Lecture 2.2: Linear Regression
CSC 84020 - Machine Learning

Andrew Rosenberg

February 5, 2009

Today

- Linear Regression

Linear Regression

Linear Regression is a **Regression** algorithm, a **supervised** technique.

In one dimension:

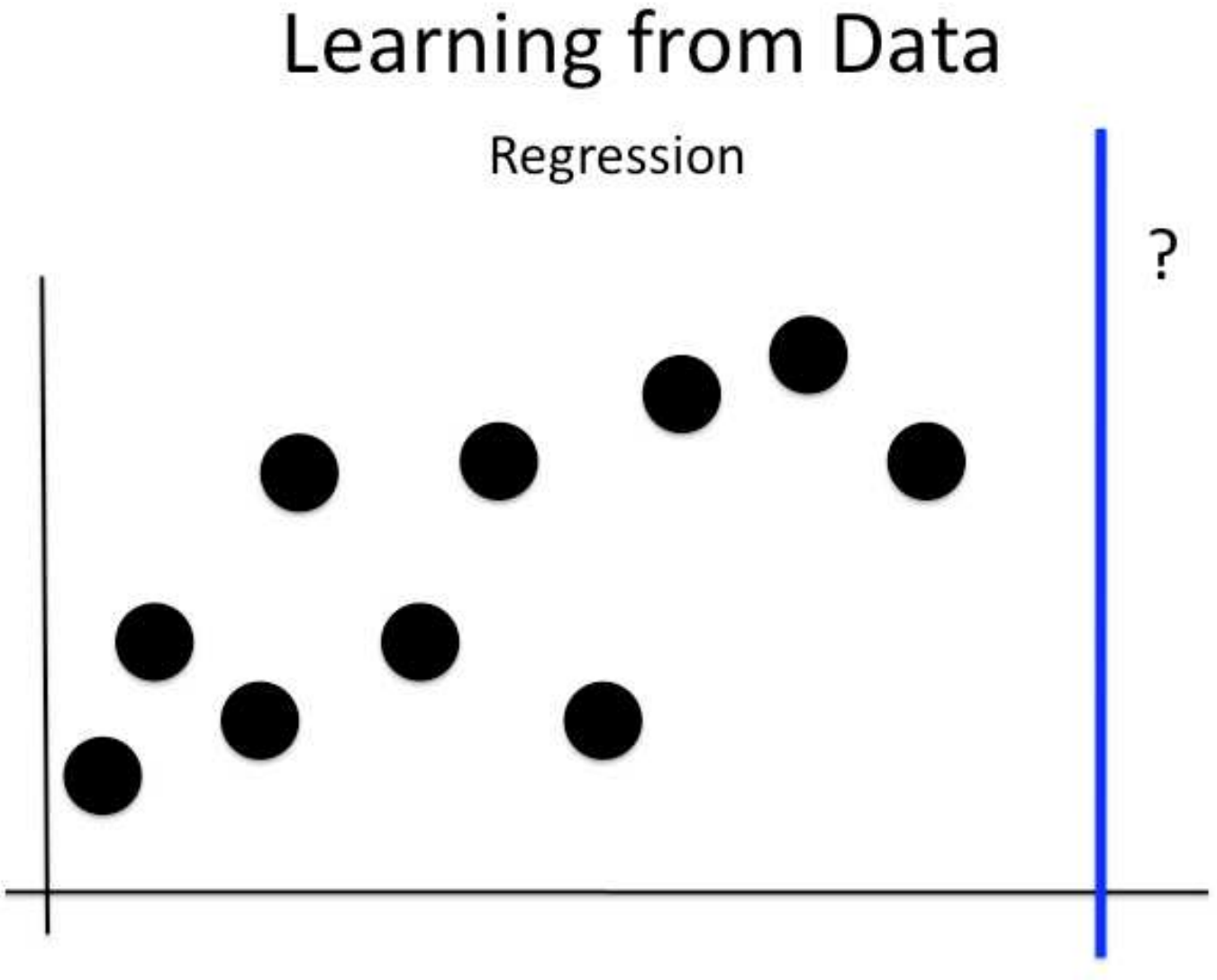
Goal: identify $y : \mathbb{R} \rightarrow \mathbb{R}$.

In D-dimensions:

