UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

POLYNOMIAL AND NON LINEAR REGRESSION

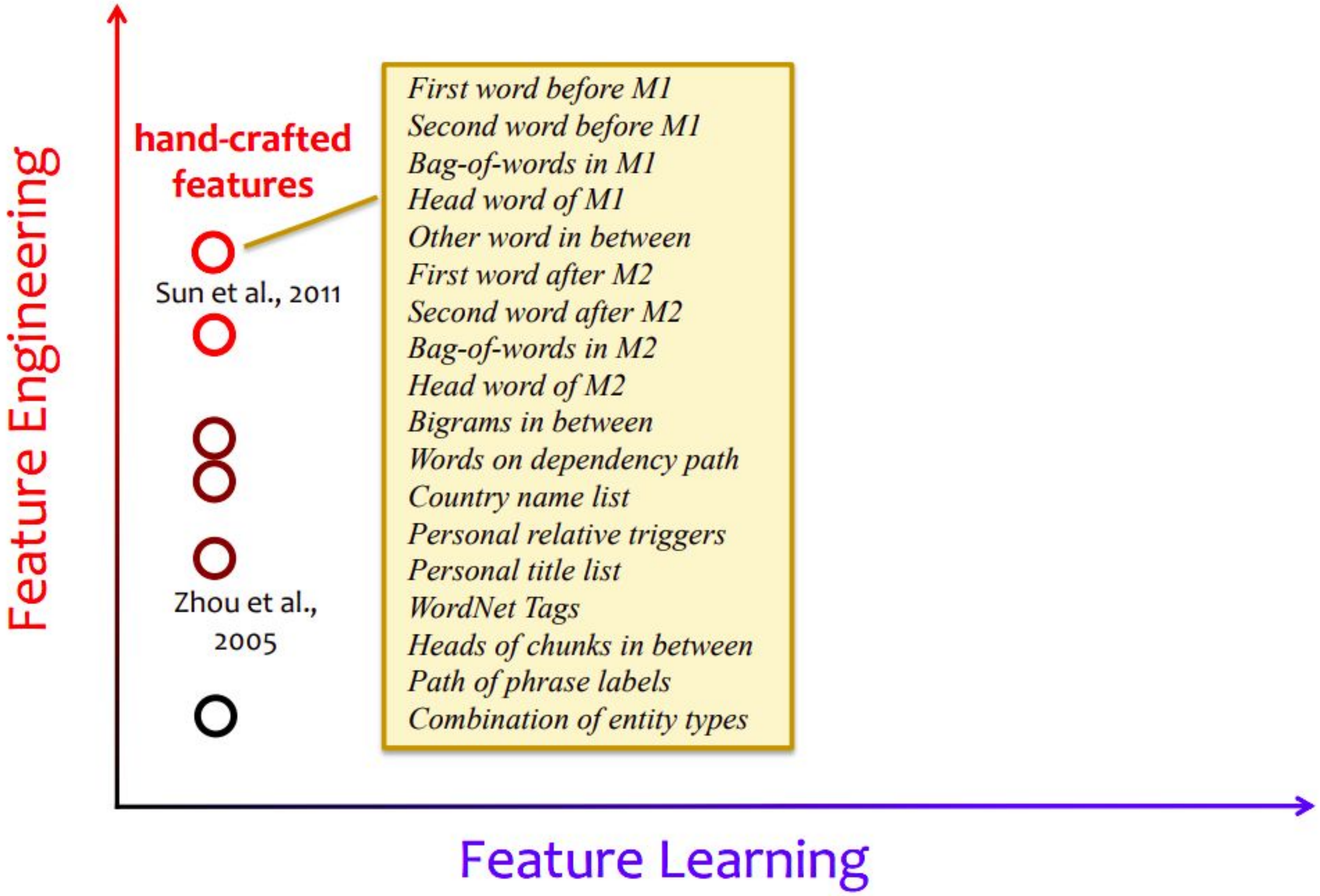
Ariana Villegas

TRANSFORMATEC

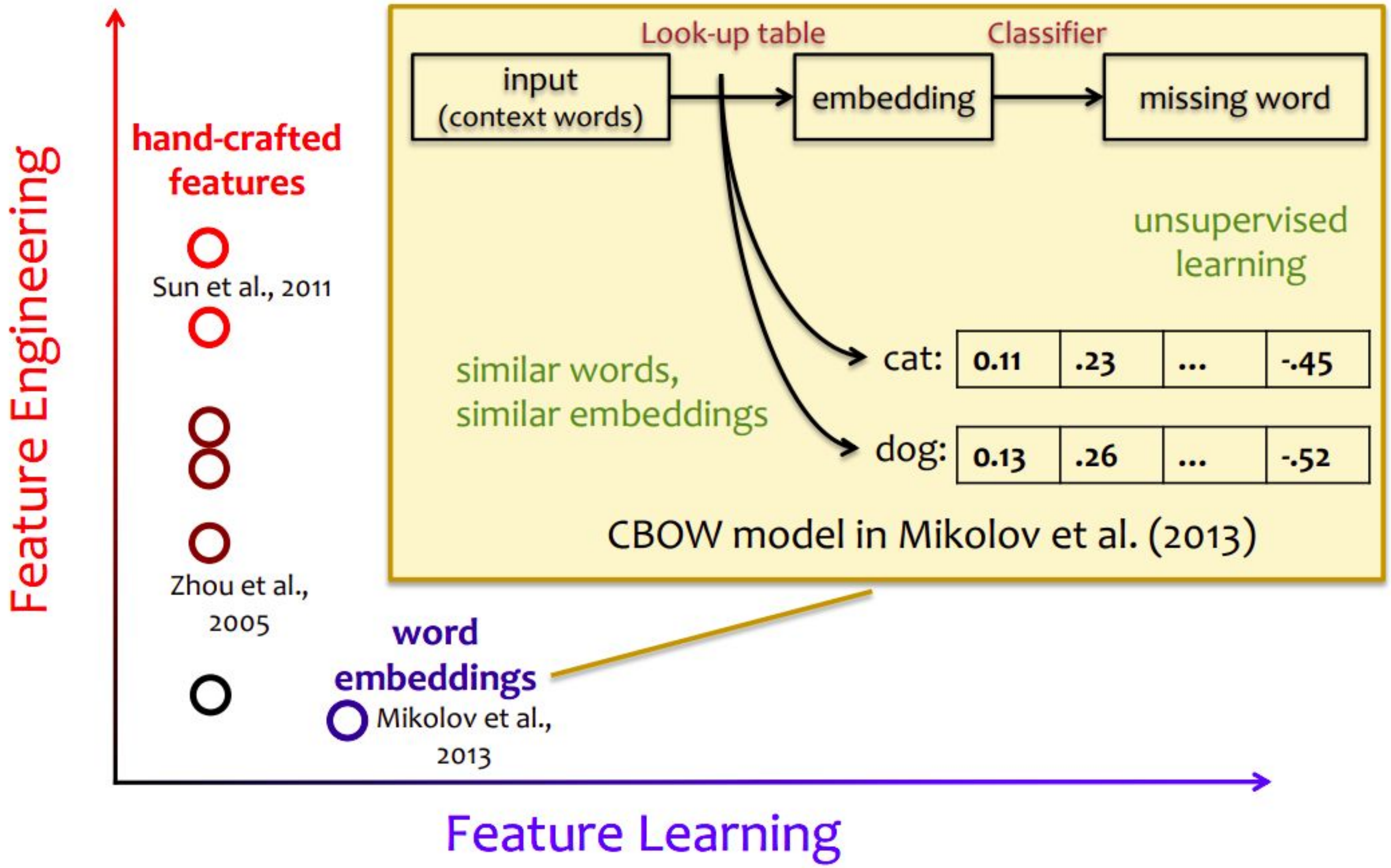
1.

Feature Engineering
NLP & CV

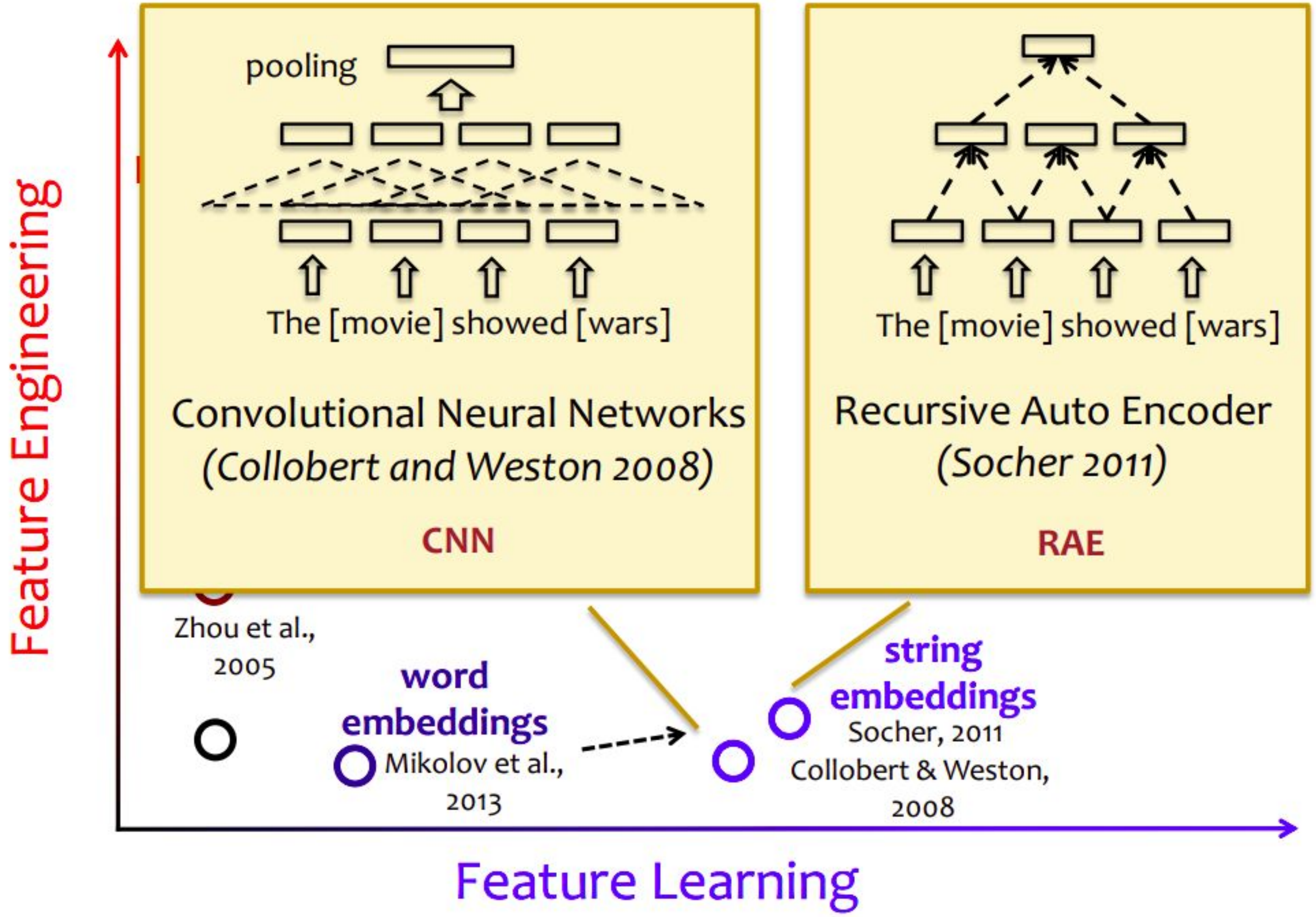
Where do features come from?



