ISTA 421 + INFO 521
Introduction to Machine Learning

Lecture 2: Linear Models

Clay Morrison
claytonm@email.arizona.edu
Harvill 437A
Phone 621-6609

23 August 2016

1

Today

- Homework 1
- Continue Introduction
- A simple learning problem
- Linear Model (what is the *model*)
- Loss Function (what is a *good* model)
- Least Squares (finding the '*best*' model)
- Prediction
- Moving to higher dimensions

2

Homework 1

- **Goal**: Set up and get comfortable with your programming environment, write some very simple scripts and recall some linear algebra.
- **DUE**: Next Friday, Sept 1, 5pm to the D2L Assignments folder (previously "Dropbox")
- Worth 24 points

Typical Supervised Machine Learning Workflow

(0) **Define the problem.** What is the task we want to teach a computer to do?

(1) **Collect data.** Gather data for training and testing sets. The larger and more diverse the data the better.

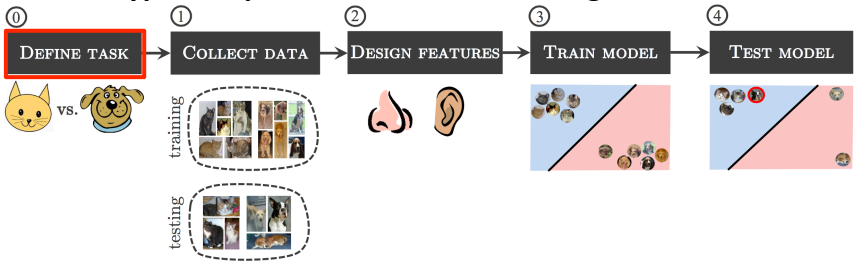
(2) **Design features.** What kind of features best describes the data?

(3) **Train the model.** Tune the parameters of an appropriate model on the training data using numerical optimization.

(4) **Test the model.** Evaluate the performance of the trained model on the testing data. If the results of this evaluation are poor, re-think the particular features used and gather more data if possible.

From forthcoming book: Watt, Borhani & Katsaggelos, *Machine Learning Refined*
http://mlrefined.wix.com/home-page

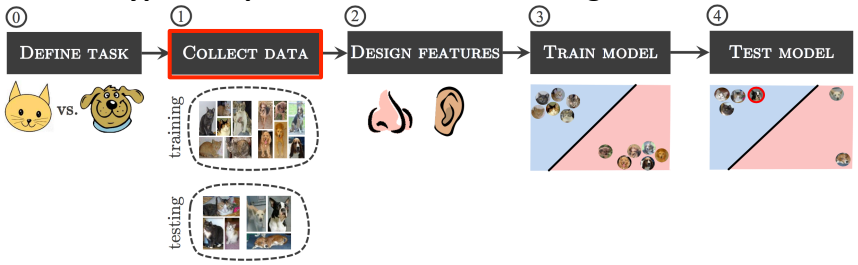
Typical Supervised Machine Learning Workflow



From forthcoming book: Watt, Borhani & Katsaggelos, *Machine Learning Refined*
http://mlrefined.wix.com/home-page

