ISTA 421 + INFO 521 Introduction to Machine Learning

Lecture 4: Higher Dimensions, Geometry of LLMS, Nonlinear response (basis fns)

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

29 August 2018 1

Next Topics

- Moving to higher dimensions
  - Linear Algebra: Quick review of some operations
  - Some Geometry of Linear Algebra
  - Least Mean Squares in Matrix formulation
  - The Geometry of LMS solution
- Nonlinear Response: Basis Functions
- Model Selection
  - Generalization and Overfitting
  - Method 1: Cross Validation
- Regularized Least Squares

2

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:

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

- 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:

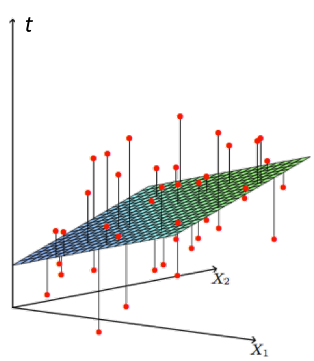
$$w_1 = \frac{\left(\frac{1}{N}\sum_{n=1}^{N} x_n t_n\right) - \left(\frac{1}{N}\sum_{m=1}^{N} t_n\right)\left(\frac{1}{N}\sum_{m=1}^{N} x_n\right)}{\left(\frac{1}{N}\sum_{n=1}^{N} x_n^2\right) - \left(\frac{1}{N}\sum_{n=1}^{N} x_n\right)^2} = \boxed{\frac{\overline{xt} - \overline{x}\overline{t}}{\overline{x^2}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

The so-called **normal equations**!

3

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. For 2 inputs it is easy to visualize the geometry: now the "line" is a plane in 3D
- In general, a (regression) 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.



4

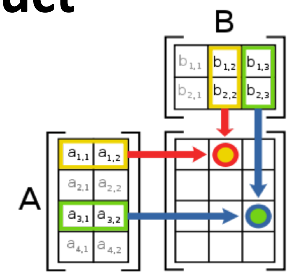
Dot product

$$\mathbf{a}^\top \mathbf{b} = \sum_{n=1}^{N} a_n b_n$$

Matrix Product

Let $\mathbf{C} = \mathbf{AB}$, where
- $\mathbf{A}$ is a $N \times P$ matrix
- $\mathbf{B}$ is a $P \times M$ matrix
- $\mathbf{C}$ is a $N \times M$ matrix

and each entry C of $\mathbf{C}$ is: $C_{ij} = \sum_k A_{ik} B_{kj}$



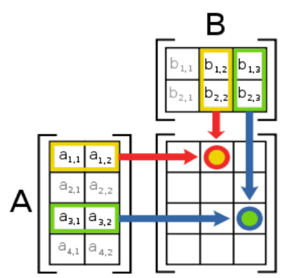
$$\begin{array}{cc} & \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \\ \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} & \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \end{bmatrix} \end{array}$$

(1st edition of FCML, has small error on p.18: copies first line, but index for a's become a_{21}, a_{22})

5

Inner versus Outer Product

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{x}^\top = [x_1, x_2, \cdots, x_n]$$



Inner (dot) product of vectors **a** and **b** (where both are size N)

$$\mathbf{a}^\top \mathbf{b} = \sum_{n=1}^{N} a_n b_n$$

Inner product is commutative:
$\mathbf{a}^\top\mathbf{b} = \mathbf{b}^\top\mathbf{a}$

Outer product of vectors vectors **a** (of size N) and **b** (of size M)

$$\mathbf{a}\mathbf{b}^\top = \begin{bmatrix} a_1b_1 & a_1b_2 & \dots & a_1b_m \\ a_2b_1 & a_2b_2 & \dots & a_2b_m \\ \vdots & \vdots & \ddots & \vdots \\ a_nb_1 & a_nb_2 & \dots & a_nb_m \end{bmatrix}$$

Outer product is *not necessarily* commutative:
$\mathbf{a}\mathbf{b}^\top \neq \mathbf{b}\mathbf{a}^\top$ (in general)

Both inner and outer products are just special cases of matrix product.

