Lecture 1: Introduction

Helen Yannakoudakis and Oana Cocarascu

Department of Informatics
King's College London

(Version 1.5)

What is Machine Learning?

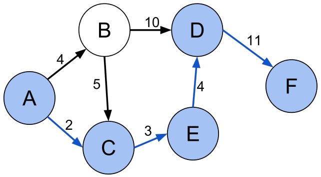
- Google search
- Google translate
- Spam filtering
- Speech recognition (eg, Amazon Alexa)
- Text prediction
- Amazon product recommendations

What is Machine Learning?

- The field of study that gives computers the ability to learn [from data] without being explicitly programmed (Samuel, 1959).

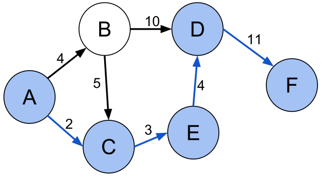
What is Machine Learning?

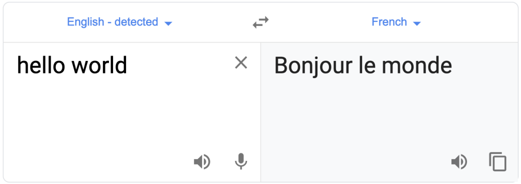
- The field of study that gives computers the ability to learn [from data] without being explicitly programmed (Samuel, 1959).



What is Machine Learning?

- The field of study that gives computers the ability to learn [from data] without being explicitly programmed (Samuel, 1959).



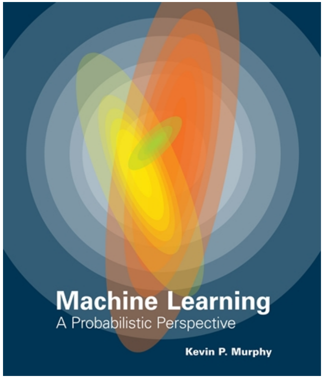


hello world

Bonjour le monde

What is Machine Learning?

- "a set of methods that can automatically detect patterns in data, and then use the . . . patterns to predict future data, or to perform other kinds of decision making. . . " (Murphy, 2012).



Will draw on this book a lot.

Types of ML

- Traditional to consider that there are three kinds of ML.
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- We will cover elements of all three.
  - Today
  - Across the module

Supervised Learning: iris example

- One of the most common types of machine learning.
- Example: given an iris flower, which species does it come from?

(http://www.statlab.uni-heidelberg.de/)

setosa, versicolor or virginica?

Supervised Learning: iris example

Successful machine learning requires 3 components:

Supervised Learning: iris example

Successful machine learning requires 3 components:

1. Representative training data: machine learning model learns from data
    - Data should be annotated (refers to meaningful labels associated with the data) / contains the right answers

Supervised Learning: iris example

- Training data: iris flower images annotated as either setosa, versicolor or virginica

(http://www.statlab.uni-heidelberg.de/)

- Use y_i to represent the type of iris (the class to which it belongs):
  - setosa
  - versicolor
  - virginica

Supervised Learning: iris example

Successful machine learning requires 3 components:

1. Representative training data: machine learning model learns from data
   - Data should be annotated (refers to meaningful labels associated with the data) / contains the right answers

Supervised Learning: iris example

Successful machine learning requires 3 components:

1. Representative training data: machine learning model learns from data
   - Data should be annotated (refers to meaningful labels associated with the data) / contains the right answers
2. Sophisticated methods for extracting features that are used to represent the data
   - Features: easily observable properties of the data based on which the model will try to make predictions

Supervised Learning: representing data as features

- The $\mathbf{x}_i$ are attributes or features of the iris (easily observable properties of the data):
  - Petal area (length x width)
  - Sepal area (length x width)
- The $\mathbf{x}_i$ are typically D dimensional vectors of numbers.
- However, could be complex objects:
  - Sets of coordinates
  - Images
  - Time series

Supervised Learning: iris example

Successful machine learning requires 3 components:

1. Representative training data: machine learning model learns from data
   - Data should be annotated (refers to meaningful labels associated with the data) / contains the right answers
2. Sophisticated methods for extracting features that are used to represent the data
   - Features: easily observable properties of the data based on which the model will try to make predictions

Supervised Learning: iris example

Successful machine learning requires 3 components:

1. Representative training data: machine learning model learns from data
   - Data should be annotated (refers to meaningful labels associated with the data) / contains the right answers
2. Sophisticated methods for extracting features that are used to represent the data
   - Features: easily observable properties of the data based on which the model will try to make predictions
3. Sophisticated choice of machine learning algorithm

Supervised Learning: learning from data

- Start from a training set that contains the right answers y_i:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$

- Training set is a set of input features $\mathbf{x}_i$, and outputs y_i.
- The $\mathbf{x}_i$ can each be a vector of values, the y_i are single elements/classes.