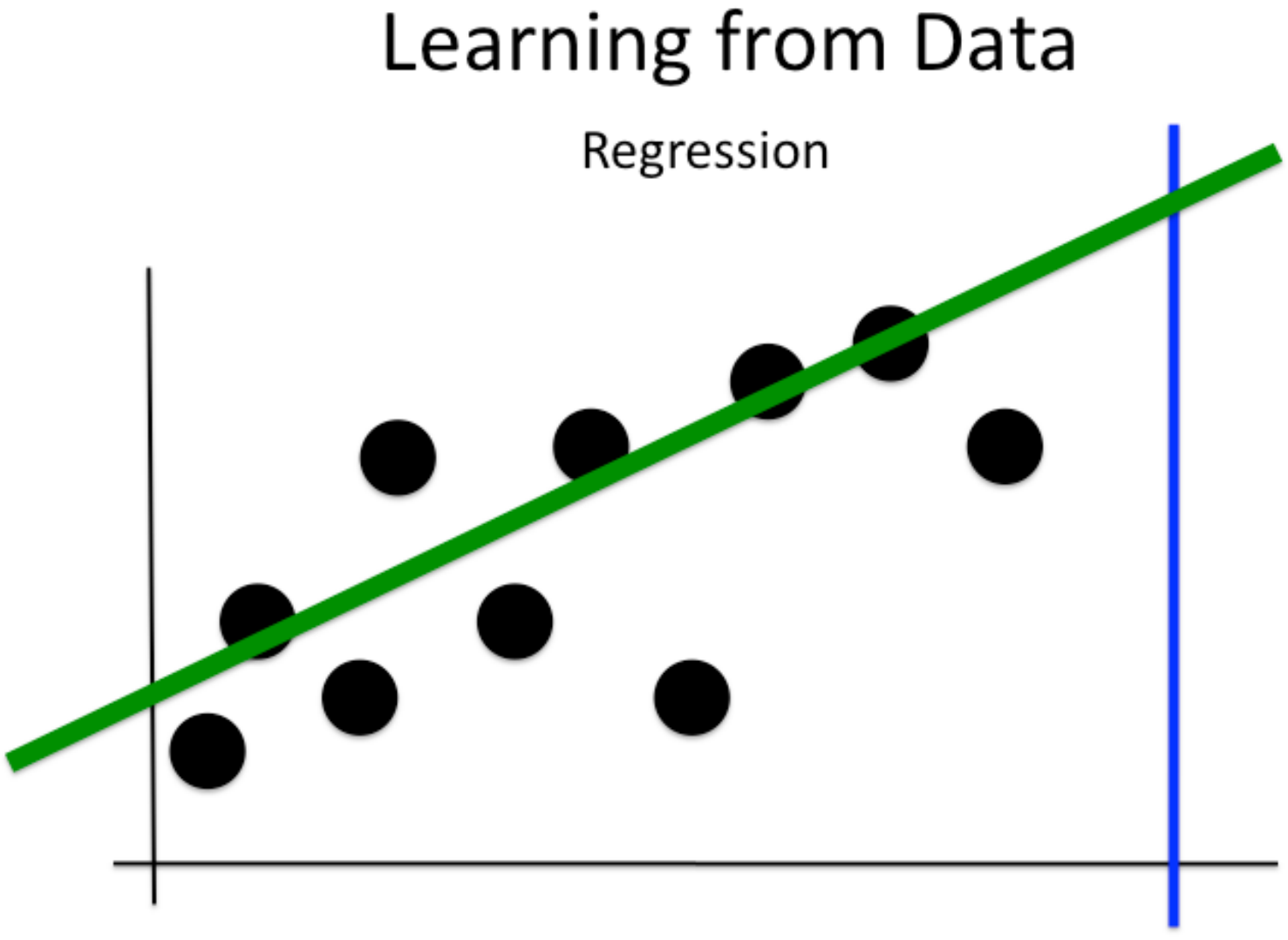
Goal: identify $y : \mathbb{R}^D \rightarrow \mathbb{R}$.

Given: a set of training data $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with targets, $\{t_0, t_1, \ldots, t_N\}$

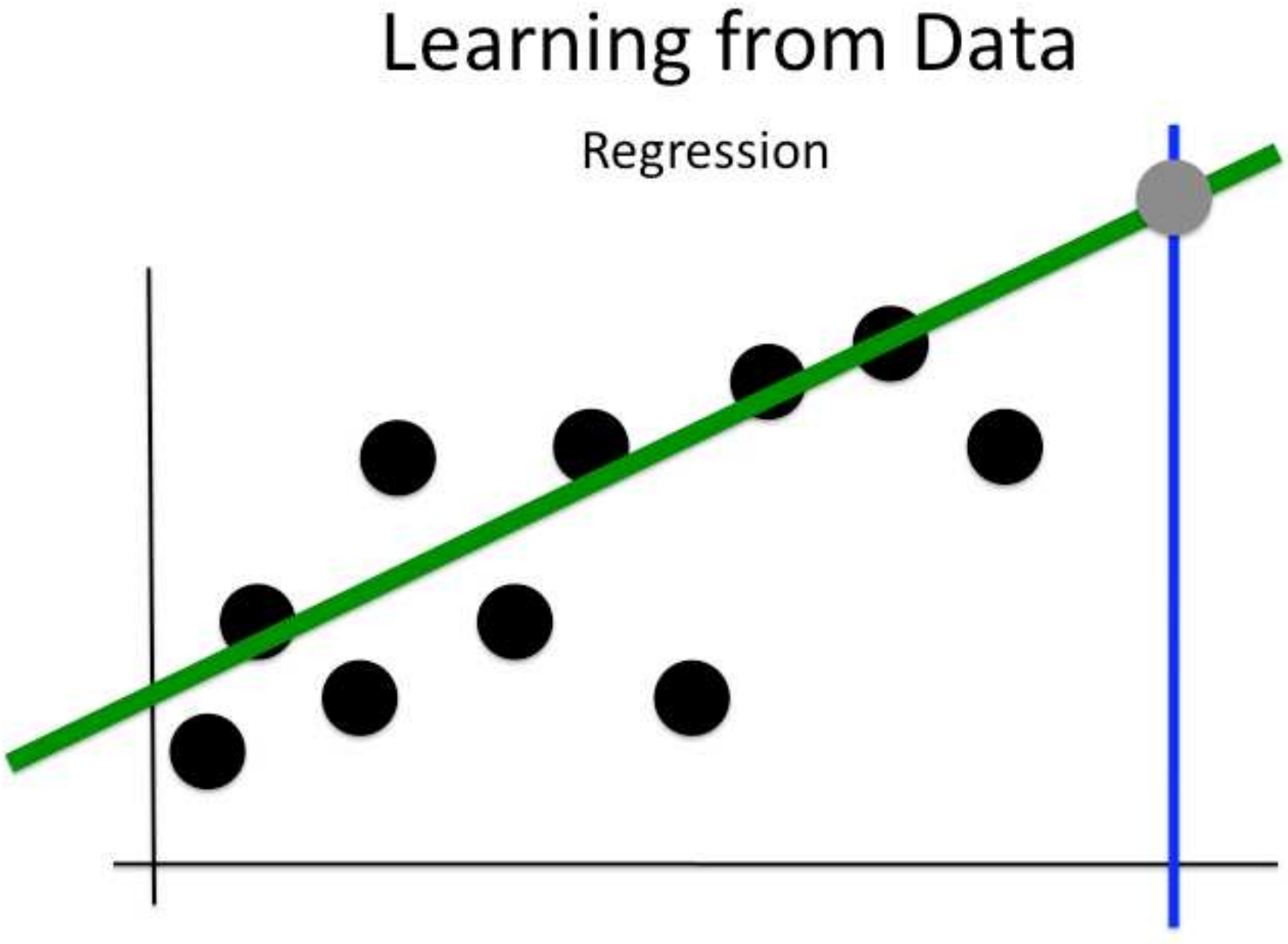
Recall Regression



Recall Regression



Recall Regression



Define the problem

In **linear regression**, we assume that the model that generates the data involves *only* a linear combination of the input variables.

