Overfitting and Regurlarization in Machine Learning

Based on [Bishop, PRML 05] Ch.1

Feedback in Learning

- Type of feedback:
  - Supervised learning: correct answers for each example
    - Discrete (categories) : classification
    - Continuous : regression
  - Unsupervised learning: correct answers not given
  - Reinforcement learning: occasional rewards

Inductive learning

- Simplest form: learn a function from examples

An example is a pair (x, y) : x = data, y = outcome
assume: y drawn from function f(x) : y = f(x) + noise

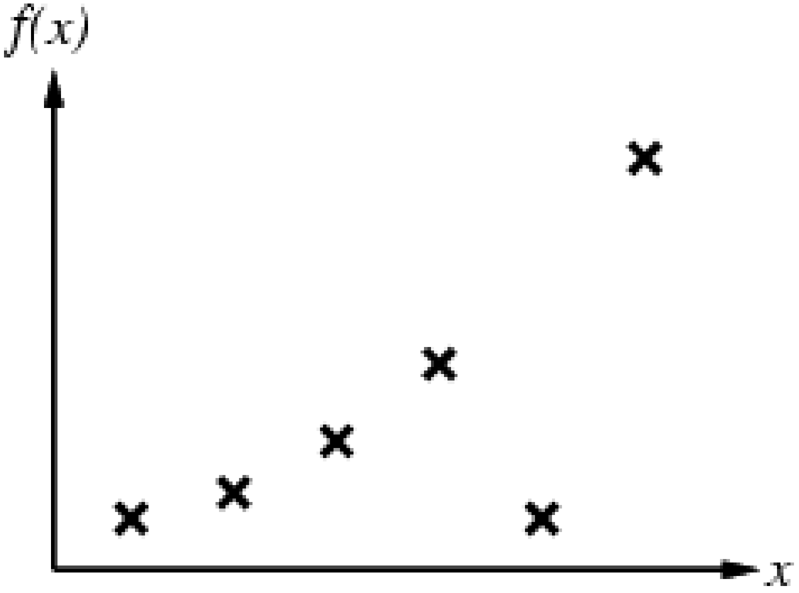
f = target function

Problem: find a hypothesis h
such that $h \approx f$
given a training set of examples

Note: highly simplified model :
- Ignores prior knowledge : some h may be more likely
- Assumes lots of examples are available
- Objective: maximize prediction for unseen data – Q. How?

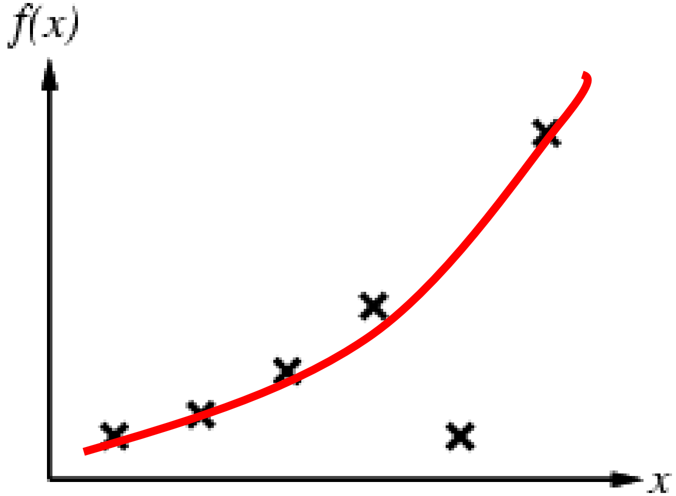
Inductive learning method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



Regression vs Classification

$y = f(x)$



Regression:

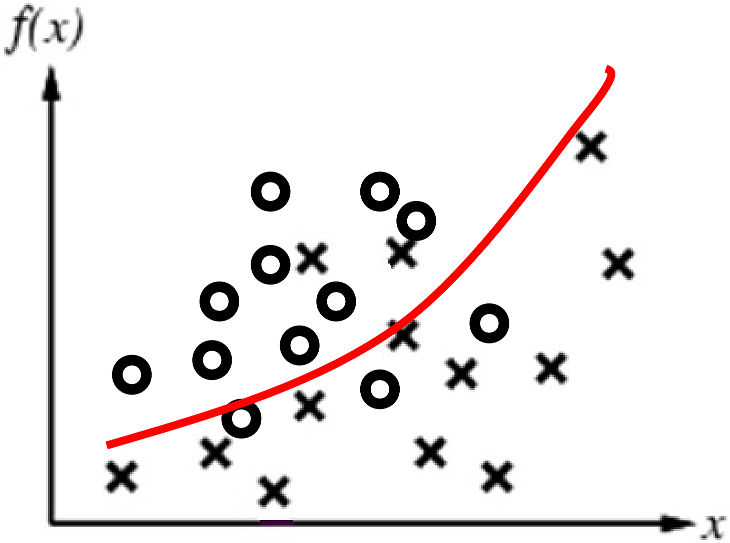
y is continuous

Classification:

y : set of discrete values

e.g. classes C_1, C_2, C_3...

