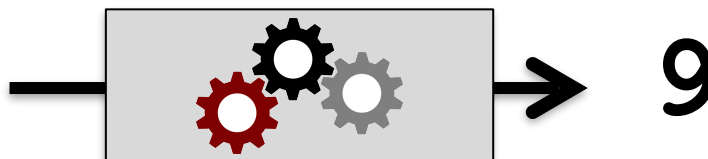
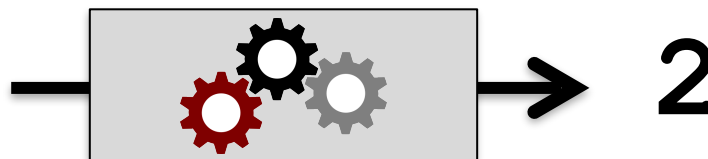


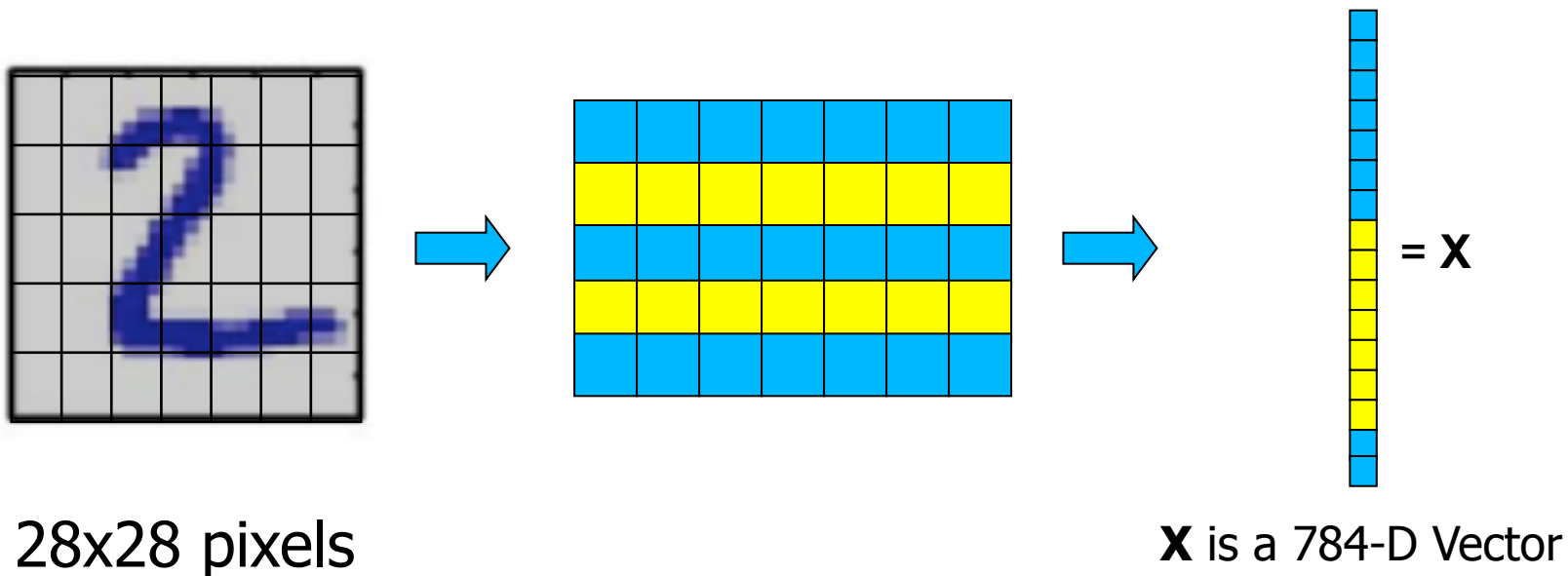
Linear Regression

Mathieu Salzmann
IC-CVLab

Reminder: Recognizing Hand-Written Digits



Reminder: Predictor and Labels

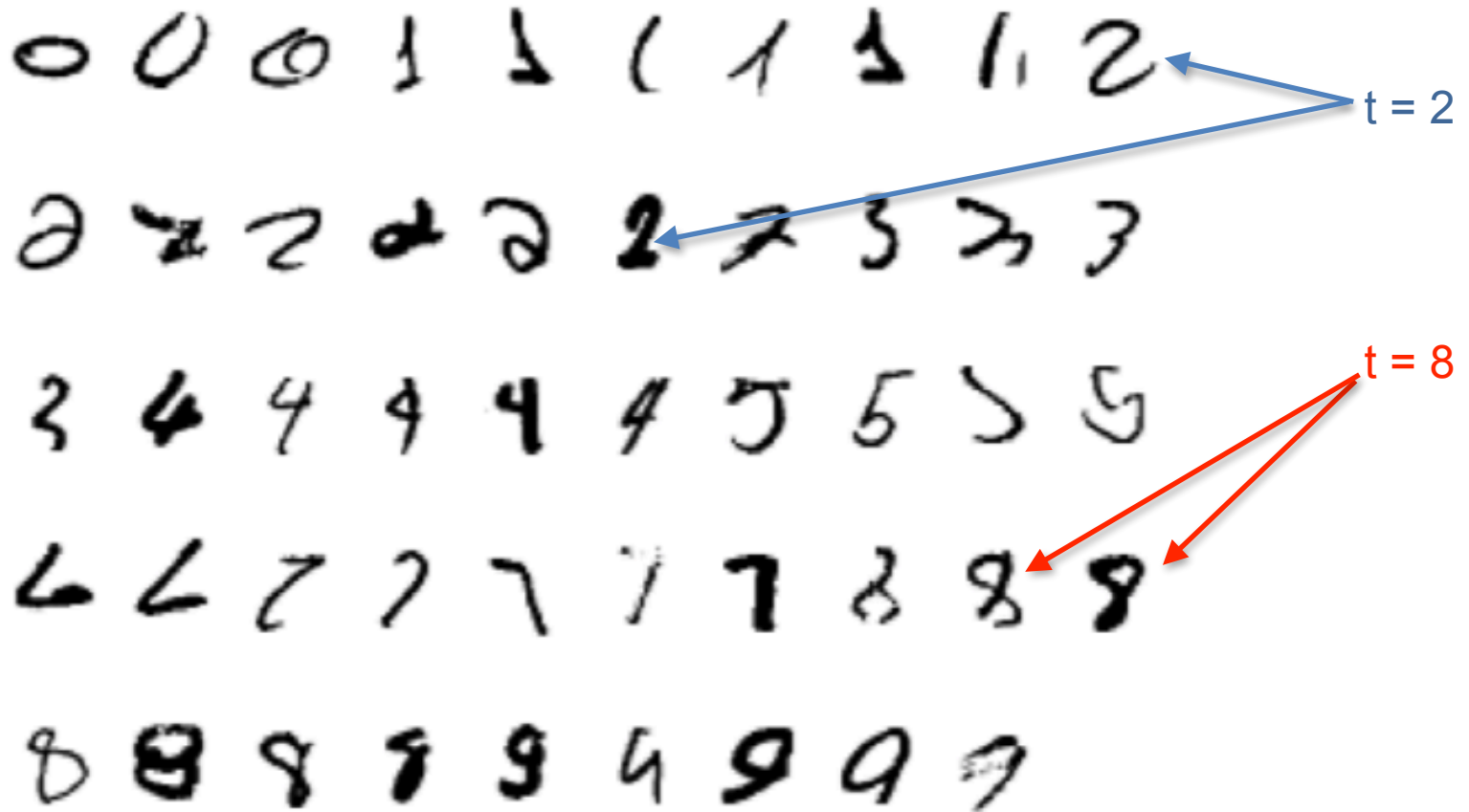


$$\boxed{y} : \mathbf{x} \in \mathbb{R}^{784} \rightarrow \boxed{\{0, 1, 2, \dots, 9\}} \quad ?$$

Predictor