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$$

Or, simplified

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

$\mathbf{w}$ is a vector of *weights* which define the M **parameters** of the model.

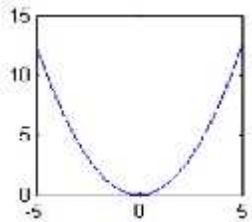
Optimization

How can we evaluate the performance of a regression solution?

Error Functions
(aka Loss Functions)

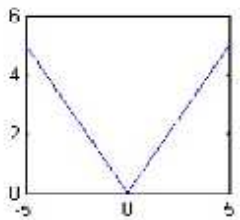
Simplest error: Squared Error from Target

$$E(t_i, y(\mathbf{x}_i, \mathbf{w})) = \frac{1}{2}(t_i - y(\mathbf{x}_i, \mathbf{w}))^2$$



Other options: Linear error

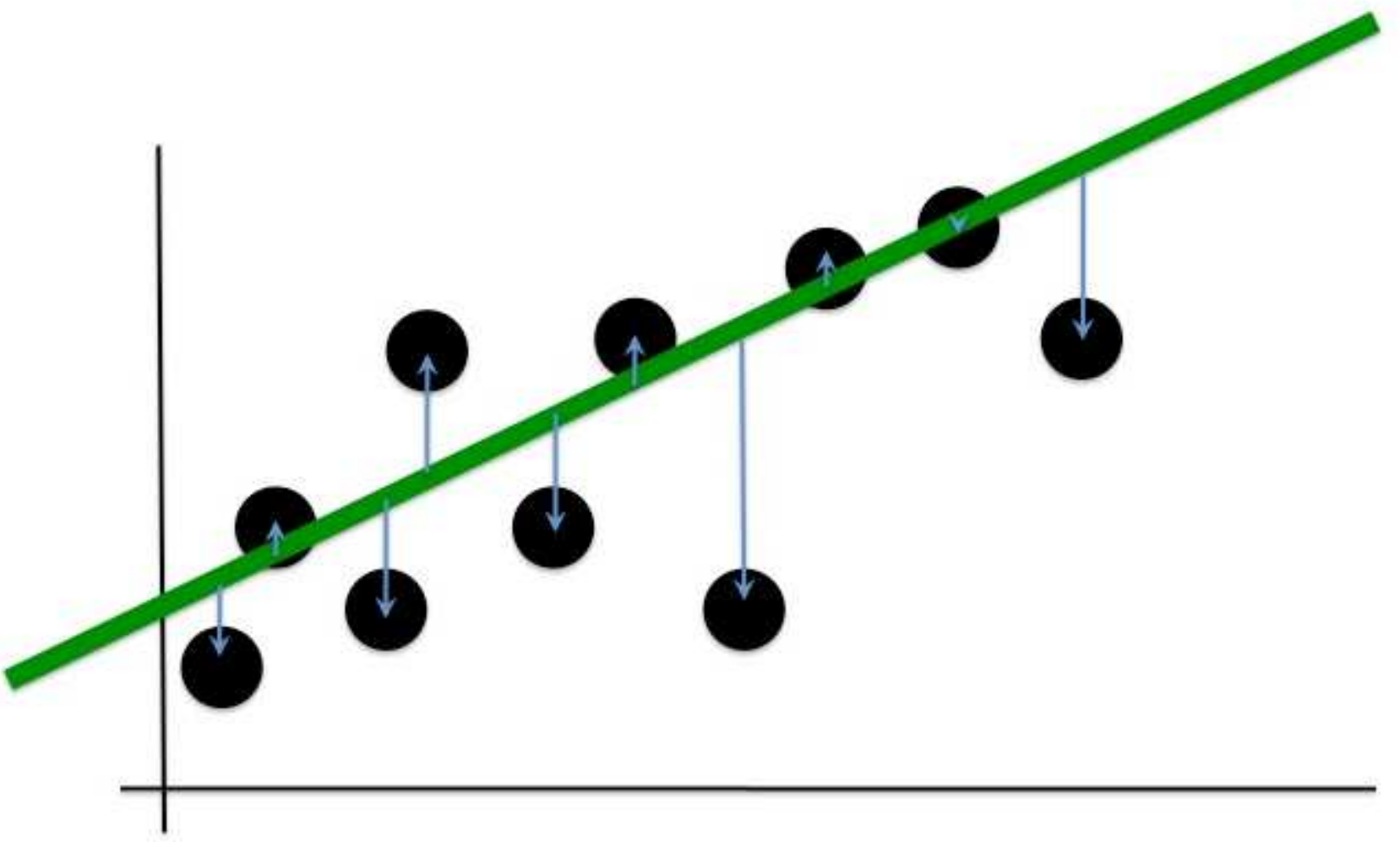
$$E(t_i, y(\mathbf{x}_i, \mathbf{w})) = |t_i - y(\mathbf{x}_i, \mathbf{w})|$$



Total Error

$$E(\mathbf{t}, y(\mathbf{x}, \mathbf{w})) = R_{emp} = \frac{1}{N}\sum_{i=1}^{N} E(t_i, y(\mathbf{x}_i, \mathbf{w}))$$

Regression Error



Likelihood of t

If we can describe the likelihood of a guess t, given a function y and training data x, we can **minimize** this risk, by setting its derivative to zero.

$$
\begin{aligned}
R_{emp} &= \frac{1}{N}\sum_{i=1}^{N} E(t_i, y(\mathbf{x}_i, \mathbf{w})) \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}(t_i - y(\mathbf{x}_i, \mathbf{w}))^2 \\
\nabla_{\mathbf{w}} R &= 0
\end{aligned}
$$

Likelihood and Risk

Brief Aside

The relationship between model likelihood and Empirical Risk.