6

Simple Linear Model in Matrix Notation

- First, express our original 1-variable, 2-param model in matrix notation:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \qquad \mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix}$$

$$f(x_n; w_0, w_1) = \mathbf{w}^\top \mathbf{x}_n = w_0 + w_1 x_n$$

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2$$

7

Simple Linear Model in Matrix Notation

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2$$

- Next, express the operations involving all of the data (the inputs $\mathbf{x}_n$ and the targets $\mathbf{t}_n$):

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad \mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

Design Matrix

$$\mathbf{X}\mathbf{w} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_1 \\ w_0 + w_1 x_2 \\ \vdots \\ w_0 + w_1 x_N \end{bmatrix}$$

$$\mathbf{t} - \mathbf{X}\mathbf{w} = \begin{bmatrix} t_1 - w_0 - w_1 x_1 \\ t_2 - w_0 - w_1 x_2 \\ \vdots \\ t_N - w_0 - w_1 x_N \end{bmatrix}$$

$$\begin{aligned}(\mathbf{t} - \mathbf{X}\mathbf{w})^\top(\mathbf{t} - \mathbf{X}\mathbf{w}) &= (t_1 - (w_0 + w_1 x_1))^2 + (t_2 - (w_0 + w_1 x_2))^2 + \ldots \\ &\quad + (t_N - (w_0 + w_1 x_N))^2 \\ &= \sum_{n=1}^{N}(t_n - (w_0 + w_1 x_n))^2 \\ &= \sum_{n=1}^{N}(t_n - f(x_n; w_0, w_1))^2\end{aligned}$$

$$\mathcal{L} = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^\top(\mathbf{t} - \mathbf{X}\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(t_n - \mathbf{w}^\top \mathbf{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(t_n - (w_0 + w_1 x_n))^2$$

Much nicer! The $\mathbf{x}^\top\mathbf{y}$ operation allows us to drop the sums!

8

Simple Linear Model in Matrix Notation

- Now that we have the matrix version of the loss function, "just" take derivative…

$$\begin{aligned}\mathcal{L} &= \frac{1}{N}(\mathbf{Xw}-\mathbf{t})^\top(\mathbf{Xw}-\mathbf{t}) \\ &= \frac{1}{N}((\mathbf{Xw})^\top-\mathbf{t}^\top)(\mathbf{Xw}-\mathbf{t}) \\ &= \frac{1}{N}(\mathbf{Xw})^\top\mathbf{Xw} \boxed{-\frac{1}{N}\mathbf{t}^\top\mathbf{Xw}-\frac{1}{N}(\mathbf{Xw})^\top\mathbf{t}} + \frac{1}{N}\mathbf{t}^\top\mathbf{t} \\ &= \frac{1}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{Xw}-\frac{2}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{t}+\frac{1}{N}\mathbf{t}^\top\mathbf{t}\end{aligned}$$

note: book accidentally drops the 1/N here

$\mathbf{t}^\top\mathbf{Xw}$ and $\mathbf{w}^\top\mathbf{X}^\top\mathbf{t}$ are the transpose of one another … and both products come out to be **scalars** (i.e., "1x1 matrices", where transpose of one is the same as the other), so the products are the same and can be combined.

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = \begin{bmatrix}\frac{\partial\mathcal{L}}{\partial w_0} \\ \frac{\partial\mathcal{L}}{\partial w_1}\end{bmatrix}$$

$f(\mathbf{w})$	$\frac{\partial f}{\partial \mathbf{w}}$
$\mathbf{w}^\top\mathbf{x}$	$\mathbf{x}$
$\mathbf{x}^\top\mathbf{w}$	$\mathbf{x}$
$\mathbf{w}^\top\mathbf{w}$	$2\mathbf{w}$
$\mathbf{w}^\top\mathbf{C}\mathbf{w}$	$2\mathbf{C}\mathbf{w}$

Some useful identities

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{w}} &= \frac{2}{N}\mathbf{X}^\top\mathbf{Xw}-\frac{2}{N}\mathbf{X}^\top\mathbf{t}=0 \\ \mathbf{X}^\top\mathbf{Xw} &= \mathbf{X}^\top\mathbf{t}.\end{aligned}$$

$$\mathbf{Iw} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{t}$$

The **matrix normal equation**! (guaranteed unique solution when the n column vectors of $\mathbf{X}$ are *linearly independent*)

But what does this *mean*??

