Introduction to machine learning I

- Example of ML in computer vision
  - Pedestrian detection
  - Pipeline of visual recognition
- Regression
  - Linear model
  - Fitting a model to data; least squares
  - Statistical interpretation: maximum likelihood estimation

Outline of this course (computer vision)

Model-based vision

1. Camera models
   - calibration
2. Two-view geometry
3. Multi-view 3D reconstruction
4. Numerical computation
   - DLT
   - robust method (RANSAC)
   - statistical inference
   - optimization (BA)

Exemplar-based vision

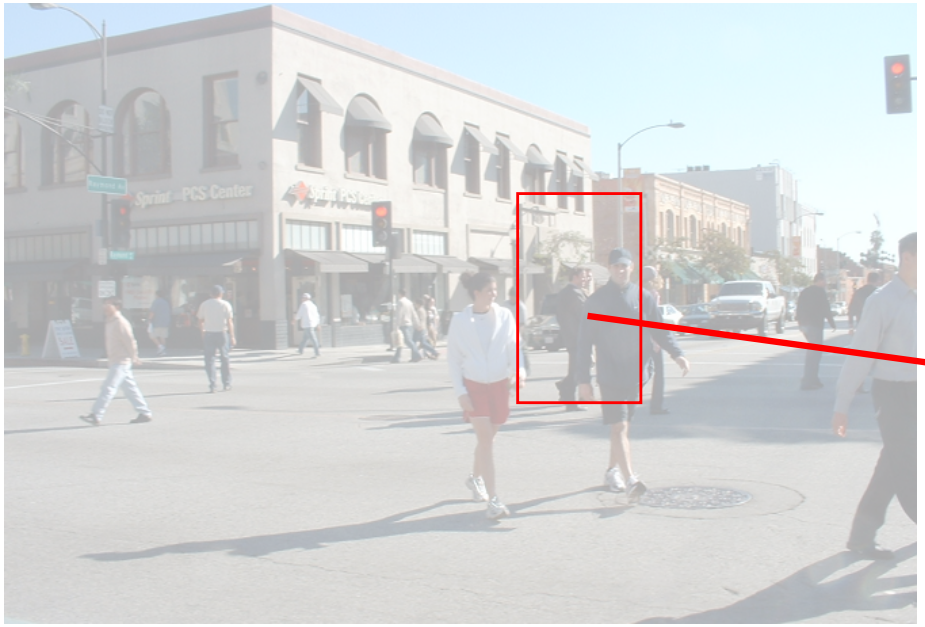
1. Introduction to machine learning I
2. Introduction to machine learning II
3. Feed-forward neural networks
4. Training neural networks
5. Convolutional neural networks
6. Feature representation and transfer learning
7. Recurrent neural networks

