Computer Science

CSC380: Principles of Data Science

Course wrap-up 2

Xinchen Yu

Announcements

- Final exam
  - Time: Dec 13, 3:30 - 5:30pm
  - Location: C E Chavez Bldg, Rm 111 (same room)
  - What you can bring:
    - one letter size cheat sheet, you can use double sides
    - calculator (not necessary)

- Fill out SCS (https://scsonline.oia.arizona.edu/) – if 80% responses, will add 5 points to the homework with lowest grade (63% right now).

Announcements

- Grades on D2L are available by this Friday.

- Final project
  - Due this Friday by 11:59 pm
  - Evaluate on test set (1,086 instances)

- Final exam practice questions out on D2L->Content

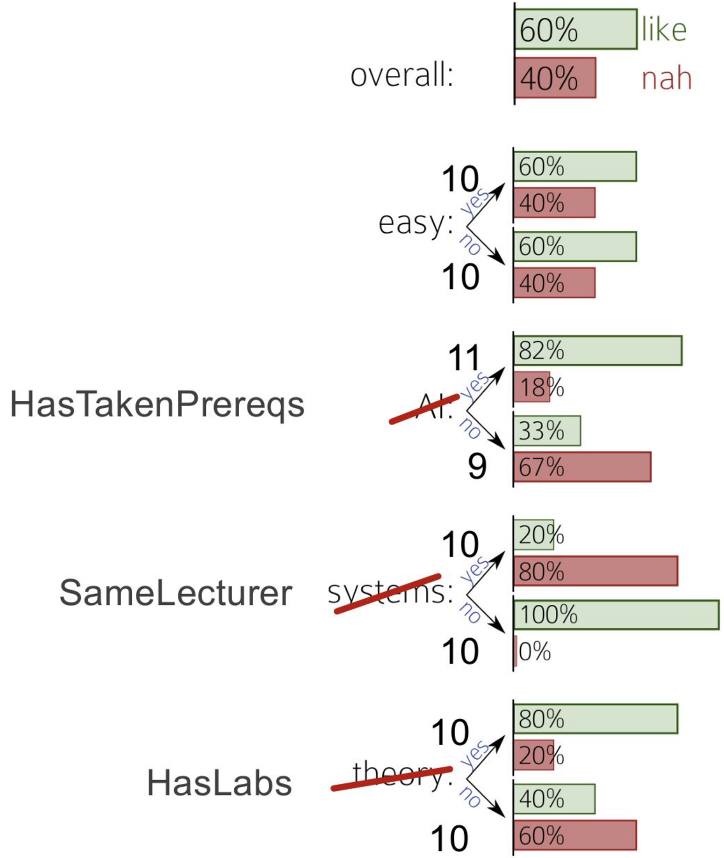
Predictive Modeling and Classification

How to construct a decision tree

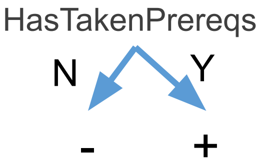
- Assign all training instances to the root of the tree. Set current node to root node.
- For each feature:
  a. Partition all data instances at the node by the value of the feature.
  b. Compute the accuracy from the partitioning.
- Identify feature that results in the highest accuracy. Set this feature to be the splitting criterion at the current node.

Rating	Easy?	Prereqs ~~AI?~~	Lecturer ~~Sys?~~	HasLabs ~~Thy?~~	Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

Decision tree: accuracy



Suppose we place the node HasTakenPrereqs at the root.
Set the prediction at each leaf node as the majority vote.



What is the train set accuracy now?

$$\frac{9}{20} \cdot \frac{6}{9} + \frac{11}{20} \cdot \frac{9}{11} = \frac{15}{20} = 0.75$$

No need to split if the leaf is pure
(all data have same labels)

Decision tree: accuracy

What is the train set accuracy now?

$$\frac{9}{20}\cdot\frac{6}{9}+\boxed{\frac{11}{20}}\cdot\frac{9}{11}=\frac{15}{20}=0.75$$

Accuracy for two groups:

- Prereqs = yes (11): 9/11
- Prereqs = no (9): 6/9

For the 11 people prereqs = y, use the majority vote label **like** (9 like, 2 dislike).

