Machine Learning

CSE 142

Xin (Eric) Wang

Monday, October 25, 2021

Today

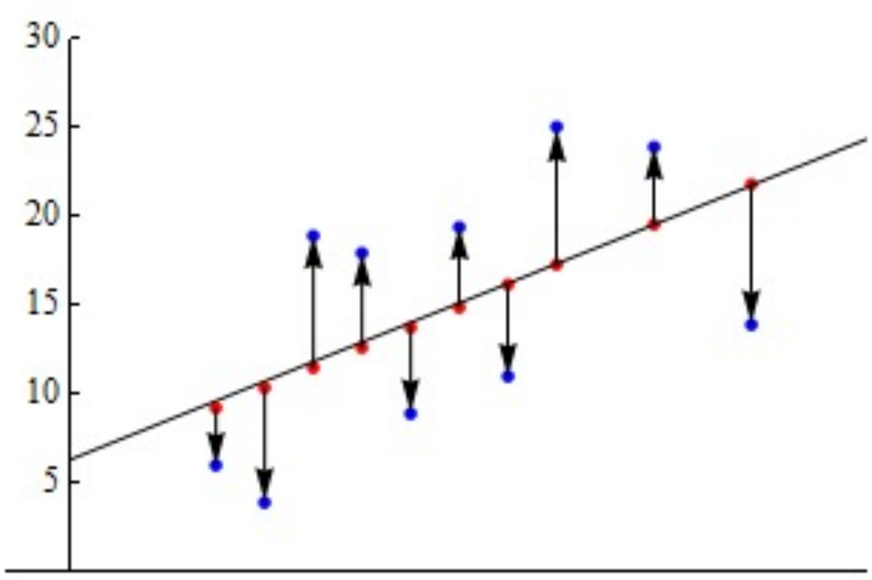
- Linear learning models (cont.)
    - ✓ Multivariate linear regression

Notes

- Sample midterm questions & answers are released on the Canvas website
- HW #2 problem 2
  - Distance from a point to a plane:
    - Plane: $f(x, y, z) = ax + by + cz + d = 0$, Point: (x_1, y_1, z_1)
      - Does $f(x_1, y_1, z_1)$ give you the distance from point to plane?
      - No – you must divide by $\|(a, b, c)\| = (a^2 + b^2 + c^2)^{1/2}$

Linear least-squares regression example

- We wish to find the relationship between the height and weight of adults
  - Data: n measurements, $(h_i, w_i) \to (input, output)$
  - Parametric linear model: $w = a + bh$ ⇨ $w_i = a + bh_i + \epsilon_i$
  - Residual: $\epsilon_i = w_i - (a + bh_i)$
  - Find (a, b) that minimizes $\sum_i [w_i - (a + bh_i)]^2$ on the training data



Linear least-squares regression example

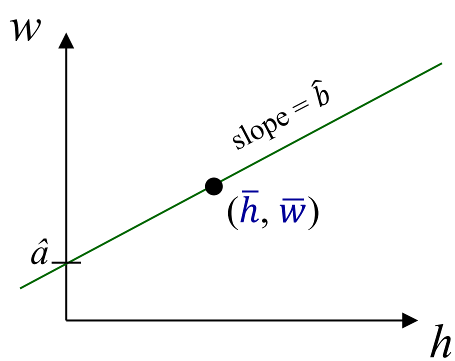
- To minimize $\sum_i [w_i - (a + bh_i)]^2$, set the partial derivatives (wrt a and b) to zero and solve for a and b

$$\frac{\partial}{\partial a}\sum_{i=1}^{n}(w_i-(a+bh_i))^2=-2\sum_{i=1}^{n}(w_i-(a+bh_i))=0 \qquad \Rightarrow \hat{a}=\overline{w}-\hat{b}\overline{h}$$

$$\frac{\partial}{\partial b}\sum_{i=1}^{n}(w_i-(a+bh_i))^2=-2\sum_{i=1}^{n}(w_i-(a+bh_i))h_i=0 \qquad \Rightarrow \hat{b}=\frac{\sum_{i=1}^{n}(h_i-\overline{h})(w_i-\overline{w})}{\sum_{i=1}^{n}(h_i-\overline{h})^2}$$

- So the regression model is $\boxed{w=\hat{a}+\hat{b}h=\overline{w}+\hat{b}(h-\overline{h})}$