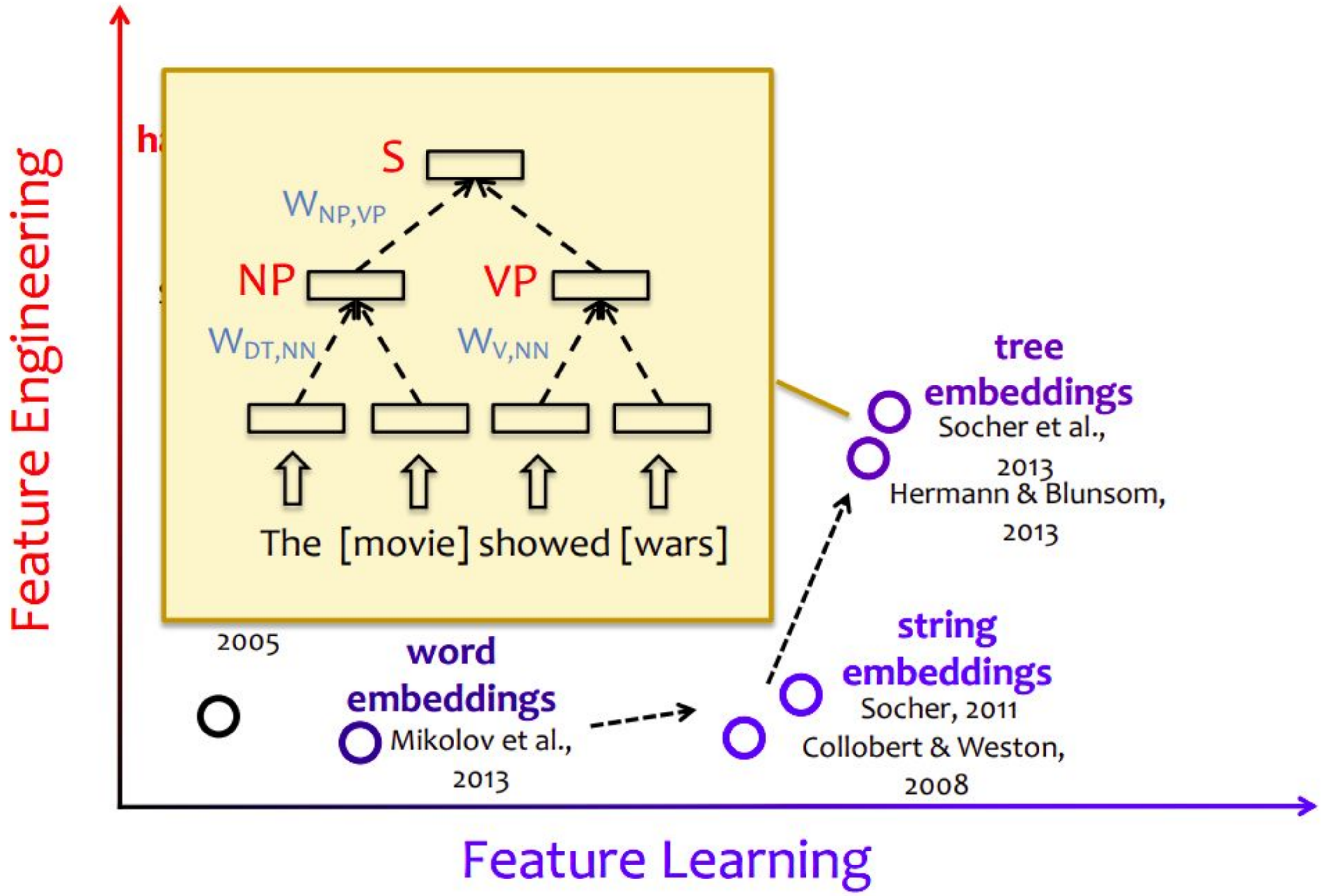
Where do features come from?



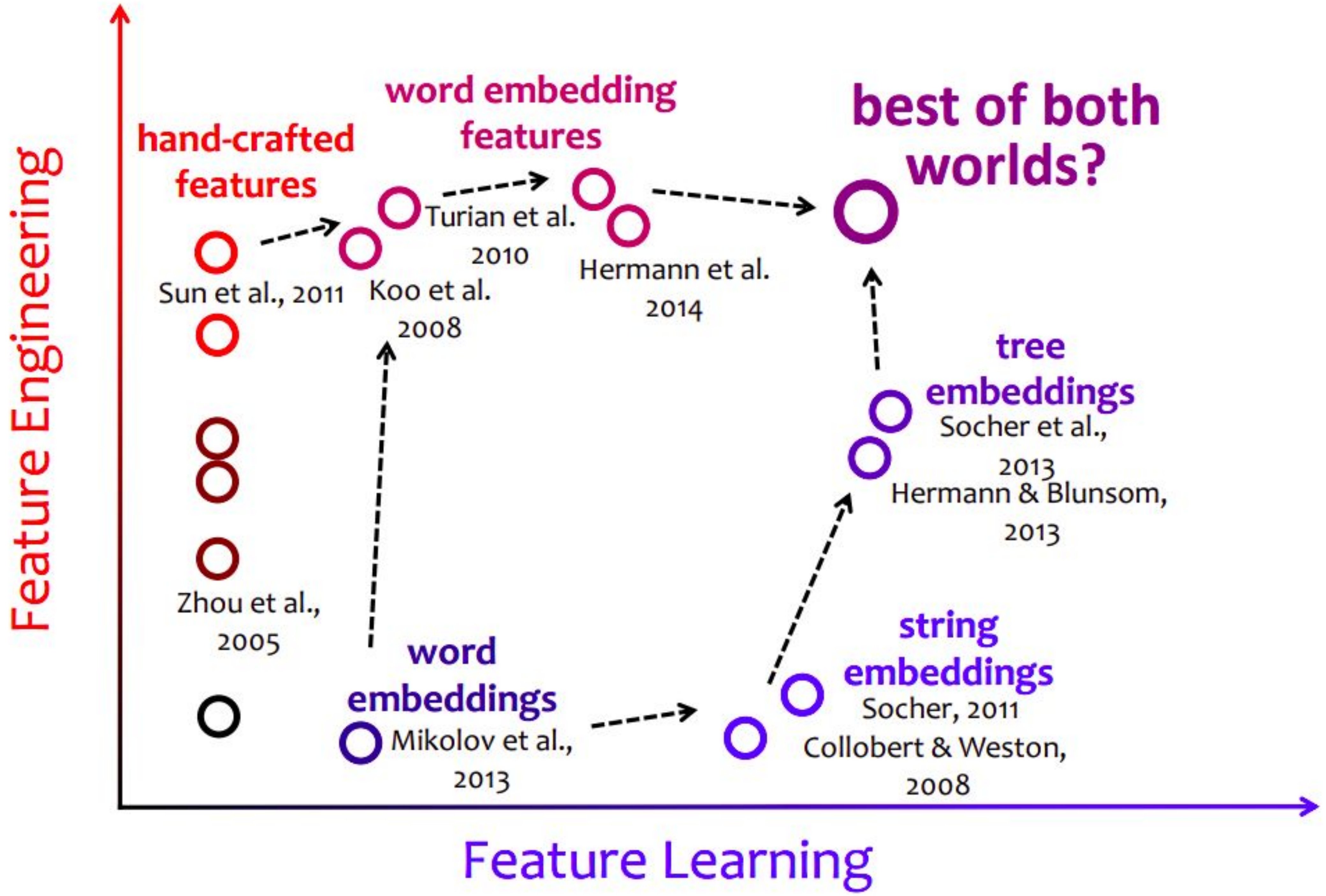
Where do features come from?



Where do features come from?



Where do features come from?



What features should you use for text?

What features should you use for text?

Per-word Features:

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(6)}$
is-capital(w_i)	1	0	1	0	0	0
endswith(w_i,"e")	1	1	0	0	0	1
endswith(w_i,"d")	0	0	0	1	1	0
endswith(w_i,"ed")	0	0	0	1	1	0
w_i == "aardvark"	0	0	0	0	0	0
w_i == "hope"	0	0	0	0	0	1
...	...	...	...	...	...	...

deter.	noun	noun	verb	verb	noun
The	*movie*	*I*	*watched*	*depicted*	*hope*

What features should you use for text?