$y \in \{1,2,3...\}$



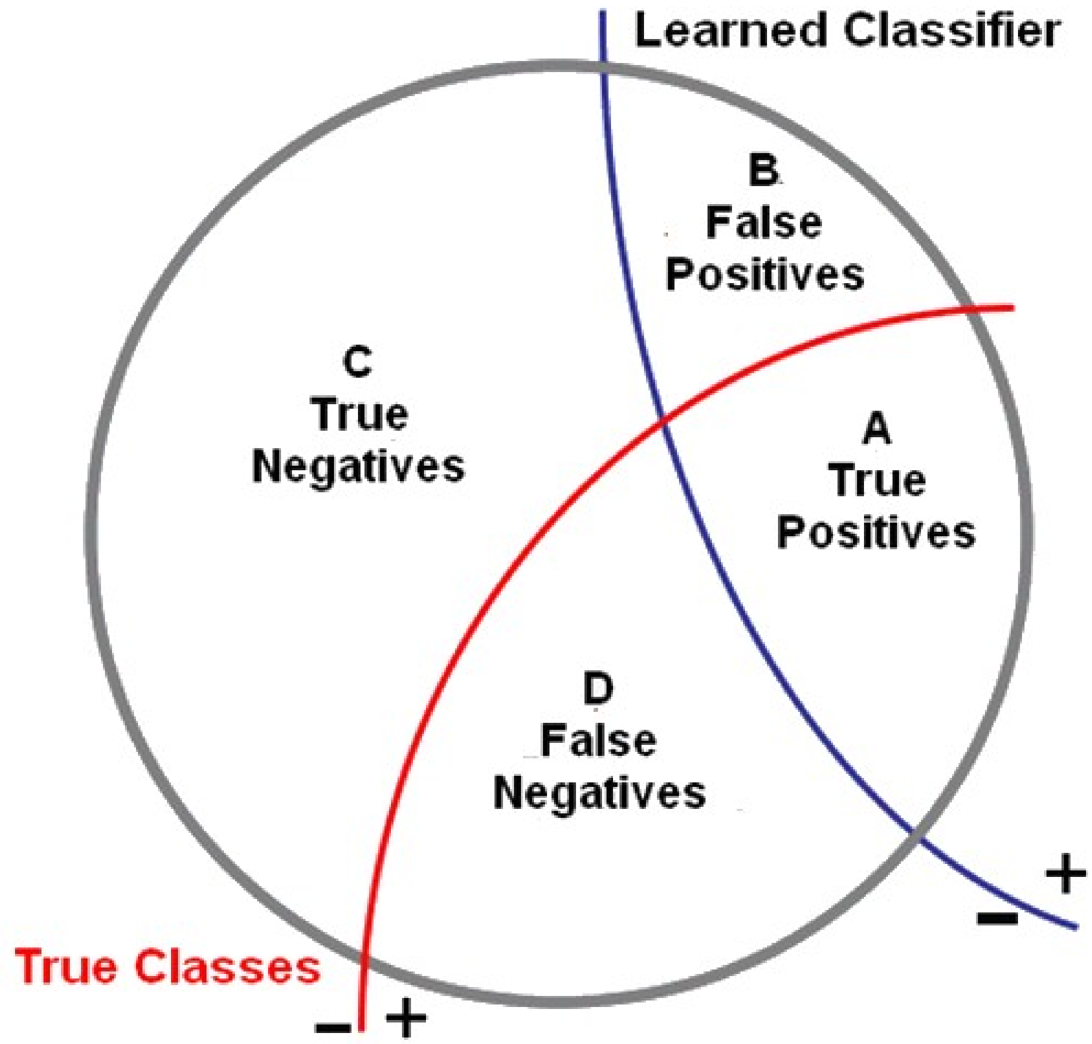
Precision vs Recall

Precision:

A / Retrieved Positives

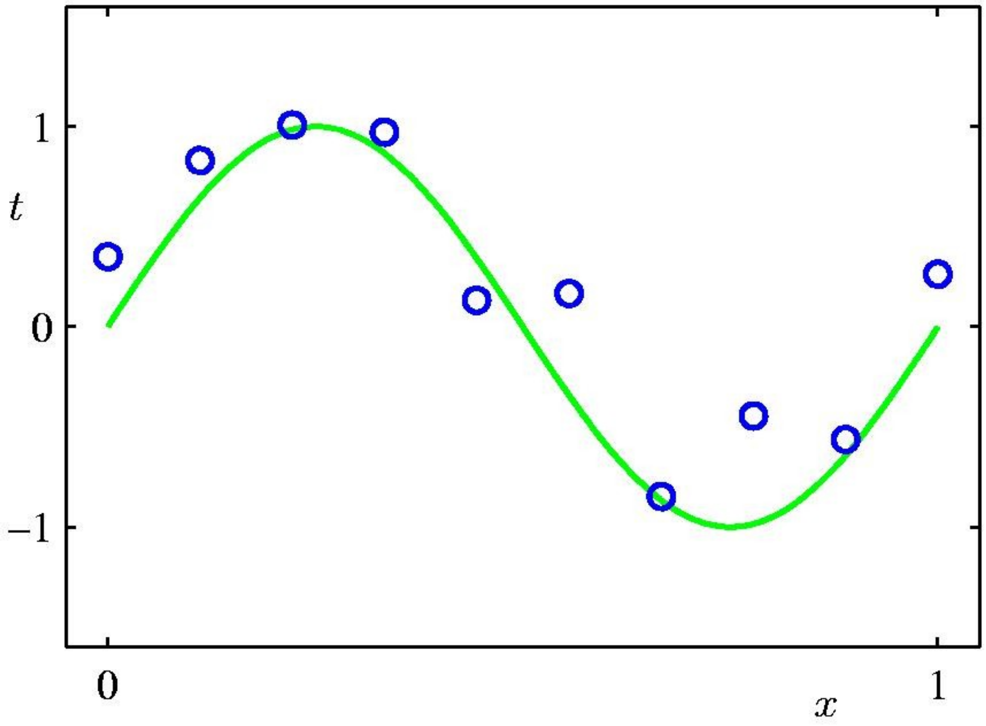
Recall:

A / Actual Positives



Regression

Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Linear Regression

$$\mathbf{y = f(x) = \Sigma_i}\, w_i \,\mathbf{.\ \varphi_i(x)}$$

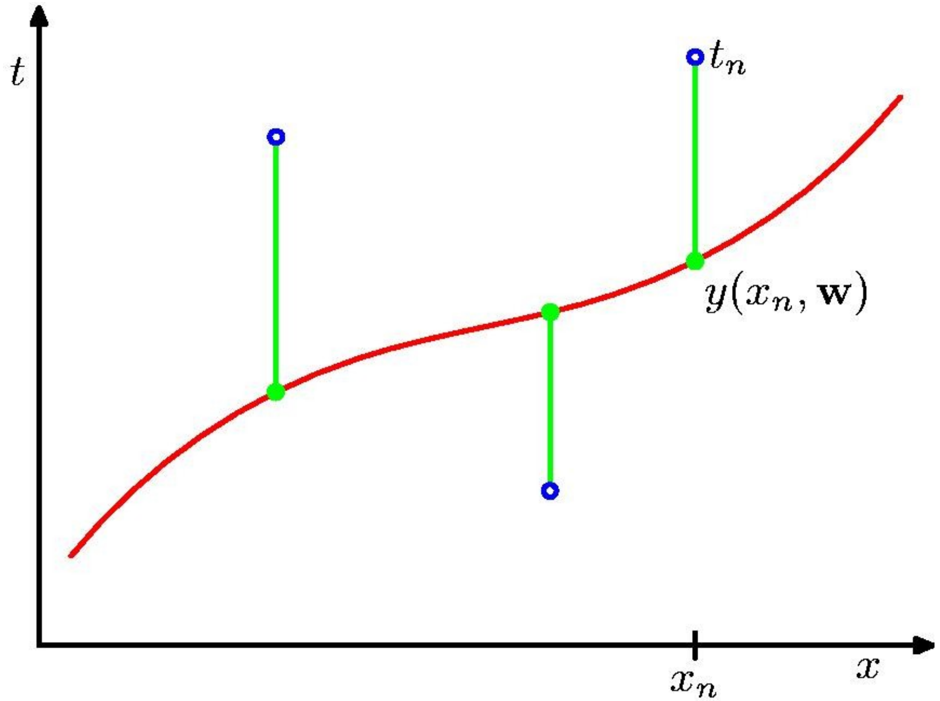
$\boldsymbol{\varphi_i(x)}$: basis function