9

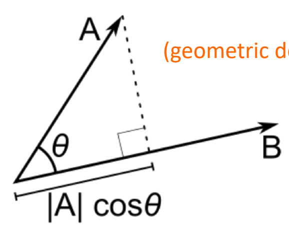
Relation of $a^{\mathsf{T}}b$ to Geometry

- **a**$^{\mathsf{T}}$**b** is special (also **a·b**), called the *dot product*
  (aka *scalar product*; the *inner product* for the Euclidean space)

$$\mathbf{a}\cdot\mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$ (algebraic def.)

- Plays a role in defining
  - Euclidean distance (norm)
  - Angles

(geometric def.)



$$\begin{aligned}\mathbf{a}\cdot\mathbf{a} &= \|\mathbf{a}\|^2 \\ \mathbf{a}\cdot\mathbf{b} &= \|\mathbf{a}\|\,\|\mathbf{b}\|\cos\theta \\ \theta &= \arccos\left(\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\,\|\mathbf{b}\|}\right).\end{aligned}$$

The dot product of vectors that are 90° (or more generally, orthogonal) is = 0

10

Geometry of Linear Systems and their Solution

A linear equation expresses a constraint between variables

A system of linear equations – more constraints!

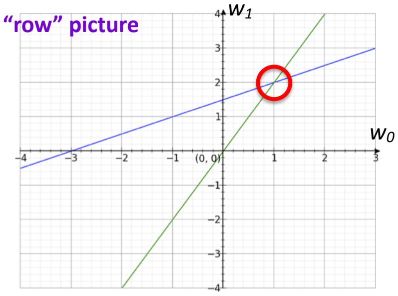
$$2w_0 - w_1 = 0$$
$$-w_0 + 2w_1 = 3$$

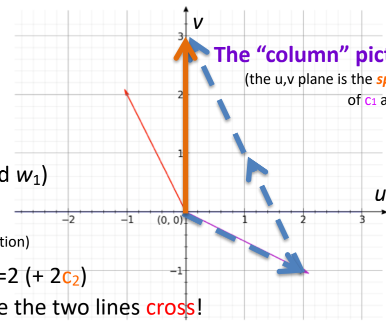
$$\begin{array}{c c} u & \\ v & \\ & \end{array}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

c_1 c_2

$\mathbf{X}\,\mathbf{w} = \mathbf{t}$ $\quad$ $\mathbf{w} = \mathbf{X}^{-1}\,\mathbf{t}$

The "row" picture





The "column" picture

(the u,v plane is the ***span*** of of c_1 and c_2)

"Solving" the linear system involves finding the linear combinations (i.e., the amounts w_0 and w_1) of c_1 and c_2 that equal the column vector $(0,3)^T$.

Solve using your favorite method (e.g., Gaussian Elimination)

Here, the solution happens to be w_0=1 (1 c_1), w_1=2 (+ 2c_2)

The w_0's and w_1's corresponds to the point where the two lines cross!

11

Geometry of Linear Systems and their Solution

But what happens if we're **over-constrained**?

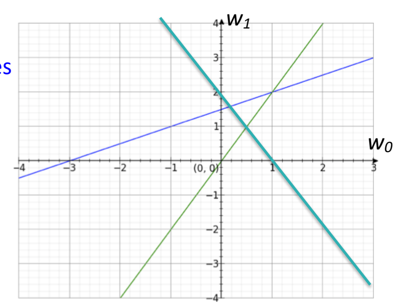
GOAL: Find a solution that is ***closest*** to (minimizes the distance between) the crossing points!

