p(X=\hat{x}, Y=C_1)+p(X=\hat{x}, Y=C_2)}$

Example



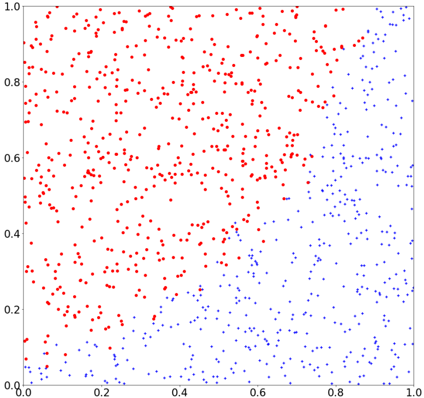
- When the decision boundary/threshold is at $x = \hat{x}$, what kind of errors are signified by the red, green and blue regions?

Example



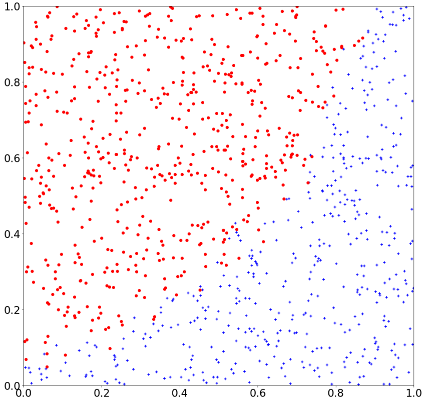
- At what point in the graph $P(y = C_1 | X = x) = 0.5$?

Underfitting & overfitting

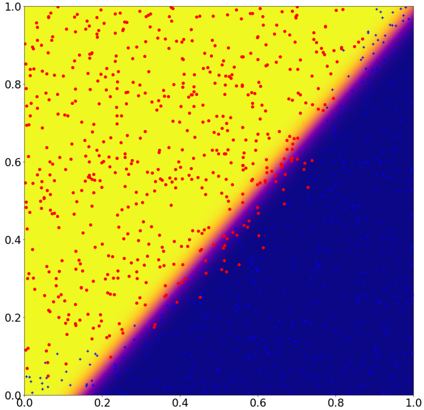


- Curved
- Classes overlap
- How would you divide them?

Underfitting & overfitting

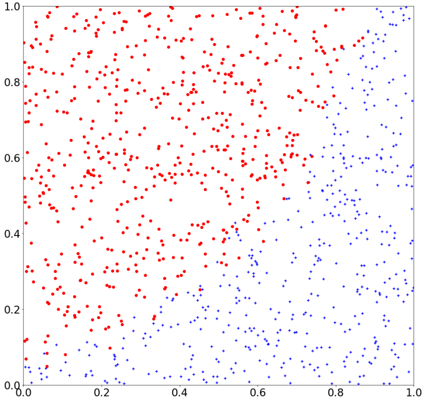


- Curved
- Classes overlap
- How would you divide them?

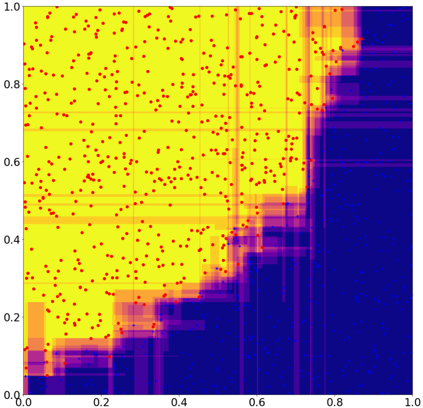


- Underfitting

Underfitting & overfitting

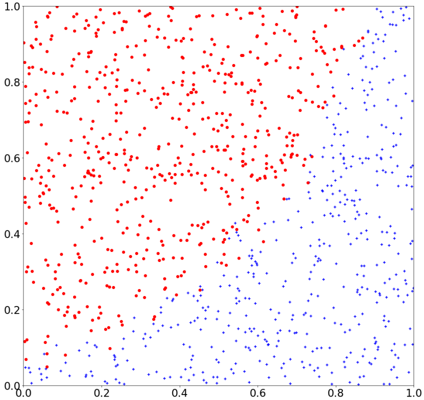


- Curved
- Classes overlap
- How would you divide them?