w_i : weights

Linear : function is linear in the weights

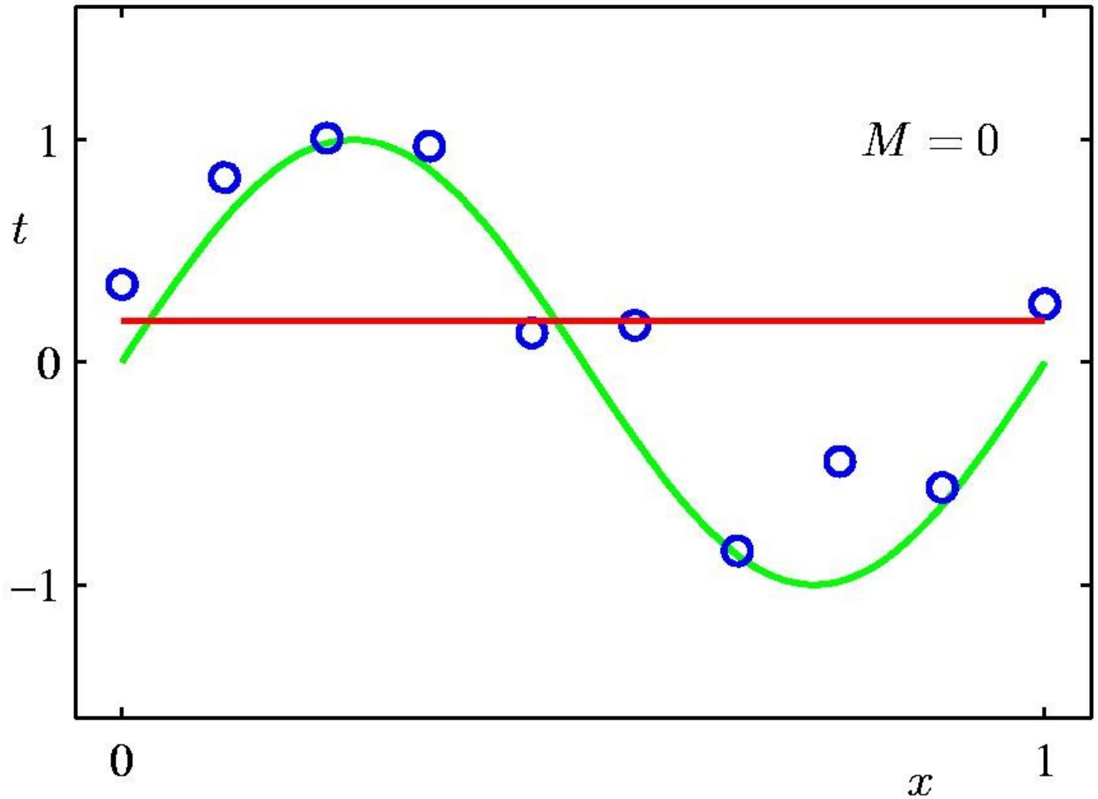
Quadratic error function --> derivative is linear in **w**