Labels

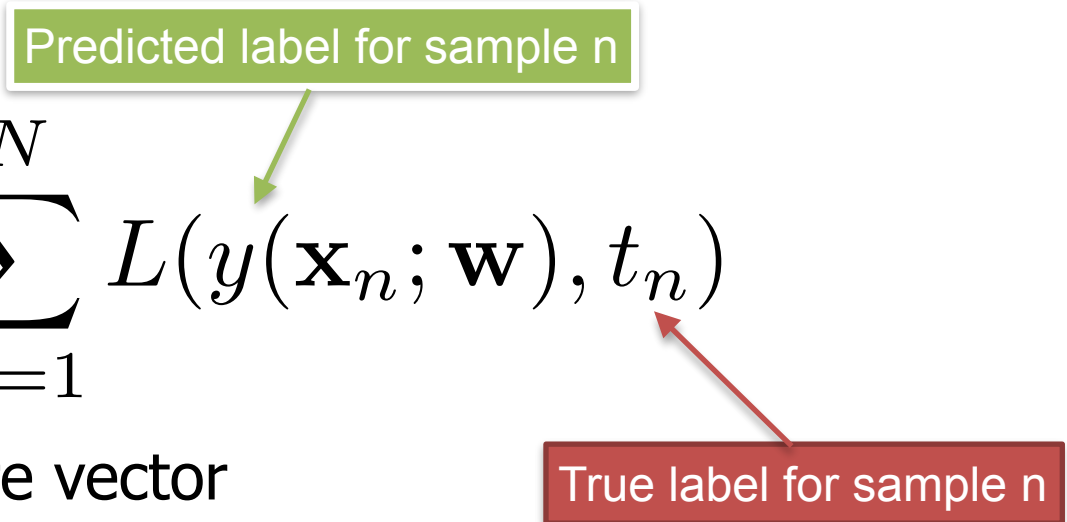
Reminder: Labeled Training Set



$$T = \{(\mathbf{x}_n, t_n) \quad \text{for} \quad 1 \leq n \leq N\}$$

Reminder: Supervised Classification

Minimize:

$$E(\mathbf{w}) = \sum_{n=1}^N L(y(\mathbf{x}_n; \mathbf{w}), t_n)$$


- **x**: Feature vector
- **w**: Model parameters
- **t**: Label
- **y**: Predictor
- **L**: Loss Function
- **E**: Error Function