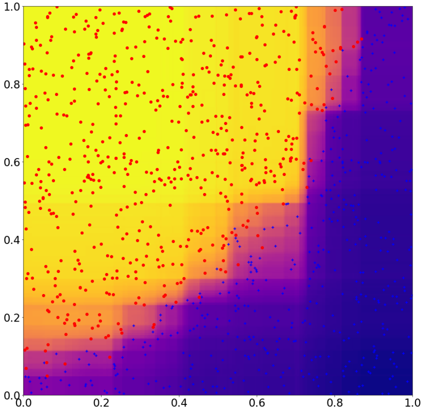


- Overfitting

Underfitting & overfitting

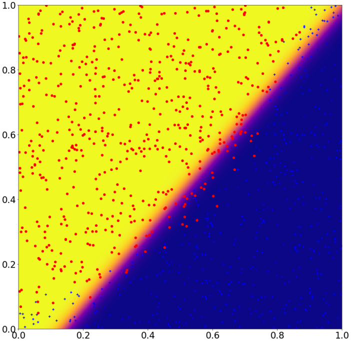


- Curved
- Classes overlap
- How would you divide them?

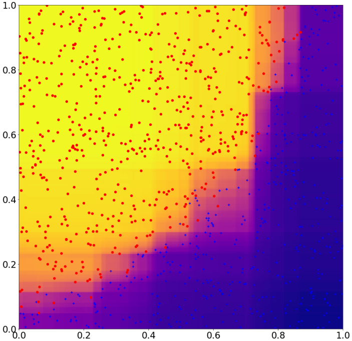


- Balanced

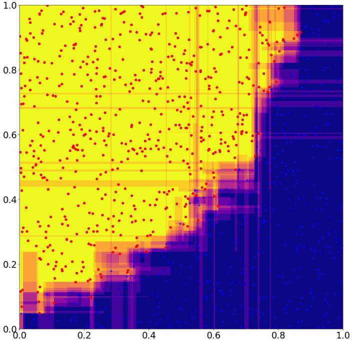
Underfitting & overfitting



- Underfitting
- Logistic regression



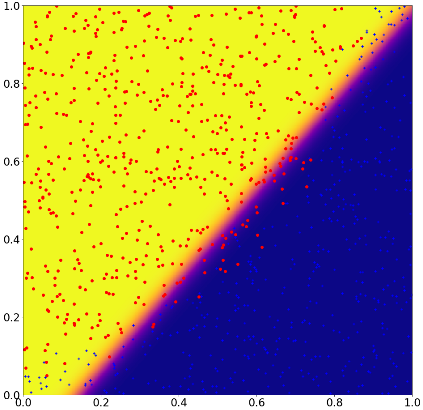
- Balanced
- Tuned random forest
- (scikit learn, `min_impurity_decrease=0.008`, `n_estimators=512`)



- Overfitting
- Badly tuned random forest
- (scikit learn, default parameters)

Underfitting causes

- Weak model
- Insufficient data



UNIVERSITY OF BATH

Overfitting causes

- Powerful model +
- Insufficient **regularisation**
  Regularisation $\sim$ smoothing out noise (so model doesn't learn it)

- How to detect?

Train & test set

- Model can't overfit on data it doesn't have!

  $\therefore$

- Split the data:
  - A **train** set, to fit the model
  - A **test** set, to verify performance

Train & test set

- Model can't overfit on data it doesn't have!
  $\therefore$
- Split the data:
  - A **train** set, to fit the model
  - A **test** set, to verify performance
- Large gap between train/test accuracy indicates overfitting (usually)

	Accuracy	
Random Forest	Train	Test
Underfitting	79.2%	79.2%
Balanced	97.6%	95.0%
Overfitting	99.6%	94.7%

Hyperparameters I

- Parameters $\rightarrow$ fit to training data
- Hyperparameters $\rightarrow$ parameter that cannot be fit to training data