Supervised Learning: learning from data

- Start from a training set that contains the right answers y_i:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$

- Training set is a set of input features $\mathbf{x}_i$, and outputs y_i.
- The $\mathbf{x}_i$ can each be a vector of values, the y_i are single elements/classes.
- A supervised learning algorithm can study the training set and learn to classify iris flowers into the three different classes y_i based on their features $\mathbf{x}_i$ (Goodfellow et al., 2016)

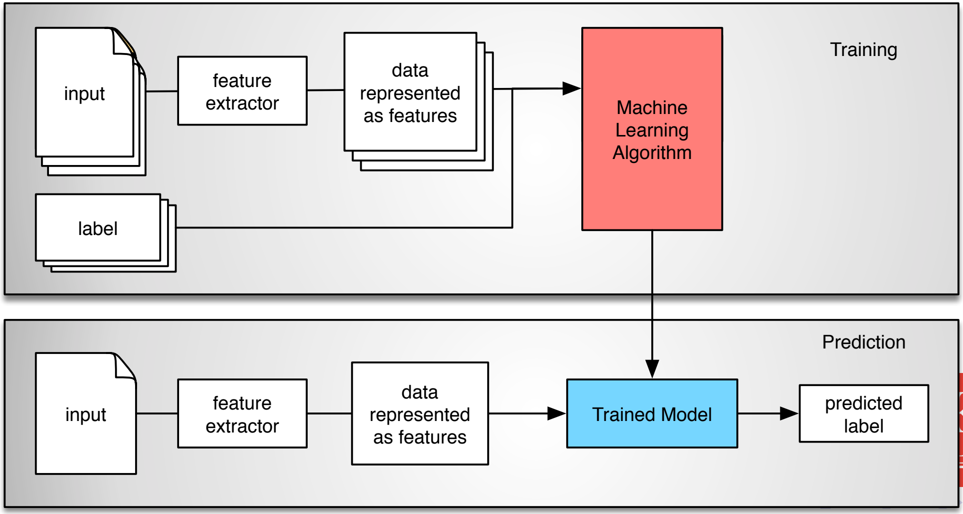
Supervised Learning: learning from data

- Start from a training set that contains the right answers y_i:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$

- Training set is a set of input features $\mathbf{x}_i$, and outputs y_i.
- The $\mathbf{x}_i$ can each be a vector of values, the y_i are single elements/classes.
- A supervised learning algorithm can study the training set and learn to classify iris flowers into the three different classes y_i based on their features $\mathbf{x}_i$ (Goodfellow et al., 2016)
- Learns a function f that maps from features $\mathbf{x}_i$ to a class y_i: $f(\mathbf{x_i}) = y_i$
  - Given the $\mathbf{x}_i$, say what the correct y_i is.
- Can predict a class using a weighted combination of the input features.

Supervised learning: training vs. prediction



Supervised learning: generalisation

- So, again, given:

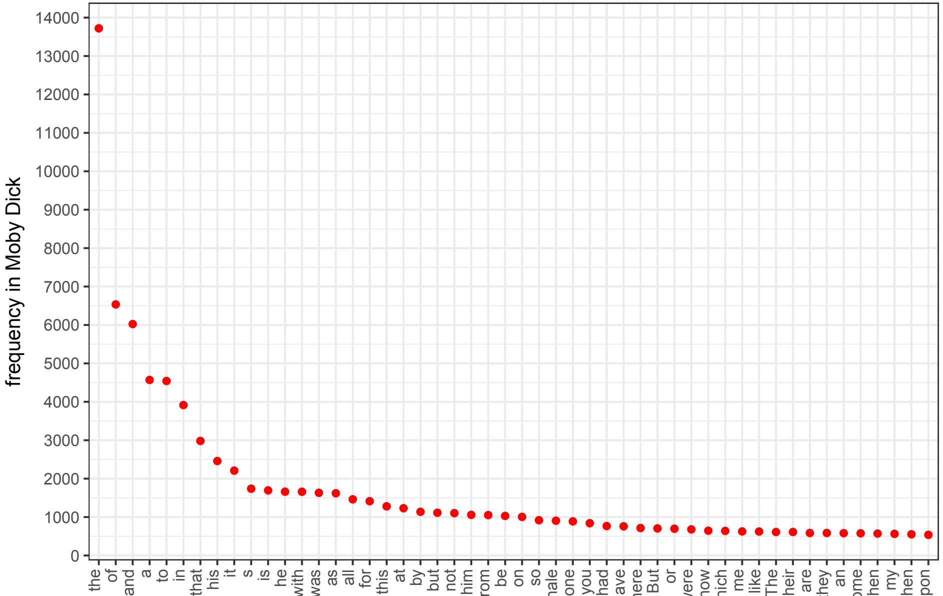
$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$

  be able to predict the y_j of some $\mathbf{x}_j$.
- Particularly interested in $\mathbf{x}_j$ such that $(\mathbf{x}_j, y_j) \notin \mathcal{D}$.

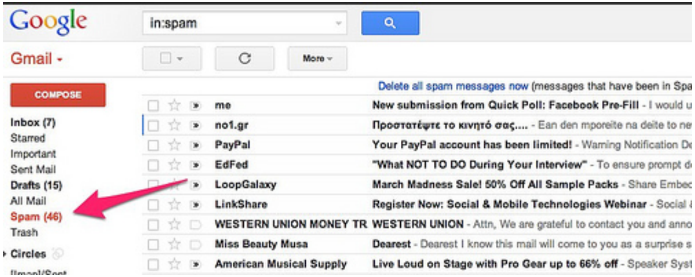
Generalisation

- We want a machine learning model that performs well on new, never-before seen data.
- That is equivalent to saying we want our model to generalise well.
- Important to generalise because of the long tail.
- In many domains, some examples are very common and some are very very rare.
- Word usage.
- So you are very unlikely to have seen all examples even with a big training set.
- For example, 20% of Google searches every day are unique.