—> ML is an optimization problem

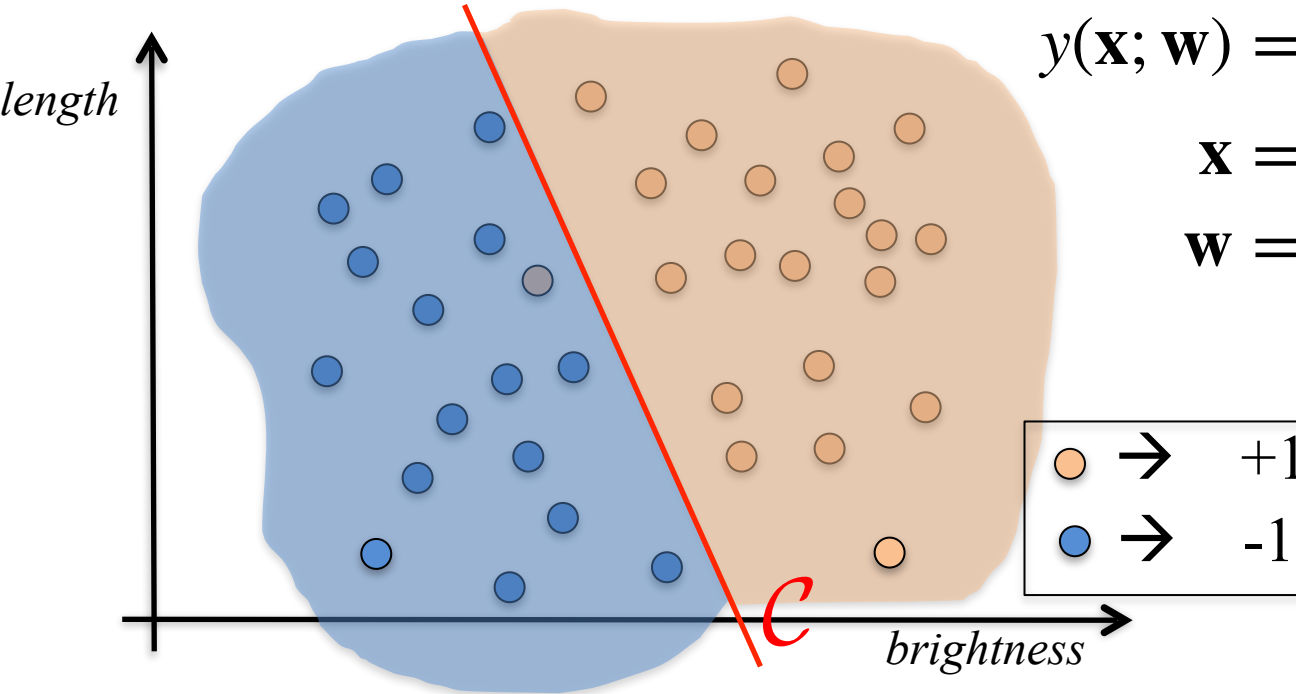
Reminder: Linear 2D Model



Some algorithm



$$\begin{pmatrix} \textit{brightness} \\ \textit{length} \end{pmatrix}$$



$$y(\mathbf{x}; \mathbf{w}) = \text{sign}(w_x b + w_y l + w_0)$$

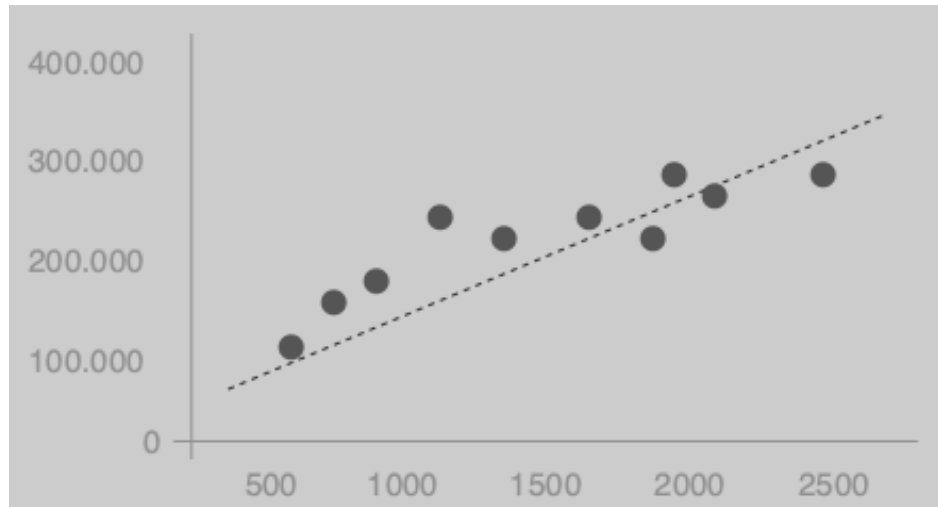
$$\mathbf{x} = [b, l]$$

$$\mathbf{w} = [w_x, w_y, w_0]$$

How do we find \mathbf{w} ?