Predicted label for 11 people is **like**, 9 people are correctly predicted.

consider it to be 'like'

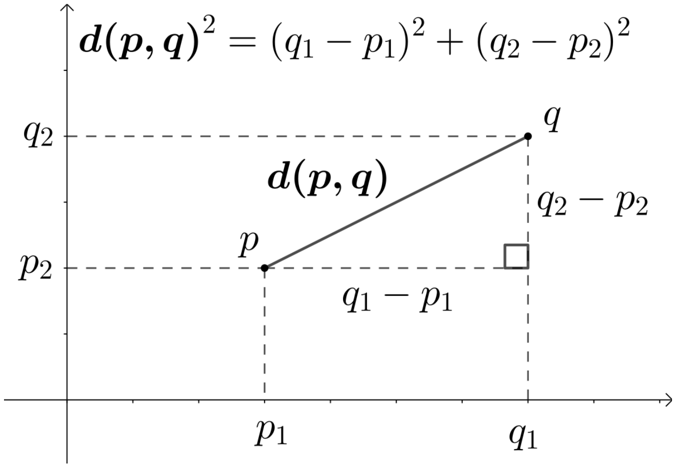
consider it to be 'dislike'

Prereqs Lecturer HasLabs

Rating	Easy?	~~AI?~~	~~Sys?~~	~~Thy?~~	Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

KNN

- Select the number K of the neighbors
- Calculate the Euclidean distance of K number of neighbors
- Take the K nearest neighbors as per the calculated Euclidean distance.
- Among these k neighbors, count the number of the data points in each category.
- Assign the new data points to that category for which the number of the neighbor is maximum.



Naïve Bayes

Training Data:

Person	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

Features

Task: Observe features $x_1, \ldots, x_D$ and predict class label $y \in \{1, \ldots, C\}$

Naïve Bayes Model: Treat features as *conditionally independent* given class label,

$$p(x, y) = p(y)p(x|y) = p(y) \prod_{d=1}^{D} p(x_d \mid y)$$

build individual models for these

To classify a given instance x: Bayes rule!

$$p(y = c \mid x) = \frac{p(y = c)p(x \mid y = c)}{p(x)}$$

Key concept in Naïve Bayes

$$p(x,y) = p(y)p(x|y) = p(y)\prod_{d=1}^{D} p(x_d \mid y)$$

Class prior distribution

Class conditional distribution

Given one data point, it has 4 features (input), and the label is 0 (output)

$$p(x_1, x_2, x_3, x_4, y = 0) = p(y = 0) \cdot p(x_1, x_2, x_3, x_4 | y = 0)$$

$$= p(y = 0) \cdot p(x_1 | y = 0) \cdot p(x_2 | y = 0) \cdot p(x_3 | y = 0) \cdot p(x_4 | y = 0)$$

Example: Naïve Bayes with Bernoulli Features

$j: feature,\ c: label,\ i: data$

$$y \sim Categorical(\pi_c):\ p(y=c) = \pi_c$$

$$p(y=1) = \pi_1$$
$$p(y=2) = \pi_2$$
$$p(y=3) = \pi_3 = 1 - \pi_1 - \pi_2$$

$$x|y \sim Bernoulli(\theta_{jc}):\ p(x|y) = \theta_{jc}{}^{x}\,(1-\theta_{jc})^{1-x}$$

$$x_{j=1}|y = 1 \sim Bernoulli(\theta_{j=1,c=1}) \qquad x_{j=2}|y = 1 \sim Bernoulli(\theta_{j=2,c=1})$$
$$x_{j=1}|y = 2 \sim Bernoulli(\theta_{j=1,c=2}) \qquad x_{j=2}|y = 2 \sim Bernoulli(\theta_{j=2,c=2})$$
$$x_{j=1}|y = 3 \sim Bernoulli(\theta_{j=1,c=3}) \qquad x_{j=2}|y = 3 \sim Bernoulli(\theta_{j=2,c=3})$$

y	x_1	x_2
1	0	1
3	1	0
3	1	1
2	0	0
1	1	0

Q: how many parameters?