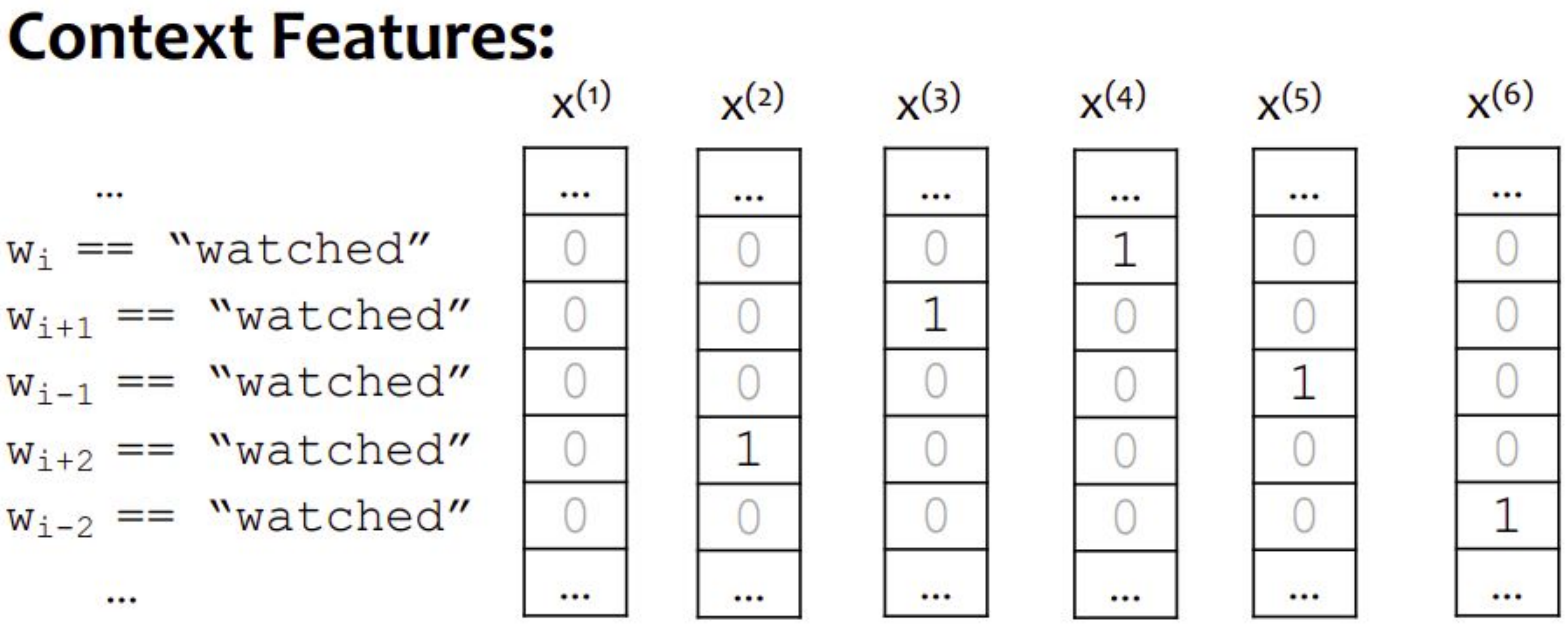


Table 3. Tagging accuracies with different feature templates and other changes on the *WSJ* 19-21 development set.

Model	Feature Templates	# Feats	Sent. Acc.	Token Acc.	Unk. Acc.
3GRAMMEMM	See text	248,798	52.07%	96.92%	88.99%
NAACL 2003	See text and [1]	460,552	55.31%	97.15%	88.61%
Replication	See text and [1]	460,551	55.62%	97.18%	88.92%
Replication′	+rareFeatureThresh = 5	482,364	55.67%	97.19%	88.96%
5W	$+\langle t_0, w_{-2}\rangle, \langle t_0, w_2\rangle$	730,178	56.23%	97.20%	89.03%
5WSHAPES	$+\langle t_0, s_{-1}\rangle, \langle t_0, s_0\rangle, \langle t_0, s_{+1}\rangle$	731,661	56.52%	97.25%	89.81%
5WSHAPESDS	+ distributional similarity	737,955	56.79%	97.28%	90.46%

What features should you use for text?

Table 3. Tagging accuracies with different feature templates and other changes on the *WSJ* 19-21 development set.

Model	Feature Templates	# Feats	Sent. Acc.	Token Acc.	Unk. Acc.
3GRAMMEMM	See text	248,798	52.07%	96.92%	88.99%
NAACL 2003	See text and [1]	460,552	55.31%	97.15%	88.61%
Replication	See text and [1]	460,551	55.62%	97.18%	88.92%
Replication′	+rareFeatureThresh = 5	482,364	55.67%	97.19%	88.96%
5W	$+\langle t_0, w_{-2}\rangle, \langle t_0, w_2\rangle$	730,178	56.23%	97.20%	89.03%
5WSHAPES	$+\langle t_0, s_{-1}\rangle, \langle t_0, s_0\rangle, \langle t_0, s_{+1}\rangle$	731,661	56.52%	97.25%	89.81%
5WSHAPESDS	+ distributional similarity	737,955	56.79%	97.28%	90.46%

Word Embeddings

One-hot vectors

- Standard representation of a word in NLP: 1-hot vector (aka. a string)
- Vectors representing related words share nothing in common

	a	and	be	cat	dog		you	zebra
cat:	0	0	0	1	0	...	0	0
dog:	0	0	0	0	1	...	0	0

Word embeddings

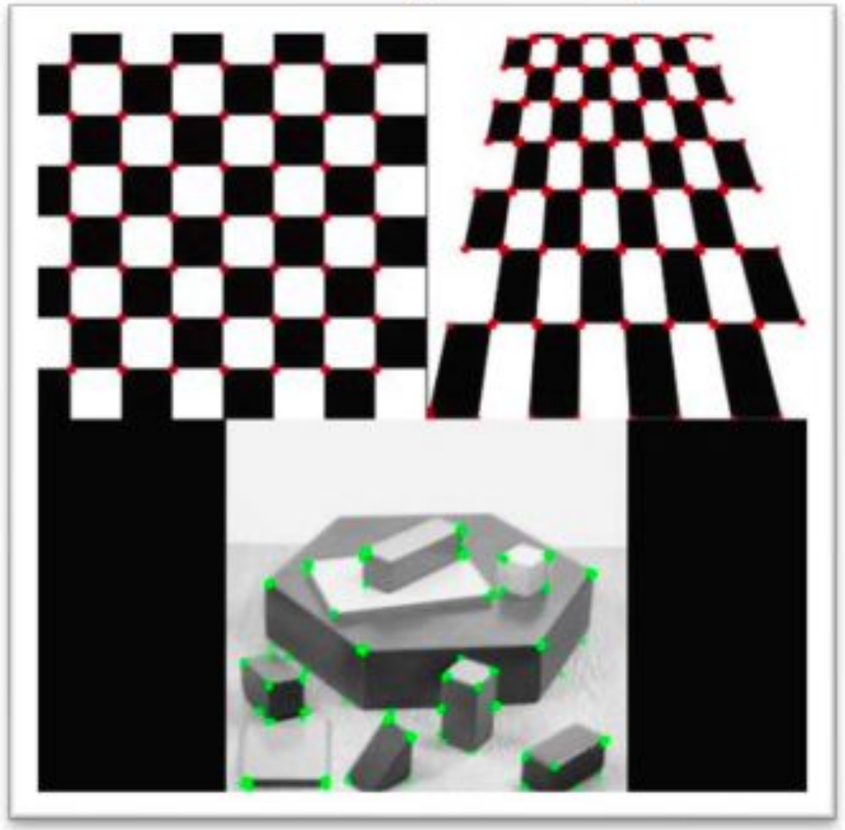
- Word embedding: real-valued vector representation of a word in *M* dimensions
- Related words have similar vectors
- Long history in NLP: Term-doc frequency matrices, Reduce dimensionality with {LSA, NNMF, CCA, PCA}, Brown clusters, Vector space models, Random projections, Neural networks / deep learning

cat:	0.13	.26	...	-.52
dog:	0.11	.23	...	-.45

What features should you use for images?

Edge detection (Canny)

Corner Detection (Harris)



UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

2.

Non-linear Regression
Definition & Examples

TRANSFORMATEC

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

What if our linear model is not good? How can we create a more complicated model?