Today: Regression

- In some cases, the value we seek to predict is not a category label
- It is rather a continuous value that follows an order

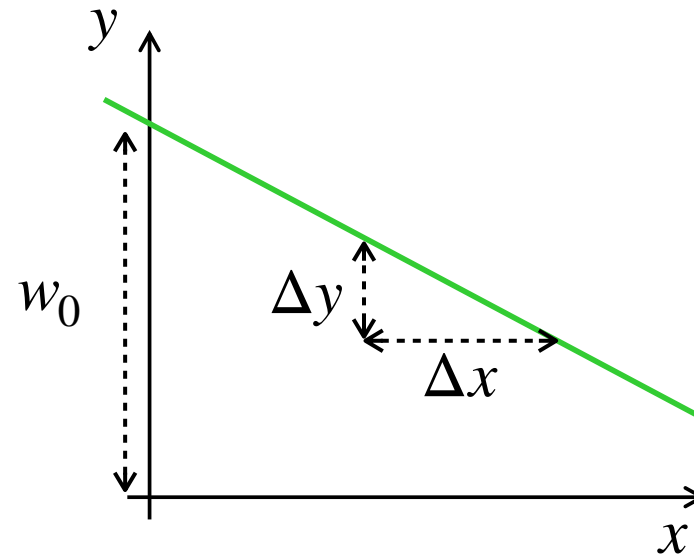


- Price of a house a function of its size

A first parametric ML model

- Let us now focus on the simplest scenario:
 - A single input dimension, i.e., $D = 1$
 - A single output dimension, i.e., $C = 1$
 - We aim to solve a regression task (the output is continuous)

A simple parametric model: The line

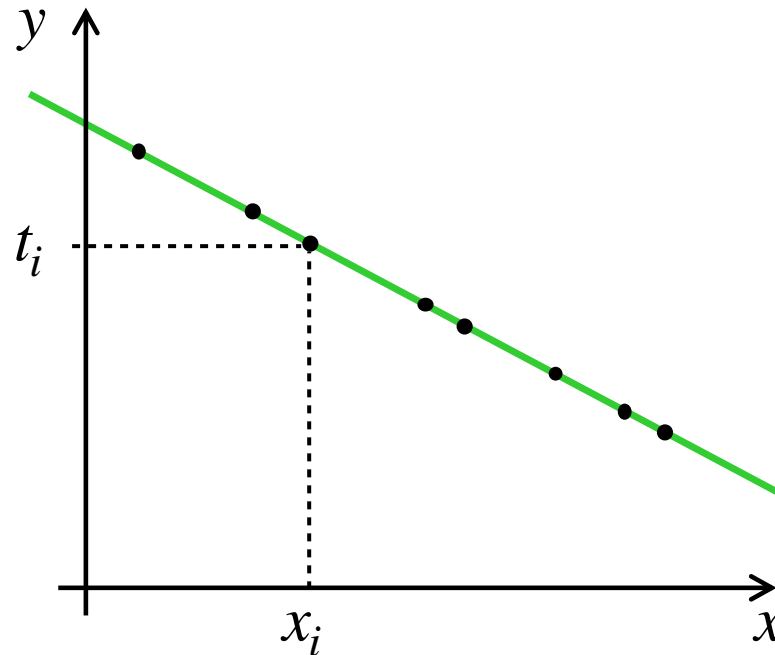


- Defined by 2 parameters
 - The y -intercept w_0
 - The slope $w_1 = \frac{\Delta y}{\Delta x}$
- Mathematically, a line is expressed as

$$y(x; \mathbf{w}) = w_0 + w_1x$$

Line fitting

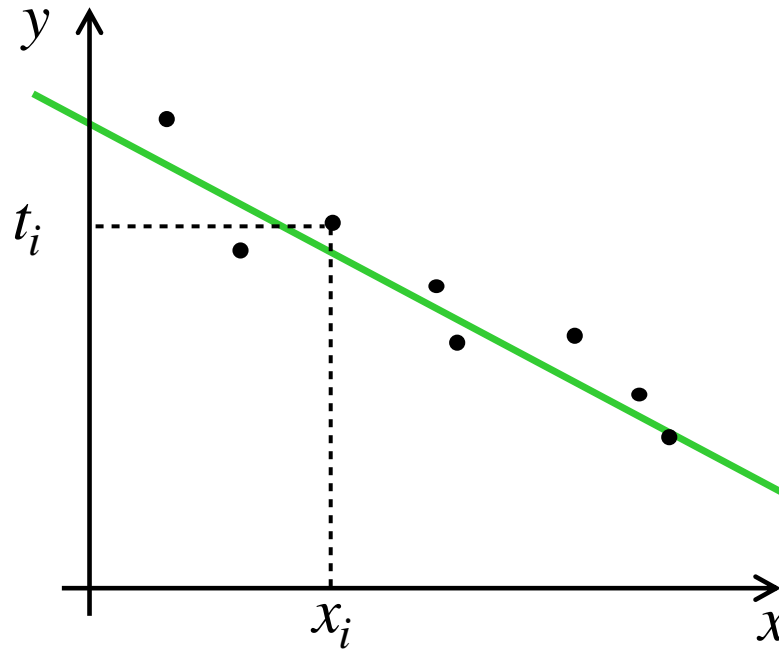
- Given N pairs $\{(x_i, t_i)\}$, find the line that passes through these observations