Example: pedestrian detection

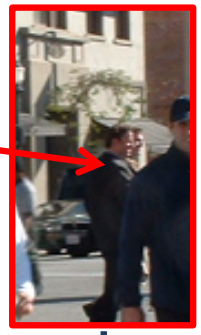
Example: pedestrian detection

Example: pedestrian detection

Example: pedestrian detection



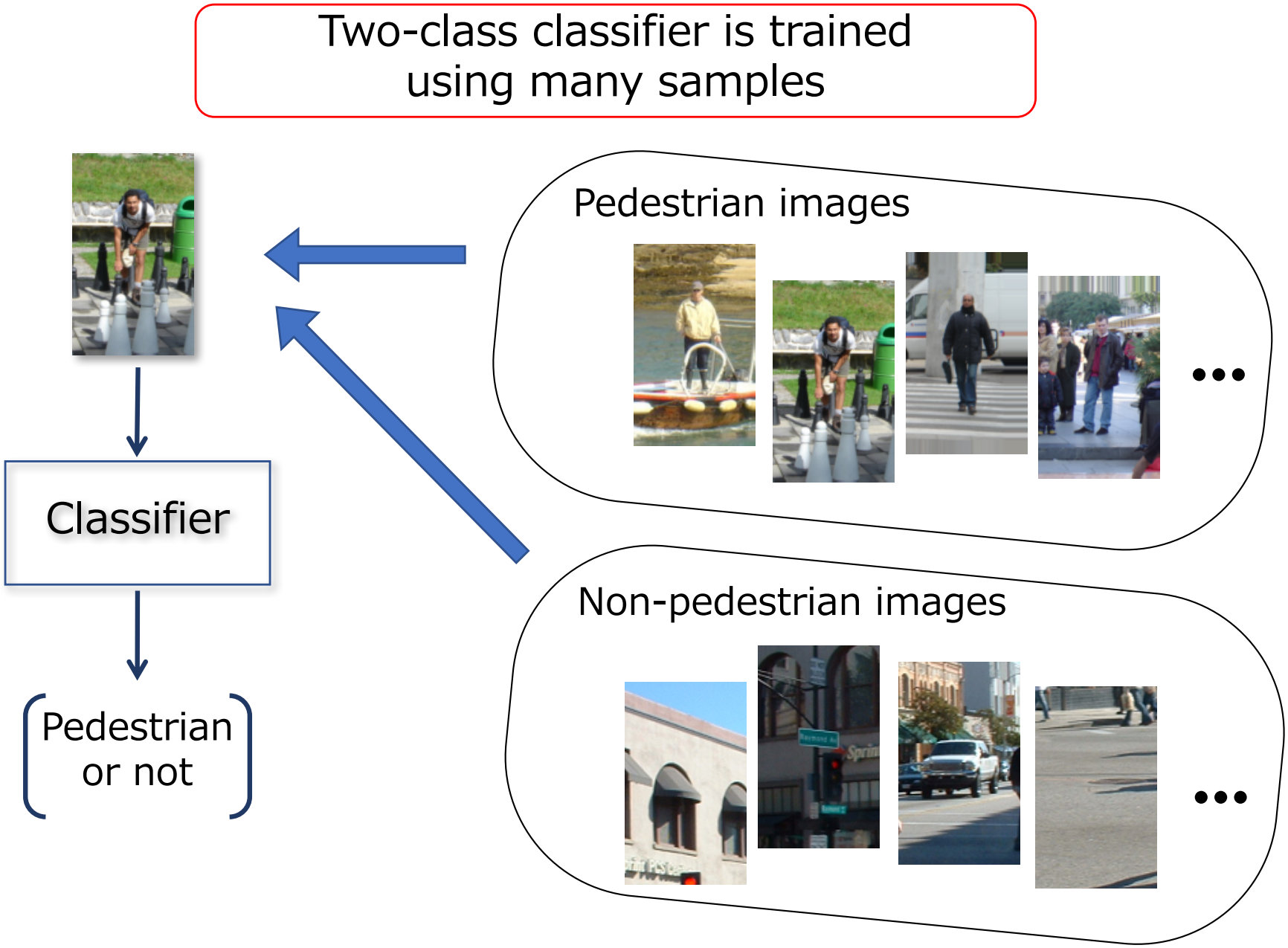
Judge if pedestrian inside a cropped subimage