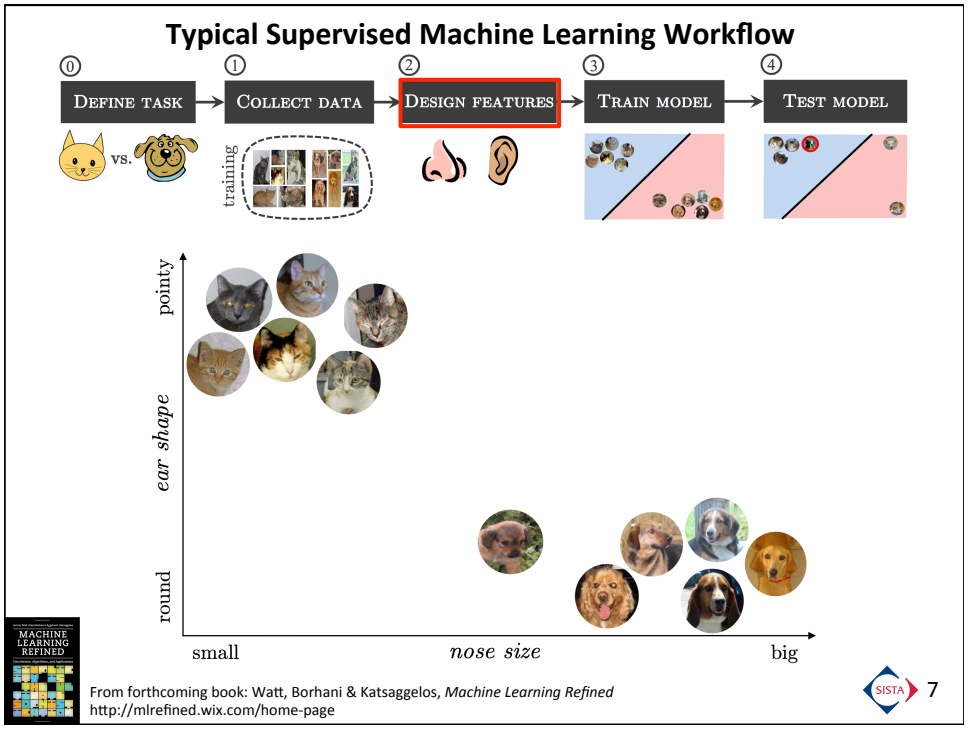
5

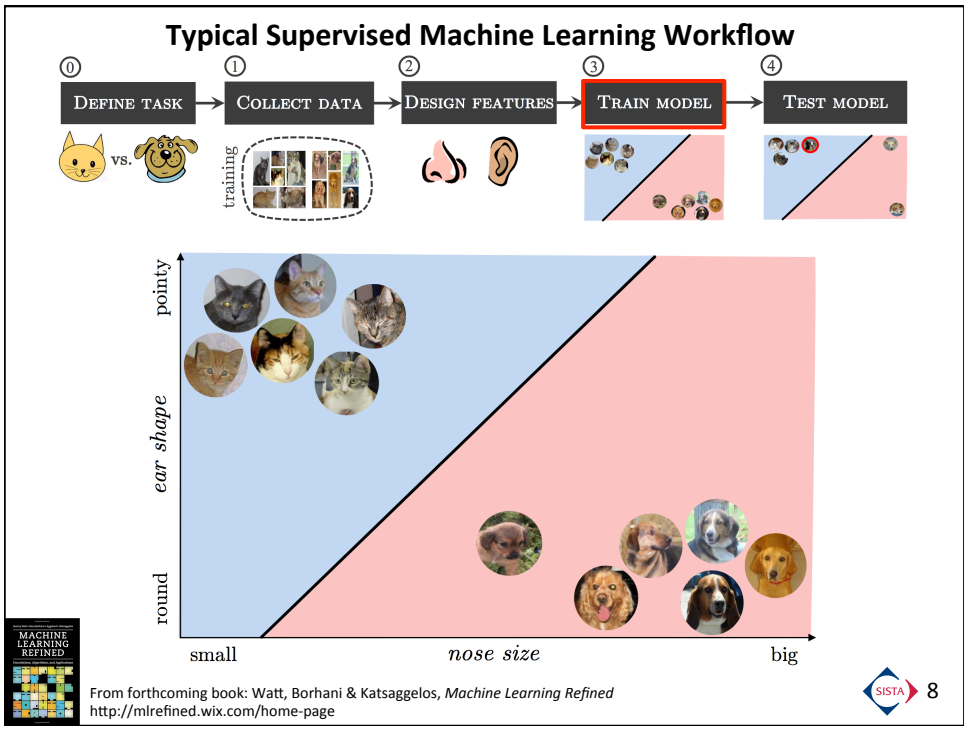
Typical Supervised Machine Learning Workflow



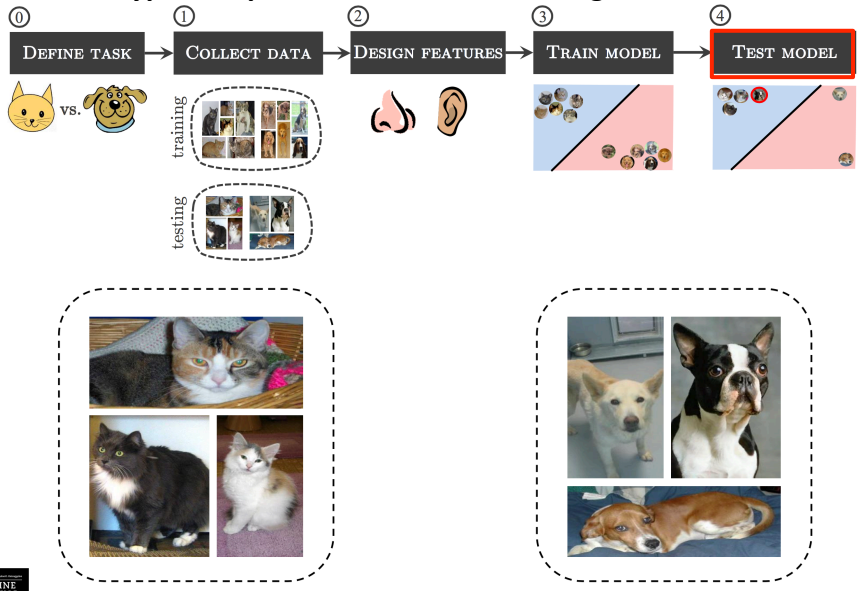
From forthcoming book: Watt, Borhani & Katsaggelos, *Machine Learning Refined*
http://mlrefined.wix.com/home-page