- This ideal case never occurs in practice

Line fitting with noise

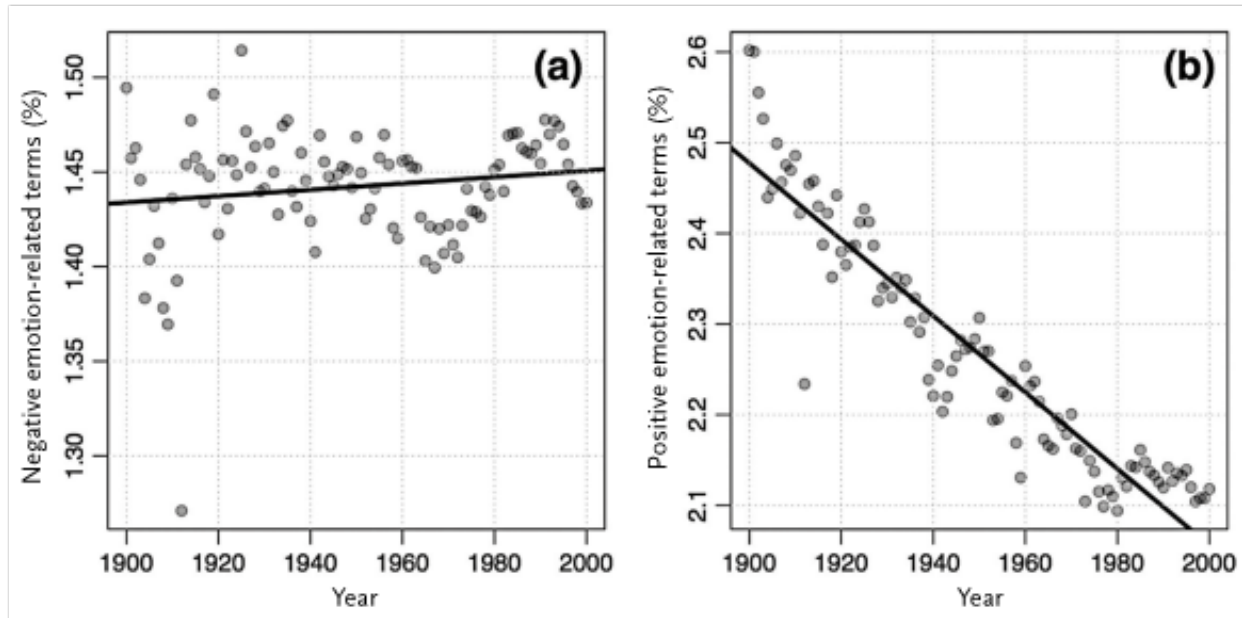
- Given N pairs $\{(x_i, t_i)\}$ of noisy measurements, find the line that best fits these observations



- This process is called linear regression

1D linear regression: Example

- Discover trends:
 - Example: Proportion of negative and positive emotions in anglophone fiction (Morin & Acerbi, 2016. Figure from Moretti & Sobchuk, 2019)



1D Linear regression: Training

- In essence, fitting a line consists of finding the best line parameters w_0^* and w_1^* for some given data
- This corresponds to the training stage:
 - Given N training pairs $\{(x_i, t_i)\}$, we aim to find (w_0, w_1) , such that the predictions of the model

$$y_i = w_0 + w_1 x_i$$

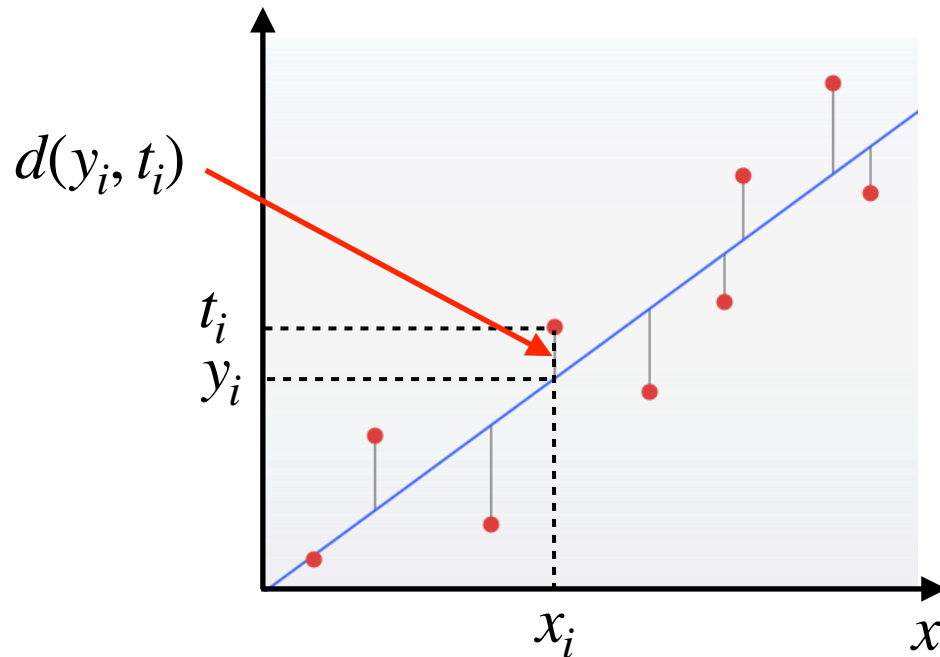
are close to the true values t_i

- We then need to define a measure of closeness, or conversely an error, between t_i and y_i