Sum-of-Squares Error Function

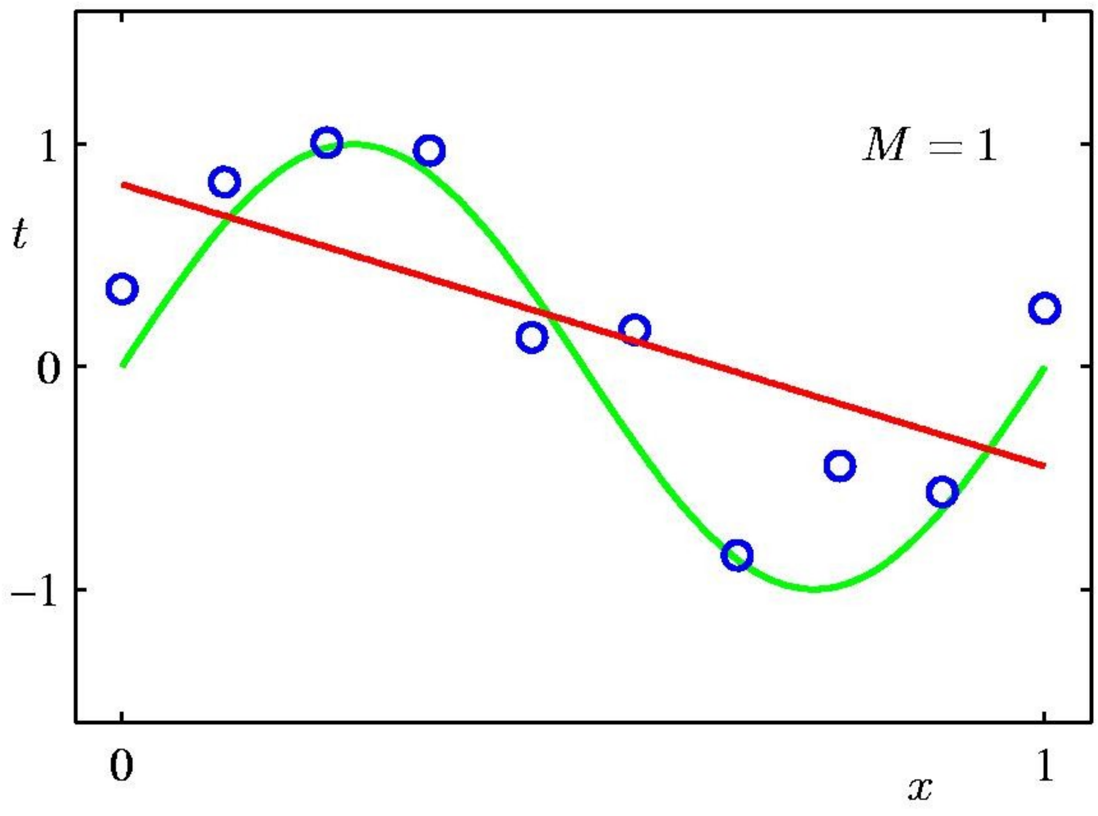


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

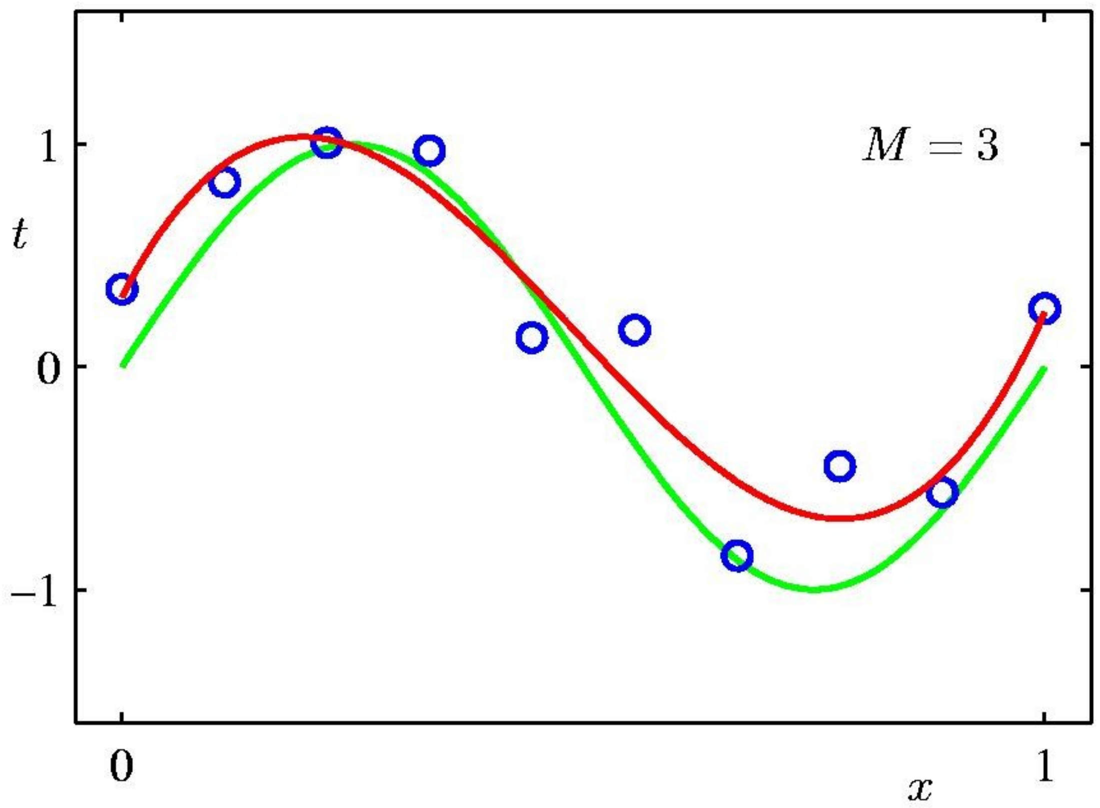
0^{th} Order Polynomial



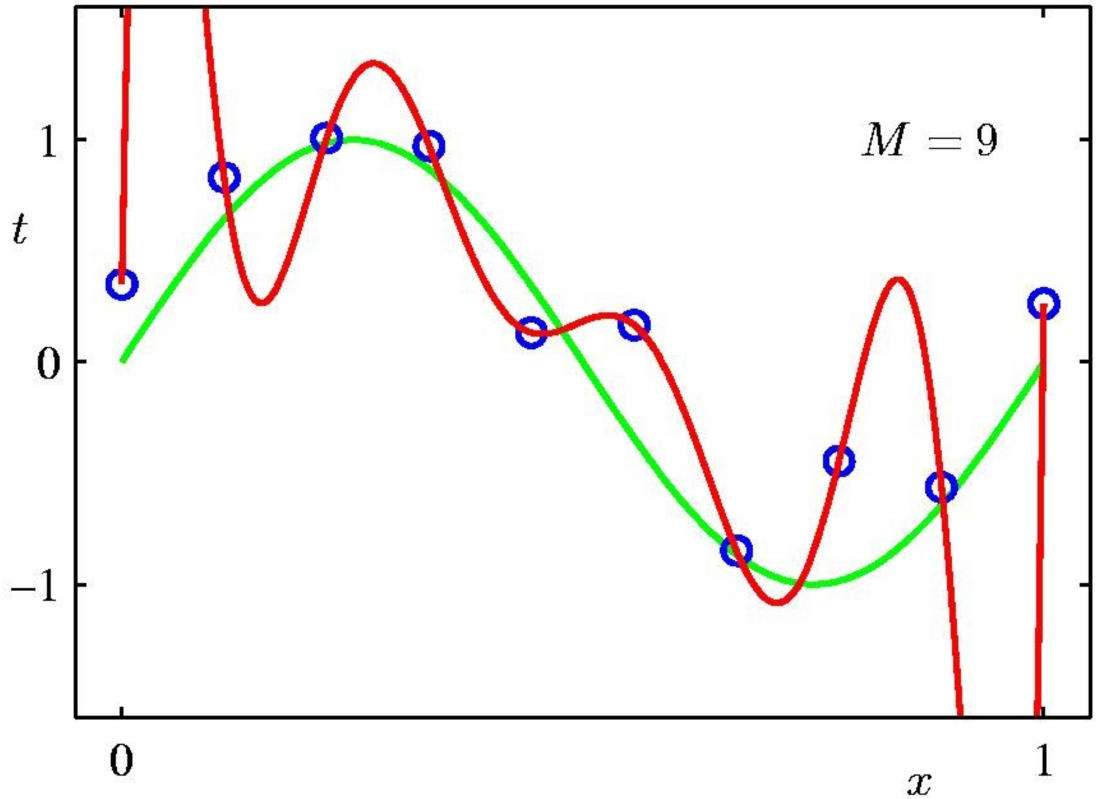
1st Order Polynomial



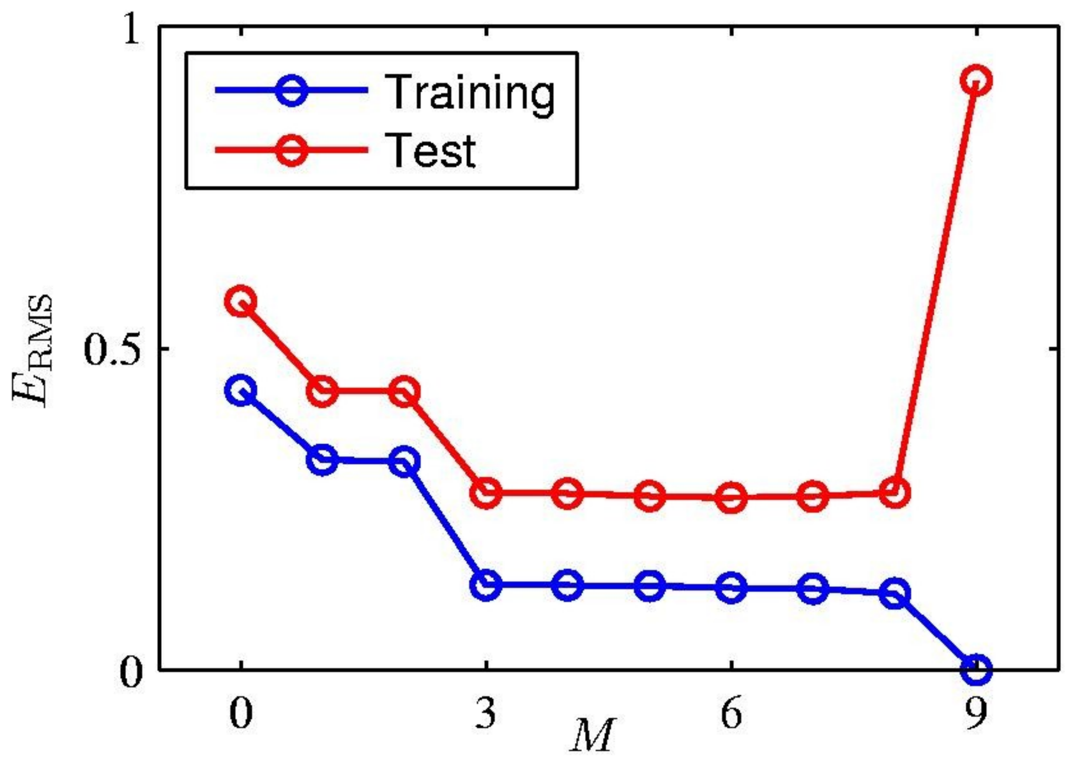