classifier

pedestrian or not

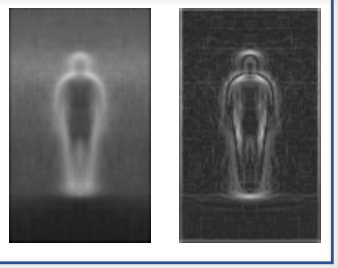
Example: pedestrian detection

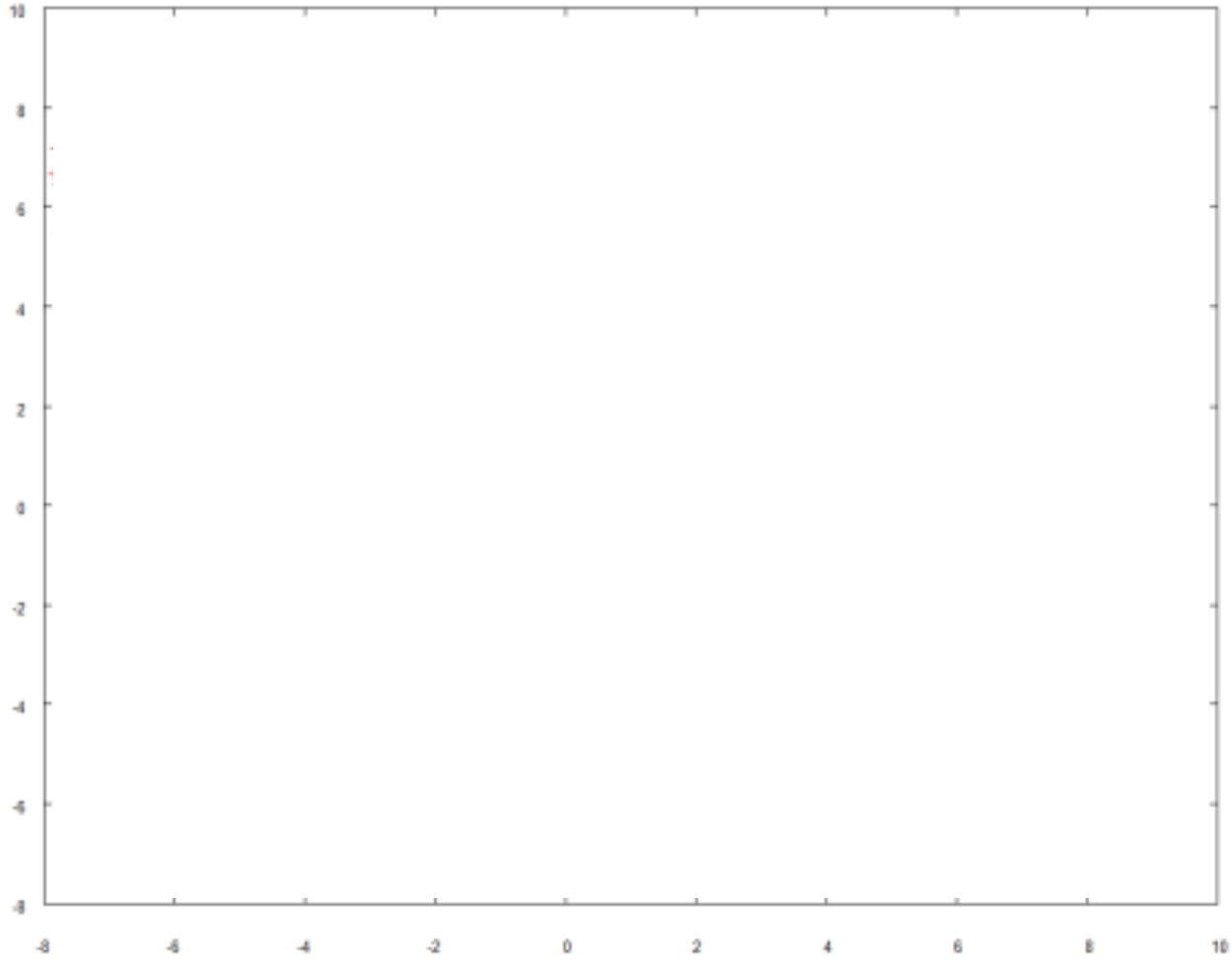


Example: pedestrian detection

Feature extraction : extracts what makes it pedestrian