Hyperparameters I

- Parameters $\rightarrow$ fit to training data
- Hyperparameters $\rightarrow$ parameter that cannot be fit to training data

- Reasons to be a hyperparameter:
  - Avoiding overfitting, e.g. decision tree depth
  - Heavy computation, e.g. ensemble size
  - Hard to optimise

Hyperparameters II

- Can still fit hyperparameters . . .
  (manually or by algorithm)

- . . . **but not to the test set!**
  (this mistake can be found in countless research papers)

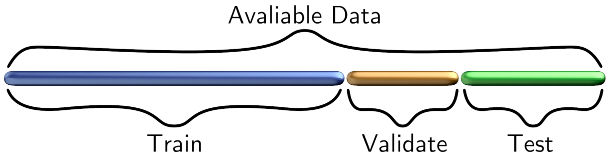
Hyperparameters II

- Can still fit hyperparameters . . .
  (manually or by algorithm)

- . . . **but not to the test set!**
  (this mistake can be found in countless research papers)

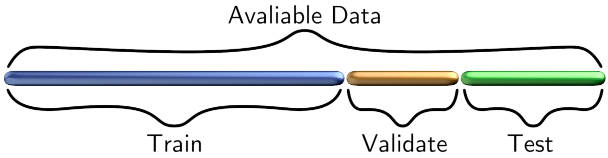
- Introduce a third set: **validation** set
  - **train** – Give to algorithm
  - **validation** – Objective of hyperparameter optimisation
  - **test** – To report final performance

Measuring performance



- How do we decide on split percentages?
  - Train large → Algorithm performs well
  - Validation large → Hyperparameter optimisation performs well
  - Test large → Accurate performance estimate

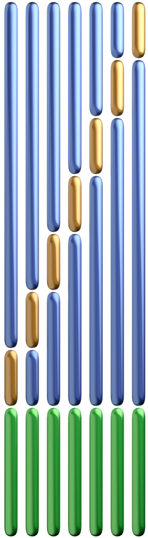
Measuring performance



- How do we decide on split percentages?
  - Train large $\rightarrow$ Algorithm performs well
  - Validation large $\rightarrow$ Hyperparameter optimisation performs well
  - Test large $\rightarrow$ Accurate performance estimate

- Good default: Validation and test small as possible to get reliable estimate, rest on train
- . . . but might shrink train due to computational cost
- "small as possible" hard to judge, but 10-15% of data for test and validation each is a good fraction

n-fold



- Validation and test used to make **measurements**
- Can average measurements! (as long as they are independent)

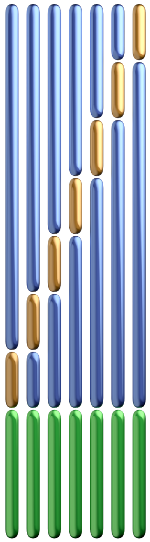
UNIVERSITY OF BATH

n-fold

- Validation and test used to make **measurements**
- Can average measurements! (as long as they are independent)

- e.g. divide train/validation into 7-fold
  - train: six parts
  - validation: one part

  Train for all seven combinations and report average performance on test



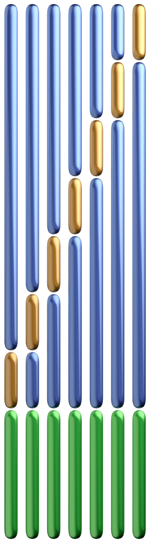
n-fold

- Validation and test used to make **measurements**
- Can average measurements! (as long as they are independent)

- e.g. divide train/validation into 7-fold
  - train: six parts
  - validation: one part

  Train for all seven combinations and report average performance on test

- n-fold = n $\times$ slower! Typically $4 \leq n \leq 20$
- General case: All combinations of train/validation/test



Final model

- May train algorithm thousands of times!

  (hyperparameter tuning and n-fold)
- Choice of n is a trade-off between accuracy / time
- Fast computer/cluster/distributed computation really help!

- Final model: Train on entire data set

  (still wise to keep a test set back to sanity check)

UNIVERSITY OF BATH

Performance?

- What do we actually measure?