3rd Order Polynomial



9^{th} Order Polynomial



Over-fitting

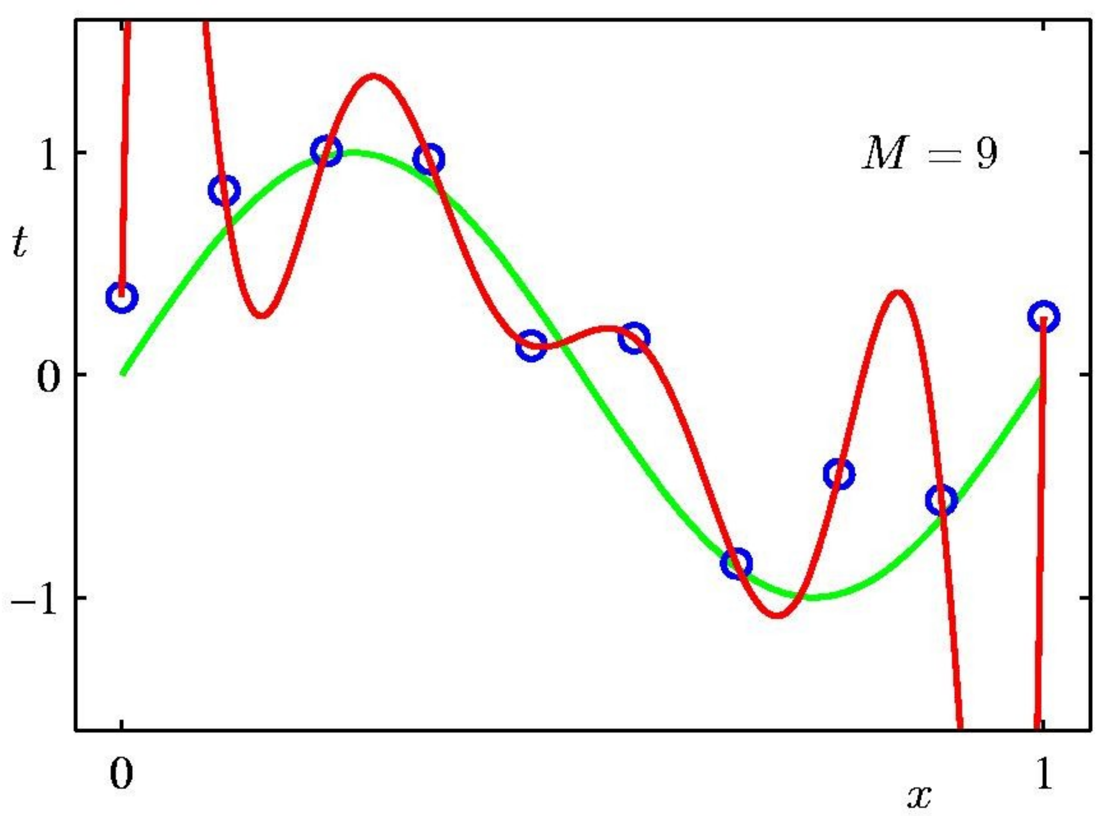


Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$

Polynomial Coefficients

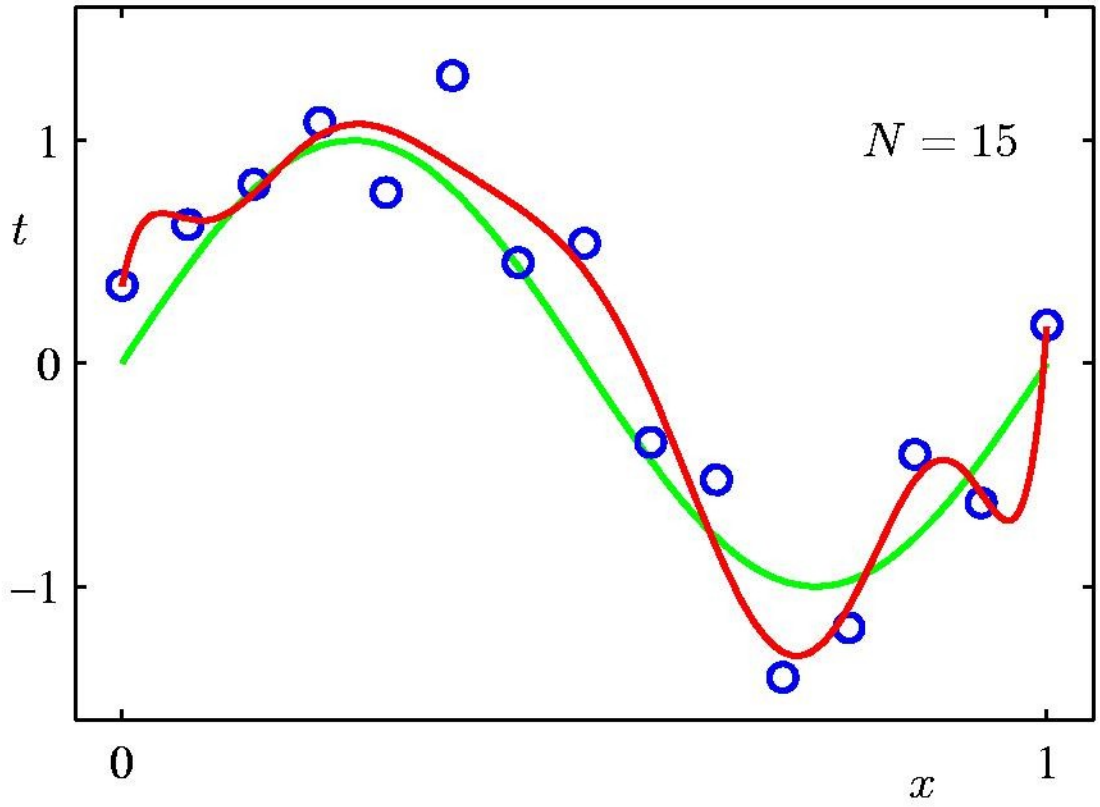
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