6


Typical Supervised Machine Learning Workflow
0 Define task
1 Collect data
2 Design features
3 Train model
4 Test model
vs.
training
pointy
ear shape
round
small
nose size
big
From forthcoming book: Watt, Borhani & Katsaggelos, Machine Learning Refined
http://mlrefined.wix.com/home-page
SISTA
7

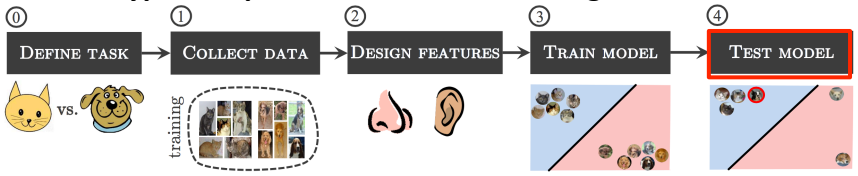

Typical Supervised Machine Learning Workflow
0 Define task
1 Collect data
2 Design features
3 Train model
4 Test model
vs.
training
pointy
ear shape
round
small
nose size
big
From forthcoming book: Watt, Borhani & Katsaggelos, Machine Learning Refined
http://mlrefined.wix.com/home-page
SISTA
8

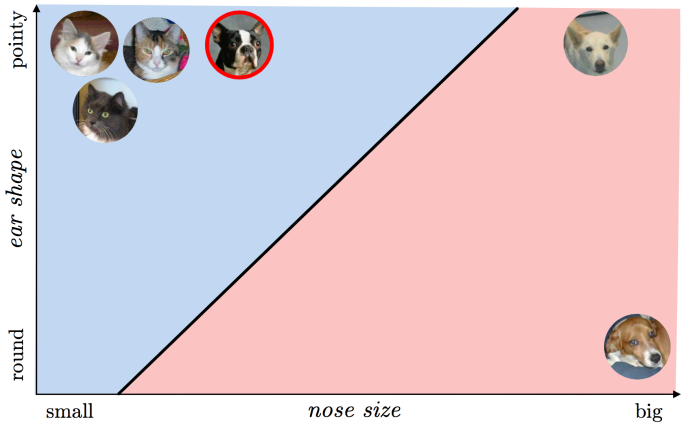
Typical Supervised Machine Learning Workflow



From forthcoming book: Watt, Borhani & Katsaggelos, *Machine Learning Refined*
http://mlrefined.wix.com/home-page

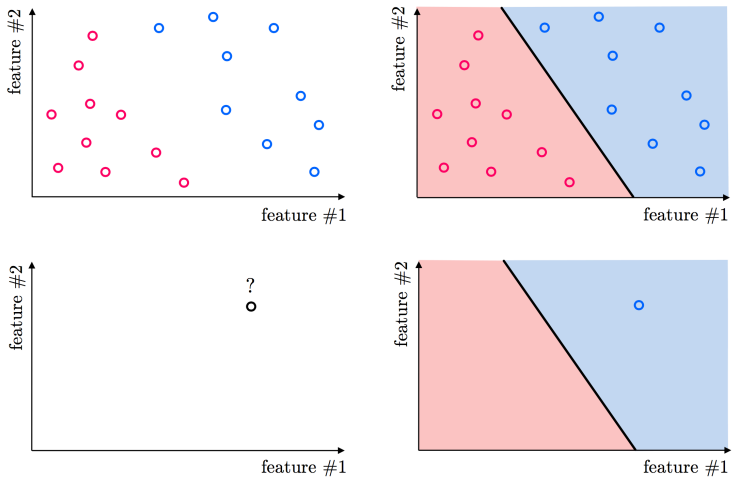
Typical Supervised Machine Learning Workflow



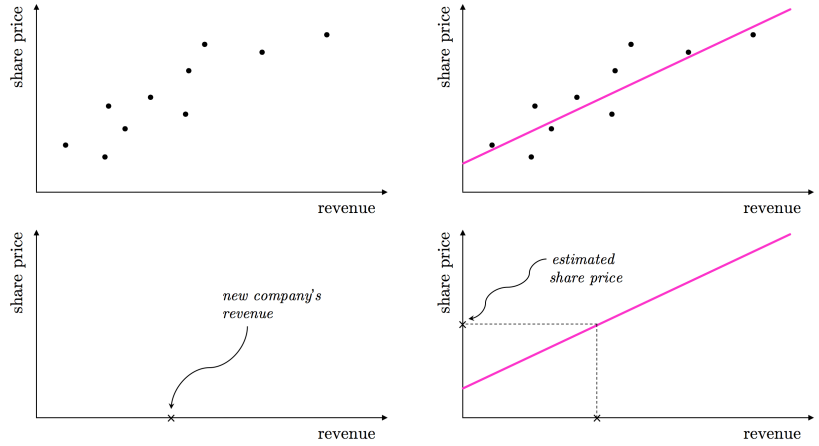


From forthcoming book: Watt, Borhani & Katsaggelos, *Machine Learning Refined*
http://mlrefined.wix.com/home-page