c-1+cj

Model Selection and Evaluation

Tuning hyperparameters

K-fold cross validation

- Randomly partition train set S into K disjoint sets; call them $\mathrm{fold}_1, \dots, \mathrm{fold}_K$
- For each hyperparameter $h \in \{1, \dots, H\}$
  - For each $k \in \{1, \dots, K\}$
    - train $\hat{f}_k^h$ with $S \setminus \mathrm{fold_k}$
    - measure error rate $e_{h,k}$ of $\hat{f}_k^h$ on fold_k
  - Compute the average error of the above: $\widehat{err}^h = \frac{1}{K}\sum_{k=1}^{K} e_{h,k}$
- Choose $\hat{h} = \arg\min_h \widehat{err}^h$
- Train $\hat{f}^*$ using S (all the training points) with hyperparameter h
- Finally, evaluate $\hat{f}^*$ on test set to estimate its future performance.

Use when (1) the dataset is small (2) ML algorithm's retraining time complexity is low (e.g., kNN)

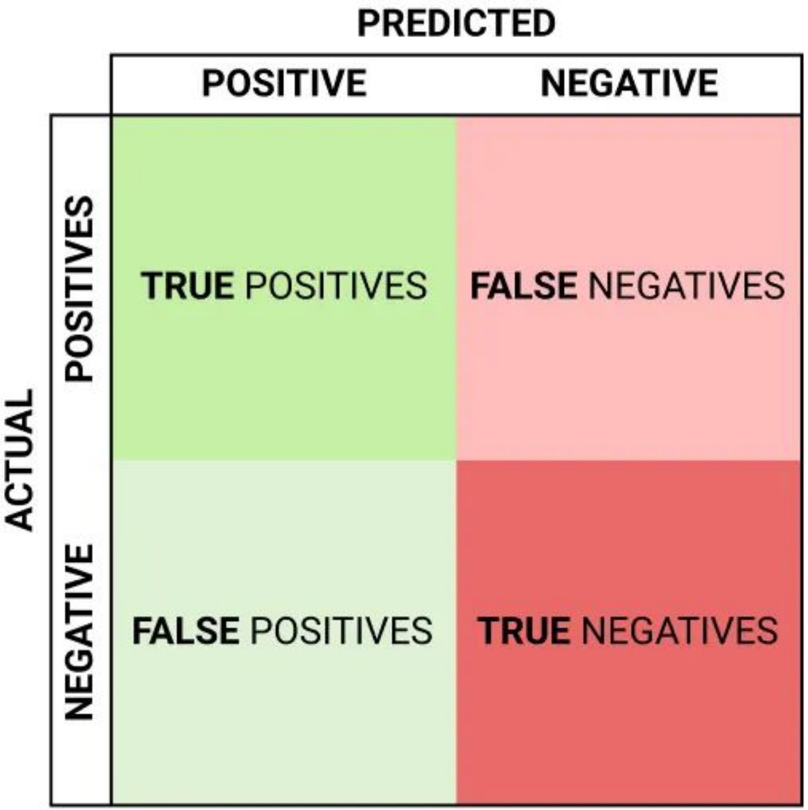
5-fold cross validation

$$Error = \frac{1}{5}\sum_{i=1}^{5} Error_i$$

Q: If we use 5-fold cross validation for KNN, how many KNNs do we need to train?

A: 5 (if excluding last retraining step).