TRANSFORMATEC

Fitting a Polynomial

We can create a more complicated model by defining input variables that are **combinations of components of x**

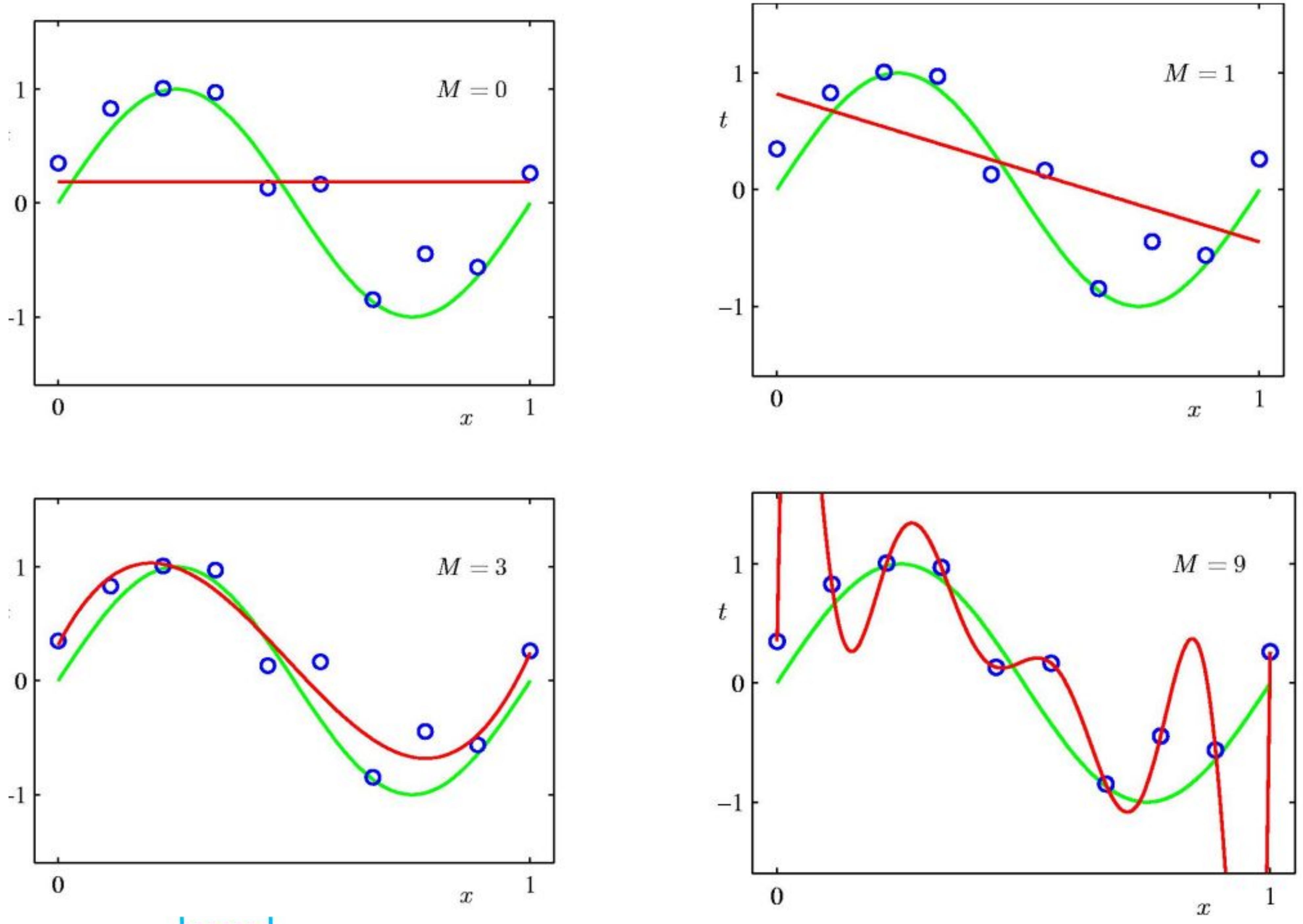
Example: an **M-th order polynomial** function of **one dimensional** feature x:

$$y(x, \mathbf{w}) = w_0 + \sum_{j=1}^{M} w_j x^j$$

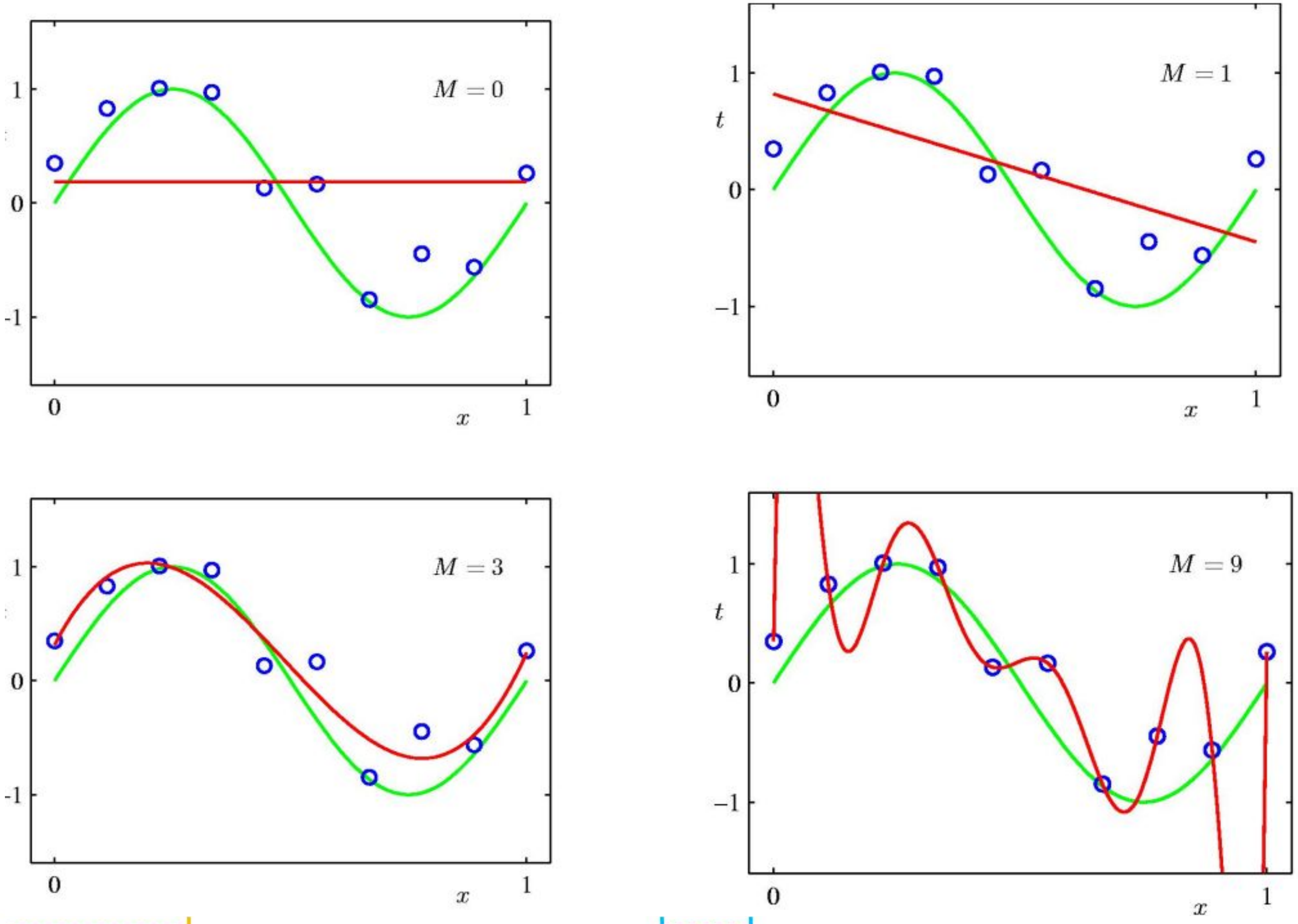
where $\mathbf{x}^j$ is the j-th power of x