9th Order Polynomial



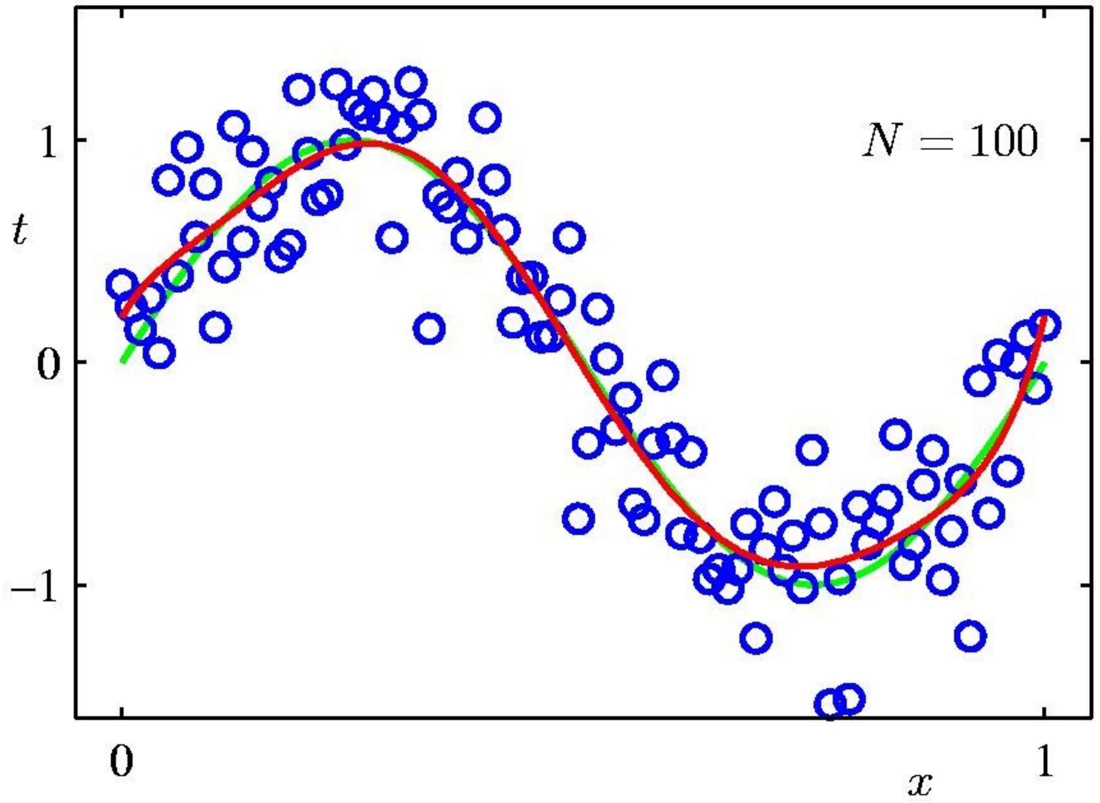
Data Set Size: $N = 15$

9^{th} Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial

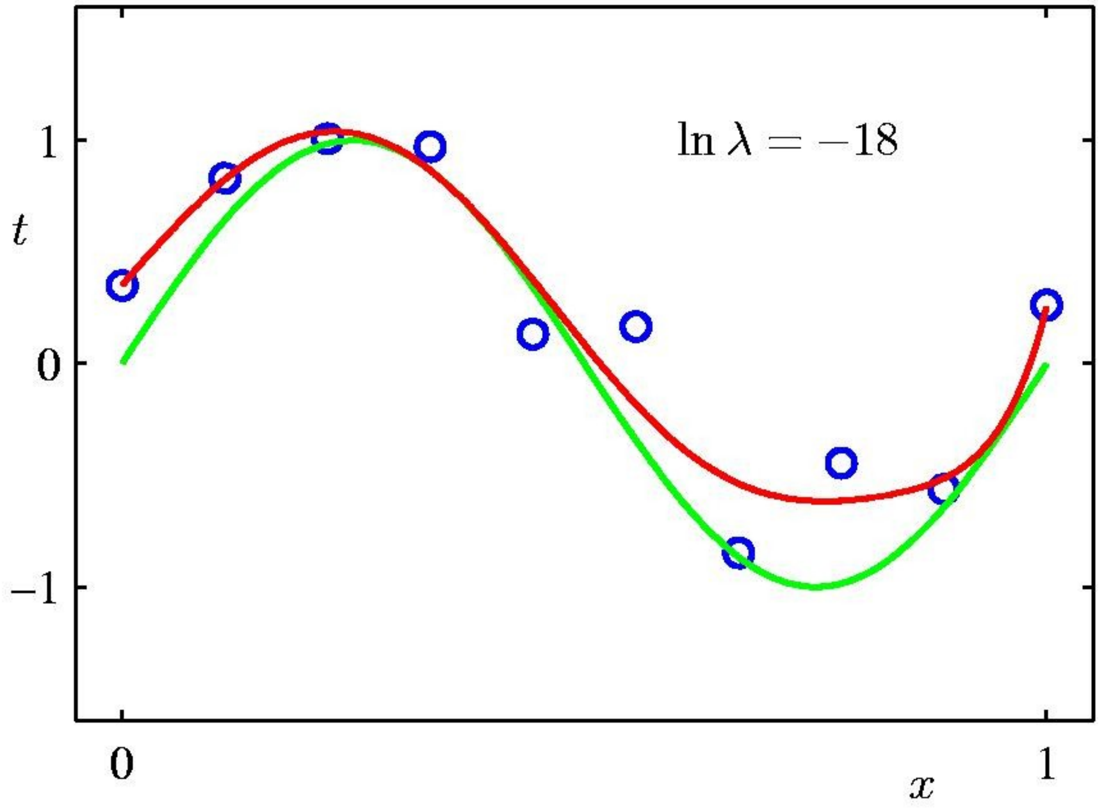


Regularization

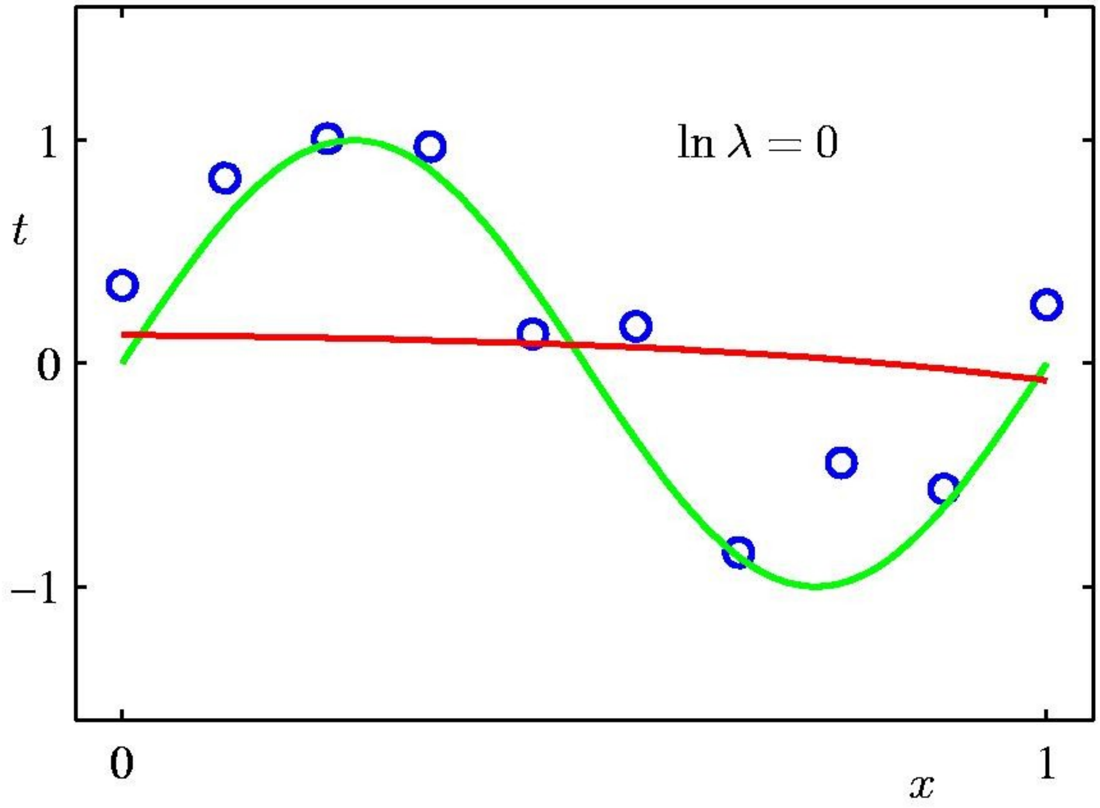
Penalize large coefficient values

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

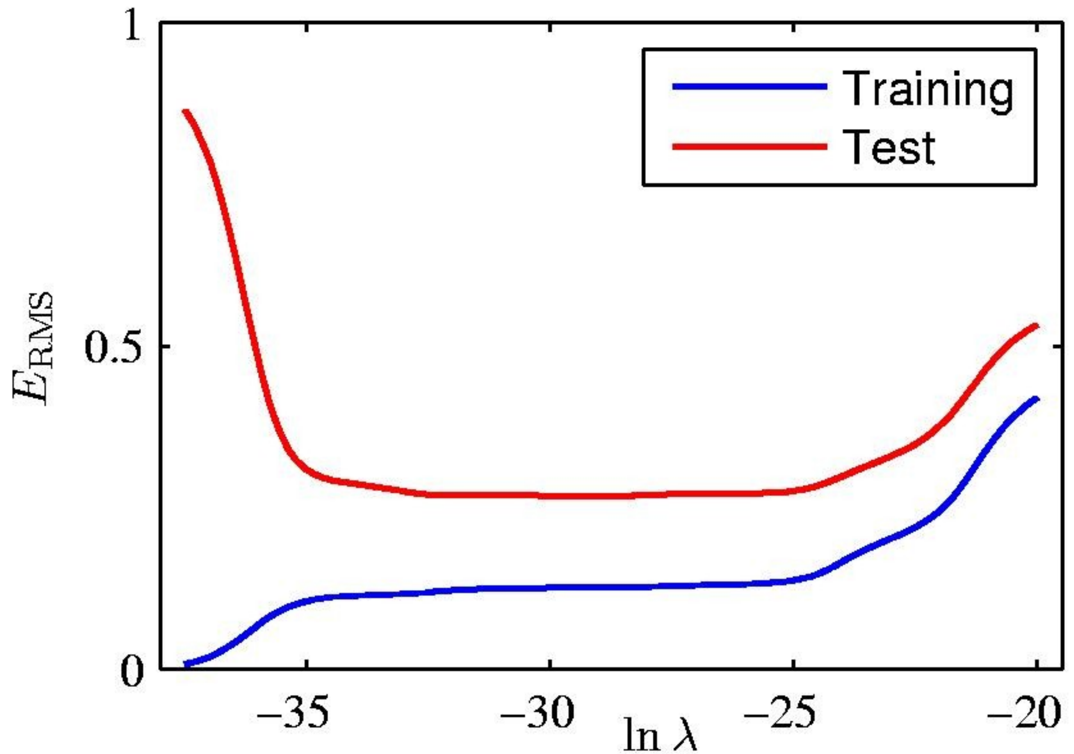
Regularization: $\ln \lambda = -18$



Regularization: $\ln \lambda = 0$



Regularization: E_{RMS} vs. $\ln \lambda$



Polynomial Coefficients

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

Information Theory

Twenty Questions

Knower: thinks of object (point in a probability space)
Guesser: asks knower to evaluate random variables

Stupid approach:

Guesser: Is it my left big toe?
Knower: No.

Guesser: Is it Valmiki?
Knower: No.

Guesser: Is it Aunt Lakshmi?
...

Expectations & Surprisal

Turn the key: expectation: lock will open

Exam paper showing: could be 100, could be zero.
random variable: function from set of marks to real interval [0,1]

