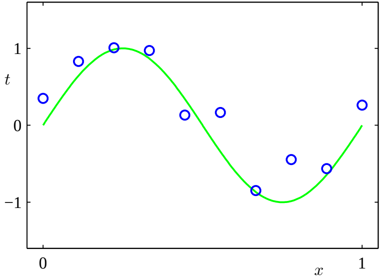
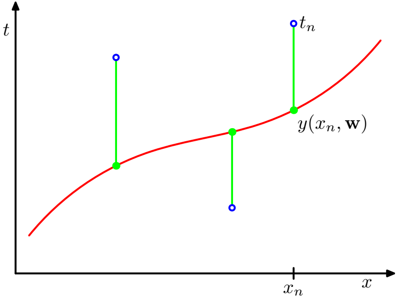
curve-fitting with polynomials...

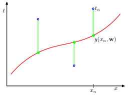
Consider data that comes from a sinusoid plus Gaussian noise.





recap...

Assuming Gaussian noise,



$$\underbrace{P(Y=T|X,w)}_{\text{Likelihood}} = \prod_{n}^{N} P(y(\mathbf{x}_n, w) = t_n)$$

$$\propto \prod_{n}^{N} \exp\left[-\frac{1}{2}\left(y(\mathbf{x}_n, w) - t_n\right)^2 \right]$$

$$\mathcal{L} = \log P(Y=T|X,w)$$

$$= \frac{1}{2}\sum_{n=1}^{N}\Big(y(\mathbf{x}_n, w) - t_n\Big)^2$$

So calculating the Max Likelihood solution amounts to minimizing the sum of squared errors on a training set of N pairs.

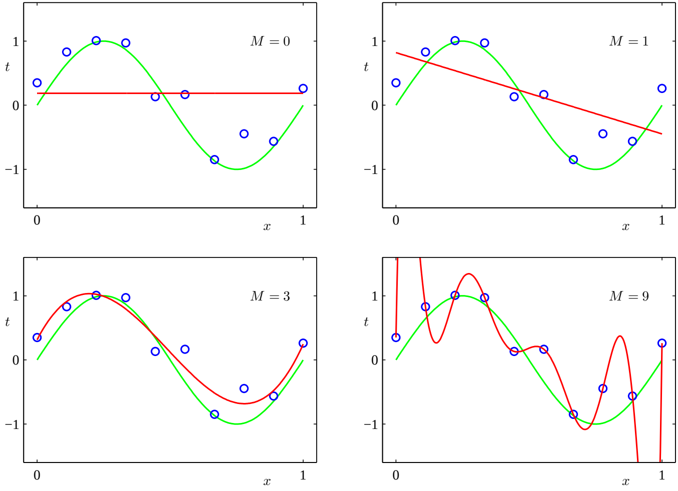
we need a form for y: how about a polynomial?

$$y(x, w) = \sum_{j=0}^{M} w_j \, x^j$$

Examples...

surprising factoid: for $y(x_n, w)$ a polynomial, the optimal values for w can be found in closed form (out of scope...). This won't usually be true!

Consider a training set with $N = 10$ pairs in it.
Best-fit curves for various values of M:

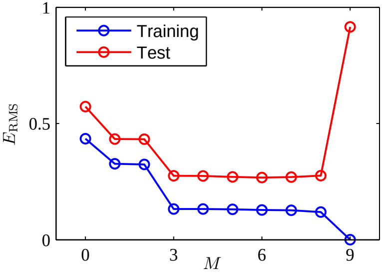


Model selection problem: which M to use?

hold-out strategies for complexity control

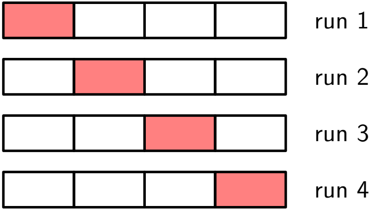
test set:

(Note both under-fitting and over-fitting)



But this “hold out” set wastes data.

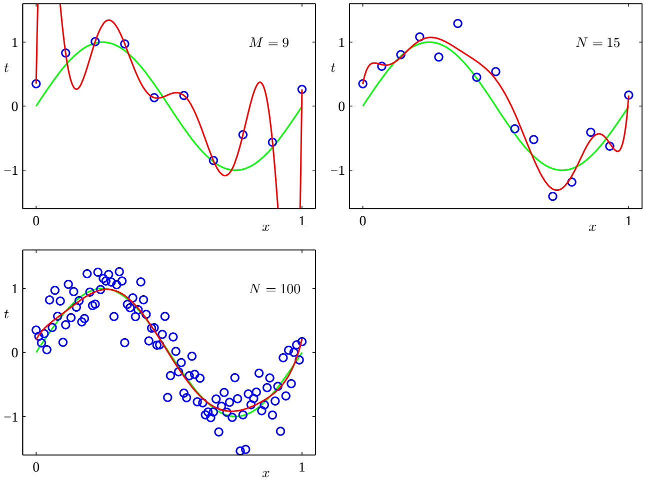
cross-validation:



(extreme case is “leave-one-out”)

- have to re-train lots of times
- multiple control parameters $\longrightarrow$ many combinations to be tried out.

More data always helps: (here $M = 9$ but N goes 10, 15, 100)



Notice: we're deciding M differently when there's more / less data.
What do we think of this?

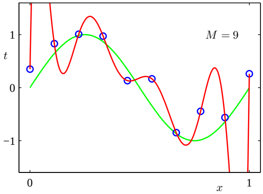
an alternative “knob” controlling complexity

Nb: the best-fit parameters w get bigger for the high-M models.
So how about using a big M but penalize big w?
We could have a new cost function:

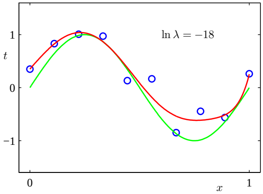
$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 \;+\; \frac{\lambda}{2}||w||^2$$

In classical stats, this is called “regularisation”, “shrinkage”, ...

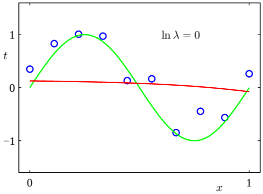
Now λ controls the model complexity, instead of M.
We *still* have a model selection problem, requiring cross-validation: what should λ be? (Here, $M = 9$ in all cases)



$\lambda = 0$

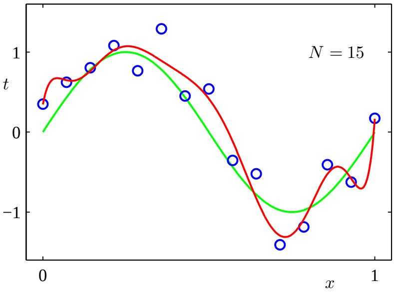


$\lambda = e^{-18}$



$\lambda = 1$

pros and cons of the polynomial family



✓ arbitrarily powerful models (as $M \to \infty$)
✓ solution for the weightings is analytic.

✗ fairly weird functional family...
✗ what's the connection with classification? (*eg.* the digits)
✗ what if input is a vector?! $\mathbf{x} = (x_1, x_2, \ldots, x_K)$. The Curse.

the curse of dimensionality

The volume of space grows rapidly with its dimensionality. You need more parameters to specify the behaviour, and more data points to constrain all those parameters.