We can use the same approach to optimize for the **weights w**

Fitting a Polynomial



Fitting a Polynomial

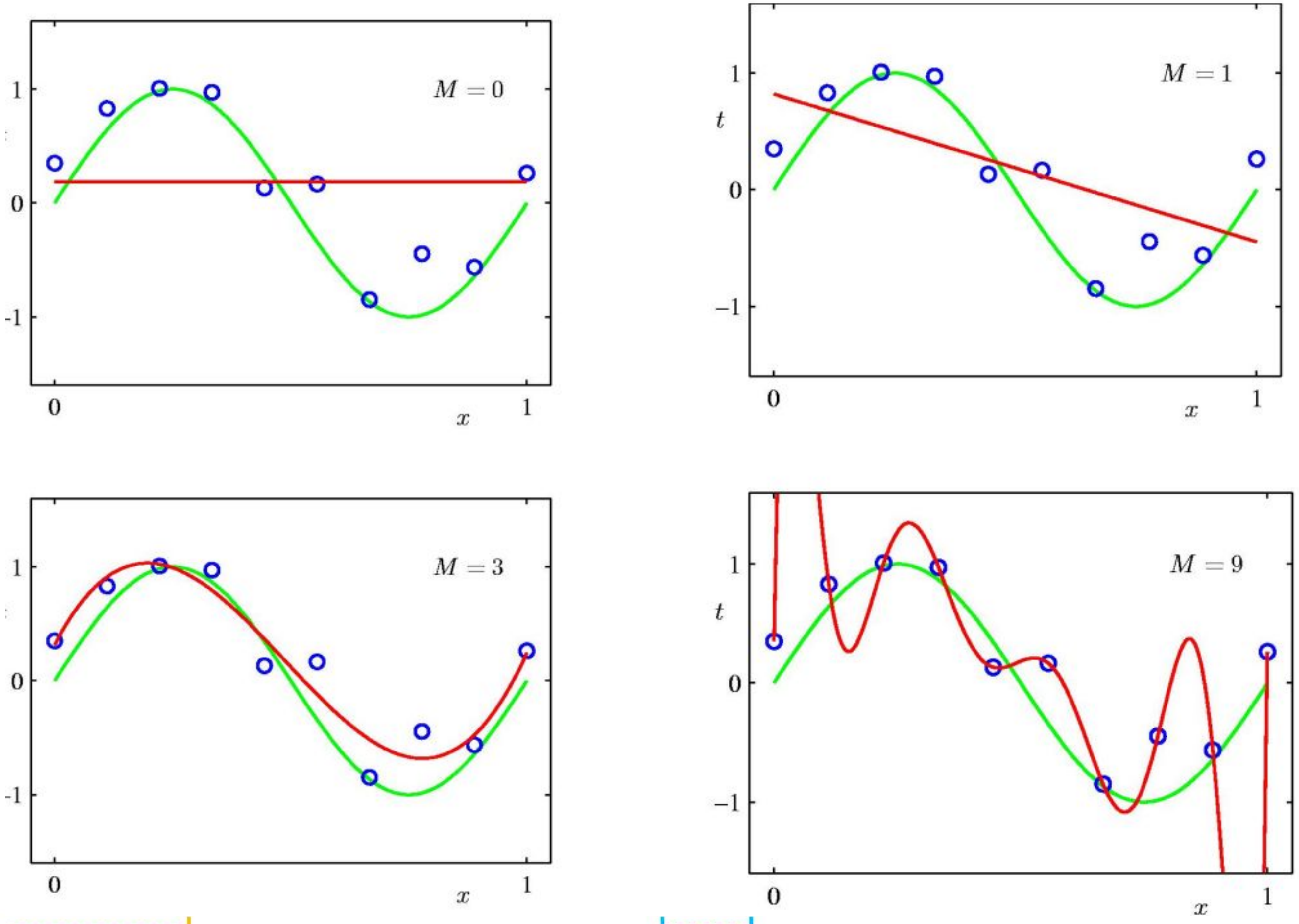


Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x))^2$$

Fitting a Polynomial



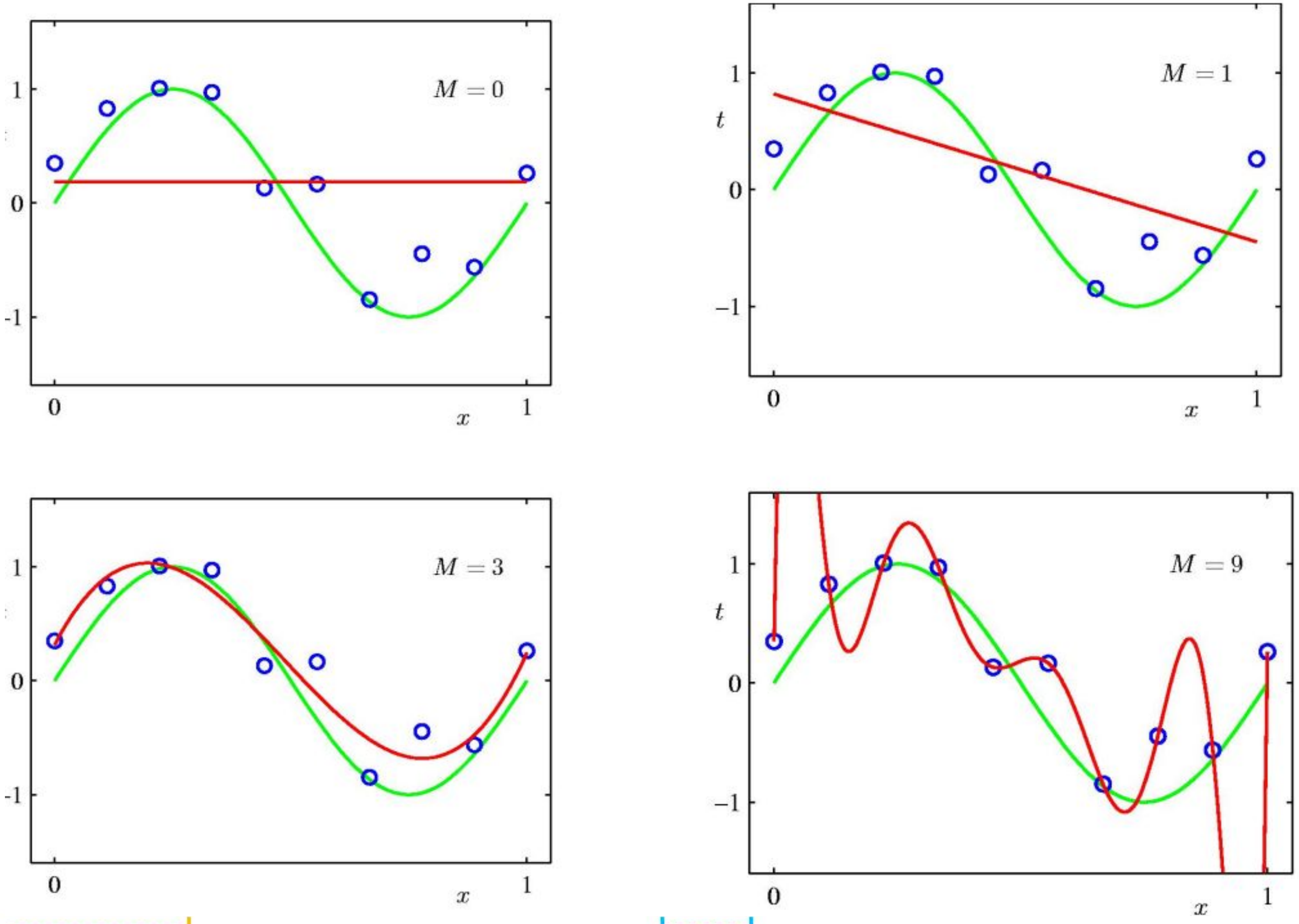
Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x))^2$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (wx+b))^2$$

Fitting a Polynomial



Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x))^2$$

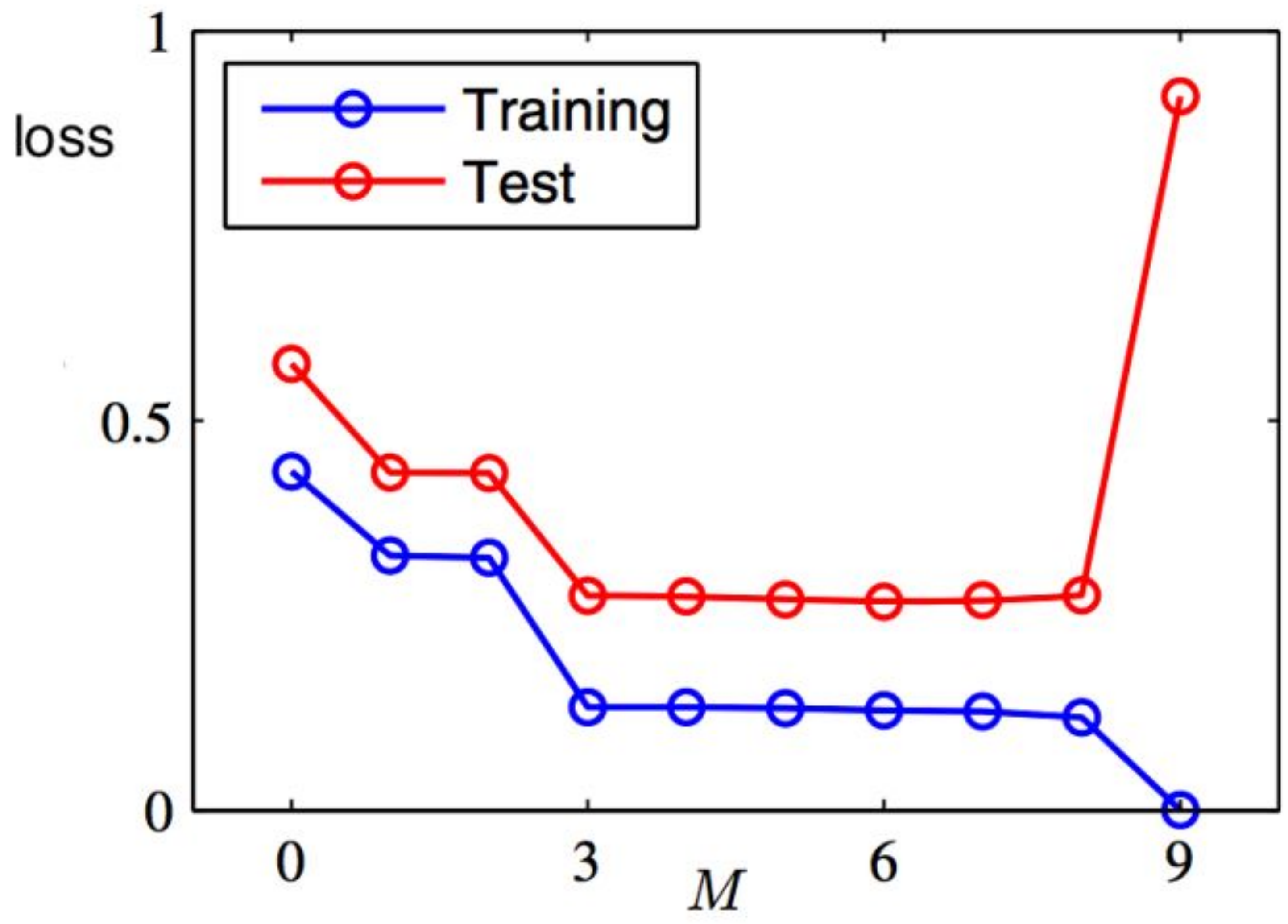
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (wx + b))^2$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0x^2 + w_1x + b))^2$$