  (and hence optimise)

Confusion matrices

- Classification only
- Random forest on breast cancer:

		Actual	
		False	True
Predicted	False	49	6
	True	14	159

Confusion matrices

- Classification only
- Random forest on breast cancer:

		Actual	
		False	True
Predicted	False	49	6
	True	14	159

- On diagonal means correct, off means wrong
- Can see which classes are confused
- An empty row is a problem

Naming the numbers

		Actual	
		False	True
Predicted	False	True Negative (TN)	False Negative (FN)
	True	False Positive (FP)	True Positive (TP)

Naming more numbers

Loads of terms are used :

$\frac{\texttt{TP}}{\texttt{TP+FN}}$	sensitivity, **recall**, hit rate, **true positive rate**
$\frac{\texttt{TN}}{\texttt{TN+FP}}$	specificity, **true negative rate**
$\frac{\texttt{TP}}{\texttt{TP+FP}}$	**precision**, positive predictive value
$\frac{\texttt{TP+TN}}{\texttt{TP+TN+FP+FN}}$	**accuracy**
$\frac{2\times\texttt{TP}}{2\times\texttt{TP+FP+FN}}$	**F1 score**

- Ideally one would like to have high values for both Precision and Recall
- However, between the two, it is typically ease to optimize one metric at the cost of the other
- Optimising F-measure, which is the harmonic mean (i.e. arithmitic mean of the inverses), of Precision and Reacll, forces to optimize the smaller one, and hence a better measure

Imbalanced data

- Imbalanced training set $\rightarrow$ Makes training difficult
- e.g. credit card fraud $\approx 0.1\%$ of transactions
- 99.9% accuracy by assuming no fraud – meaningless!

- Training : often need to adjust training process (e.g. oversampling)
- Evaluation : F1 score is a better measure

- Balanced accuracy:

$$= \frac{1}{|C|} \sum_{c \in C} \frac{|\{y_i = c \wedge f_\theta(x_i) = c\}|}{|\{y_i = c\}|}$$

  - C = set of classes, of size $|C|$
  - (y_i, x_i) = data points
  - $f_\theta(\cdot)$ = machine learning model

Performance

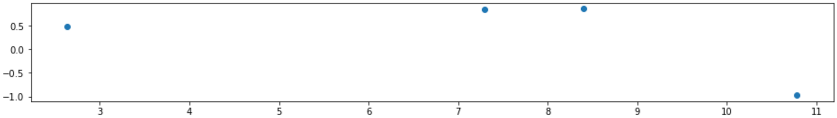
- Previous are intermediates
- Need a problem specific function of the confusion matrix (for classification)

- Depending on problem might be better to think in terms of:
    - Cost / Loss
    - Gain
    - Error
    - Risk
    - Ranking

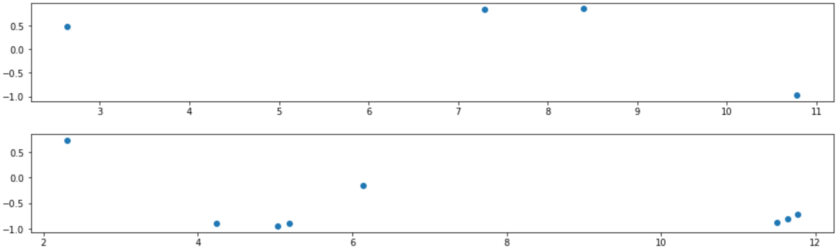
- Test complete system!