Classification



Regression



Three General Classes of ML

- **Supervised learning** – model $p(y|\mathbf{x})$
  - Given data and model, or data with correct output (label)
  - Regression, Classification, etc.
- **Unsupervised Learning** – model $p(\mathbf{x})$
  - Only given input data (no output)
  - Clustering, Latent Models, Projection methods, etc.
- **Reinforcement Learning** – model $p(\mathbf{s}_{t+1}|\mathbf{s}_t, a)$
  - Given input data, *some* output, and *grade* for output
  - Learning to choose better actions
  - Markov decision processes, POMDPs, planning

13

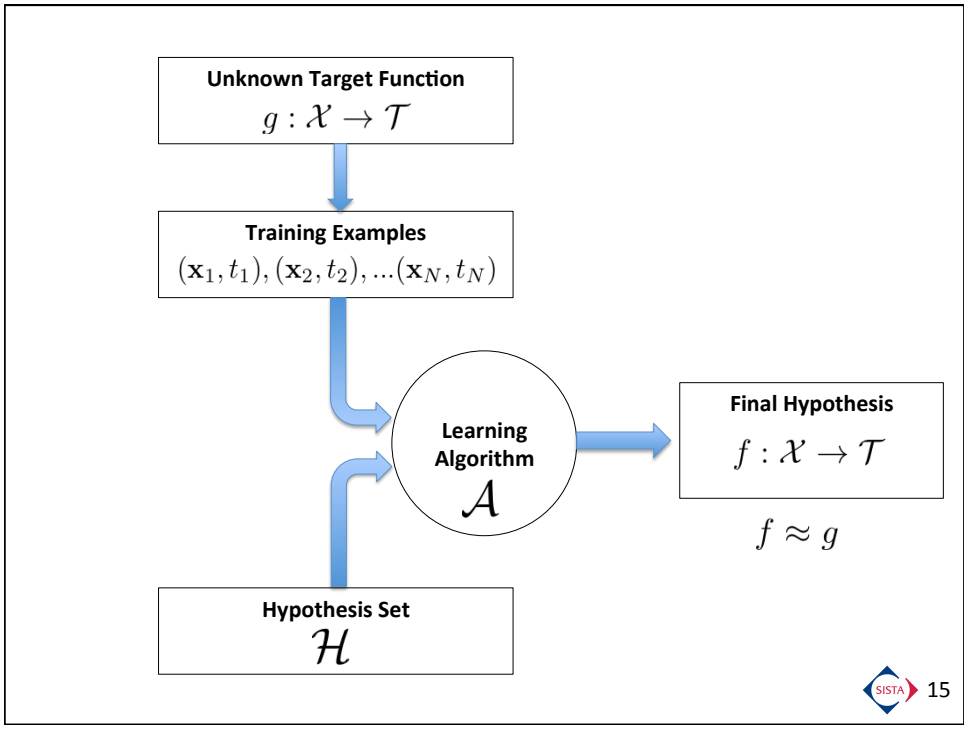
Supervised Learning: Some Terminology

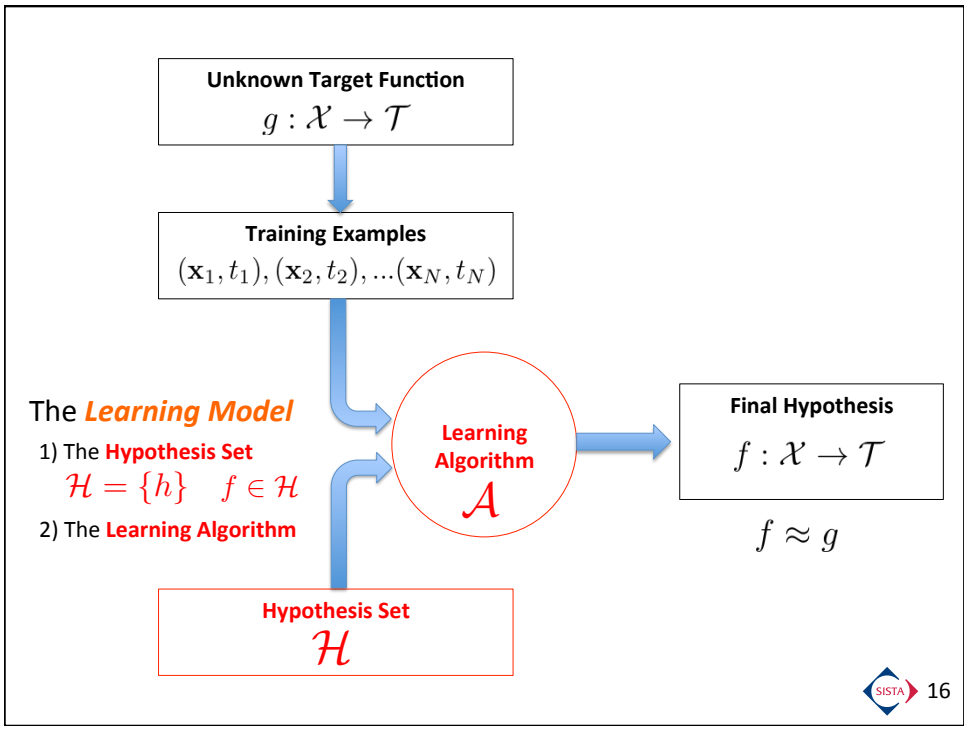
- Input: $\mathbf{x}$ (customer application)
- Output: t (good/bad customer?)
- Target function: $g : \mathcal{X} \rightarrow \mathcal{T}$ (*ideal* credit approval fn)
- Data: $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots (\mathbf{x}_N, t_N)$ (historical records)

- Hypothesis: $f : \mathcal{X} \rightarrow \mathcal{T}$ (formula to be used)

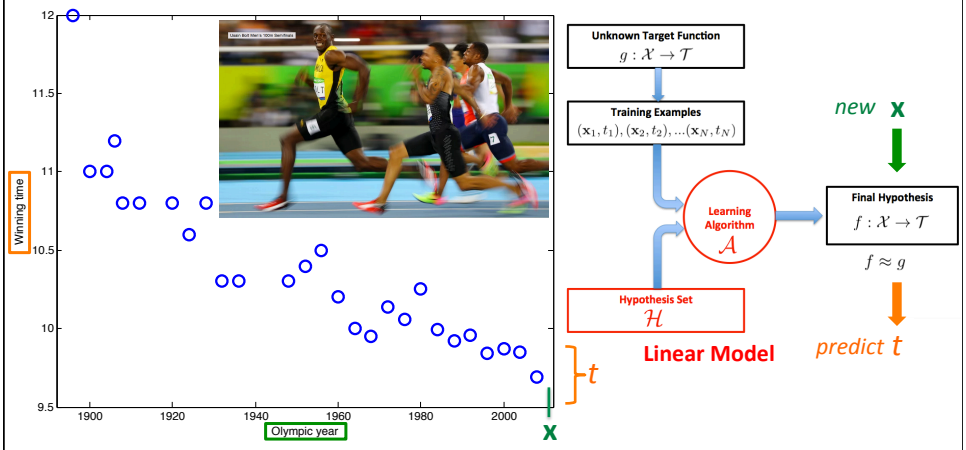
Adapted from Yaser S. Abu-Mostafa et al., *Learning from Data*

14