Generalization

Generalization = model's **ability** to predict the **held out data**

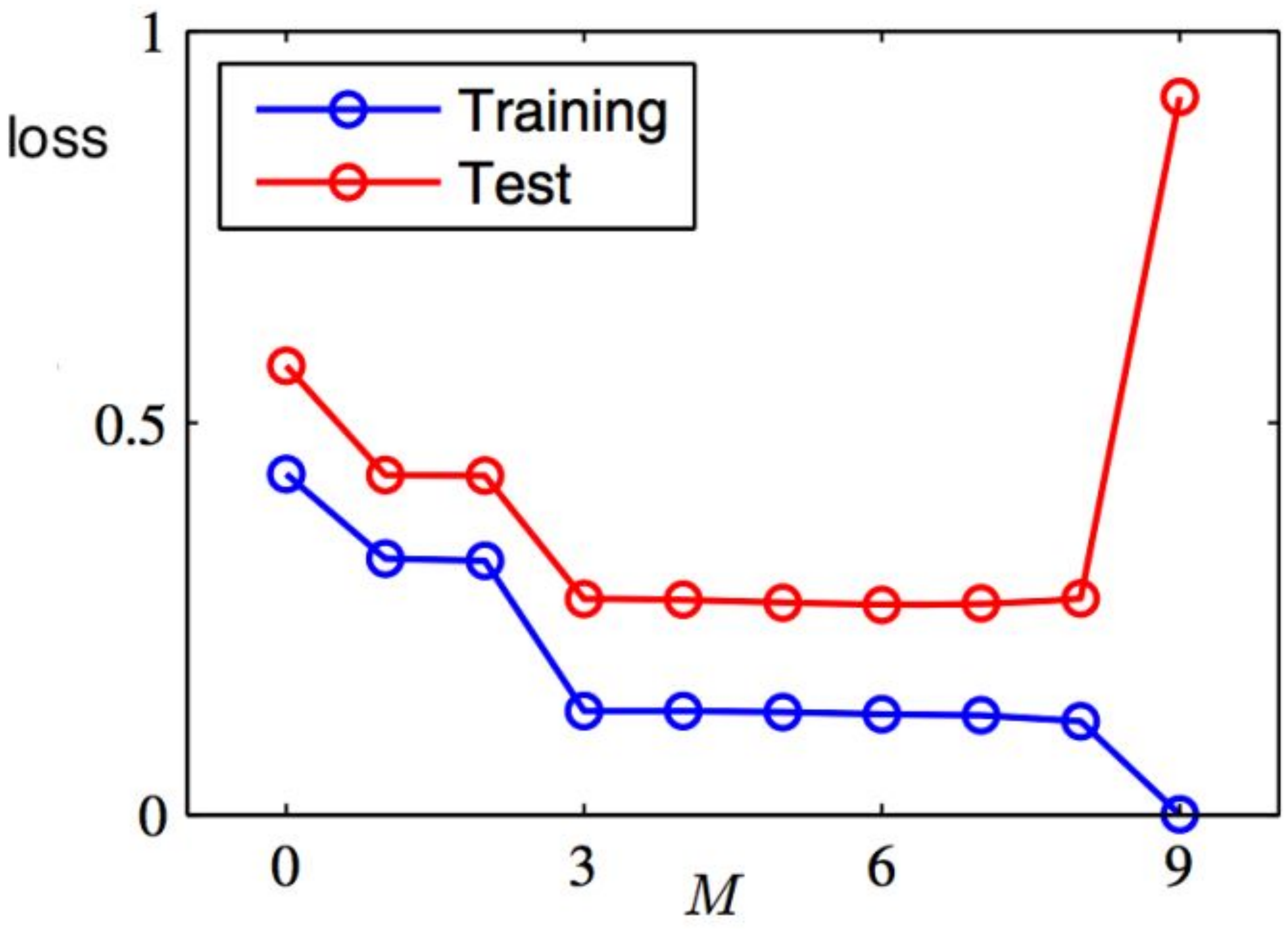
What is happening?



Generalization

Generalization = model's **ability** to predict the **held out data**

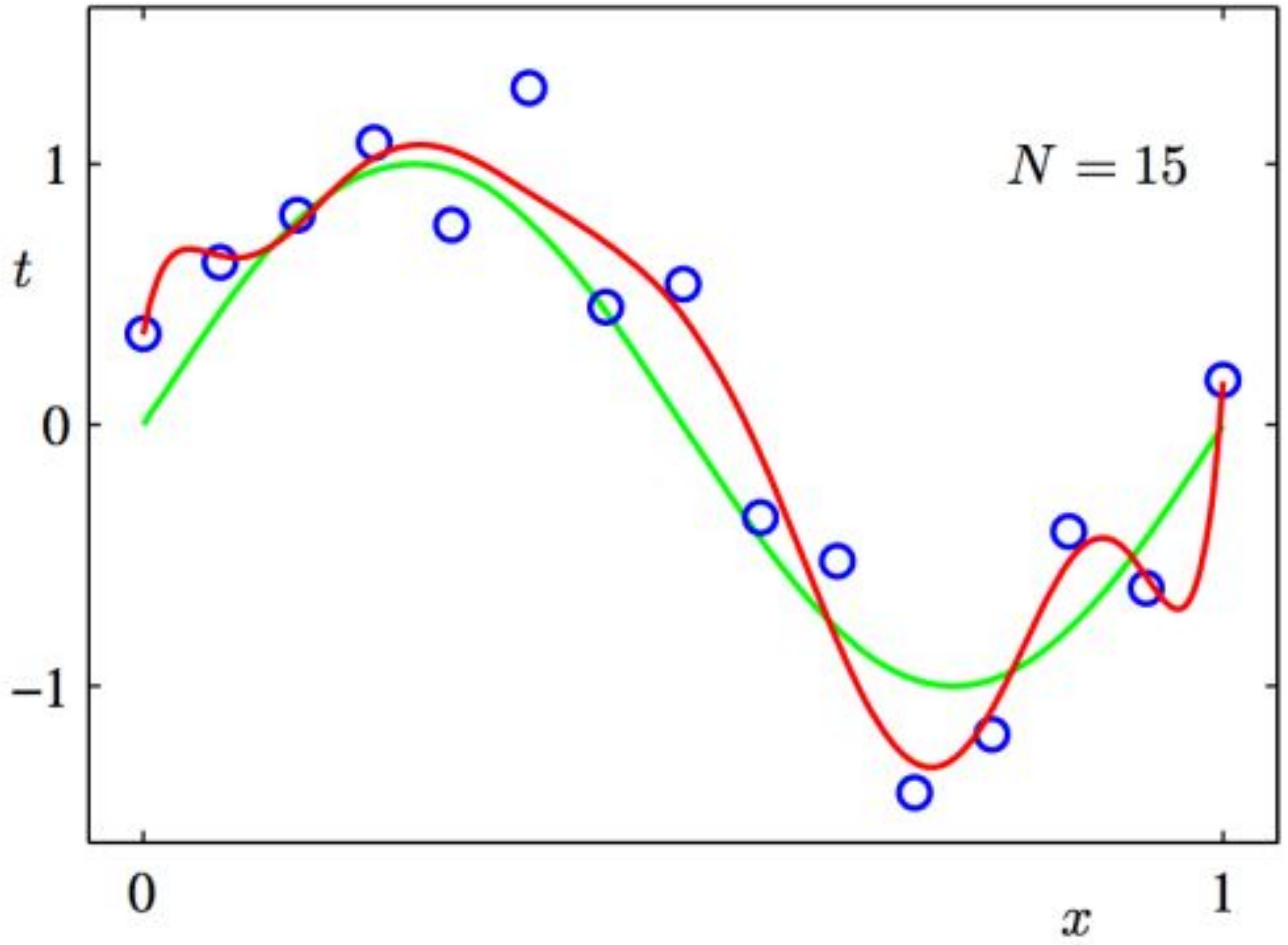
What is happening?

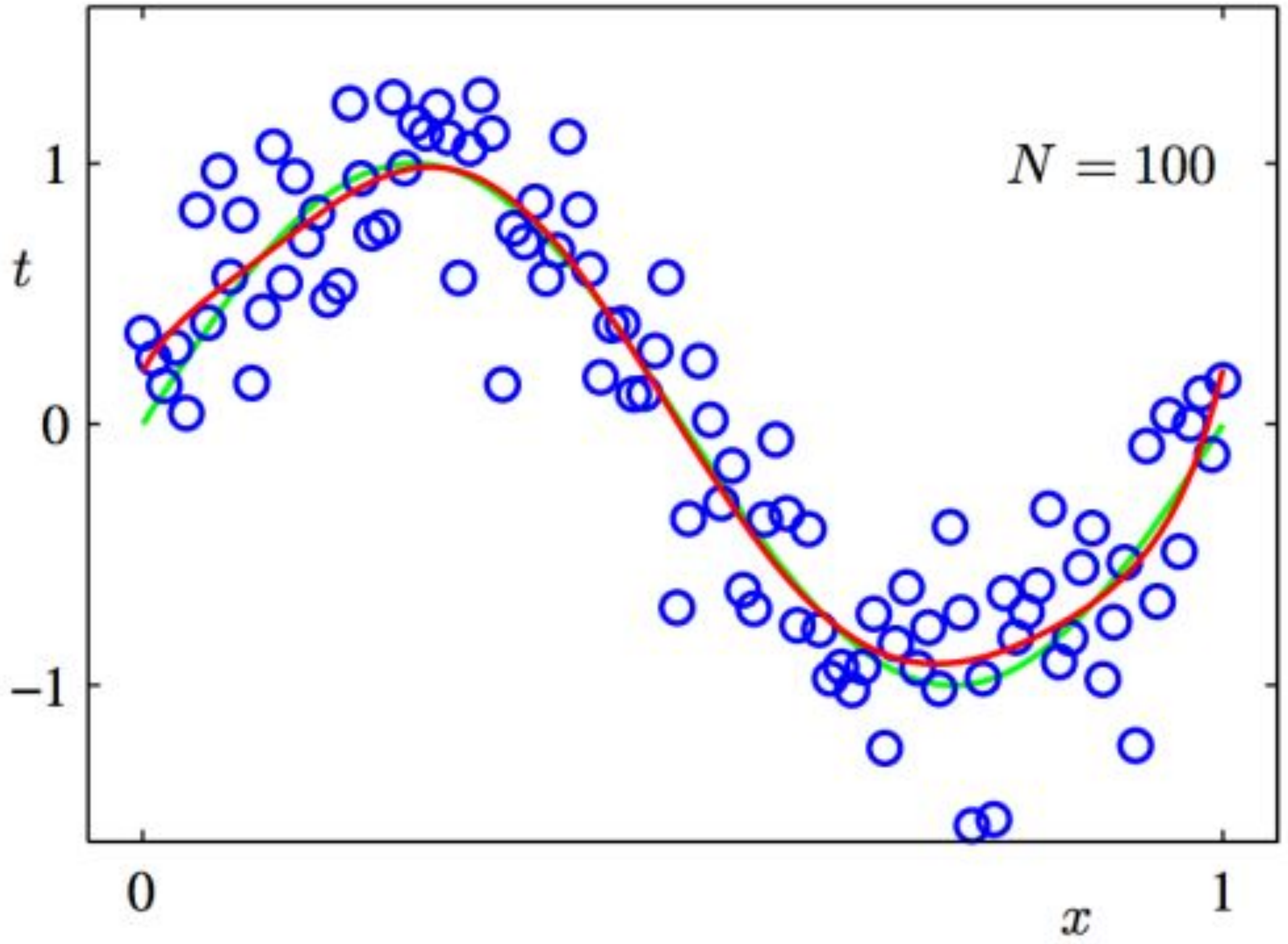


Our model with M = 9 **overfits** the data (it models also noise)