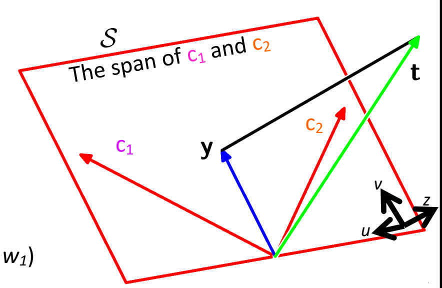
$$2w_0 - w_1 = 0$$
$$-w_0 + 2w_1 = 3$$
$$2w_0 + w_1 = 2$$

$$\begin{array}{c} u \\ v \\ z \end{array}\begin{bmatrix} 2 & -1 \\ -1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix}$$

c_1 c_2 $\quad$ $\mathbf{t}$

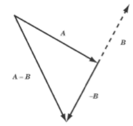
$\mathbf{X}\,\mathbf{w} = \mathbf{t}$ $\quad$ $\mathbf{X}\,\hat{\mathbf{w}} = \mathbf{y}$





"Solving" the linear system involves finding the linear combinations (i.e., the amounts w_0 and w_1) of c_1 and c_2 that equal the column vector $(0,3,2)^T$.

Finding the projection

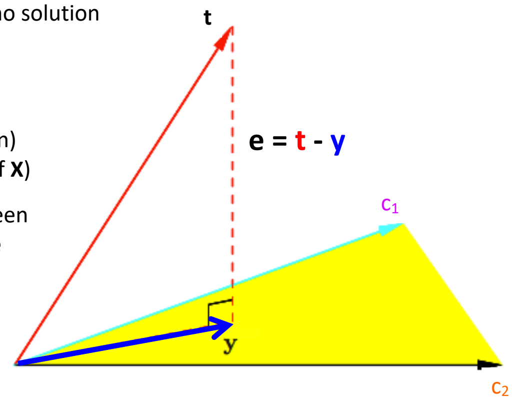


X w = **t** ...what we started with, but no solution

X ŵ = **y** ...something we can solve

In particular, we want **y** to be the ***closest*** to **t** but in the column space (the span) of $\mathbf{c}_1$ and $\mathbf{c}_2$ (which are the columns of **X**)

But we know the shortest distance between the end of **t** and the span is a vector **e** that is ***orthogonal*** (right angles!) to $\mathbf{c}_1$ and $\mathbf{c}_2$ (i.e., orthogonal to **X**)

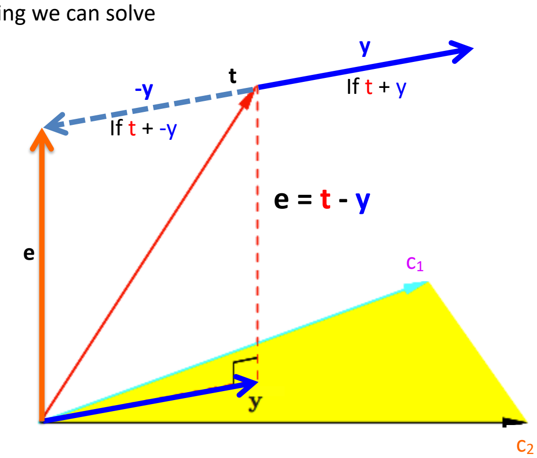


13

In case you need to remember vector subtraction...

X w = **t** ...what we started with, but no solution

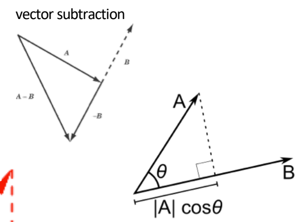
X ŵ = **y** ...something we can solve



Now we want to enforce: **e**'s <u>length</u> to be *minimal* and <u>direction</u> *orthogonal* to **X**

14

Finding the projection



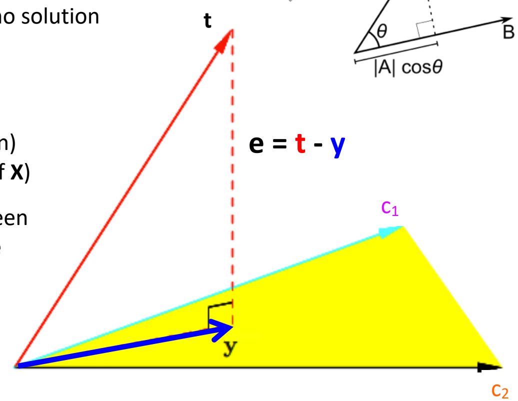
X w = **t** ...what we started with, but no solution