Interestingness $\propto$ unpredictability

$$\textit{surprisal}\ (r.v. = x) = -\log_2 p(x)$$

$= 0$ when $p(x) = 1$
$= 1$ when $p(x) = ½$
$= \infty$ when $p(x) = 0$

Expectations in data

A: 00010001000100010001. . . 0001000100010001000100010001

B: 01110100110100100110. . . 1010111010111011000101100010

C: 00011000001010100000. . . 0010001000010000001000110000

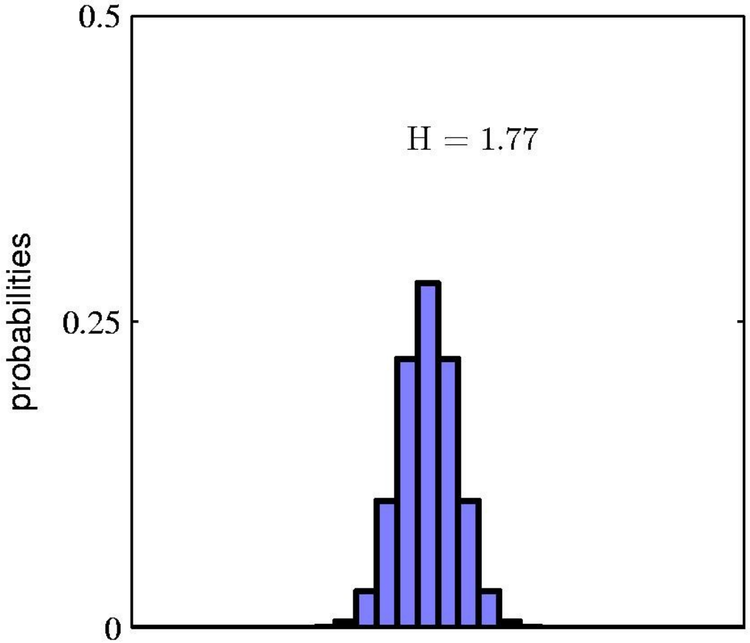
Structure in data → easy to remember

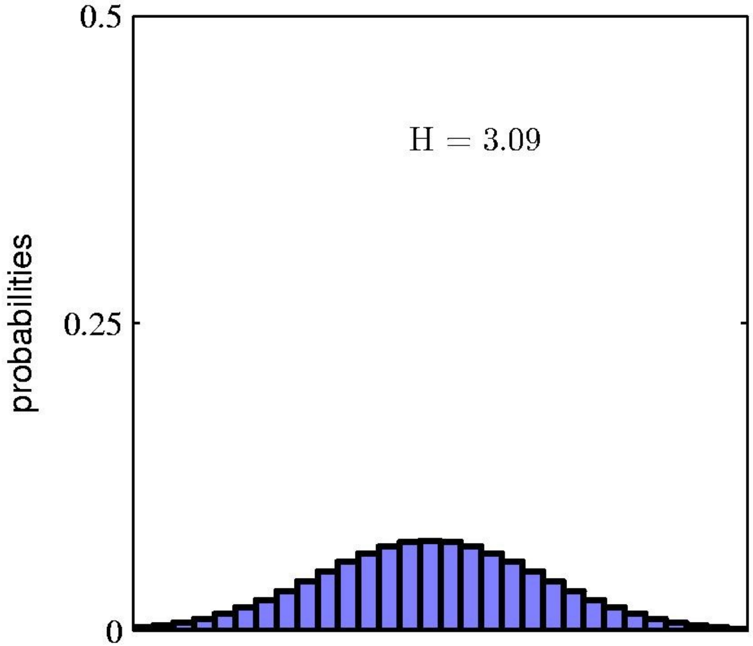
Entropy

$$\mathrm{H}[x] = -\sum_{x} p(x) \log_2 p(x)$$

Used in
- coding theory
- statistical physics
- machine learning

Entropy





Entropy

In how many ways can N identical objects be allocated M bins?

$$W = \frac{N!}{\prod_i n_i!}$$

$$\mathrm{H} = \frac{1}{N} \ln W \simeq -\lim_{N \to \infty} \sum_i \left(\frac{n_i}{N}\right) \ln \left(\frac{n_i}{N}\right) = -\sum_i p_i \ln p_i$$

Entropy maximized when $\forall i : p_i = \frac{1}{M}$

Entropy in Coding theory

x discrete with 8 possible states; how many bits to transmit the state of x?

All states equally likely

$$\mathrm{H}[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits.}$$

Coding theory

x	a	b	c	d	e	f	g	h
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$
code	0	10	110	1110	111100	111101	111110	111111

$$\begin{aligned}
\mathrm{H}[x] &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} \\
&= 2 \text{ bits}
\end{aligned}$$

$$\begin{aligned}
\text{average code length} &= \frac{1}{2}\times 1 + \frac{1}{4}\times 2 + \frac{1}{8}\times 3 + \frac{1}{16}\times 4 + 4\times\frac{1}{64}\times 6 \\
&= 2 \text{ bits}
\end{aligned}$$

Entropy in Twenty Questions

Intuitively : try to ask q whose answer is 50-50

Is the first letter between A and M?

question entropy = p(Y)logp(Y) + p(N)logP(N)

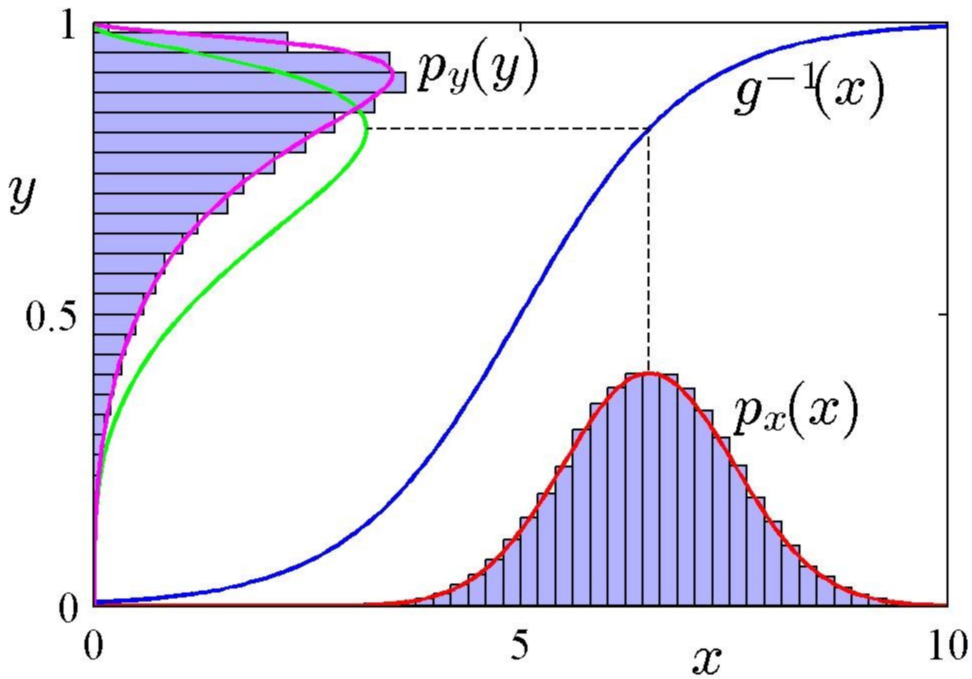
For both answers equiprobable:

$$\text{entropy} = - ½ * \log_2(½) - ½ * \log_2(½) = 1.0$$

For P(Y)=1/1028

entropy = - 1/1028 * -10 - eps = 0.01

Change of variable x=g(y)



$$p_y(y) = p_x(x)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| = p_x(g(y))\,|g'(y)|$$