The likelihood of a target given a model is:

$$p(t|x, \mathbf{w}, \beta) = N(t|y(x, \mathbf{w}), \beta - 1)$$

where $\beta = \frac{1}{\sigma^2}$ – the inverse variance.

So...

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{i=0}^{N-1} N(t_i|y(x_i, \mathbf{w}), \beta - 1)$$

assuming Independent Identically Distributed (iid) data.

Likelihood and Risk

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{i=0}^{N-1} N(t_i|y(x_i, \mathbf{w}), \beta - 1)$$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{i=0}^{N-1} \sqrt{\beta}\frac{1}{\sqrt{2\pi}} \exp\left\{\frac{-\beta}{2}(y(x_i, \mathbf{w}) - t_i)^2\right\}$$

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) &= \ln \prod_{i=0}^{N-1} \sqrt{\beta}\frac{1}{\sqrt{2\pi}} \exp\left\{\frac{-\beta}{2}(y(x_i, \mathbf{w}) - t_i)^2\right\} \\ &= -\frac{\beta}{2}\sum_{i=0}^{N-1}\left\{(y(x_i, \mathbf{w}) - t_i)^2\right\} + \frac{N}{2}\ln\beta - \frac{N}{2}\ln 2\pi \end{aligned}$$

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2} \sum_{i=0}^{N-1} \left\{ (y(x_i, \mathbf{w}) - t_i)^2 \right\} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln 2\pi$$

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \nabla_{\mathbf{w}} \left(-\frac{\beta}{2} \sum_{i=0}^{N-1} \left\{ (y(x_i, \mathbf{w}) - t_i)^2 \right\} \right)$$

To Maximize log likelihood:

$$\nabla_{\mathbf{w}} \left(-\frac{\beta}{2} \sum_{i=0}^{N-1} \left\{ (y(x_i, \mathbf{w}) - t_i)^2 \right\} \right) = 0$$

Maximizing (log) likelihood (under a gaussian) is **equivalent** to minimizing sum of squares error.

Maximize the log likelihood

Optimize the weights in one dimension.

In one dimension $\mathbf{w} = \begin{bmatrix} w_0 & w_1 \end{bmatrix}^T$

$$\nabla_{\mathbf{w}} R = \mathbf{0}$$

$$\begin{bmatrix} \frac{\partial R}{\partial w_0} \\ \frac{\partial R}{\partial w_1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$R(\mathbf{w}) = \frac{1}{2N} \sum_{i=0}^{N-1} (t_i - w_1 x_i - w_0)^2$$

Maximize the log likelihood

$$\nabla_{\mathbf{w}} R(\mathbf{w}) = \frac{1}{2N} \sum_{i=0}^{N-1} (t_i - w_1 x_i - w_0)^2$$

Set each partial to 0. First w_0.

$$\begin{aligned}
\frac{\partial R}{\partial w_0} &= \frac{1}{N} \sum_{i=0}^{N-1} (t_i - w_1 x_i - w_0)(-1) \\
\frac{1}{N} \sum_{i=0}^{N-1} (t_i - w_1 x_i - w_0)(-1) &= 0 \\
\frac{1}{N} \sum_{i=0}^{N-1} w_0 &= \frac{1}{N} \sum_{i=0}^{N-1} (t_i - w_1 x_i) \\
w_0 &= \frac{1}{N} \sum_{i=0}^{N-1} (t_i - w_1 x_i) \\
w_0 &= \frac{1}{N} \sum_{i=0}^{N-1} t_i - w_1 \frac{1}{N} \sum_{i=0}^{N-1} x_i
\end{aligned}$$

Maximize the log likelihood

$$\nabla_{\mathbf{w}} R(\mathbf{w}) = \frac{1}{2N} \sum_{i=0}^{N-1} (t_i - w_1 x_i - w_0)^2$$

Set each partial to 0. Now w_1.

$$\begin{aligned}
\frac{\partial R}{\partial w_1} &= \frac{1}{N} \sum_{i=0}^{N-1} (t_i - w_1 x_i - w_0)(-x_i) \\
\frac{1}{N} \sum_{i=0}^{N-1} (t_i - w_1 x_i - w_0)(-x_i) &= 0 \\
\frac{1}{N} \sum_{i=0}^{N-1} -(t_i x_i - w_1 x_i^2 - w_0 x_i) &= 0 \\
\frac{1}{N} \sum_{i=0}^{N-1} w_1 x_i^2 &= \frac{1}{N} \sum_{i=0}^{N-1} t_i x_i - \frac{1}{N} \sum_{i=0}^{N-1} w_0 x_i \\
w_1 \sum_{i=0}^{N-1} x_i^2 &= \sum_{i=0}^{N-1} t_i x_i - w_0 \sum_{i=0}^{N-1} x_i
\end{aligned}$$

Maximize the log likelihood

Substitute in w_0^* and Simplify.