Feature extraction

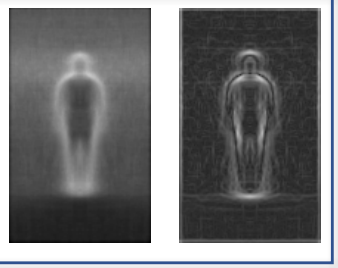


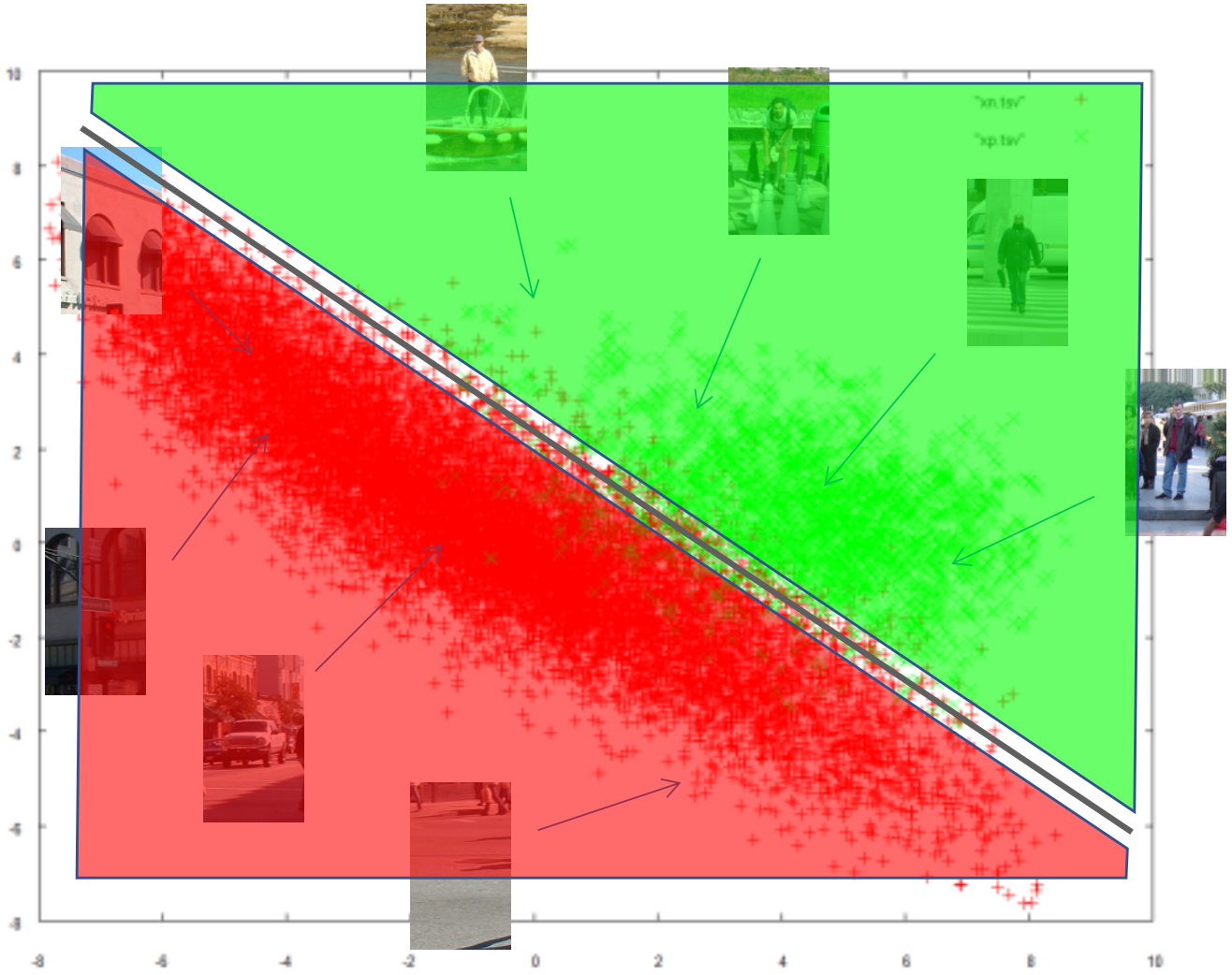


Example: pedestrian detection

Feature extraction : extracts what makes it pedestrian

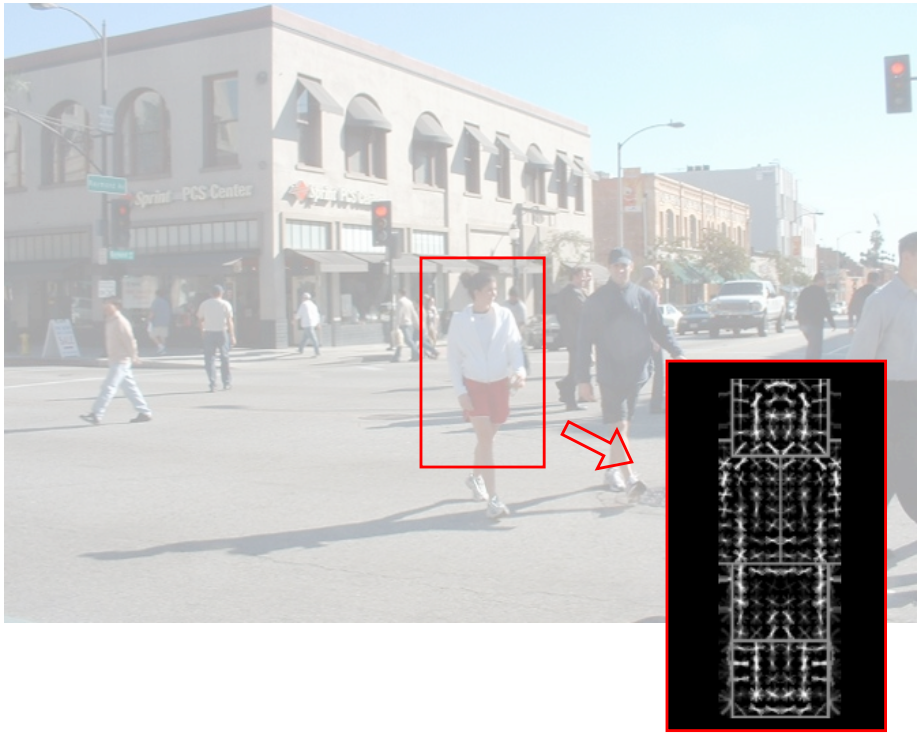