    intermediate proxies can sometimes be misleading

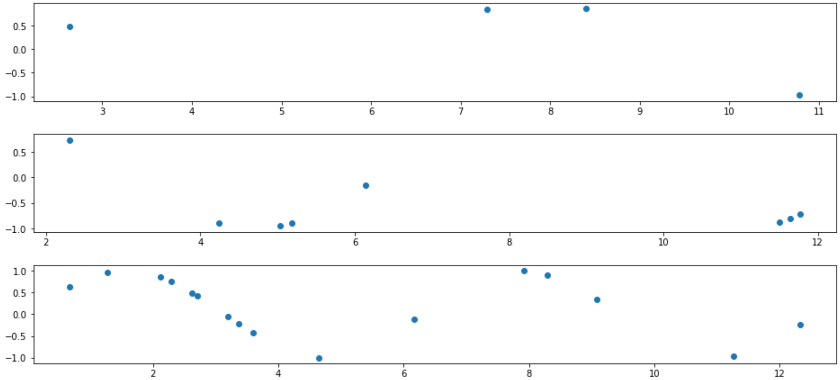
Bad data: Insufficient



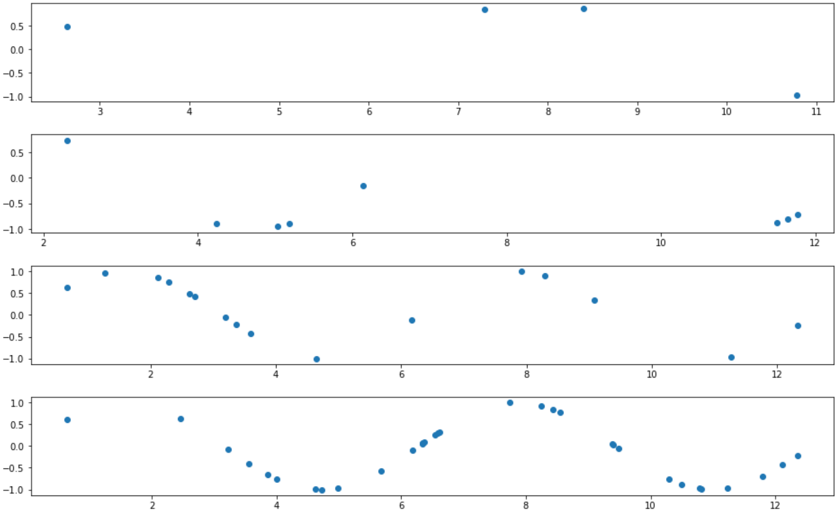
Bad data: Insufficient



Bad data: Insufficient



Bad data: Insufficient



Bad data: Spurious correlation

- 1964: Researcher was spotting M-48 tanks in images
- Got a near perfect score

- Problem:
  - Tank photos taken on a cloudy day
  - Not-tank photos taken on a sunny day
  - . . . so it was checking the brightness (b&w so no colour)

- Original paper (probably!): `https://dl.acm.org/citation.cfm?doid=800257.808903`

Bad data: No correlation

- Problem: Estimate when next bus will arrive
- Input:
  - Current height of the fountain
  - Number of purple cars on campus
  - How many bats are in the bat cave
- What's the problem?
- There is nothing to learn – no correlation – it's impossible!

UNIVERSITY OF BATH

Bad data: Unbalanced

- When you train with 1000 examples of one class and 10 of another
- Classifier can get 99% by always predicting the larger class...

  ...and often does

- Good example: `https://arxiv.org/pdf/1606.08390.pdf`
  - Visual question answering:

    $$f(\text{image}, \text{question about image}) \rightarrow \text{answer}$$
  - Always giving most common answer ("*yes*") beat sophisticated approaches!

UNIVERSITY OF BATH

Bad data: Runtime mismatch

Volvo's driverless cars 'confused' by kangaroos

27 June 2017

There are more than 20,000 kangaroo strikes each year in Australia

Bad data: Biased data

- Amazon has too many job applications
- ML system sorts best to worst to save time

- Misogynistic
- Trained on past applicants... mostly male

- Spurious correlation as well as bad data

- Article: `https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G`

Summary

- Overfitting/underfitting
- Train/test/verification
- Measuring success
- Misinterpretation
- (many kinds of) bad data

- Spotting issues is a skill... takes practise

UNIVERSITY OF BATH

Further reading

- Blog breaking down why a medical data set is useless: https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/
- "Concrete Problems in AI Safety"
  by **Amodei, Olah, Steinhardt, Christiano, Schulman and Mane**
  https://arxiv.org/pdf/1606.06565.pdf
- "How to recognize AI snake oil"
  by **Narayanan** (slides)
  https://www.cs.princeton.edu/~arvindn/talks/MIT-STS-AI-snakeoil.pdf