$$
\begin{aligned}
w_0^* &= \frac{1}{N}\sum_{i=0}^{N-1} t_i - w_1 \frac{1}{N}\sum_{i=0}^{N-1} x_i \\
w_1 \sum_{i=0}^{N-1} x_i^2 &= \sum_{i=0}^{N-1} t_i x_i - w_0 \sum_{i=0}^{N-1} x_i \\
w_1 \sum_{i=0}^{N-1} x_i^2 &= \sum_{i=0}^{N-1} t_i x_i - \left(\frac{1}{N}\sum_{i=0}^{N-1} t_i - w_1 \frac{1}{N}\sum_{i=0}^{N-1} x_i \right) \sum_{i=0}^{N-1} x_i
\end{aligned}
$$

$$
\begin{aligned}
w_1 \left(\sum_{i=0}^{N-1} x_i^2 - \frac{1}{N}\sum_{i=0}^{N-1} x_i \sum_{i=0}^{N-1} x_i \right) &= \sum_{i=0}^{N-1} t_i x_i - \frac{1}{N}\sum_{i=0}^{N-1} t_i \sum_{i=0}^{N-1} x_i \\
w_1 &= \frac{\sum_{i=0}^{N-1} t_i x_i - \frac{1}{N}\sum_{i=0}^{N-1} t_i \sum_{i=0}^{N-1} x_i}{\sum_{i=0}^{N-1} x_i^2 - \frac{1}{N}\sum_{i=0}^{N-1} x_i \sum_{i=0}^{N-1} x_i}
\end{aligned}
$$

Maximized Log likelihood

Thus:

$$\begin{bmatrix} w_0^* \\ w_1^* \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{i=0}^{N-1} t_i - w_1^* \frac{1}{N}\sum_{i=0}^{N-1} x_i \\ \frac{\sum_{i=0}^{N-1} t_i x_i - \frac{1}{N}\sum_{i=0}^{N-1} t_i \sum_{i=0}^{N-1} x_i}{\sum_{i=0}^{N-1} x_i^2 - \frac{1}{N}\sum_{i=0}^{N-1} x_i \sum_{i=0}^{N-1} x_i} \end{bmatrix}$$

Done.

But this is a little clunky. Let's use linear algebra to generalize.

Extend to multiple dimensions

Maximum Log Likelihood calculation as vectors and matrices.

$$
\begin{aligned}
R_{emp}(\mathbf{w}) &= \frac{1}{2N}\sum_{i=0}^{N-1}(t_i - w_1 x_i - w_0)^2 \\
&= \frac{1}{2N}\sum_{i=0}^{N-1}\left(t_i - \begin{bmatrix}1 & x_i\end{bmatrix}\begin{bmatrix}w_0 \\ w_1\end{bmatrix}\right)^2 \\
&= \frac{1}{2N}\left\|\begin{bmatrix}t_0 \\ t_1 \\ \vdots \\ t_{N-1}\end{bmatrix} - \begin{bmatrix}1 & x_0 \\ 1 & x_1 \\ \vdots & \\ 1 & x_{N-1}\end{bmatrix}\begin{bmatrix}w_0 \\ w_1\end{bmatrix}\right\|^2 \\
&= \frac{1}{2N}\left\|\mathbf{t} - \mathbf{X}\mathbf{w}\right\|^2
\end{aligned}
$$

Extend to multiple dimensions

Now that we have a general form of the empirical Risk, we can easily extend to higher dimensions.

$$R_emp(\mathbf{w}) = \frac{1}{2N} \|\mathbf{t} - \mathbf{X}\mathbf{w}\|^2$$

Now...

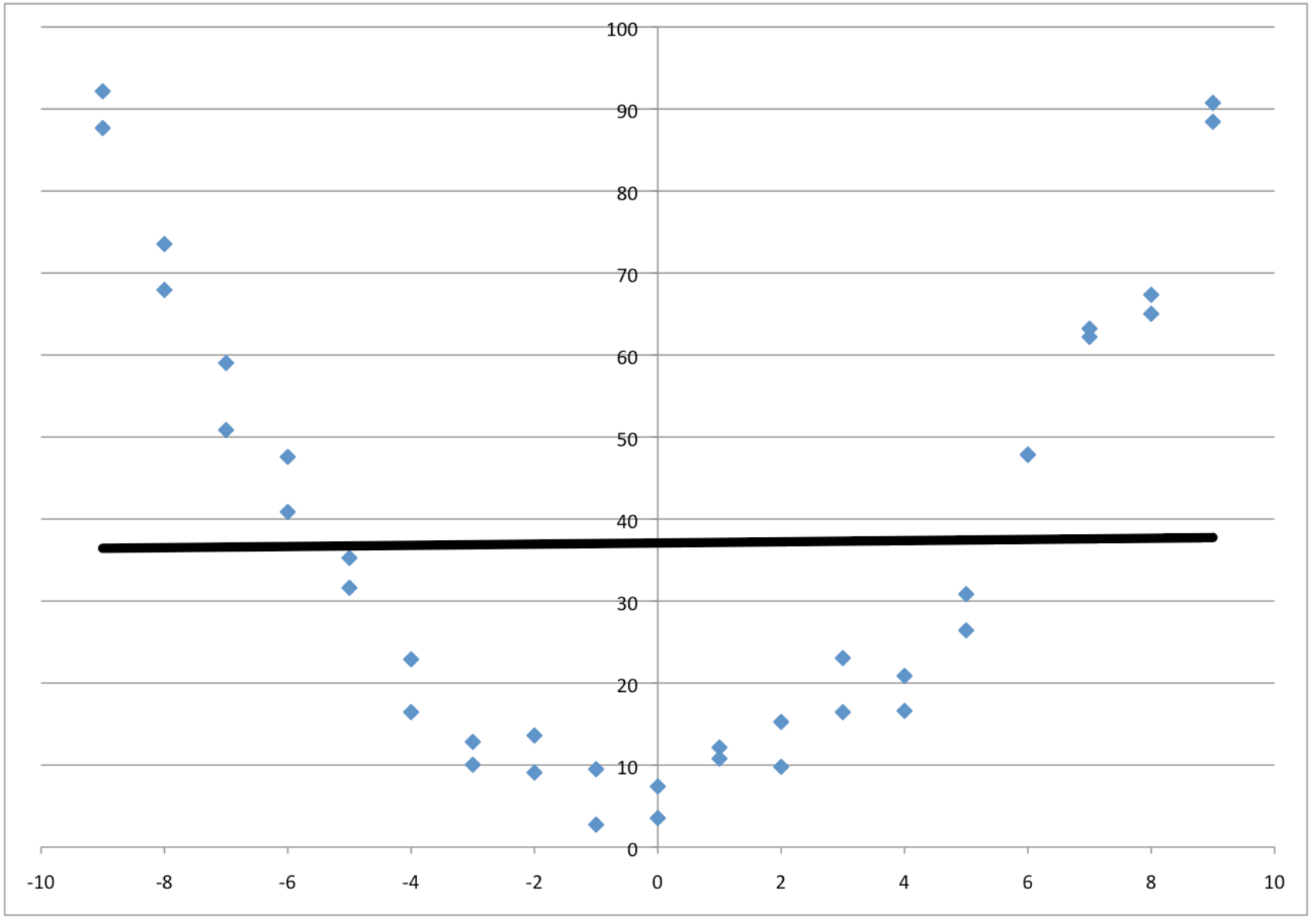
$$\begin{aligned} \nabla_{\mathbf{w}} R_{emp}(\mathbf{w}) &= 0 \\ \nabla_{\mathbf{w}} \left(\frac{1}{2N} \|\mathbf{t} - \mathbf{X}\mathbf{w}\|^2 \right) &= 0 \end{aligned}$$

