

CS-XXX: Introduction to Machine Learning

Curve Fitting

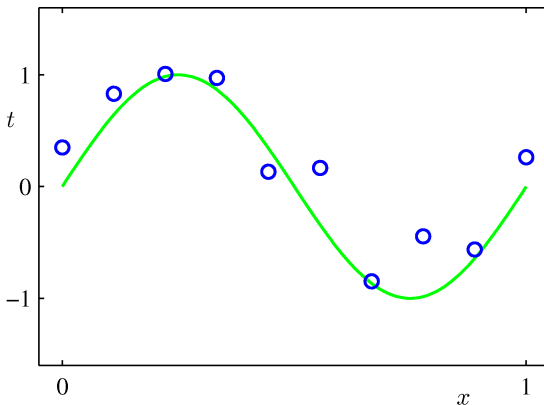
Arbish Akram

Department of Computer Science
Government College University

Slides by Dr. Nazar Khan

Example: Polynomial Curve Fitting

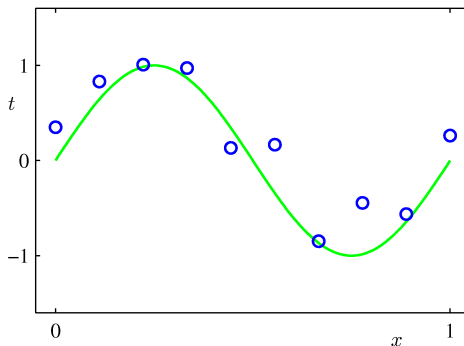
Problem: Given N observations of input x_i with corresponding observations of output t_i , find function $f(x)$ that predicts t for a new value of x .



Real-world Data

Real-world data has 2 important properties

1. underlying regularity,
2. individual observations are corrupted by noise.



Learning corresponds to discovering the underlying regularity of data (the $\sin(\cdot)$ function in our example).

Polynomial curve fitting

- ▶ We will fit the points (x, t) using a polynomial function

$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \cdots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

where M is the *order* of the polynomial.

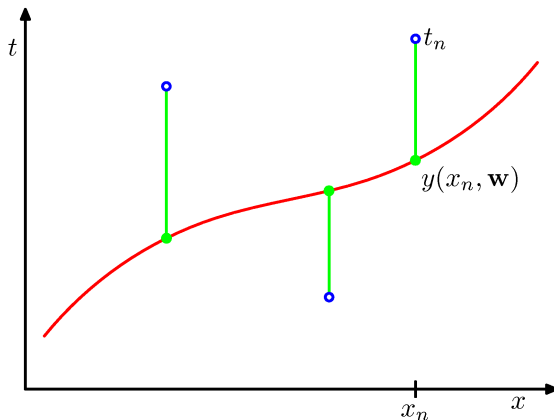
- ▶ Function $y(x, \mathbf{w})$ is a
 - ▶ non-linear function of the input x , but
 - ▶ a linear function of the parameters $\mathbf{w}$.
- ▶ So our model $y(x, \mathbf{w})$ is a *linear model*.