Unknown Target Function
$g : \mathcal{X} \rightarrow \mathcal{T}$
Training Examples
$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ...(\mathbf{x}_N, t_N)$
Learning Algorithm
$\mathcal{A}$
Final Hypothesis
$f : \mathcal{X} \rightarrow \mathcal{T}$
$f \approx g$
Hypothesis Set
$\mathcal{H}$
15


Unknown Target Function
$g : \mathcal{X} \rightarrow \mathcal{T}$
Training Examples
$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ...(\mathbf{x}_N, t_N)$
The *Learning Model*
1) The Hypothesis Set
$\mathcal{H} = \{h\} \quad f \in \mathcal{H}$
2) The Learning Algorithm
Learning Algorithm
$\mathcal{A}$
Final Hypothesis
$f : \mathcal{X} \rightarrow \mathcal{T}$
$f \approx g$
Hypothesis Set
$\mathcal{H}$
16

A simple learning problem



Want to describe **Winning time (*t*)** as a function of **Olympic year (x)**

Defining a Model

- Define function that maps inputs (Olympics year, x_i) to output or target values (Winning times, t_i)

  $t = f(x)$

- The model itself likely has parameters, which we'll generically refer to as 'θ' here. It is common to make them explicit within a function:

  $t = f(x ; \theta)$

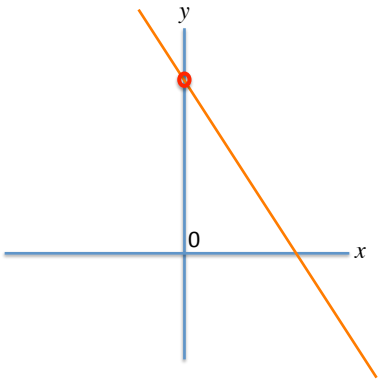
Lines!