General form of Risk minimization

Solve the Gradient $= 0$

$$
\begin{aligned}
\nabla_{\mathbf{w}} R_{emp}(\mathbf{w}) &= 0 \\
\nabla_{\mathbf{w}} \left(\frac{1}{2N} \|\mathbf{t} - \mathbf{X}\mathbf{w}\|^2 \right) &= 0 \\
\frac{1}{2N} \nabla_{\mathbf{w}} \left((\mathbf{t} - \mathbf{X}\mathbf{w})^T (\mathbf{t} - \mathbf{X}\mathbf{w}) \right) &= 0 \\
\frac{1}{2N} \nabla_{\mathbf{w}} \left((\mathbf{t}^T\mathbf{t} - \mathbf{t}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{t} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}) \right) &= 0 \\
\frac{1}{2N} \left(-\mathbf{X}^T\mathbf{t} - \mathbf{X}^T\mathbf{t} + 2\mathbf{X}^T\mathbf{X}\mathbf{w}^* \right) &= 0 \\
\frac{1}{2N} \left(-2\mathbf{X}^T\mathbf{t} + 2\mathbf{X}^T\mathbf{X}\mathbf{w}^* \right) &= 0 \\
\mathbf{X}^T\mathbf{X}\mathbf{w}^* &= \mathbf{X}^T\mathbf{t} \\
\mathbf{w}^* &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}
\end{aligned}
$$

Extension to fitting a line to a curve



Polynomial Regression

Polynomial Regression in One dimension.

$$y(x, \mathbf{w}) = \sum_{d=1}^{D} w_d x^d + w_0$$

Risk:

$$R = \frac{1}{2} \left\| \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_{n-1} \end{bmatrix} - \begin{bmatrix} 1 & x_0 & \dots & x_0^p \\ 1 & x_1 & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & \dots & x_{n-1}^p \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} \right\|^2$$

But this is just the same as **linear** regression in P dimensions.

Polynomial Regression as Linear Regression

To fit a P dimensional polynomial, create a P-element vector from x_i

$$\mathbf{x}_i = \begin{bmatrix} x_i^0 & x_i^1 & \dots & x_i^P \end{bmatrix}^T$$

Then linear regression in P dimensions.

How is this Linear regression?

- The regression is *linear* in the parameters.
- Despite manipulating x_i from one dimension to P dimensions.
- Now we fit a plane (or hyperplane) to a representation of x_i in a higher dimensional feature space.

How else can we use this method?
This generalizes to any set of functions $\phi_i : \mathbb{R} \to \mathbb{R}$.

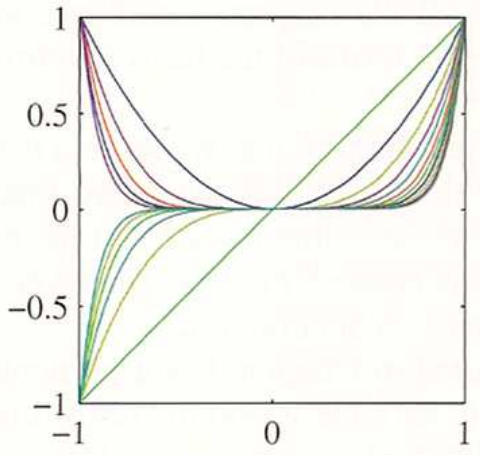
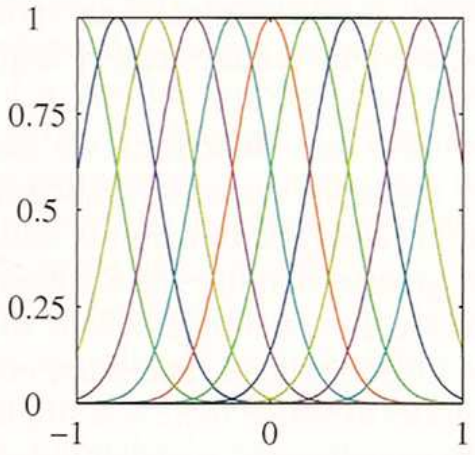
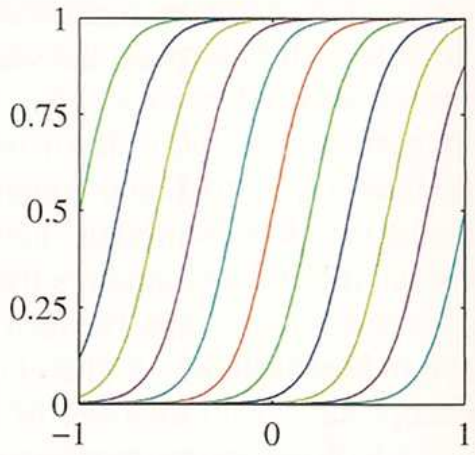
$$\mathbf{x}_i = \begin{bmatrix} \phi_0(x_i) & \phi_1(x_i) & \dots & \phi_P(x_i) \end{bmatrix}^T$$

Basis functions as Feature Extraction

These $\phi_i(x)$ functions are called **basis functions**, as they define the bases of the feature space.