There are a small number of high-frequency words...



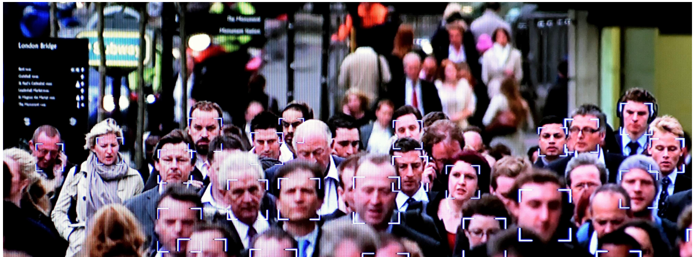
Supervised learning



Delete all spam messages now (messages that have been in Spa

me	New submission from Quick Poll: Facebook Pre-Fill - I would u
no1.gr	Προστατέψτε το κινητό σας.... - Ean den mporeite na deite to ne
PayPal	Your PayPal account has been limited! - Warning Notification De
EdFed	"What NOT TO DO During Your Interview" - To ensure prompt d
LoopGalaxy	March Madness Sale! 50% Off All Sample Packs - Share Embec
LinkShare	Register Now: Social & Mobile Technologies Webinar - Social &
WESTERN UNION MONEY TR	WESTERN UNION - Attn, We are grateful to contact you and anno
Miss Beauty Musa	Dearest - Dearest I know this mail will come to you as a surprise s
American Musical Supply	Live Loud on Stage with Pro Gear up to 66% off - Speaker Syst

(www.whatcounts.com)



Supervised learning: classification vs regression

- The response variable y_i is typically a categorical value from some finite set, like “setosa” or “virginica”.
  A set of labels or classes.
- Formally, $y \in \{1, \ldots, C\}$.
- In this case, the learning problem is classification or pattern recognition.

- y_i can also be a real-valued scalar (eg, income level, marks 1–10).
- In that case the problem is regression.

- In ordinal regression, y_i is a set of labels with some order (eg, grades A–F).

Classification

- Form of supervised learning.
- Here the y are class labels:

$$y \in \{1, \ldots, C\}$$

  where C is the number of classes.

- Common version of the problem is binary classification in which $C = 2$.
- In this case we often write the labels as $\{0, 1\}$ (eg, spam or no spam).
- In binary classification we are often thinking “is the example in the class or not”.
- If $C > 2$ we have multiclass classification (eg, iris example or sentiment classification).

Classification

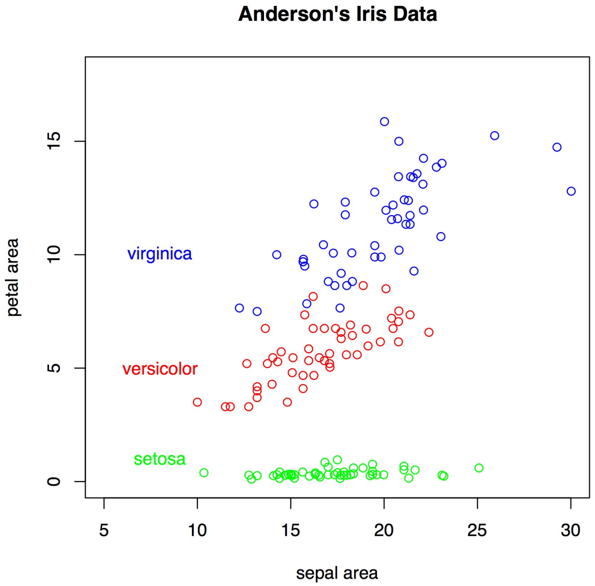
- One way to think of classification is as function approximation.
- That is, we assume that:

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

- Learning is then trying to estimate f, creating an estimate $\hat{f}$.
- After learning we are predicting an estimate of y using:

$$\hat{y} = \hat{f}(\mathbf{x})$$

Classification – the Iris example as multiclass classification



Probabilistic prediction

- Much classification is not very clear at the boundaries.
- So we cast the problem such that it returns a probability:

  $$p(y_i|\mathbf{x}_i, \mathcal{D})$$

  the probability of being in each class, given the input and the set of training data.
- Probabilistic classifiers provide a distribution over classes.
- If we have C classes, we get a probability for each class.
- Sometimes we will explicitly consider the model that we are using to make the prediction:

  $$p(y_i|\mathbf{x}_i, \mathcal{D}, M)$$

Probabilistic prediction

- We can decide on a single class by choosing the one with the highest probability given the input:

$$\hat{y} = \arg\max_{c=1}^{C} p(y = c|\mathbf{x}_i, \mathcal{D})$$

- This is the most likely class label, the mode of the distribution.
- This is called the MAP estimate, the "maximum a posteriori" estimate.
- Other times we want the probability of each y_i, then sample the class.

Regression

- Regression is similar to classification, but the output is continuous.
- Eg, models that can predict income level, marks in range 1–10, etc.

Linear regression

- Assumes that the output y is a linear combination of the input x.
- In other words:

$$y(\mathbf{x}) = \sum_{j=1}^{D} w_j x_j + \epsilon$$

  where the w_j make up a weight vector $\mathbf{w}$, and ϵ is the residual error.