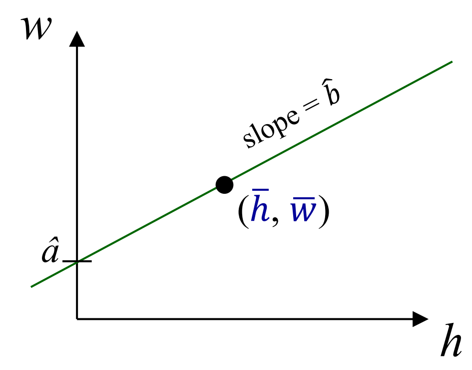
Note that the regression line goes through $(\bar{h}, \bar{w})$



The regression coefficient

- The slope ($\hat{b}$) is the regression coefficient

$$\hat{b} = \frac{\sum_{i=1}^{n}(h_i - \overline{h})(w_i - \overline{w})}{\sum_{i=1}^{n}(h_i - \overline{h})^2} = \frac{n\sigma_{hw}}{n{\sigma_h}^2} = \frac{\sigma_{hw}}{{\sigma_h}^2}$$



- In general, the regression coefficient for a feature x and a target variable y is

$$\hat{b} = \frac{\sigma_{xy}}{{\sigma_x}^2}$$

covariance(x, y)

variance(x)

- We often simplify the problem by first normalizing the data
  - Find the data averages $(\overline{h}, \overline{w})$
  - Subtract the averages from the data: $h_i \leftarrow h_i - \overline{h}$
    $w_i \leftarrow w_i - \overline{w}$

- This makes $\hat{a} = 0$, so we're just left with estimating the regression coefficient $\hat{b}$

Multivariate linear regression

- Most linear regression problems involve multiple (N) input variables, ***x***
  - E.g., estimate a patient's cholesterol level from N (N > 1) input variables
- In multivariate LR, there are N+1 regression parameters

- Linear regression function:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \qquad y(x_2, x_1) = w_2x_2 + w_1x_1 + w_0$$

- Using homogeneous coordinates:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} \qquad y(x_2, x_1, x_0) = w_2x_2 + w_1x_1 + w_0x_0$$

$x_0 = 1$

Multivariate linear regression

Linear regression equations:

$x_{i0} = 1$ (homogeneous notation)

Univariate

$$y_i = w_1 x_i + w_0 + \epsilon_i \Rightarrow$$

Multivariate

$$y_i = w_2 x_{i2} + w_1 x_{i1} + w_0 x_{i0} + \epsilon_i$$

$$\begin{aligned} y_1 &= w_2 x_{12} + w_1 x_{11} + w_0 + \boldsymbol{\epsilon}_1 \\ y_2 &= w_2 x_{22} + w_1 x_{21} + w_0 + \boldsymbol{\epsilon}_2 \\ y_3 &= w_2 x_{32} + w_1 x_{31} + w_0 + \boldsymbol{\epsilon}_3 \\ &\ldots \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} y_1 &= w_2 x_{12} + w_1 x_{11} + w_0 x_{10} + \boldsymbol{\epsilon}_1 \\ y_2 &= w_2 x_{22} + w_1 x_{21} + w_0 x_{20} + \boldsymbol{\epsilon}_2 \\ y_3 &= w_2 x_{32} + w_1 x_{31} + w_0 x_{30} + \boldsymbol{\epsilon}_3 \\ &\ldots \end{aligned}$$

Column of 1s

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} x_{12} & x_{11} & \boxed{\begin{matrix} x_{10} \\ x_{20} \\ \vdots \end{matrix}} \\ x_{22} & x_{21} & \\ \vdots & \vdots & \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} w_2 \\ w_1 \\ w_0 \end{bmatrix} \qquad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \end{bmatrix}$$

Labels | Data (homogeneous) | Regression parameters | Residuals

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$$

Multivariate least-squares in matrix form

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$$

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\,(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \Longleftarrow$$

Least-squares solution $\widehat{\boldsymbol{w}}$

Note: Here, $\boldsymbol{X}$ contains the inputs as row vectors. $\boldsymbol{X}$ is often written with the inputs as column vectors, so in that case:

$$\boldsymbol{y} = \boldsymbol{X}^T\boldsymbol{w} + \boldsymbol{\epsilon}$$

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{y}$$

Need to understand in context

Linear regression function

$$y(\boldsymbol{x}) = \widehat{\boldsymbol{w}}^T\boldsymbol{x}$$

Using homogeneous coordinates

Simple linear regression example

Training set:

(-1, 0)

(0, 1)

(1, 1)

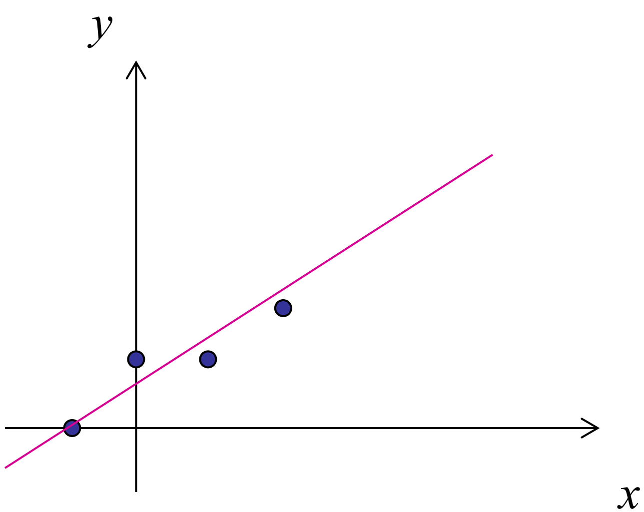
(2, 2)

inputs (x) outputs (y)

Learn the regression function $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{w} = w_1 x + w_0$

$$\boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} -1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_0 \end{bmatrix}$$

Homogeneous representation



$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$= \left(\begin{bmatrix} -1 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} = \begin{bmatrix} 6 & 2 \\ 2 & 4 \end{bmatrix}^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$= \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.3 \end{bmatrix}\begin{bmatrix} -1 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \end{bmatrix}$$

$$\widehat{\boldsymbol{w}} = \begin{bmatrix} 0.6 \\ 0.7 \end{bmatrix} = \begin{bmatrix} \text{slope} \\ \text{y−intercept} \end{bmatrix}$$

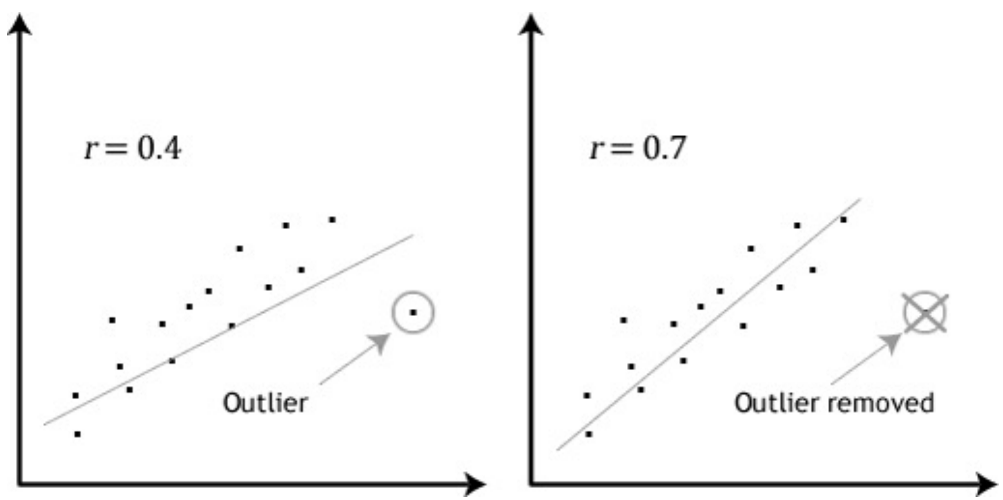
$$y(\boldsymbol{x}) = \widehat{\boldsymbol{w}}^T\boldsymbol{x} = \begin{bmatrix} 0.6 & 0.7 \end{bmatrix}\begin{bmatrix} x \\ 1 \end{bmatrix} = 0.6x + 0.7$$

Feature correlation

- If the features in a multivariate regression problem with *d* input features are uncorrelated ($\sigma_{x_i x_j} = 0$ if $i \neq j$) then the problem reduces to *d* univariate (one variable) problems
  - This relates to the task of feature construction – construct uncorrelated features to simplify the problem!
  - We may come back to this in Chapter 10 on features

Outliers

- An outlier is a measurement/observation that is distant from other observations
  - Could be due to measurement error or "heavy-tailed distribution" events
  - In other words, experimental anomalies
- In some machine learning problems we're very interested in such outliers (e.g., anomaly detection)
  - But in linear regression, they can be problematic – linear regression is sensitive to outliers



There is a lot of research on reducing sensitivity to outliers.