- **Slope-intercept form**

$$y = mx + b^*$$

- General (standard) form

$$ax + by + c = 0$$



slope $m = -\frac{a}{b}$

y-intercept $b^* = -\frac{c}{b}$

x-intercept $= -\frac{c}{a}$

Linear Relationship

- ***y* = *m*x + *b*** **(or $t = w_1x + w_0$)**
  - the classic line (in 2D space)
  - For a given line, *m* and *b* are the ***parameters*** and *x* is a ***variable*** in the relationship: *y* = *f*(*x* ; *m*, *b*)
  - When considering alternate lines, we are adjusting *m* and *b*
- Generally, as long as the values that vary (assuming the others are constant) are not themselves involved in anything more than
  (1) addition and
  (2) scalar multiplication,
  … then the relationship is ***linear***.

Let's consider the relationship between *y* and *x*

$y = mx^2 + c$ $y = \sin(x)$ $\sqrt{y} = mx + c$ Not linear rel. btwn *x*,*y*

$y = mx + c^2$ $y = x\sin(m) + c$ Is linear rel. btwn *x*,*y*

What about the relationship between *y* and *m* (*m* is a parameter!)

Linear *Models*

- ***y* = *m*x + *b*** **(or $t = w_1x + w_0$)**
  - the classic line (in 2D space)
  - For a given line, *m* and *b* are the ***parameters*** and *x* is a ***variable*** in the relationship: $y = f(x\,;\, m, b)$
  - When considering alternate lines, we are adjusting *m* and *b*
- Generally, as long as the *parameters* are not themselves involved in anything more than
  (1) addition and
  (2) scalar multiplication,
  … then the relationship (as a function of parameters) is ***linear***
  and we are working with a **linear model**.

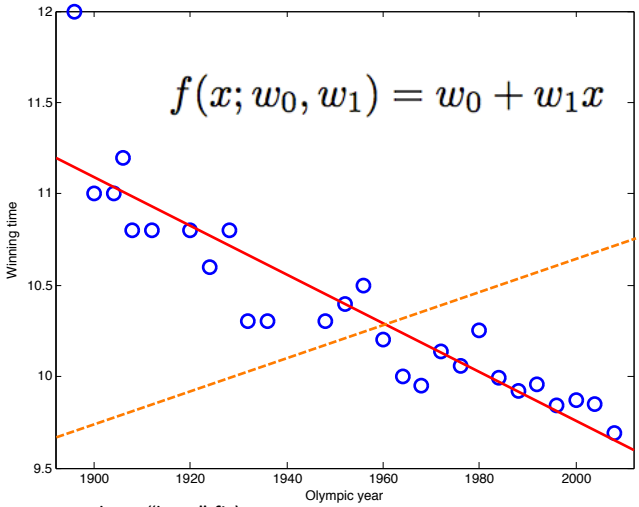
$y = mx^2 + c$ $y = \sin(x)$ $\sqrt{y} = mx + c$ Not linear rel. btwn x,y

$y = mx + c^2$ $y = x\sin(m) + c$ Is linear rel. btwn x,y ; **but not parameters!**

Now, suppose we hold *x* and *y* constant and treat the parameters *m*, *c* as *variable*?
(This is what we do when we are searching for "best fit" parameters…)
Which functions are then linear functions of the parameters m and c?

21

Data with line (particular w_0 & w_1)



(The red line happens to be a "best" fit)
(The dashed orange line does not describe the trend in the data very well; not a good fit)


22

Loss Function

$$\mathcal{L}_n()$$

Squared Error:

$$(t_n - f(x_n; w_0, w_1))^2$$

$$\mathcal{L}_n(t_n, f(x_n; w_0, w_1)) = (t_n - f(x_n; w_0, w_1))^2$$

Mean Squared Error:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(t_n, f(x_n; w_0, w_1))$$

23

Goal: Find the "best" values for the parameters of model according to the loss fn

- If we're lucky (i.e., the model and the loss fn are "well-behaved") we can derive an ***analytic*** solution. Otherwise, we'll pick some (iterative) optimization method that is appropriate.
- Our first example, using a **mean squared error loss function** with a **linear model** permits a nice analytic solution!
  - Here (and in the book) we'll first look at the direct, analytic method.
  - Another method: gradient descent
    - Same loss function, but iterative algorithm and can be used in cases where we don't have an analytic solution for the parameters