- “Learning” is the process of figuring out what the w_j should be to give a good mapping.
- With different w_j we get different functions f.

Linear regression

- For a one dimensional input x, we write:

$$\begin{aligned}\mathbf{x} &= (x_0, x_1)\\ x_0 &= 1\\ x_1 &= x\end{aligned}$$

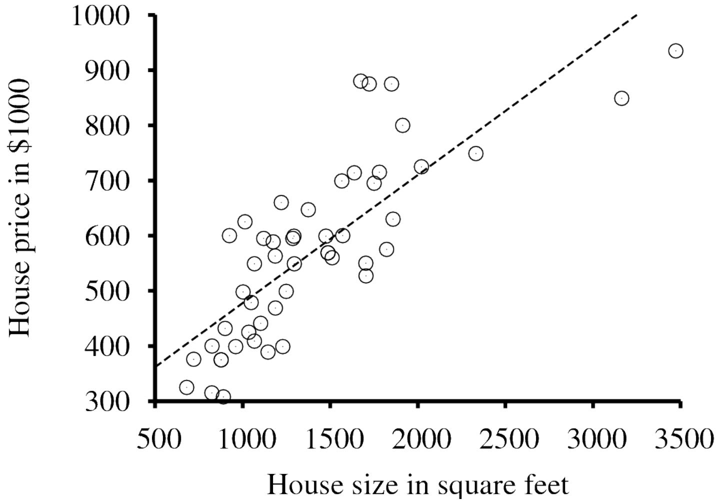
which is the same as saying $\mathbf{x} = (1, x)$. Then

$$\hat{y}(\mathbf{x}) = w_0 + w_1 x_1$$

- We call w_0 the intercept or bias and w_1 is the slope.
- Adding the intercept into $\mathbf{x}$ makes it simpler to learn.

Regression

- Here $x_i \in \mathbb{R}$ and $y_i \in \mathbb{R}$, and the function $f(\mathbf{x})$ is a linear mapping from input to output.



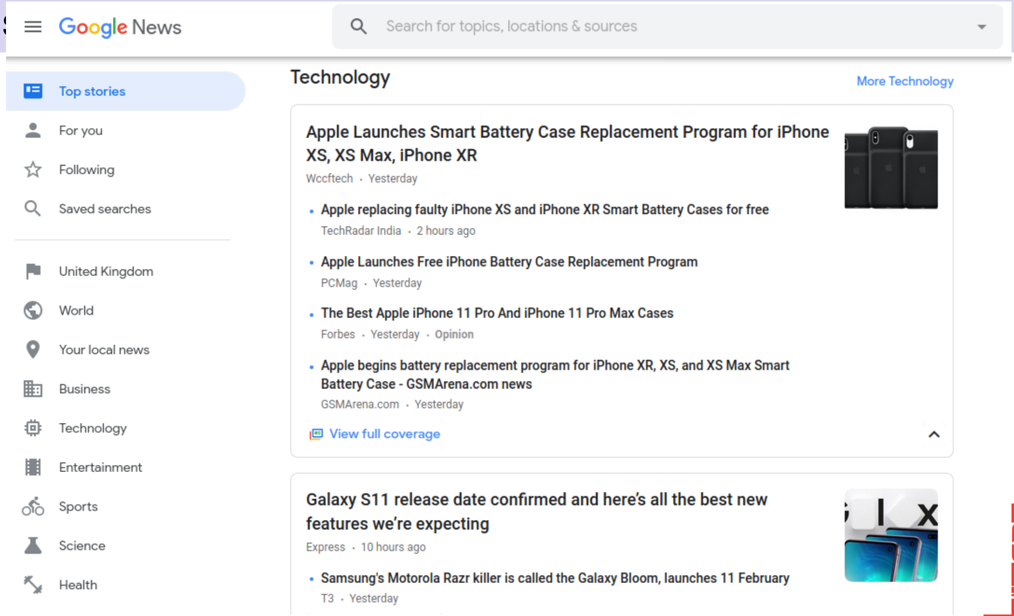
Unsupervised Learning

- Unsupervised = there are no labeled or annotated data.
- Unlike supervised learning, here we are not given the right answer / not told what the desired output is for each input.
- Start from a set of examples:

$$\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$$

  and look for patterns / interesting "structure" in the data.
- The most common form of unsupervised learning is clustering.
- Looking to partition data into groups / clusters: data within a cluster should be similar.
- Clusters are inferred from the data.