E.g., robust loss functions, probabilistic modeling methods such as RANSAC

Regularization

In a typical polynomial regression, $r(\boldsymbol{w}) = \|\boldsymbol{w}\|^2$ to discourage large coefficients

- Formulate the multivariate least-squares problem via *optimization*:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$$

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}}\,(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

(least squares minimization)

- Sometimes we'd like to provide constraints on the optimization problem in order to avoid overfitting to the data
  - E.g., if we think the training data may not be representative, or we have external knowledge about the problem beyond the data

- One way to do this is through regularization

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}}\,(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \lambda\,\underbrace{r(\boldsymbol{w})}_{\text{Regularization function}}$$

λ is a scalar determining the amount of regularization

  - So now when we optimize (minimize) to choose $\boldsymbol{w}^*$, λ is involved

Least-squares regression for classification

- We can use regression techniques to learn a binary classifier by encoding the two classes as real numbers, learning a regression function, and then thresholding the output
  - Label positive examples with +1 and negative examples with –1
  - I.e., $y_i \epsilon \{+1, -1\}$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- Assign class $\hat{y} = \text{sign}(\boldsymbol{w} \cdot \boldsymbol{x})$

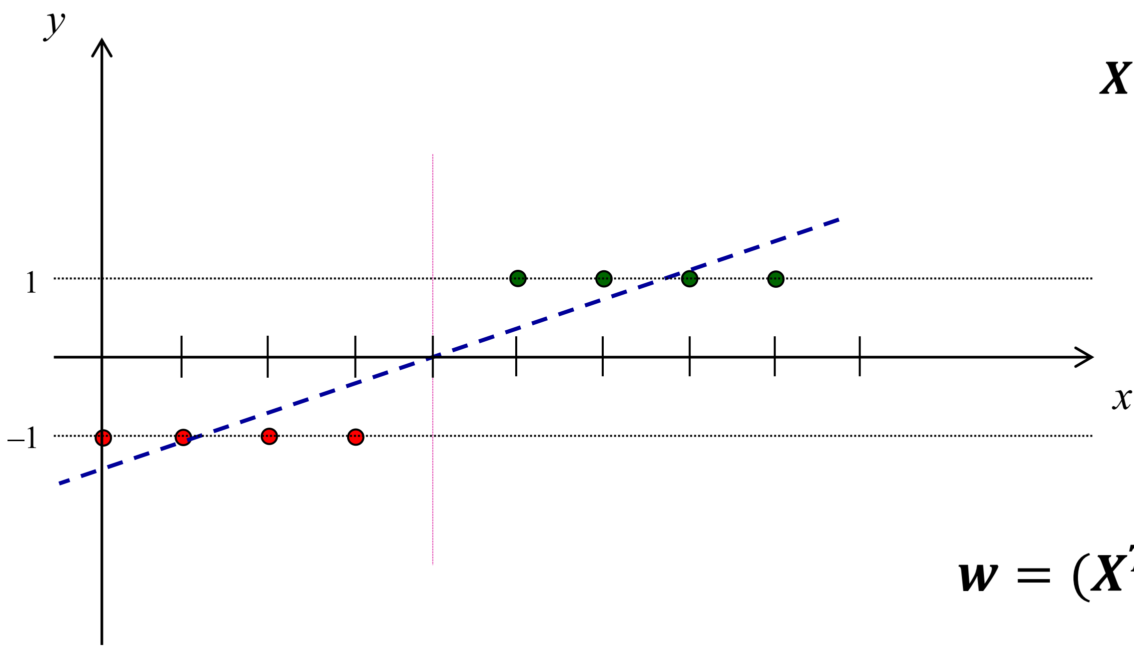
$$\text{sgn}(x) = \text{sign}(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ \cancel{0} & \cancel{\text{if } x = 0} \\ -1 & \text{if } x < 0 \end{cases}$$

(alternatively)