Polynomial curve fitting

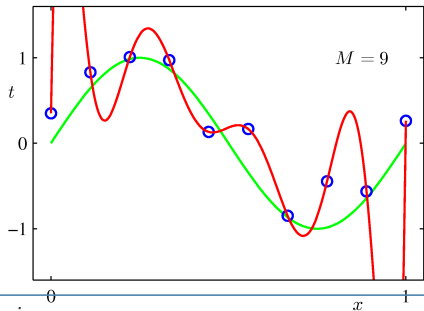
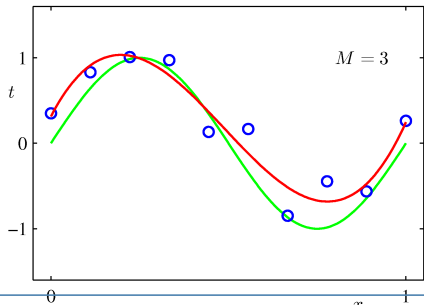
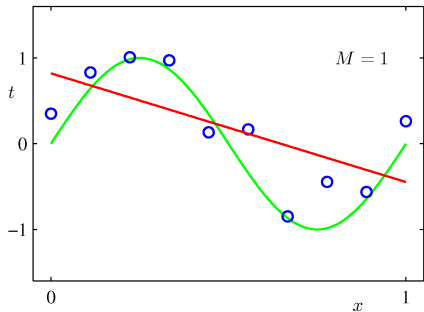
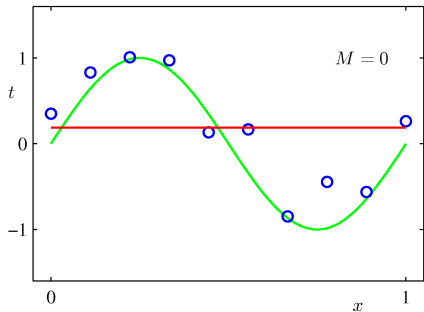
- ▶ Fitting corresponds to finding the optimal $\mathbf{w}$. We denote it as $\mathbf{w}^*$.
- ▶ Optimal $\mathbf{w}^*$ can be found by *minimising* an *error function*

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

- ▶ Why does minimising $E(\mathbf{w})$ make sense?
- ▶ Can $E(\mathbf{w})$ ever be negative?
- ▶ Can $E(\mathbf{w})$ ever be zero?



Geometric interpretation of the sum-of-squares error function.



Over-fitting

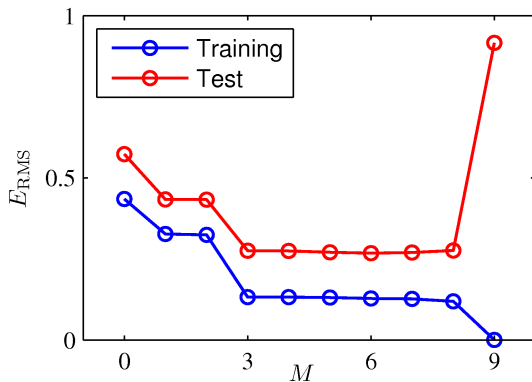
- ▶ Lower order polynomials can't capture the variation in data.
- ▶ Higher order leads to *over-fitting*.
 - ▶ Fitted polynomial passes *exactly* through each data point.
 - ▶ But it oscillates wildly in-between.
 - ▶ Gives a very poor representation of the real underlying function.
- ▶ Over-fitting is bad because it gives bad generalization.

Over-fitting

- ▶ To check generalization performance of a certain $\mathbf{w}^*$, compute $E(\mathbf{w}^*)$ on a *new* test set.
- ▶ Alternative performance measure: root-mean-square error (RMS)

$$E_{RMS} = \sqrt{\frac{2E(\mathbf{w}^*)}{N}}$$

- ▶ Mean ensures datasets of different sizes are treated equally. (How?)
- ▶ Square-root brings the *squared* error scale back to the scale of the target variable t .



Root-mean-square error on training and test set for various polynomial orders M .

Paradox?

- ▶ A polynomial of order M contains all polynomials of lower order.
- ▶ So higher order should *always* be better than lower order.
- ▶ **But**, it's not better. Why?
 - ▶ Because higher order polynomial starts fitting the noise instead of the underlying function.