Uns



Technology

More Technology

Apple Launches Smart Battery Case Replacement Program for iPhone XS, XS Max, iPhone XR
Wccftech · Yesterday

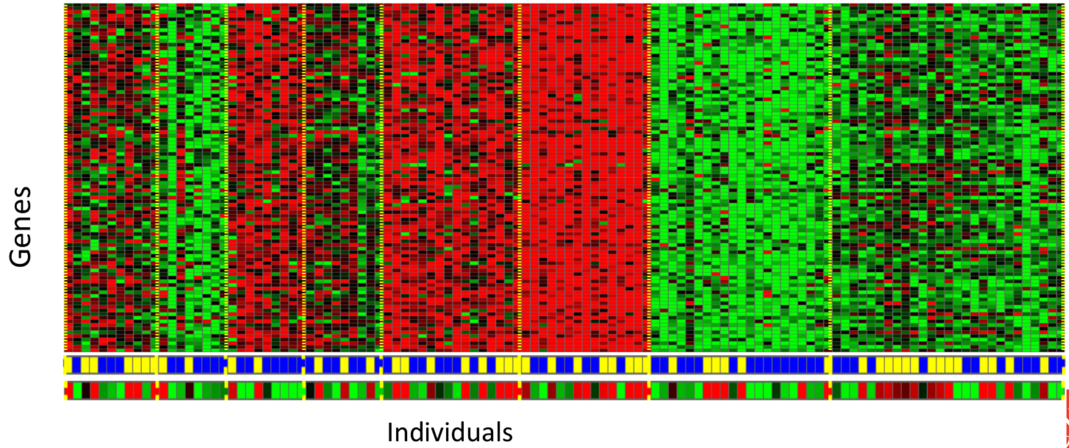
- **Apple replacing faulty iPhone XS and iPhone XR Smart Battery Cases for free**
  TechRadar India · 2 hours ago
- **Apple Launches Free iPhone Battery Case Replacement Program**
  PCMag · Yesterday
- **The Best Apple iPhone 11 Pro And iPhone 11 Pro Max Cases**
  Forbes · Yesterday · Opinion
- **Apple begins battery replacement program for iPhone XR, XS, and XS Max Smart Battery Case - GSMArena.com news**
  GSMArena.com · Yesterday

View full coverage

Galaxy S11 release date confirmed and here's all the best new features we're expecting
Express · 10 hours ago

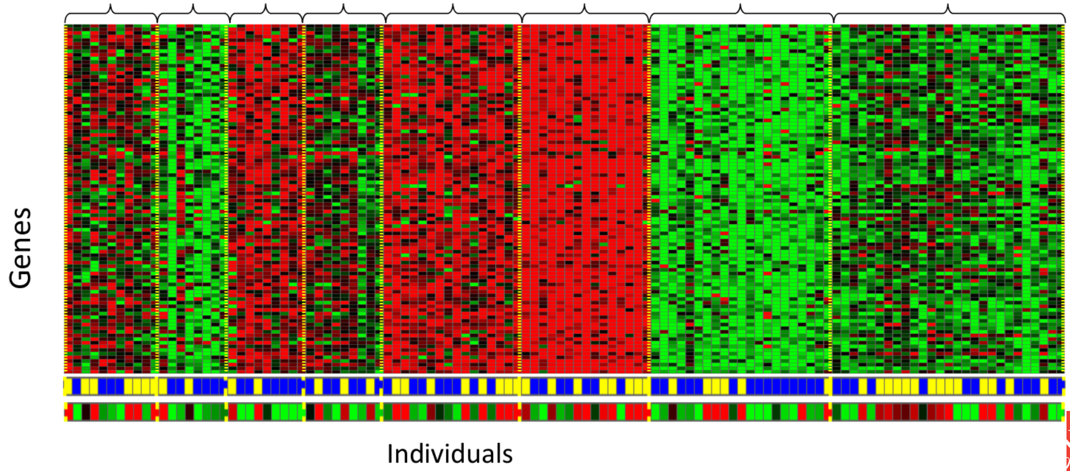
- **Samsung's Motorola Razr killer is called the Galaxy Bloom, launches 11 February**
  T3 · Yesterday

Unsupervised learning: clustering gene expression data



[Andrew Ng; Daphne Koller]

Unsupervised learning: clustering gene expression data



[Andrew Ng; Daphne Koller]

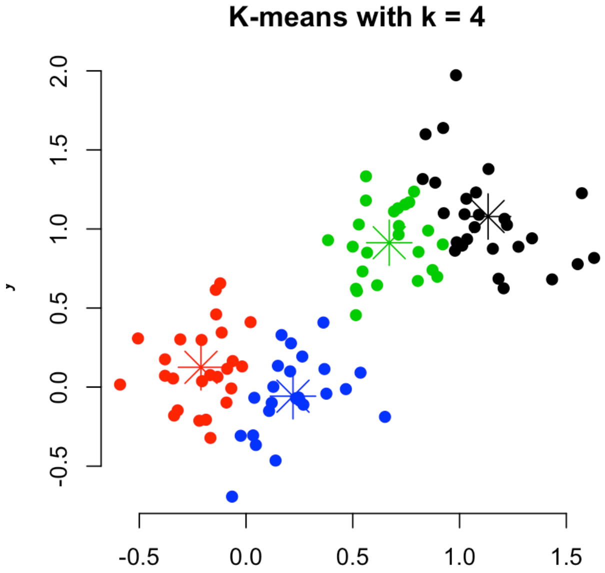
Unsupervised Learning

- Sometimes called knowledge discovery.
- One way that we will think of unsupervised learning is as density estimation.
- We take the task to be to build models of the form:

$$p(\mathbf{x}_i|\mathcal{D})$$

- Two differences from the supervised case:
  1. We don't write $p(y_i|\mathbf{x}_i, \mathcal{D})$, because we don't have a set of class labels to predict.
  2. $\mathbf{x}_i$ is typically a vector of features, so our prediction is a multivariate probability distribution (vs univariate probability models). That makes it more complex.
- Labelled data is not common "in the wild", so arguably unsupervised learning is more natural than supervised learning.