Feature extraction

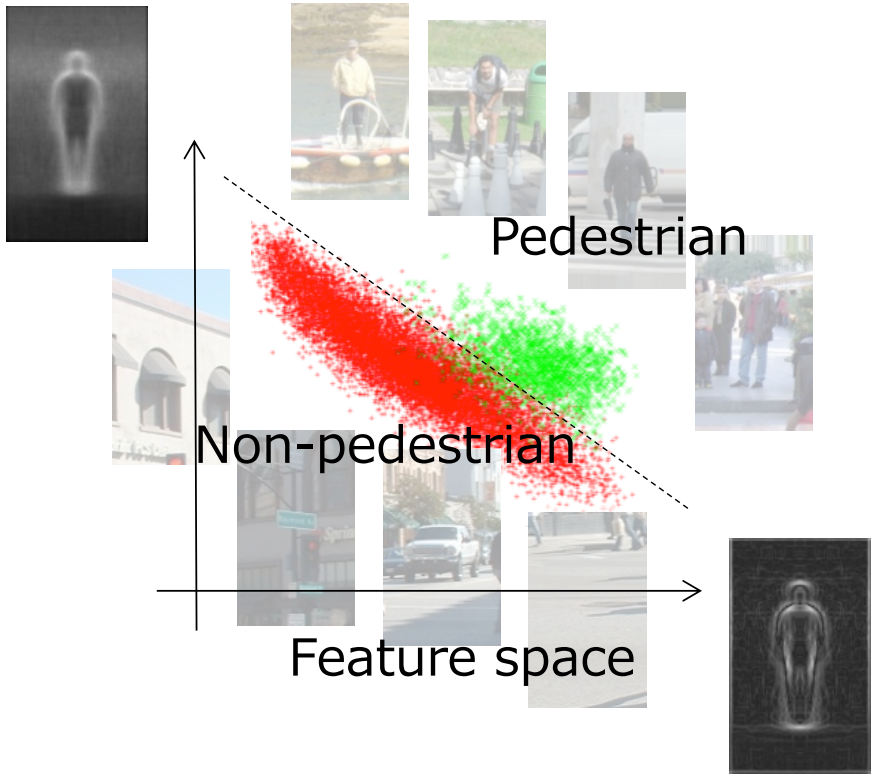




Example: pedestrian detection

Standard pipeline of visual recognition





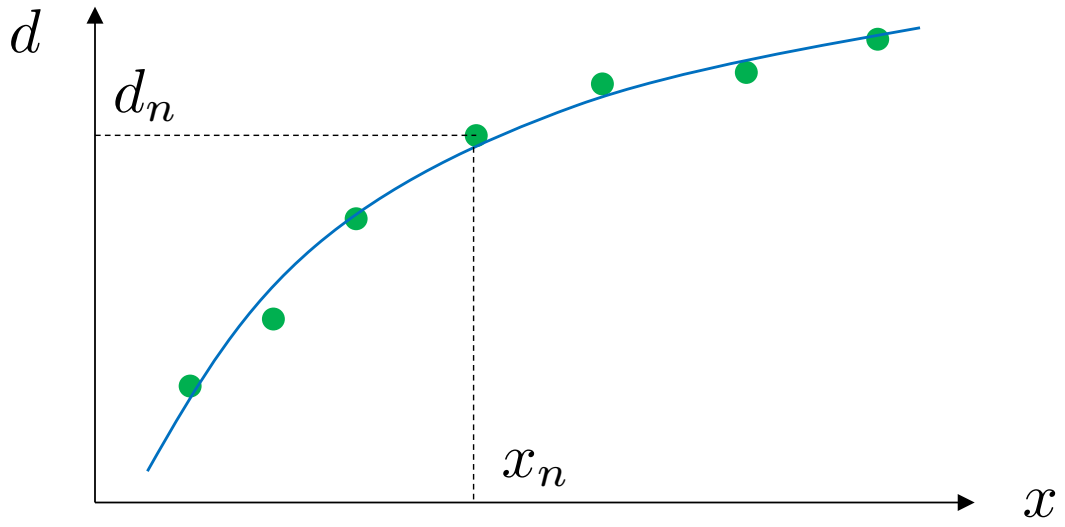
Regression

- Suppose we have N pairs of samples (data)