This allows us to fit a linear decomposition of any type of function to data points.

Common Choices include: Polynomials, Gaussians, Sigmoids (we'll cover them) and Wave (sine waves) Functions.


1
0.5
0
−0.5
−1
−1
0
1

1
0.75
0.5
0.25
0
−1
0
1

1
0.75
0.5
0.25
0
−1
0
1

Training data v. testing data

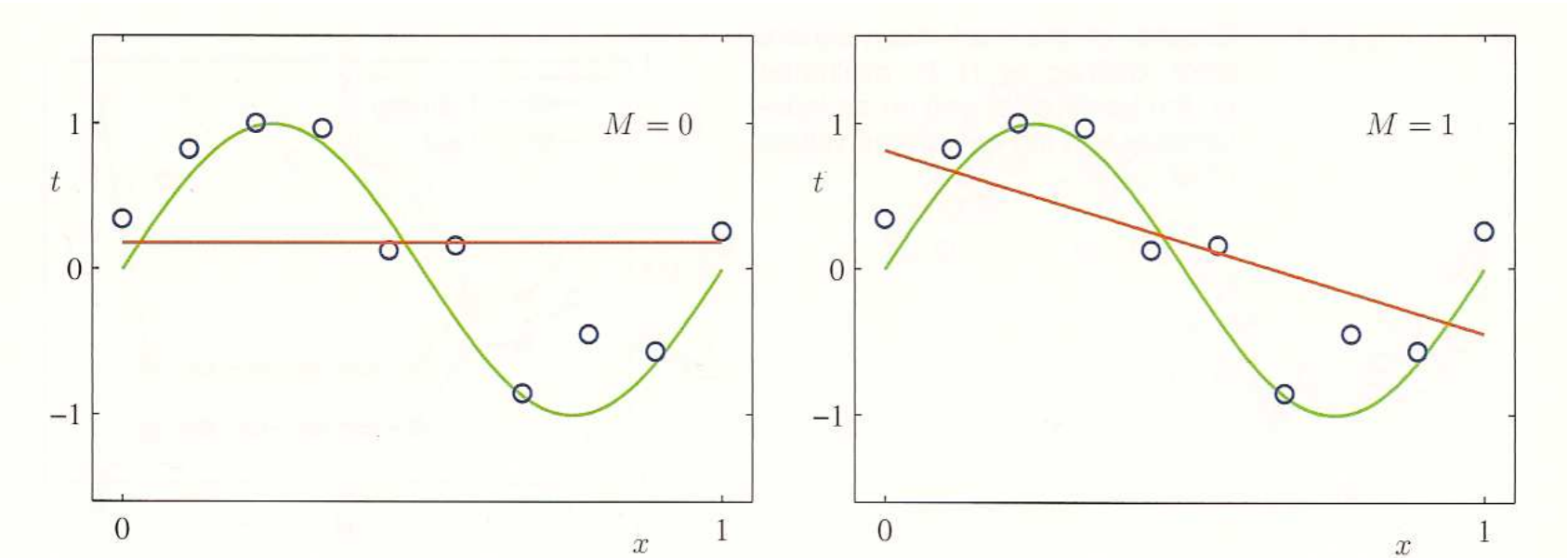
Evaluation.

- Evaluating the performance of a classifier on training data is meaningless.
- With enough parameters, a model can simply memorize (**encode**) every training point.

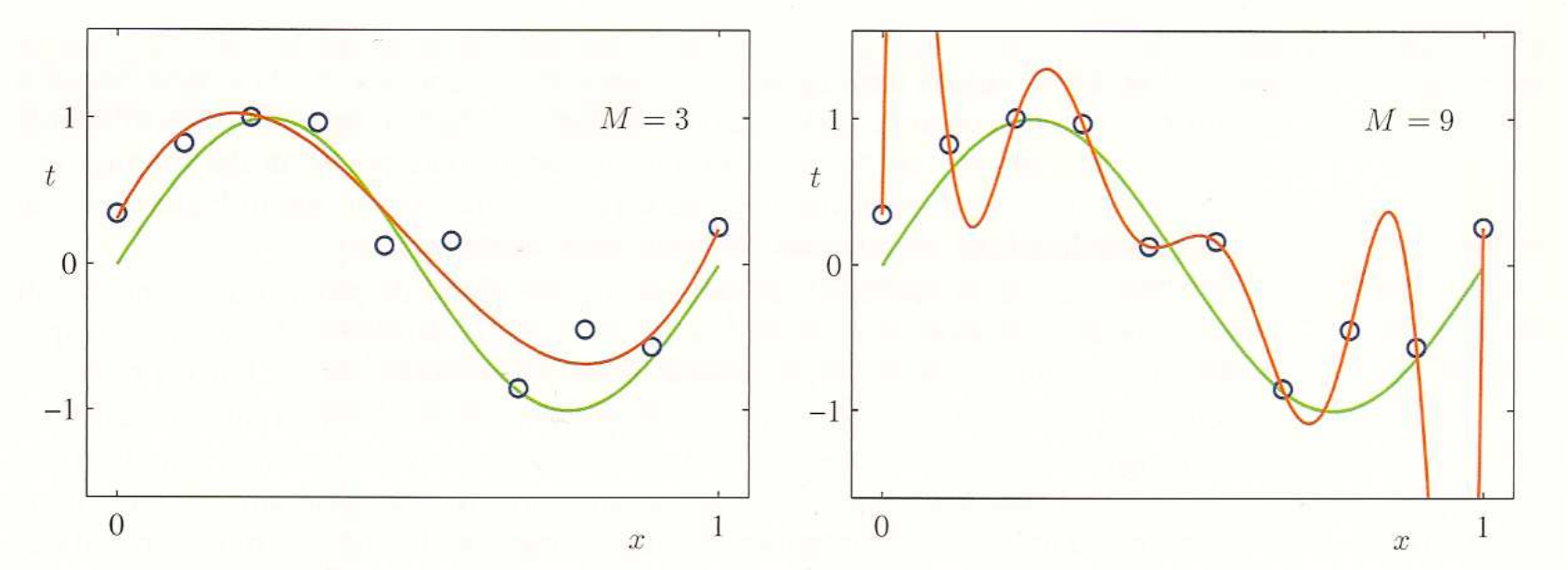
Therefore data is typically divided into two sets, **training** data and **testing** or **evaluation** data.

