» Overview

- Feature engineering
- Mapping model output
- Over-fitting and under-fitting

» Fitting a nonlinear curve using a linear model?

* Linear model:

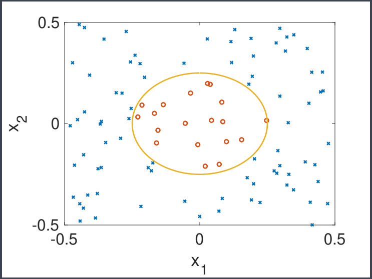
$$\hat{y} = \theta^T x \qquad \text{(regression)}$$
$$\hat{y} = sign(\theta^T x) \qquad \text{(classification)}$$

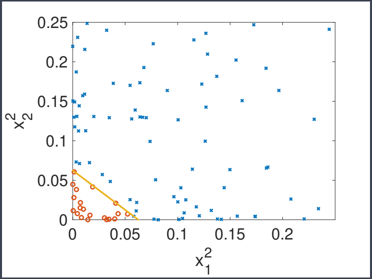
* *Linear* because plotting $\theta^T x$ vs x gives a straight line when we have only one feature (x is a scalar), a plane when we have two features and a hyperplane when > 2 features.
* Linear model seems simple but we have freedom in how we map from inputs to features x ...

» Separating by Nonlinear Curves: Choosing Features

Example (classification):

- We have training with two features x_1, x_2 and ± 1 labelled outputs.
- Suppose when data is not linearly separable but rather has a circle separating the two classes
- Equation for a circle of radius R is $x_1^2 + x_2^2 = R^R$ ...
- ... so change our features to be $x_1^{new} = x_1^2$, $x_2^{new} = x_2^2$
- Using this new feature vector our model can be rewritten as $sign(\theta_0 + \theta_1 x_1^{new} + \theta_2 x_2^{new})$, which is now linear.

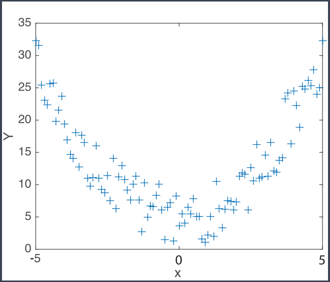


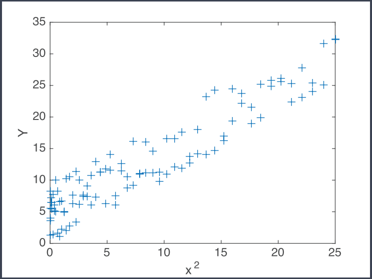


» Fitting Nonlinear Curves: Choosing Features

Example (regression):

* Data is generated by quadratic function of input x, with added noise
* Define feature vector x^{new} with $x^{new} = x^2$. This new x^{new} can be computed given input x, so its known.
* Using this new feature vector our prediction can be rewritten as $h_\theta(x^{new}) = \theta_0 + \theta_1 x^{new}$, which is a linear model in x^{new}.





» Feature Engineering

- Basic idea:
  - Start with some features
  - Process or transform them to produce new (“engineered”) features
  - Use these new features in your predictor/model
- Was it a good idea? Did new features improve predictions?
  - Train model with original features, evaluate performance
  - Train model with new features, evaluate performance
  - Compare e.g. using cost function ... much more on this shortly

» Feature Engineering

- Modify individual feature
  - E.g. replace feature x_i by normalised version $x_i^{new} = (x_i - \mu_i)/\sigma_i$, μ_i the mean of x_i in training data, σ_i the standard deviation.
- Create multiple features from one original feature
  - E.g. powers, replace x_i by $(x_i, x_i^2, x_i^3, \dots, x_i^q)$
- Create new feature from multiple original features
  - E.g. product, $x_i^{new} = x_j x_k$

» Creating new feature from multiple original features

* Products can be used to model interactions between features
* E.g. suppose original features x_1, x_2, x_3 are boolean with values 0 or 1 (perhaps representing medical symptoms or whether a shopping basket contains an item).
* model: $\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1 x_2 + \theta_5 x_1 x_3 + \theta_6 x_2 x_3$
    * θ_1 is the amount our prediction increases when $x_1 = 1$, similarly θ_2, θ_3
    * θ_4 is the amount our prediction increases when $x_1 = 1$ *and* $x_2 = 1$ e.g. when θ_4 is large then prediction increases a lot when $x_1 = 1$ and $x_2 = 1$
    * Similarly, θ_5 is the amount our prediction increases when $x_1 = 1$ and $x_3 = 1$, θ_5 when $x_2 = 1$ and $x_3 = 1$.