Signum function

Least-squares regression for classification

Training data: $\{(x_i, y_i)\}$ = { (0, -1), (1, -1), (2, -1), (3, -1), (5, 1), (6, 1), (7, 1), (8, 1) }



$$\boldsymbol{X} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \begin{bmatrix} 0.33 \\ -1.33 \end{bmatrix}$$

Test data point $\boldsymbol{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$ $\Rightarrow$ $\hat{y} = \mathrm{sign}(\boldsymbol{w} \cdot \boldsymbol{x}) = \mathrm{sign}(0.33x - 1.33)$

Quiz questions

- For data with N input features, what is the dimensionality of the linear regression function?
  - N (fit a line to 1D data, fit a plane to 2D data, etc.)
- For data with N features, what is the dimensionality of the linear classification boundary?
  - N-1 (a line separates 2D data, a plane separates 3D data, etc.)
- For data with N features, what is (nonhomogeneous) $\boldsymbol{w}$?
  - An N-dimensional vector
- What's the output/result of linear classifier training?
  - $\boldsymbol{w}$ (homogeneous) or ($\boldsymbol{w}$, t) (non-homogeneous)

By the way…

- The book is sometimes unclear when they're using homogeneous notation and when they're not
- For example, $\boldsymbol{w}^T\boldsymbol{x}$ can mean either

Nonhomogeneous

$$[w_1 \quad w_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

or

Homogeneous

$$[\,w_1 \quad w_2 \quad -t\,]\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

$$\boldsymbol{w}^T\boldsymbol{x} - t > 0$$

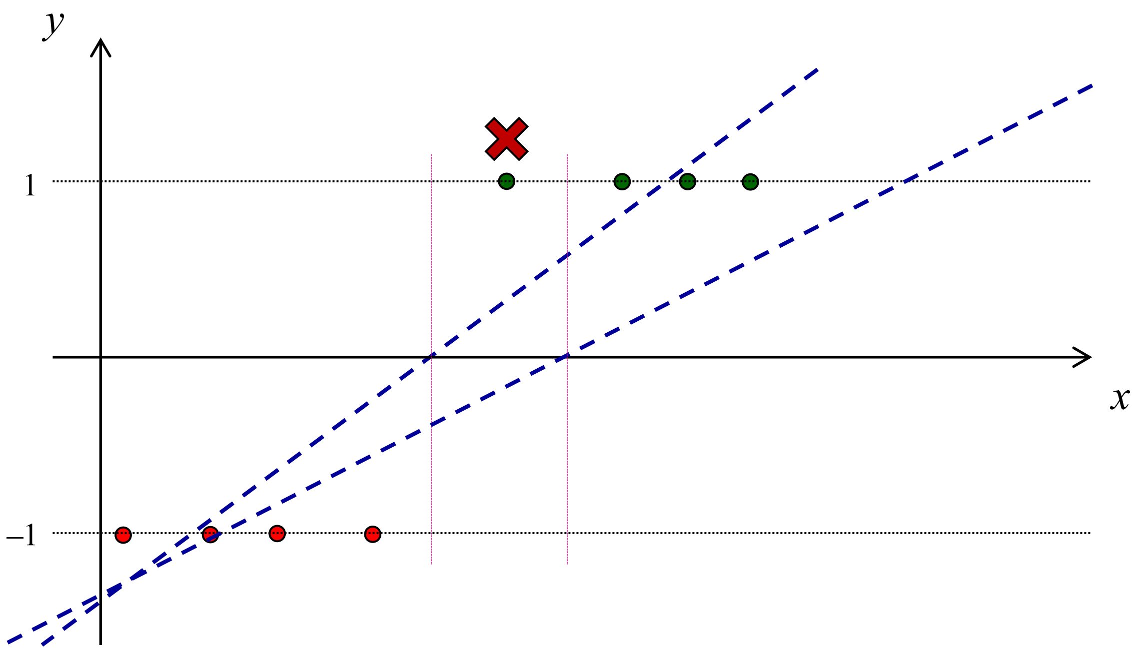
$$\boldsymbol{w}^T\boldsymbol{x} > 0$$

$$w_1x_1 + w_2x_2 - \mathrm{t} \ > 0$$

Interpret in context…

The perceptron

- A least squared classifier is not guaranteed to find a perfect decision boundary for linearly separable data



Next

- Perceptron, Chapter 7.2