Generalization

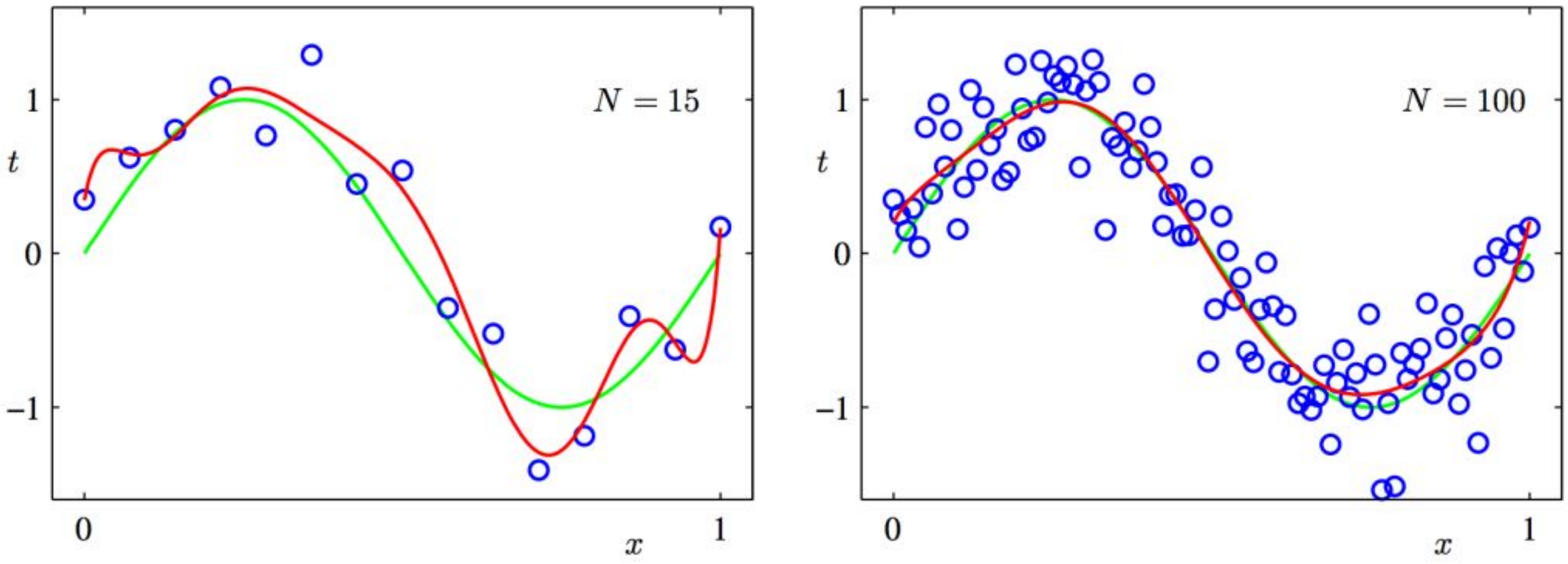
What can we do?





Generalization

What can we do? Not a problem if we have lots of training examples



Generalization

Let's look at the **estimated weights** for **various M** in the case of fewer examples

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	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

Generalization

Let's look at the **estimated weights** for **various M** in the case of fewer examples

The weights are becoming huge to compensate for the noise

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^\star$	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

How can we deal with this?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

3.

Regularization

L0 & L1 & L2

TRANSFORMATEC

Regularization

Occam's Razor: prefer the simplest hypothesis

What does it mean for a hypothesis (or model) to be simple?

1. small number of features **(model selection)**

2. small number of "important" features **(shrinkage)**

Regularization

Given objective function: J(θ)

Goal is to find

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

Key idea: Define regularizer r(θ) s.t. we tradeoff between fitting the data and keeping the model simple

Choose form of regularizer

Common choice: p-norm

$$r(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_p = \left[\sum_{m=1}^{M} |\theta_m|^p\right]^{\left(\frac{1}{p}\right)}$$

Regularization

Given objective function: J(θ)

Goal is to find

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}) \qquad r(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_p = \left[\sum_{m=1}^{M} |\theta_m|^p\right]^{\left(\frac{1}{p}\right)}$$

p	$r(\boldsymbol{\theta})$	**yields parameters that are...**	**name**	**optimization notes**
0	$\lVert\boldsymbol{\theta}\rVert_0 = \sum \mathbb{1}(\theta_m \neq 0)$	zero values	L0 reg.	no good computational solutions
1	$\lVert\boldsymbol{\theta}\rVert_1 = \sum \lvert\theta_m\rvert$	zero values	L1 reg.	subdifferentiable
2	$(\lVert\boldsymbol{\theta}\rVert_2)^2 = \sum \theta_m^2$	small values	L2 reg.	differentiable

L2+L1 Regularization

L2 and L1 regularization can be **combined**

$$R(\mathbf{w}) = \lambda_2 ||\mathbf{w}||^2 + \lambda_1 ||\mathbf{w}||_1$$

- Also called **ElasticNet**
- Can work better than either type alone
- Can **adjust hyperparameters** to control which of the two penalties is more important

4.