24

Least Mean Squares Solution

(for single variable, 2 parameter linear model)

$$f(x; w_0, w_1) = w_0 + w_1 x$$

$$\begin{aligned}
\mathcal{L} &= \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_n(t_n, f(x_n; w_0, w_1)) \\
&= \frac{1}{N}\sum_{n=1}^{N} (t_n - f(x_n; w_0, w_1))^2 \qquad \text{The specific loss fn we're working with here} \\
&= \frac{1}{N}\sum_{n=1}^{N} (t_n - (w_0 + w_1 x_n))^2 \qquad \text{The specific model we're working with here} \\
&= \frac{1}{N}\sum_{n=1}^{N} (w_1^2 x_n^2 + 2 w_1 x_n w_0 - 2 w_1 x_n t_n + w_0^2 - 2 w_0 t_n + t_n^2) \qquad \text{Multiply out and re-arrange to put into an easier-to deal with form.} \\
&= \frac{1}{N}\sum_{n=1}^{N} (w_1^2 x_n^2 + 2 w_1 x_n (w_0 - t_n) + w_0^2 - 2 w_0 t_n + t_n^2)
\end{aligned}$$

25

Least Mean Squares Solution

(for single variable, 2 parameter linear model)

$$f(x; w_0, w_1) = w_0 + w_1 x$$ ← Our model family

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} (w_1^2 x_n^2 + 2 w_1 x_n (w_0 - t_n) + w_0^2 - 2 w_0 t_n + t_n^2)$$ ← Our loss fn

Our goal: We want values for w_0 and w_1 that will ***minimize*** this loss function

I.e., we seek values for w_0 and w_1 that will make the loss function be the smallest when we actually sum over all the values of *x* and *t* in the dataset.

Because the loss function happens to be quadratic (in the two parameters) we can use a standard method from calculus for finding minima (maxima) directly: taking the derivative of the function and setting it to zero.

Our loss function has **two** parameters that we're trying set to minimize the loss fn, so we need to take the partial derivative (w.r.t. w_0 and w_1)

What we end up with are two functions, one for w_0 and one for w_1, and both will work with **any** data and give the best **least mean square** (LMS) fit!

Side note: One way to tell we have a unique extreme

First and Second Derivative
around particular x value (x=a)

Stationary point	*Sign diagram of $f'(x)$ near $x = a$*	*Shape of curve near $x = a$*
local maximum	+ | − at a	$x = a$
local minimum	− | + at a	$x = a$
horizontal inflection	+ + at a or − − at a	$x = a$ or $x = a$

Second derivative

Constant negative
(if *f* is quadratic)

Constant positive
(if *f* is quadratic)

No longer a constant fn
(if *f* higher order than quad)

Least Mean Squares Solution

(for single variable, 2 parameter linear model)

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(w_1^2 x_n^2 + 2w_1 x_n(w_0 - t_n) + w_0^2 - 2w_0 t_n + t_n^2)$$

- Partial derivative for w_0

First, since we're taking the partial w.r.t. w_0, can drop any terms without w_0.

$$\frac{1}{N}\sum_{n=1}^{N}\left[w_0^2 + 2w_1 x_n w_0 - 2w_0 t_n\right]$$

$$w_0^2 + 2w_0 w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - 2w_0 \frac{1}{N}\left(\sum_{n=1}^{N} t_n\right)$$

Next, move sums inward to put in easier form

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right)$$

Finally, take deriv. w.r.t. w_0

Continued…

- Solve for $\frac{\partial \mathcal{L}}{\partial w_0} = 0$

$$2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) = 0$$

$$2w_0 = \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) - w_1 \frac{2}{N}\left(\sum_{n=1}^{N} x_n\right)$$

$$w_0 = \frac{1}{N}\left(\sum_{n=1}^{N} t_n\right) - w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right)$$

29

Least Mean Squares Solution

(for single variable, 2 parameter linear model)

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(w_1^2 x_n^2 + 2w_1 x_n(w_0 - t_n) + w_0^2 - 2w_0 t_n + t_n^2)$$

- Partial derivative for w_1

Do the same for w_1 …

$$\frac{1}{N}\sum_{n=1}^{N}\left[w_1^2 x_n^2 + 2w_1 x_n w_0 - 2w_1 x_n t_n\right]$$ only keep terms with w_1

$$w_1^2 \frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)$$ move sums inside

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)$$ now take partial derivative

$$= w_1 \frac{2}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n\left(\overbrace{\bar{t} - w_1\bar{x}}^{\text{solution for } w_0} - t_n\right)\right)$$ plug in solution for w_0

… and rearrange terms

$$= w_1 \frac{2}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \bar{t}\frac{2}{N}\left(\sum_{n=1}^{N} x_n\right) - w_1\bar{x}\frac{2}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} x_n t_n\right)$$