1D Linear regression: Training

- A natural measure of error is the Euclidean distance

$$d(y_i, t_i) = \sqrt{(y_i - t_i)^2}$$



- The difference between y_i and t_i is often referred to as the residual

1D Linear regression: Training

- In practice, one often prefers using the squared Euclidean distance

$$d^2(y_i, t_i) = (y_i - t_i)^2$$

- Training can then be expressed as the least-squares minimization problem

$$\min_{w_0, w_1} \frac{1}{N} \sum_{i=1}^N d^2(y_i, t_i)$$

where y_i depends on w_0 and w_1

1D linear regression: Demo

- http://digitalfirst.bfwpub.com/stats_applet/stats_applet_5_correg.html

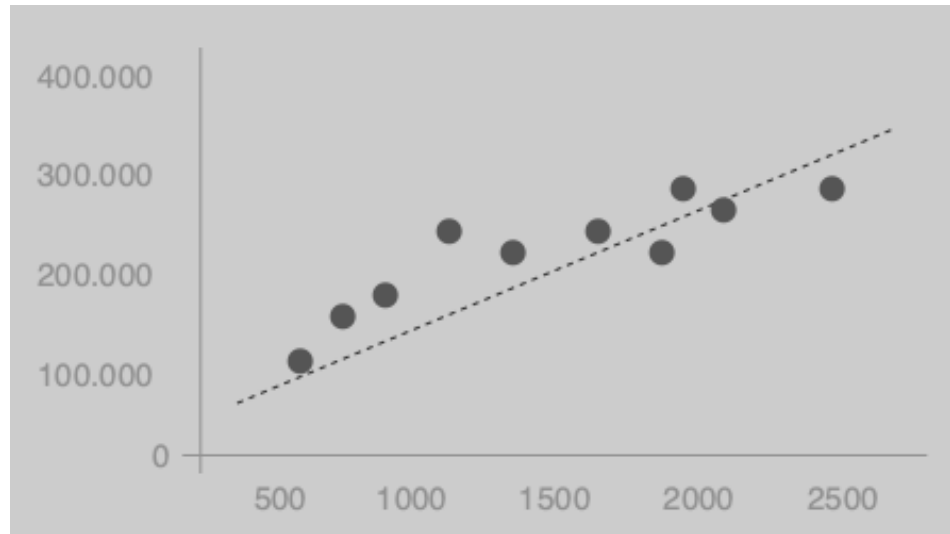
1D Linear regression: Prediction

- Once you found the best line for the given N observations, you can use it to predict a y value for a new x
- Let w_0^* and w_1^* be the best line parameters given the observations
- Then, for any test value x_t , you can predict an estimate of the corresponding y_t as

$$y_t = w_0^* + w_1^* x_t$$

1D linear regression: Example

- Predict quantities
 - Predict the price of a house based on its size (example from <https://www.internalpointers.com/post/linear-regression-one-variable>)



- With temporal trends, one can predict what will happen in the future

Dealing with multiple input dimensions

- In general, an input observation \mathbf{x}_i is not represented by a single value
 - E.g., a grayscale image can be represented by an $W \cdot H$ dimensional vector

$$\mathbf{x}_i = \text{vectorize}(\text{2}) \in \mathbb{R}^{28 \cdot 28} = \mathbb{R}^{784}$$

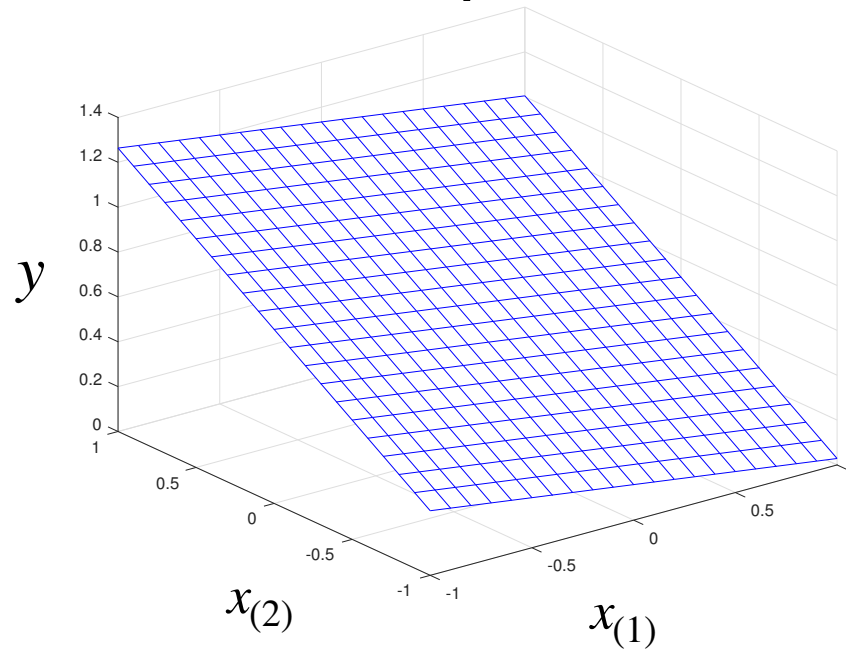
- E.g., with attribute-based representations, multiple attributes are often given