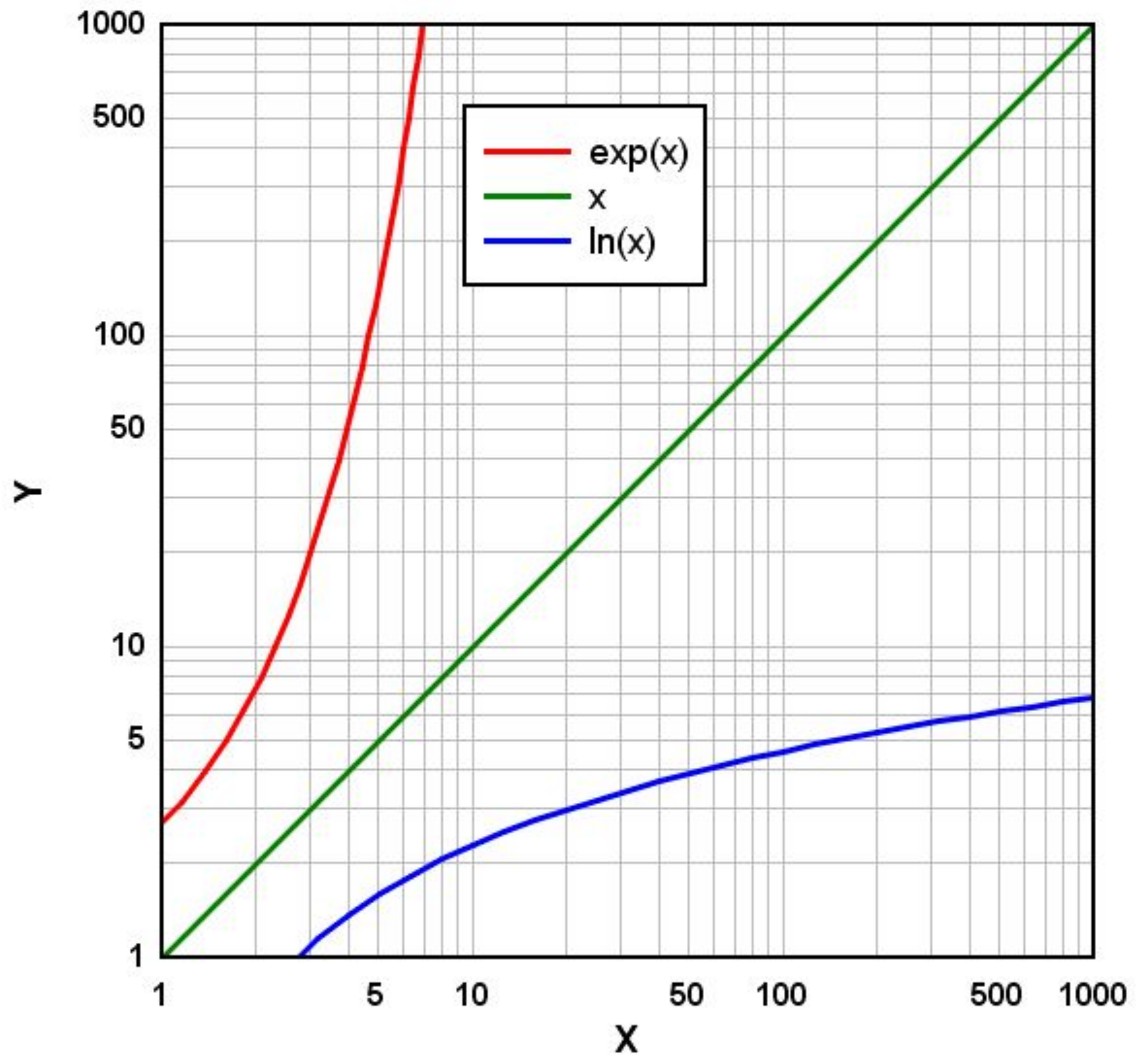
More non linear

Non polynomial

TRANSFORMATEC

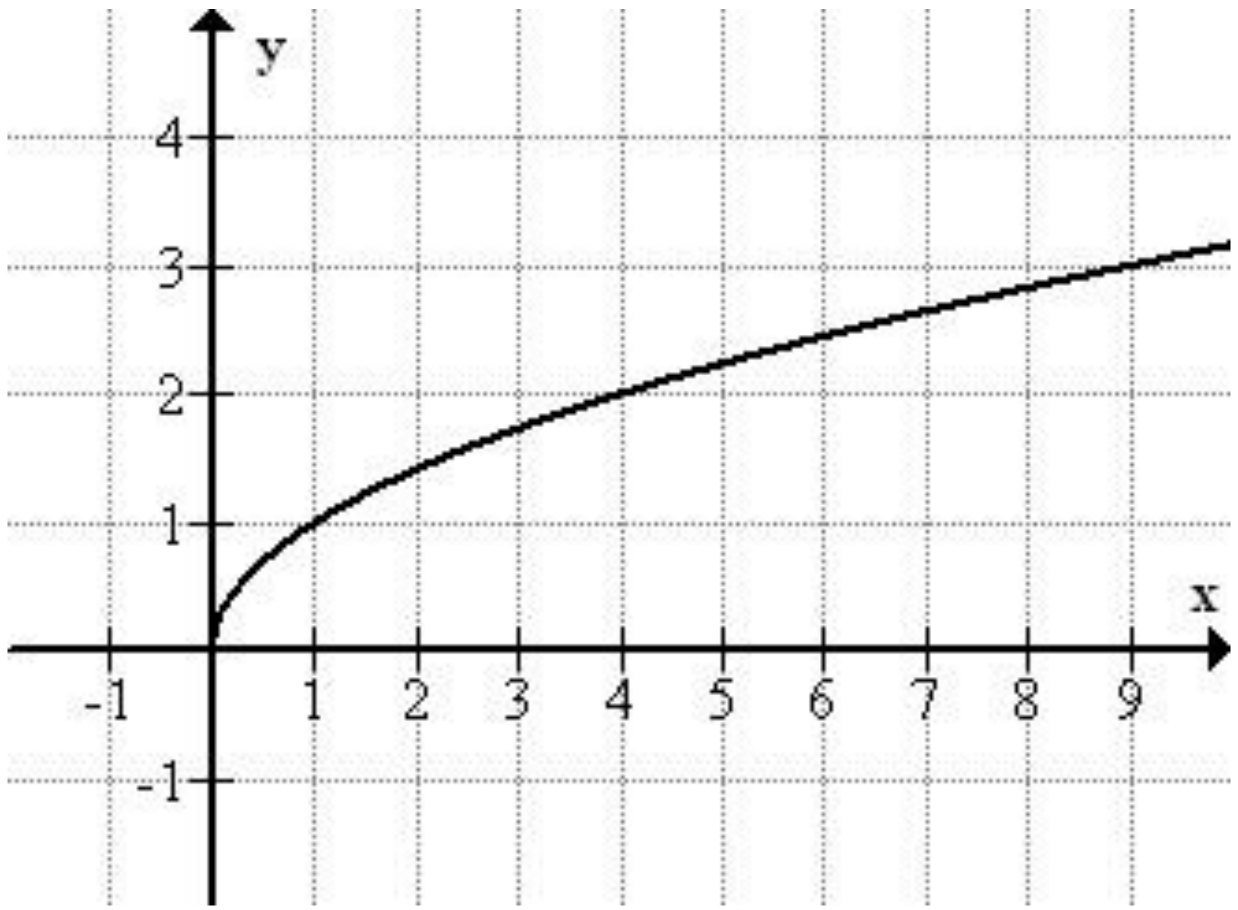
More non linear functions...

Logarithmic and exponential function



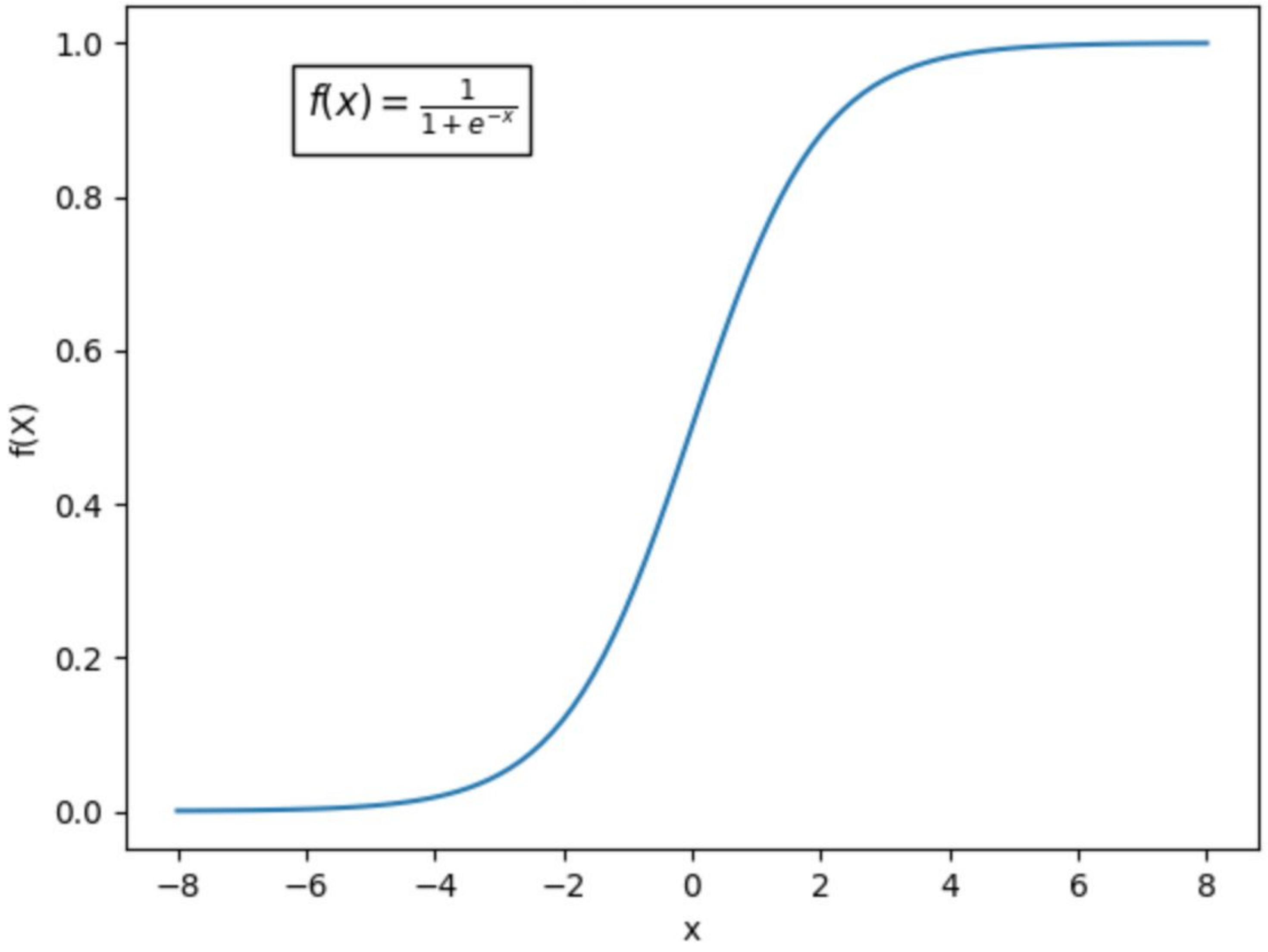
More non linear functions...

Square root function



More non linear functions...

Sigmoid function



What about more dimensions?

$$f(x) = w_0 + x_1 w_1 + x_2 w_2$$

What about more dimensions?

$$f(x) = w_0 + x_1w_1 + x_2w_2$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_2w_3$$

What about more dimensions?

$$f(x) = w_0 + x_1w_1 + x_2w_2$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_2w_3$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_1^3w_3 + x_2^2w_4 + x_2w_5$$

What about more dimensions?

$$f(x) = w_0 + x_1w_1 + x_2w_2$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_2w_3$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_1^3w_3 + x_2^2w_4 + x_2w_5$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_1^3w_3 + x_2^2w_4 + x_2w_5 + x_1x_2w_6$$

What about more dimensions?

$$f(x) = w_0 + x_1 w_1 + x_2 w_2$$

$$f(x) = w_0 + x_1 w_1 + x_1^2 w_2 + x_2 w_3$$

$$f(x) = w_0 + x_1 w_1 + x_1^2 w_2 + x_1^3 w_3 + x_2^2 w_4 + x_2 w_5$$

$$f(x) = w_0 + x_1 w_1 + x_1^2 w_2 + x_1^3 w_3 + x_2^2 w_4 + x_2 w_5 + x_1 x_2 w_6$$

$$f(x) = w_0 + \exp(x_1) w_1 + x_2 w_2$$

What about more dimensions?

$$f(x) = w_0 + x_1w_1 + x_2w_2$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_2w_3$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_1^3w_3 + x_2^2w_4 + x_2w_5$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_1^3w_3 + x_2^2w_4 + x_2w_5 + x_1x_2w_6$$

$$f(x) = w_0 + \exp(x_1)w_1 + x_2w_2$$

$$f(x) = w_0 + \exp(x_1)w_1 + x_2^2w_2 + x_2w_3$$

What about more dimensions?

$$f(x) = w_0 + x_1w_1 + x_2w_2$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_2w_3$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_1^3w_3 + x_2^2w_4 + x_2w_5$$

$$f(x) = w_0 + x_1w_1 + x_1^2w_2 + x_1^3w_3 + x_2^2w_4 + x_2w_5 + x_1x_2w_6$$

$$f(x) = w_0 + \exp(x_1)w_1 + x_2w_2$$

$$f(x) = w_0 + \exp(x_1)w_1 + x_2^2w_2 + x_2w_3$$

$$\vdots$$

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

5.

Non Linear Regression
Example

TRANSFORMATEC

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

**https://colab.research.google.com
/drive/1-ZJqtjlM_PwAdN2vMmYzD
RDv-DH9ILdj?usp=sharing**

TRANSFORMATEC

More Exercises?

MACHINE LEARNING WORKOUT: Chapter 5 and 6

Kahoot Time!


> Reinventa el mundo <
TRANSFORMATEC