Clustering – grouping data into clusters



(www.sthda.com)

Clustering

- Let K be the number of clusters.
- Need to estimate the distribution over the number of clusters $p(K|\mathcal{D})$.
- Often this is simplified by approximating $p(K|\mathcal{D})$ by its mode:

$$K^* = \arg\max_K p(K|\mathcal{D})$$

- Clearly picking the right K is important.
- This is an example of model selection

Parametric vs. non-parametric

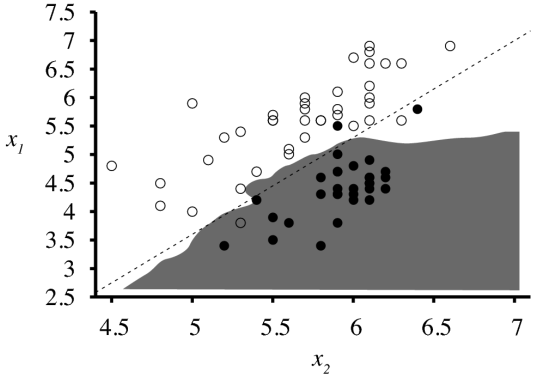
- Basic distinction between types of ML technique.
- Does the technique make assumptions about the structure of the data?
  - Like, “there are four Gaussian clusters”.
- If so, it is parametric.
- If not, it is non-parametric.

Parametric vs. non-parametric: pros and cons

- Parametric models can be computationally simpler (linear regression)
- But, the assumptions that they make can lead to inaccuracy.
- Non-parametric models are more flexible (no assumptions).
- But, they can be intractable for large datasets.

k-nearest neighbour (kNN)

- Simple non-parametric classifier.
- Looks at the *k* points in the training set that are nearest to the test input **x**.



(Russell & Norvig)

- The simplest version of kNN chooses the class with the most points in the *k* nearest set.

K-nearest neighbour

- A more sophisticated version uses the *k* nearest points to estimate the probability of class membership:

$$p(y = c|\mathbf{x}, \mathcal{D}, K) = \frac{1}{K} \sum_{i \in N_K(\mathbf{x}, \mathcal{D})} \mathbb{I}(y_i = c)$$

where:

- $N_K(\mathbf{x}, \mathcal{D})$ are the indices of the *k* nearest points to $\mathbf{x}$ in $\mathcal{D}$, and
- $\mathbb{I}(e)$ is an indicator function such that:

$$\mathbb{I}(e) = \begin{cases} 1 & \textit{if e is true} \\ 0 & \textit{if e is false} \end{cases}$$

- Counts how many members of each class are in the *k* nearest set.

K-nearest neighbour

- Obviously “nearest” needs a notion of distance.
- Typically Euclidian distance (in a suitable dimension) is used.
- This limits the data to being real-valued.

Scaleability

- kNN classifers are simple and can work well.
- However, they scale badly: do not work well with high dimensional inputs.
- To establish the nearest neighbours we need to run through the set of examples computing distance.
    - Distance can be a nasty computation for high dimension/feature data.
- Can address this by clever storage of examples.
- In general, get better performance by summarising the examples rather than using them directly.
  (As other classifiers do).

Curse of dimensionality

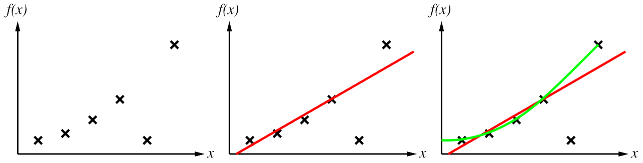
- The scaleability of kNN is related to the more general problem of the curse of dimensionality.
- To better distinguish between examples, add more features (dimensions).
- However this leads to a need for more examples.
- With more features we need more examples to determine how the features distinguish.
- Otherwise we overfit.
- Requirement for examples grows exponentially.

Parametric models

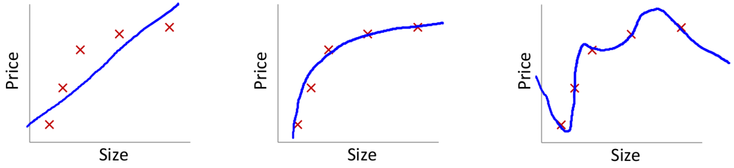
- One way to address the curse is to use parametric models.
- Make some assumption about the best way to distinguish between examples.
- Means assuming something about the way examples are distributed/generated.
- We assume something about $p(y|\mathbf{x})$ and $p(\mathbf{x})$

Overfitting

- When we learn very flexible models, it is possible that the result is too specialised to the data.
- This is overfitting: modeling minor variations in the input.
- Results in low errors on the training data and high errors on new examples.



(Russell & Norvig)

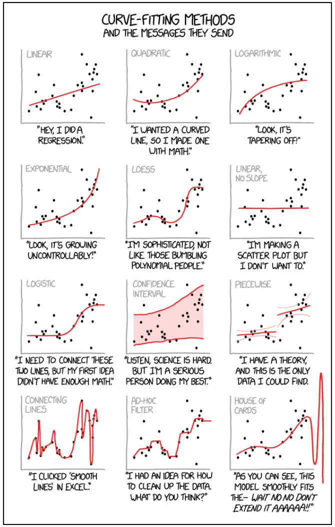


(Andrew Ng)

Overfitting

- How can we tell if we have overfit?
- Can answer that if we know if we have learnt well.