» Fitting Nonlinear Curves: Choosing Outputs

Covid Example

* We're not confined to playing around with the mapping from inputs to features, we can also change the mapping from the model output to the *actual* output of interest.
* Growth of an infection within the population, e.g. Covid, is often modelled as being exponential i.e.

$$z_k = e^{ak} z_0$$

  where z_k is the number of infected people at day k and a is a growth parameter. When $a < 0$ then the infection decays over time, when $a > 0$ then it grows.
* Taking logs:

$$\log z_k = ak + \log z_0$$

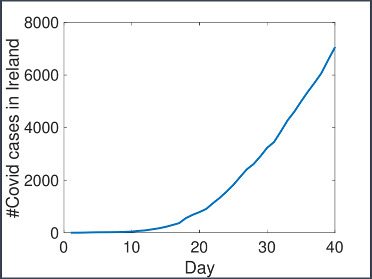
* Define model output $y_k = \log z_k$. Then

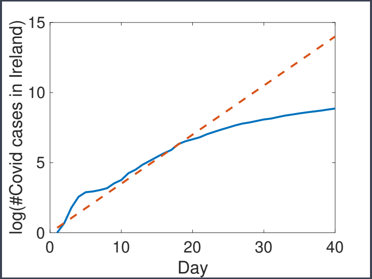
$$y_k = ak + y_0 = \theta_1 k + \theta_0$$

  which is a linear model with unknown parameters θ_0 and θ_1. Use linear regression to estimate these from data.

» Fitting Nonlinear Curves: Choosing Outputs

Covid Example (cont):





- Covid-19 data for Ireland, starting 3rd March 2020
- Red dashed line is linear regression fit to transformed data

» Hand-crafted vs automatic features

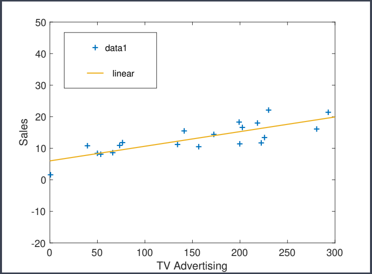
* Feature selection and feature engineering generally done by hand, using insight into task and experience (this is part of the "art" of machine learning).
* Can also develop feature mappings automatically by training on data ...
* E.g. word2vec[1], vgg16 (neural net) were developed automatically from v large data sets. Will come back to these later.
* ... tricky to get right and often v time consuming

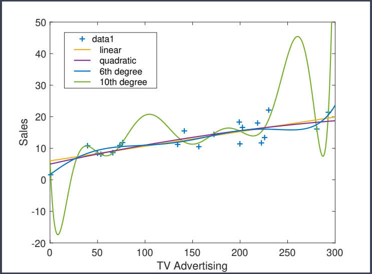
[1]https://en.wikipedia.org/wiki/Word2vec

» Model Selection: Why Not Include Every Possible Feature?

Advertising example again. Thin out data by taking every 10th point. Try a few different models:

- $h_\theta(x) = \theta_0 + \theta_1 x$
- $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
- $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_6 x^6$
- $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$

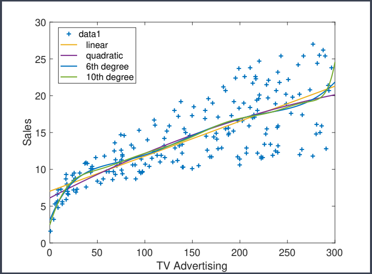


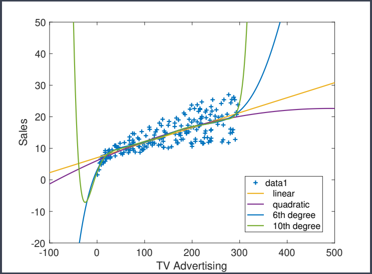


- As we add more parameters, we start to fit the “noise” in the training data, called *overfitting*.
- At points where we don’t have training data the predictions of the model get pretty bad (the model *generalises* poorly).

» Model Selection

More data can help, e.g. when don't thin out data:

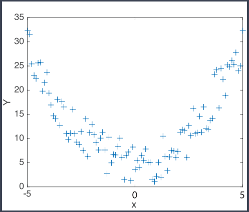




* But even with more data, still our hypothesis doesn't generalise well i.e. doesn't predict well for data outside the training set.

» Model Selection: Underfitting

- We don't want to make our model overly complex, since then it suffers from over-fitting and generalises poorly
- But we also don't want to make it so simple that it fails to capture the behaviour of the data. E.g. if data is quadratic then trying to fit it with a straight line will give poor predictions $\rightarrow$ *under-fitting*



- Striking the right balance between over-fitting and under-fitting is a key task in supervised machine learning, and is intrinsic to all supervised learning approaches i.e it can't be avoided.
- Also referred to as the *bias-variance* trade-off.