- **Training** data is used to learn model parameters.
- **Testing** data is used to evaluate the model.

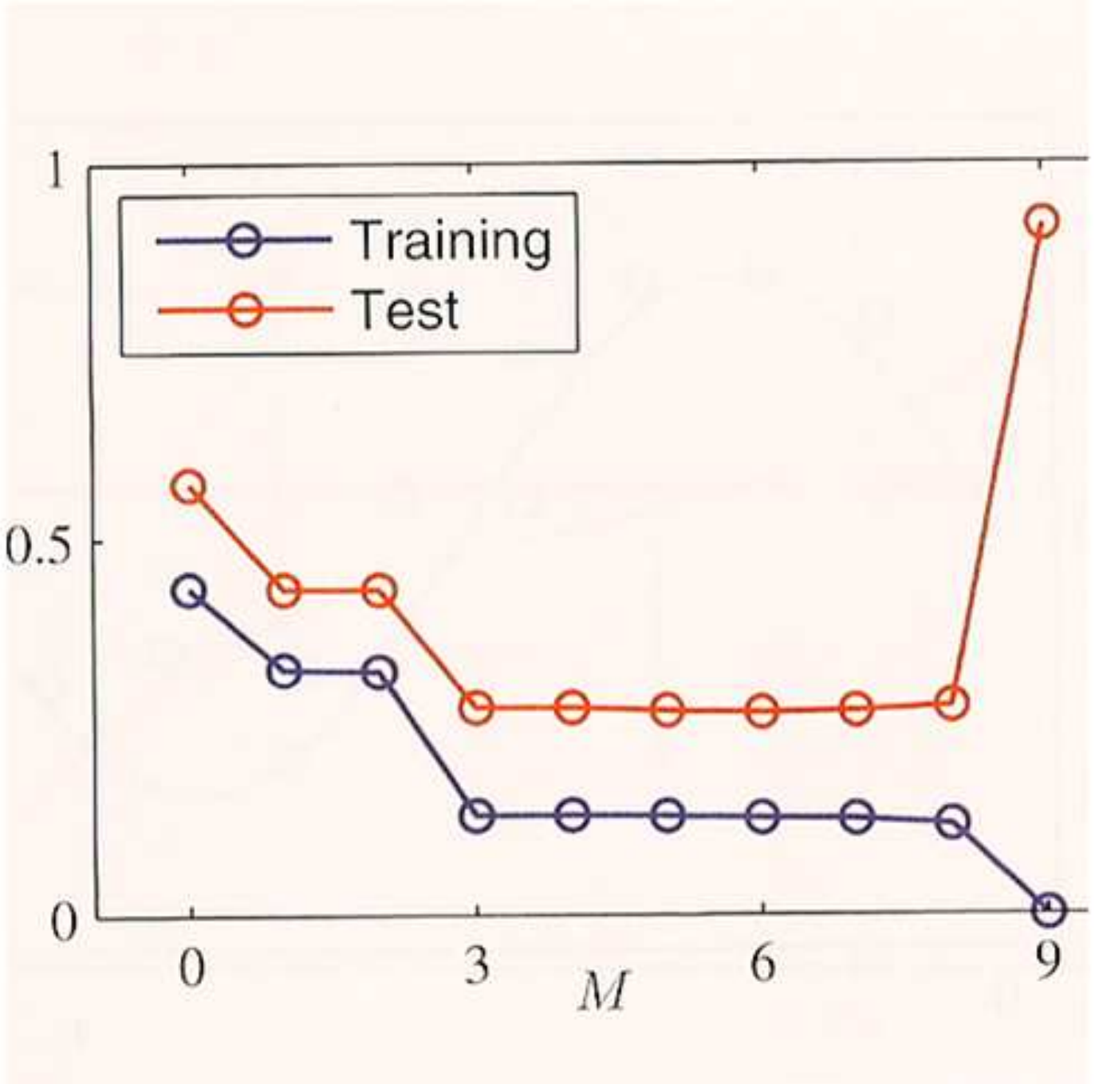
Overfitting 1/2



Overfitting 2/2



Overfitting



What is the **correct** model size?

The **best** model size is the size that generalizes to unseen data the best.

We approximate this by the testing error.

One way to optimize the parameters is to minimize the testing error.

This makes the testing data a *tuning* set.

However, this reduces the amount of *training* data in favor of parameter optimization.

Can we do this directly without sacrificing training data?

Regularization.

Context

Who cares about Linear Regression?

- It's a simple modeling approach that learns efficiently.
- By extensions to the basis functions, its very extensible.
- With regularization we can construct efficient models.

Bye

- Next
  - Regularization in Linear Regression.