Overfitting


CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
LINEAR, NO SLOPE
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO."
LOGISTIC
"I NEED TO CONNECT THESE TWO LINES, BUT MY FIRST IDEA DIDN'T HAVE ENOUGH MATH."
CONFIDENCE INTERVAL
"LISTEN, SCIENCE IS HARD. BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"I HAVE A THEORY, AND THIS IS THE ONLY DATA I COULD FIND."
CONNECTING LINES
"I CLICKED 'SMOOTH LINES' IN EXCEL."
AD-HOC FILTER
"I HAD AN IDEA FOR HOW TO CLEAN UP THE DATA. WHAT DO YOU THINK?"
HOUSE OF CARDS
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE– WAIT NO NO DON'T EXTEND IT AAAAAA!!"

xkcd.com/2048/

Performance measurement

- How do we know if we have learnt well?
- In supervised learning, this means "can we predict y_i well, given x_i?".
- Compute the misclassification rate

$$err(F, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(F(x_i) \neq y_i)$$

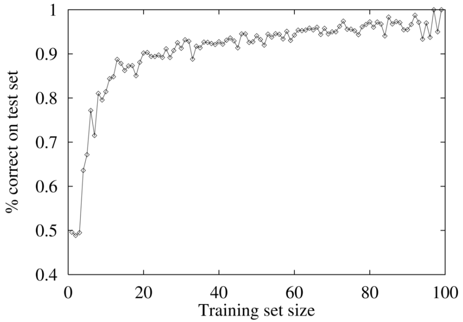
  on the data we have as examples to learn from.
- Or its converse, the proportion of correctly classified examples.

Unseen test set

- Misclassification rate gives us some idea whether we have learnt well from the training data.
- But it doesn't say anything about new data.
- What we care about is generalization error.
- That is misclassification rate on data we haven't seen.
- Can estimate this by trying the classifier on test data that we didn't see in training.
- Assuming test data is manually annotated with (gold) labels.

Learning curve

- Learning curve = % correct on test set as a function of training set size



(Russell & Norvig)

- A (somewhat real) example.

Unseen test set and seen validation set

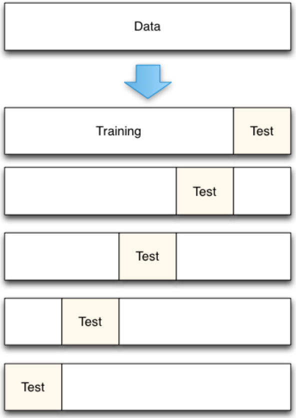
- Partition the training set into three subsets: training set, validation set (aka development set), and test set.
- Use validation set for model selection and/or parameter tuning.
- Test set to evaluate generalisation performance of best model according to validation set.
- Typical split of the data: 80% train; 10% dev; 10% test.
- Need to ensure enough data for training but also enough data for tuning and testing.

Unseen test set and seen validation set

- Partition the training set into three subsets: training set, validation set (aka development set), and test set.
- Use validation set for model selection and/or parameter tuning.
- Test set to evaluate generalisation performance of best model according to validation set.
- Typical split of the data: 80% train; 10% dev; 10% test.
- Need to ensure enough data for training but also enough data for tuning and testing.
- But, what happends if the usable number of training examples is very small?

K-fold cross-validation (CV)

- Split data into k equal subsets / folds to test on. Train on $k-1$ sets and test on the remainder fold only.
- Repeat k times (test on each fold only once).
- Computer misclassification error averaged over all k test folds.

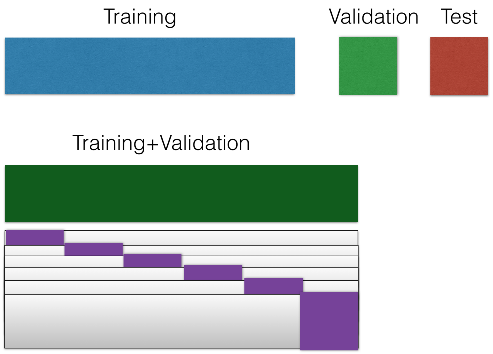


K-fold cross-validation

- The final performance is the average of the performances for each fold.
- Average test set score is a better estimate of the error rate than a single score.
- Common values of k are 5 and 10, both giving error estimates that are very likely to be accurate.
- The extreme case is when $k = n$, the number of data points.
- Leave-one-out cross validation.

Model selection and/or parameter tuning

- How about model selection and/or parameter tuning?
- Use CV + separate unseen test set to measure generalisation performance:



No free lunch theorem

- There is no best model for all scenarios.
- So, we have to do model selection.
- And models may have different algorithms for learning, which also have tradeoffs.
  Speed, accuracy, complexity.

More performance metrics

- Accuracy
- Mainly aimed at classification problems.

Accuracy

- Accuracy measures the proportion of “correct” predictions.

$$accuracy = \frac{correct}{total}$$

- Dual of the misclassification rate.
- Higher is better.
- But what is “good” depends on the number of classes.