Birth weight prediction

	Age at delivery	Weight prior to pregnancy (pounds)	Smoker	Doctor visits during 1 st trimester	Race	Birth Weight (grams)
Patient 1	29	140	Yes	2	Caucasian	2977
Patient 2	32	132	No	4	Caucasian	3080
Patient 3	36	175	No	0	African-Am	3600
*	*	*	*	*	*	*
*	*	*	*	*	*	*
Patient 189	30	95	Yes	2	Asian	3147

Plane

- When there are two input dimensions, instead of a line, we can define a plane

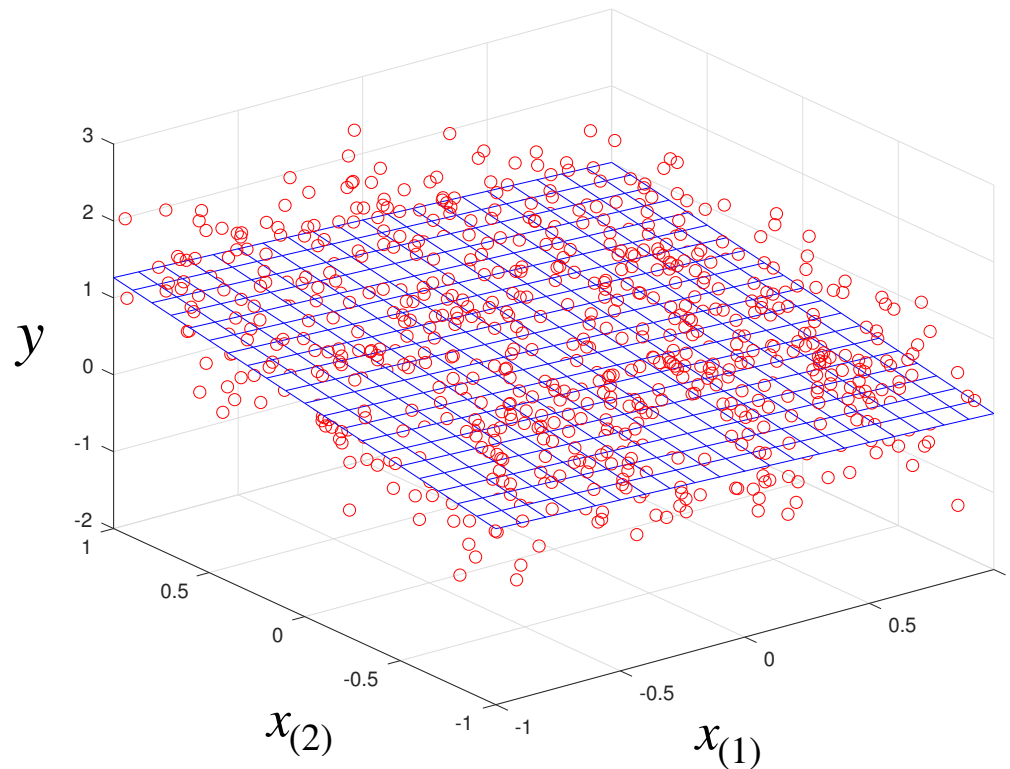


- Mathematically, a plane is expressed as

$$y = w_0 + w_1x_{(1)} + w_2x_{(2)} = \mathbf{w}^T \begin{bmatrix} 1 \\ x_{(1)} \\ x_{(2)} \end{bmatrix}$$

Plane fitting

- Given N noisy pairs $\{(\mathbf{x}_i, t_i)\}$, where $\mathbf{x}_i \in \mathbb{R}^2$, find the plane that best fits these observations



Hyperplane

- This can be generalized to higher dimensions
- In dimension D , we can write

$$y = w_0 + w_1x_{(1)} + w_2x_{(2)} + \dots + w_Dx_{(D)} = \mathbf{w}^T \begin{bmatrix} 1 \\ x_{(1)} \\ x_{(2)} \\ \vdots \\ x_{(D)} \end{bmatrix}$$

- Ultimately, whatever the dimension, we can write

$$y = \mathbf{w}^T \mathbf{x}$$

with $\mathbf{x} \in \mathbb{R}^{D+1}$, where the extra dimension contains a 1 to account for w_0

Multi-input linear regression: Training

- Because the output remains 1D, we can use the same least-square loss function as before, and write training as

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^N d^2(y_i, t_i) \Leftrightarrow \min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^N (\mathbf{w}^T \mathbf{x}_i - t_i)^2$$

- Note that this has the same form as in the 1D input case
- Solving this minimization problem involves computing its derivatives w.r.t. the model parameters (\mathbf{w})
 - This will be discussed in detail in a future lecture on optimization

Multi-input linear regression: Training

- In short, there are two approaches to solving the minimization problem
 1. Gradient descent: Iterative update of the parameters based on the gradient of the function
 - This is generally applicable to many ML training problems
 2. Closed-form solution: Express the optimal parameters in an algebraic form
 - This is much less generally applicable, but it does apply to linear regression

Linear regression: Closed-form Solution

- The closed-form solution is obtained by writing the equation

$$\nabla_{\mathbf{w}} E(\mathbf{w}) = \mathbf{0}$$

where $E(\mathbf{w})$ is the error function we aim to minimize

- Expanding this gives

$$\left(\sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T \right) \mathbf{w}^* = \sum_{i=1}^N \mathbf{x}_i t_i$$

$(D + 1) \times (D + 1)$ matrix $(D + 1) \times 1$ vector $(D + 1) \times 1$ vector