X $\hat{\mathbf{w}}$ = **y** ...something we can solve

In particular, we want **y** to be the ***closest*** to **t** but in the column space (the span) of $\mathbf{c}_1$ and $\mathbf{c}_2$ (which are the columns of **X**)

But we know the shortest distance between the end of **t** and the span is a vector **e** that is ***orthogonal*** (right angles!) to $\mathbf{c}_1$ and $\mathbf{c}_2$ (i.e., orthogonal to **X**)

That only happens when $\mathbf{e}^T\mathbf{X} = \mathbf{X}^T\mathbf{e} = 0$



Let's solve for when $\mathbf{X}^T\mathbf{e} = 0$

$\mathbf{X}^T(\mathbf{t} - \mathbf{y}) = \mathbf{0}$ plug in for **e**

$\mathbf{X}^T(\mathbf{t} - \mathbf{X}\hat{\mathbf{w}}) = \mathbf{0}$ substitute original $\mathbf{X}\hat{\mathbf{w}}=\mathbf{y}$ (since we don't know **y**)

$\mathbf{X}^T\mathbf{t} - \mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} = \mathbf{0}$ multiply through...

$\mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} = \mathbf{X}^T\mathbf{t}$

$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}$ ... ah hah! $\mathbf{I}\mathbf{w} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{t}$

15

Finding the projection

Parameters of hyperplane

$$2w_0 - w_1 = 0$$
$$-w_0 + 2w_1 = 3$$
$$2w_0 + w_1 = 2$$

inputs targets

X w = **t** ...what we started with, but no solution

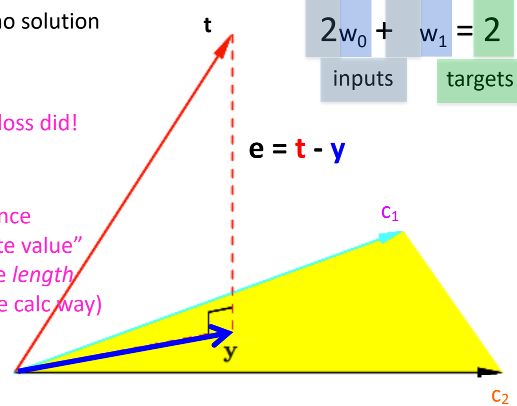
X $\hat{\mathbf{w}}$ = **y** ...something we can solve

This is just what minimizing the squared loss did!

$$\mathcal{L} = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^{\top}(\mathbf{t} - \mathbf{X}\mathbf{w})$$

This "error" vector **e** is the shortest distance
(Note: we can't differentiate the "absolute value" so the squaring of the difference – of the *length* of **e** – was a better choice for doing it the calc way)

That only happens when $\mathbf{e}^T\mathbf{X} = \mathbf{X}^T\mathbf{e} = 0$



Let's solve for when $\mathbf{X}^T\mathbf{e} = 0$

$\mathbf{X}^T(\mathbf{t} - \mathbf{y}) = \mathbf{0}$ plug in for **e**

$\mathbf{X}^T(\mathbf{t} - \mathbf{X}\hat{\mathbf{w}}) = \mathbf{0}$ substitute original $\mathbf{X}\hat{\mathbf{w}}=\mathbf{y}$ (since we don't know **y**)

$\mathbf{X}^T\mathbf{t} - \mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} = \mathbf{0}$ multiply through...

$\mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} = \mathbf{X}^T\mathbf{t}$

$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}$ ... ah hah! $\mathbf{I}\mathbf{w} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{t}$

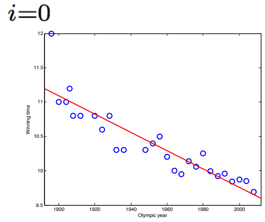
16

The Normal Equations

For model: $t = f(x_1, \ldots, x_k; w_0, \ldots, w_k) = \sum_{i=0}^{k} x_i w_i$

$$w_0 = \bar{t} - w_1 x$$

$$w_1 = \frac{\overline{xt} - \overline{x}\,\overline{t}}{\overline{x^2} - (\overline{x})^2}$$



$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{t}$$

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad \mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$