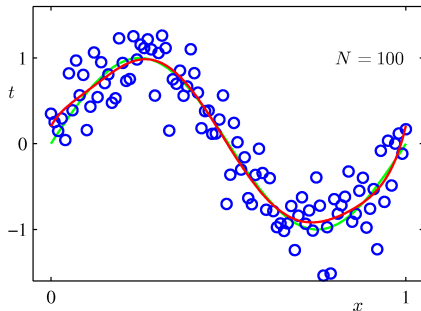
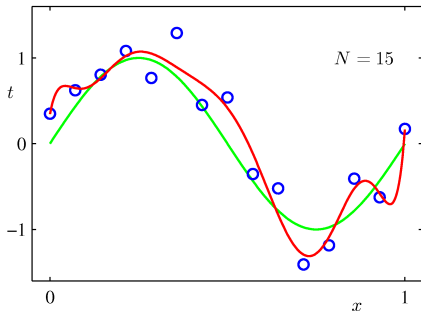
Over-fitting

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

- ▶ Typical magnitude of the polynomial coefficients is increasing dramatically as M increases.
- ▶ This is a sign of over-fitting.
- ▶ The polynomial is trying to fit the data points exactly by having larger coefficients.

Over-fitting

- ▶ Large $M \implies$ more flexibility $\implies$ more tuning to noise.
- ▶ **But**, if we have more data, then over-fitting is reduced.



- ▶ Fitted polynomials of order $M = 9$ with $N = 15$ and $N = 100$ data points. More data reduces the effect of over-fitting.
- ▶ Rough heuristic to avoid over-fitting: Number of data points should be greater than $k|\mathbf{w}|$ where k is some multiple like 5 or 10.

How to avoid over-fitting

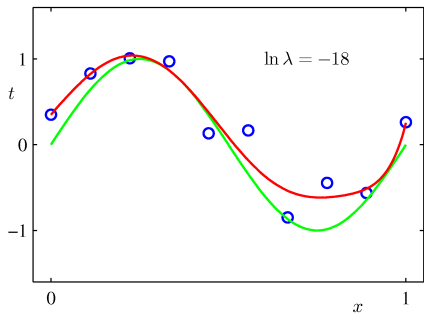
- ▶ Since large coefficients $\implies$ over-fitting, *discourage large coefficients* in $\mathbf{w}$.

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

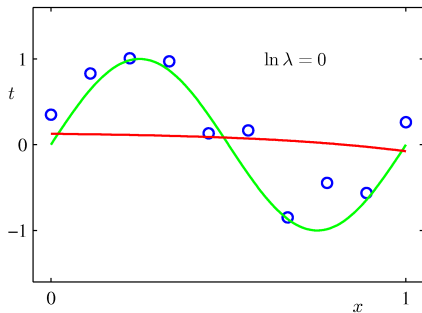
where $\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$ and λ controls the relative importance of the regularizer compared to the error term.

- ▶ Also called *regularization, shrinkage, weight-decay*.

For a polynomial of order 9



For $\lambda = e^{-18}$
No over-fitting



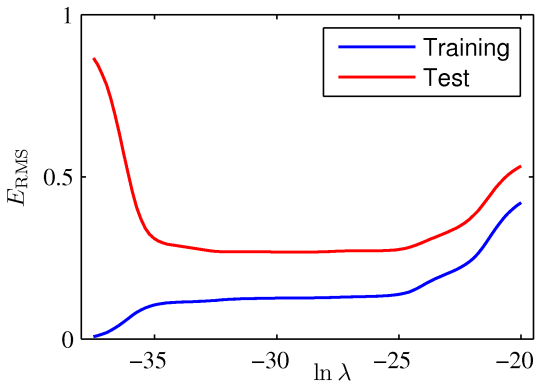
For $\lambda = 1$
Too much smoothing (no fitting)

Effect of regularization

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

- ▶ As λ increases, the typical magnitude of coefficients gets smaller.
- ▶ We go from over-fitting ($\lambda = 0$) to no over-fitting ($\lambda = e^{-18}$) to poor fitting ($\lambda = 1$).
- ▶ Since $M = 9$ is fixed, regularization controls the degree of over-fitting.

Effect of regularization



Graph of root-mean-square (RMS) error of fitting the $M = 9$ polynomial as λ is increased.