$$\{\mathbf{x}_n\}(n = 1, \ldots, N) \qquad \{d_n\}(n = 1, \ldots, N)$$

- Regression is to predict the value *d* for a new input *x*
  - To do this we wish to have a function

$$y(\mathbf{x}_n) \sim d_n$$



Linear regression

- Linear regression model: represents an output by a linear sum of inputs

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_I x_I \qquad y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$$

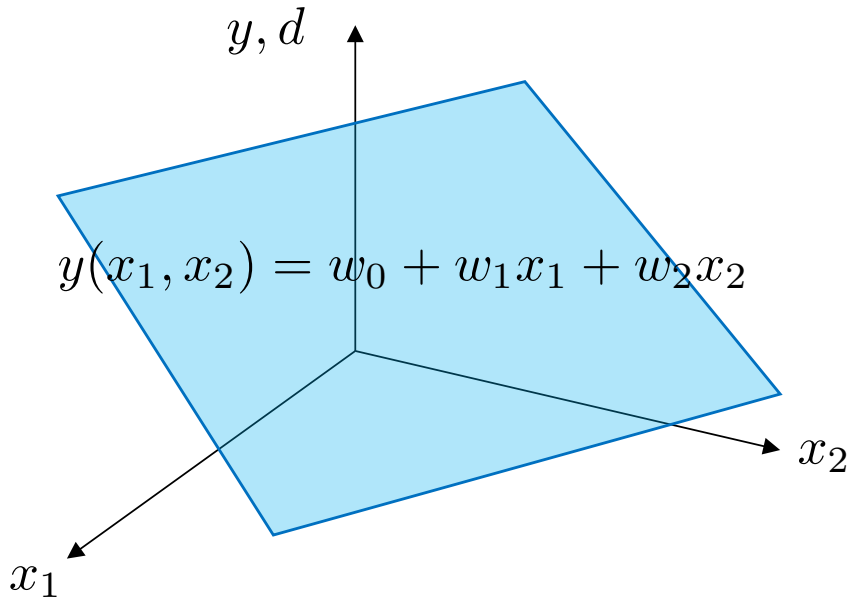
$$\left(\quad \mathbf{x} = [x_1, \ldots, x_I]^\top \qquad \mathbf{w} = [w_0, w_1, \ldots, w_I]^\top \quad \right)$$

- When the target (output) is multi-dimensional

$$\mathbf{d}_n = [d_{n1}, \ldots, d_{nJ}]^\top$$

$$\mathbf{y}(\mathbf{x}) = [y_1(\mathbf{x}), \ldots, y_J(\mathbf{x})]^\top$$

$$\left(\quad y_j(\mathbf{x}) = y_j(\mathbf{x}, \mathbf{w}_j) \quad \right)$$



Fitting a model (=training/learning)

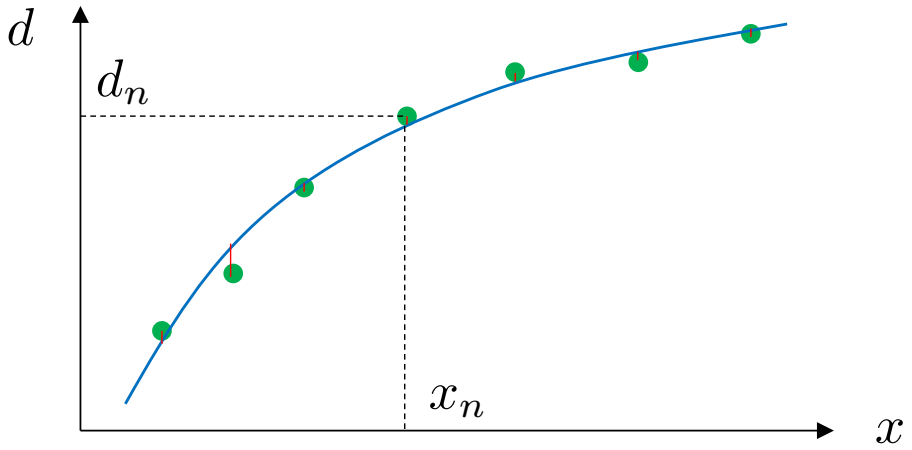
- *N* pairs of observation and its associated target value are given