Linear regression: Matrix form

- Let us now group all $\{\mathbf{x}_i\}$ and all $\{t_i\}$ in a matrix and a vector

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_{1,(1)} & x_{1,(2)} & \cdots & x_{1,(D)} \\ 1 & x_{2,(1)} & x_{2,(2)} & \cdots & x_{2,(D)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N,(1)} & x_{N,(2)} & \cdots & x_{N,(D)} \end{bmatrix} \in \mathbb{R}^{N \times (D+1)}$$

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} \in \mathbb{R}^N$$

Linear regression: Closed-form Solution

- Then, we can re-write the solution as

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

- This finally gives us

$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t} = \mathbf{X}^\dagger \mathbf{t}$$

where \mathbf{X}^\dagger is known as the Moore-Penrose pseudo-inverse of \mathbf{X}

- In other words, we can obtain the solution to linear regression algebraically

Linear regression: Test time

- As in the 1D input case, once we have the optimal parameters \mathbf{w}^* , we can predict y_t for any new test input \mathbf{x}_t
- The predicted value is given by

$$y_t = (\mathbf{w}^*)^T \begin{bmatrix} 1 \\ \mathbf{x}_t \end{bmatrix}$$

Linear regression: Demo

- [https://playground.tensorflow.org/
#activation=linear&batchSize=10&dataset=circle®Dataset=reg-plane&learningRate=0.03®ularizationRate=0&noise=0&networkShape=&seed=0.45772&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=regression&initZero=false&hideText=false](https://playground.tensorflow.org/#activation=linear&batchSize=10&dataset=circle®Dataset=reg-plane&learningRate=0.03®ularizationRate=0&noise=0&networkShape=&seed=0.45772&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=regression&initZero=false&hideText=false)

Model evaluation

- Once an ML model is trained, one would typically understand how well it performs on unseen test data
 - At this stage, the parameters of the model are fixed
 - Recall that the training and testing data must be separated!
- During this evaluation, one compares the predictions of the model with the true annotations of the test data
 - In contrast to the training stage, the model parameters are not updated
 - The evaluation metric may directly be the loss function, but may also differ from it

Evaluation metrics for regression

- Mean Squared Error (MSE)
 - Same as the loss function but for N_t test samples

$$MSE = \frac{1}{N_t} \sum_{i=1}^{N_t} (y_i - t_i)^2$$

where y_i is the prediction for test sample i and t_i the corresponding ground-truth value

- Root Mean Squared Error (RMSE)
 - Square-root of the MSE

Evaluation metrics for regression

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{N_t} \sum_{i=1}^{N_t} |y_i - t_i|$$

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{N_t} \sum_{i=1}^{N_t} \left| \frac{y_i - t_i}{t_i} \right|$$

Taking a percentage w.r.t. the true value might be easier to interpret

Interlude

Interpreting a Linear Model

Linear regression: Example

- UCI Wine Quality dataset:
 - Predict the quality of wine based on several attributes (5 samples shown below)

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

- Final RMSE:
 - 0.65 for training data
 - 0.63 for test data

Example from <https://medium.com/datadriveninvestor/regression-from-scratch-wine-quality-prediction-d61195cb91c8>

Linear regression: Example

- One can then look at the coefficient values (i.e., the $\{w_i\}$) to see the influence of each attribute

	Coefficient
fixed acidity	0.017737
volatile acidity	-0.992560
citric acid	-0.139629
chlorides	-1.590943
free sulfur dioxide	0.005597
total sulfur dioxide	-0.003520
density	0.768590
pH	-0.437414
sulphates	0.812888
alcohol	0.301484

Linear regression: Example

- Author age prediction from text
 - N'guyen et al., Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, 2011
 - Influence of features (words)

Younger people

like	-1.295
gender-male	-0.539
LIWC-School	-0.442
just	-0.354
LIWC-Anger	-0.303
LIWC-Cause	-0.290
mom	-0.290
so	-0.271
definitely	-0.263
LIWC-Negemo	-0.256

Older people

years	0.601
POS - dt	0.485
LIWC - Incl	0.483
POS - prp vbp	0.337
granddaughter	0.332
grandchildren	0.293
had	0.277
daughter	0.272
grandson	0.245
ah	0.243

Linear regression: Example

- Example: True age 17, predicted age 16.48

“I can’t sleep, but this time I have school tommorow, so I have to try I guess. My parents got all pissed at me today because I forgot how to do the homework [...]. Really mad, I ended it pissing off my mom and [...] NOTHING! Damn, when I’m at my cousin’s I have no urge to use the computer like I do here, [...]”

- Example: True age 70, predicted age 71.53

“[...] I was a little bit fearful of having surgery on both sides at once (reduction and lift on the right, tissue expander on the left) [...] On the good side, my son and family live near the plastic surgeon’s office and the hospital, [...], at least from my son and my granddaughter [...]”

Interpreting a linear model

- Warning: The magnitude of a coefficient will depend on the magnitude of the corresponding feature/attribute
 - A coefficient might be very small simply to compensate for the fact that the range of the feature is very large
 - E.g., looking at 2 attributes from the UCI Wine Quality example

chlorides	free sulfur dioxide
0.076	11.0
0.098	25.0
0.092	15.0

- This can be addressed by normalizing the data