30

- Partial derivative for w_1 continued…

$$\frac{\partial \mathcal{L}}{\partial w_1} = w_1 \frac{2}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \bar{t}\frac{2}{N}\left(\sum_{n=1}^{N} x_n\right) - w_1 \bar{x}\frac{2}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} x_n t_n\right)$$

$$= 2w_1\left[\left(\frac{1}{N}\sum_{n=1}^{N} x_n^2\right) - \bar{x}\,\bar{x}\right] + 2\bar{t}\bar{x} - 2\frac{1}{N}\left(\sum_{n=1}^{N} x_n t_n\right)$$

replace remaining mean x with $\bar{x}$ and group w_1 terms

$$2w_1\left[\left(\frac{1}{N}\sum_{n=1}^{N} x_n^2\right) - \bar{x}\,\bar{x}\right] + 2\bar{t}\bar{x} - 2\frac{1}{N}\left(\sum_{n=1}^{N} x_n t_n\right) = 0$$

Solve for w_1 with $\frac{\partial \mathcal{L}}{\partial w_1} = 0$ …

$$2w_1\left[\left(\frac{1}{N}\sum_{n=1}^{N} x_n^2\right) - \bar{x}\,\bar{x}\right] = 2\frac{1}{N}\left(\sum_{n=1}^{N} x_n t_n\right) - 2\bar{t}\bar{x}$$

$$w_1 = \frac{\frac{1}{N}\left(\sum_{n=1}^{N} x_n t_n\right) + \bar{t}\bar{x}}{\left(\frac{1}{N}\sum_{n=1}^{N} x_n^2\right) - \bar{x}\,\bar{x}} = \frac{\left(\frac{1}{N}\sum_{n=1}^{N} x_n t_n\right) - \left(\frac{1}{N}\sum_{m=1}^{N} t_n\right)\left(\frac{1}{N}\sum_{m=1}^{N} x_n\right)}{\left(\frac{1}{N}\sum_{n=1}^{N} x_n^2\right) - \left(\frac{1}{N}\sum_{n=1}^{N} x_n\right)^2} = \frac{\overline{xt} - \bar{x}\bar{t}}{\overline{x^2} - (\bar{x})^2}$$

31

Solving LMS: Method 1 (analytic)

(for single variable, 2 parameter linear model)

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(w_1^2 x_n^2 + 2w_1 x_n(w_0 - t_n) + w_0^2 - 2w_0 t_n + t_n^2)$$

- Partial derivative for w_0:

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1\frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right)$$

- Set $\frac{\partial \mathcal{L}}{\partial w_0} = 0$ and solve for w_0:

$$w_0 = \frac{1}{N}\left(\sum_{n=1}^{N} t_n\right) - w_1\frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) = \bar{t} - w_1\bar{x}$$

The so-called **normal equations**!

- Partial derivative for w_1:

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1\frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)$$

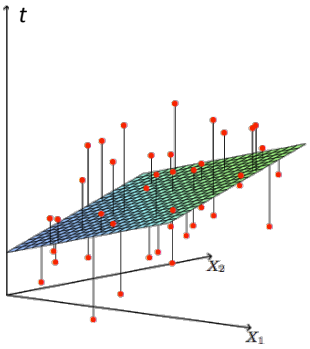
- Plug in w_0, set $\frac{\partial \mathcal{L}}{\partial w_1} = 0$ and solve for w_1:

$$w_1 = \frac{\left(\frac{1}{N}\sum_{n=1}^{N} x_n t_n\right) - \left(\frac{1}{N}\sum_{m=1}^{N} t_n\right)\left(\frac{1}{N}\sum_{m=1}^{N} x_n\right)}{\left(\frac{1}{N}\sum_{n=1}^{N} x_n^2\right) - \left(\frac{1}{N}\sum_{n=1}^{N} x_n\right)^2} = \frac{\overline{xt} - \bar{x}\bar{t}}{\overline{x^2} - (\bar{x})^2}$$

32

How about more than 1 input?

- Most problems will involve more than just the relationship between 1 input attribute and a target.
- Extending our linear models to higher dimensions is desirable, and at least for 2 inputs (now the "line" is a plane in 3D) it is easy to visualize the geometry.
- In general, a linear model with ***n*** input variables and ***n***+1 parameters (the ***w***'s, with their values determined) is an ***n***-dimensional "hyperplane" embedded in ***n***+1 dimensions.



Things quickly get messy as we increase the dimensions…

- Suppose we want a richer predictive model for the Olympic data: not only include the best overall time for the gold, but also the best times of each sprinter that raced (s_1,…,s_8)
- This is a 9 dimensional hyperplane with 10 parameters:

$$t = f(x, s_1, ..., s_8; w_0, ..., w_9) = w_0 + w_1 x + w_2 s_1 + w_3 s_2 + w_4 s_3 + w_5 s_4 + w_6 s_5 + w_7 s_6 + w_8 s_7 + w_9 s_8$$

- The math is fundamentally the same, but to derive the normal equations, we need to take 10 partial derivatives, then have 10 equations to re-arrange and substitute back in…