$$\{\mathbf{x}_n\}(n = 1, \ldots, N) \qquad \{d_n\}(n = 1, \ldots, N)$$

- We want to have **w** such that the model best explains the *N* pairs of samples

- We choose the value **w** minimizing the sum of squared errors

$$E(\mathbf{w}) = \sum_{n=1}^{N} (d_n - y(\mathbf{x}_n, \mathbf{w}))^2$$



Computing the optimal parameter

- Linear least squares

$$E(\mathbf{w}) = \sum_{n=1}^{N} (d_n - y(\mathbf{x}_n, \mathbf{w}))^2 \qquad y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$$

$$(d_n - y(\mathbf{x}_n, \mathbf{w}))^2 = \left(d_n - \mathbf{w}^\top \begin{bmatrix} 1 \\ \mathbf{x}_n \end{bmatrix} \right)^2$$

$$= \sum_{n=1}^{N} \left(d_n - \begin{bmatrix} 1 & \mathbf{x}_n^\top \end{bmatrix} \mathbf{w} \right)^2$$

$$= \left\| \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} - \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_N^\top \end{bmatrix} \mathbf{w} \right\|^2$$

$$\|\mathbf{x}\| \equiv \sqrt{\sum_{i=1}^{I} x_i^2}$$

Computing the optimal parameter

$$E(\mathbf{w}) = \left\| \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} - \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_N^\top \end{bmatrix} \mathbf{w} \right\|^2$$

w minimizing the above is given by:

$$\hat{\mathbf{w}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{d}$$

$$\mathbf{d} \equiv \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} \qquad \mathbf{A} \equiv \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_N^\top \end{bmatrix}$$

$\hat{\mathbf{w}}$ is a stationary point of $E(\mathbf{w})$

$$E(\mathbf{w}) = (\mathbf{d} - \mathbf{A}\mathbf{w})^\top (\mathbf{d} - \mathbf{A}\mathbf{w}) \quad \Rightarrow \quad \frac{dE}{d\mathbf{w}} = -2\mathbf{A}^\top (\mathbf{d} - \mathbf{A}\mathbf{w}) = \mathbf{0}$$

Interpretation from statistical point of view: Maximum likelihood estimation

- Assuming the target *d* for a given **x** is generated as follows:

$$d = y(\mathbf{x}, \mathbf{w}) + \varepsilon$$

  - where ε is a random noise with a Gaussian distribution

$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

- The posterior density of *d* when x has been observed is given by

$$p(d_n \,|\, \mathbf{x}_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d_n - y(\mathbf{x}_n, \mathbf{w}))^2)}{2\sigma^2}\right)$$

- Joint posterior distribution of all *N* samples

$$p(d_1, \ldots, d_N \,|\, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(d_n \,|\, \mathbf{x}_n)$$

Interpretation from statistical point of view: Maximum likelihood estimation

- Given *N* pairs of an observation and target value, we wish to determine w such that the model best explains them

- If we employ maximum likelihood estimation, the answer is to choose **w** maximizing the likelihood of the data (*N* samples)

$$l(\mathbf{w}) \equiv p(d_1, \ldots, d_N \,|\, \mathbf{x}_1, \ldots, \mathbf{x}_N; \mathbf{w}) = \prod_{n=1}^{N} p(d_n \,|\, \mathbf{x}_n; \mathbf{w})$$

- Equivalent to minimization of its negative log-likelihood, which is more convenient

$$-\log l(\mathbf{w}) = -\sum_{n=1}^{N} \log p(d_n \,|\, \mathbf{x}_n; \mathbf{w})$$

$$= \frac{1}{2\sigma^2} \sum_{n=1}^{N} (d_n - y(\mathbf{x}_n, \mathbf{w}))^2 + \text{const.}$$

$E(\mathbf{w})$

$$\log\{A \exp(B/C)\} = \log A + \log \exp(B/C) = \log A + B/C$$