Accuracy

- Consider a two class problem with an equal number of examples in each class.
- Picking randomly would give 50% accuracy.
- Consider a problem with 286 examples, 201 in class “no” and 85 in class “yes”
- A classifier that always said “no” (majority classifier) would be 70% accurate.
- So maybe a classifier is only good if it gets more than 70% right in this case.

Accuracy

- But what if the “yes” cases really mattered? (eg, a breast cancer detection, abusive language detection).
- False negatives might be more important than false positives
- (Or vice versa).

Confusion matrix

- Method to help better understand what is going on.
- Compare actual class labels with predicted labels.

		actual	
		yes	**no**
predicted	**yes**	*true positive*	false positive
	no	false negative	*true negative*

- Can expand for any number of class labels.

Confusion matrix

- Here is a silly “all no” breast cancer classifier in a confusion matrix:

		actual	
		yes	**no**
predicted	**yes**	0	0
	no	85	201

- The lack of true positives, and false negatives should make us think again.

Confusion matrix

- Here is a more reasonable classifier, albeit one with a lower accuracy:

		actual	
		yes	**no**
predicted	**yes**	10	13
	no	75	188

- You can calculate the accuracy for comparison.
- I might prefer fewer false negatives still. . . though there will typically be a tradeoff in pushing the false negative rate down.

Precision

- Precision is the number of true positives divided by the sum of the true positives and the false positives:

$$Precision = \frac{TP}{TP + FP}$$

- How many of the “yes” predictions we made are correct?
- Positive predictive value.

Recall

- Recall is the number of true positives divided by the sum of true positives and false negatives.

$$Recall = \frac{TP}{TP + FN}$$

- How many of all the true “yes” examples we picked correctly.
- Sensitivity, or true positive rate.

F_1 Score

- Balances precision and recall, weighting them equally:

$$F = 2\left(\frac{precision \times recall}{precision + recall}\right)$$

- Harmonic mean of precision and recall.
- One of a family of measures:

$$F_\beta = (1 + \beta^2)\left(\frac{precision \times recall}{\beta^2 precision + recall}\right)$$

- F_2 and $F_{0.5}$ are also commonly used.
- $\beta = 2$ weights recall higher. $\beta = 0.5$ weights precision higher.

ROC curve

- So far we have talked as if a classifier can make a crisp distinction between "yes" and "no".
- As discussed above, most classification problems can be cast as a probabilistic prediction returning:

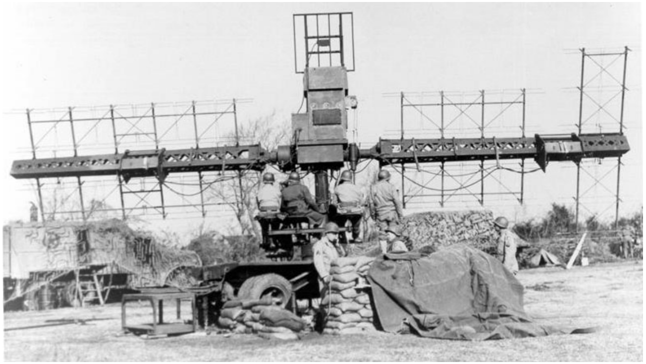
$$p(y_i|\mathbf{x}_i, \mathcal{D})$$

and we turn this into "yes" and "no" using:

$$classification = \begin{cases} yes & if\ p(y_i|\mathbf{x}_i, \mathcal{D}) > threshold \\ no & otherwise \end{cases}$$

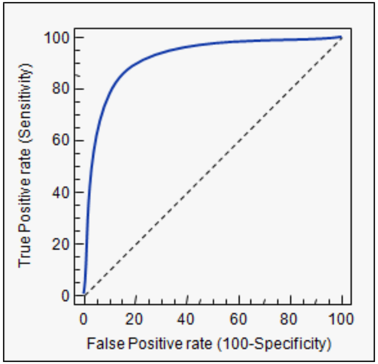
- Moving the threshold changes *TP* and *FP*.

ROC Curve


(US Army Signal Corps)

- Receiver operating characteristic.
- First used in WWII to analyse radar signals.

ROC Curve



(medcalc.org)

- Plot pairs of TP and FP (as we vary the threshold).
- Dotted line is performnce of a random classifier (on average).
- For a good system, the graph climbs steeply on the left side.
- Area under the curve is related to the probability that the classifier will correctly classify a randomly chosen example.

Evaluating clusters

- What we have seen so far works for classification problems.
- What about clustering?
- If you know what the clusters should be, then the classification measures can be used.
- But most of the time we don't know what the clusters should be.

Evaluating clusters

- External evaluation: eg, how well the number of clusters serves a downstream task (eg, market segmentation).
- Internal evaluation:

  Try to establish how coherent the clusters are and how well they are separated from each other.

Evaluating clusters

- For example, the Davis-Bouldin index for n clusters:

$$DB = \frac{1}{N}\sum_{i=1}^{n} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

  where c_i is the centroid of cluster i, σ_i is the average distance of points in cluster i to the centroid of the cluster, and $d(\cdot, \cdot)$ is a distance metric.
- The DB index will be small when clusters are tightly grouped and far from each other.

Summary

- Types of ML
  - Supervised
  - Unsupervised
  - Reinforcement
- Some simple models.
- Basic concepts
  - Parametric
  - Non-parametric
- ML in practice
  - Training/testing
  - Cross-validation
  - Precision/recall
  - ROC curve