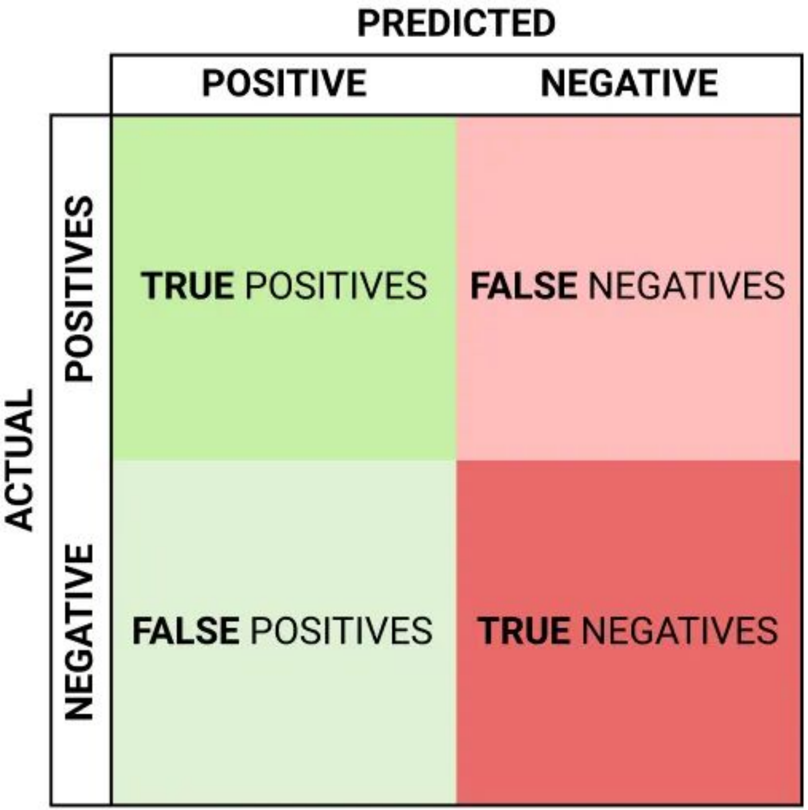
Evaluating Classifiers - Precision



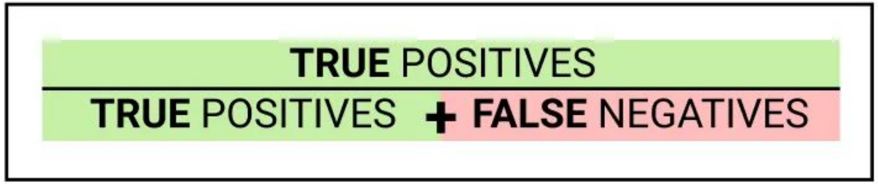
Precision: dividing the true positives by anything that was predicted as a positive.

$$\frac{\text{TRUE POSITIVES}}{\text{TRUE POSITIVES} + \text{FALSE POSITIVES}}$$

Evaluating Classifiers - Recall



Recall (or True Positive Rate): dividing the true positives by anything that should have been predicted as positive.



Evaluating Classifiers

F1 score symmetrically represents both precision and recall in one metric.

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}$$

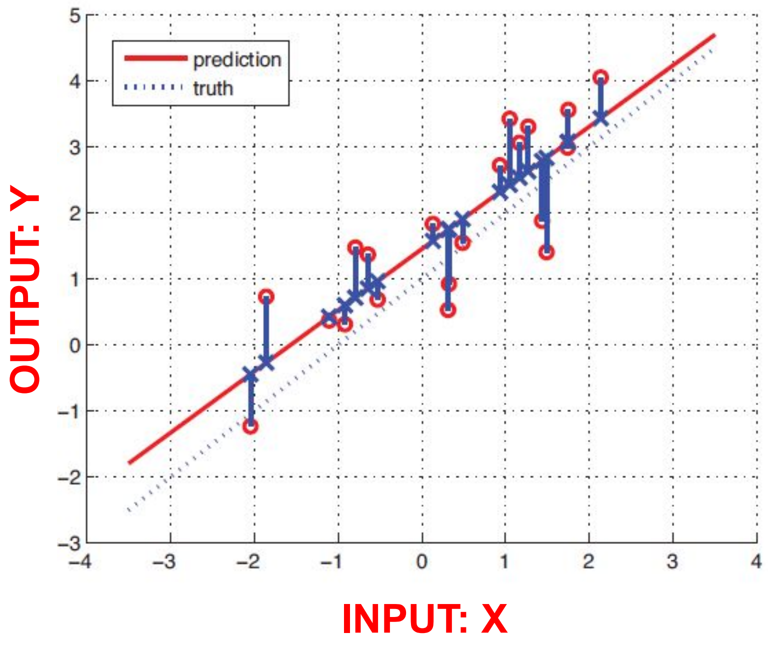
- This is the *harmonic mean* of precision and recall
  - harmonic_mean(x,y)

$$\frac{1}{\frac{1}{2}\left(\frac{1}{x} + \frac{1}{y}\right)}$$

- Gives equal importance to precision and recall – F1 may not be best when you care about one more than the other (e.g., in medical tests we care about recall)

Linear Models

Linear Regression



Regression Learn a function that predicts outputs from inputs,

$$y = f(x)$$

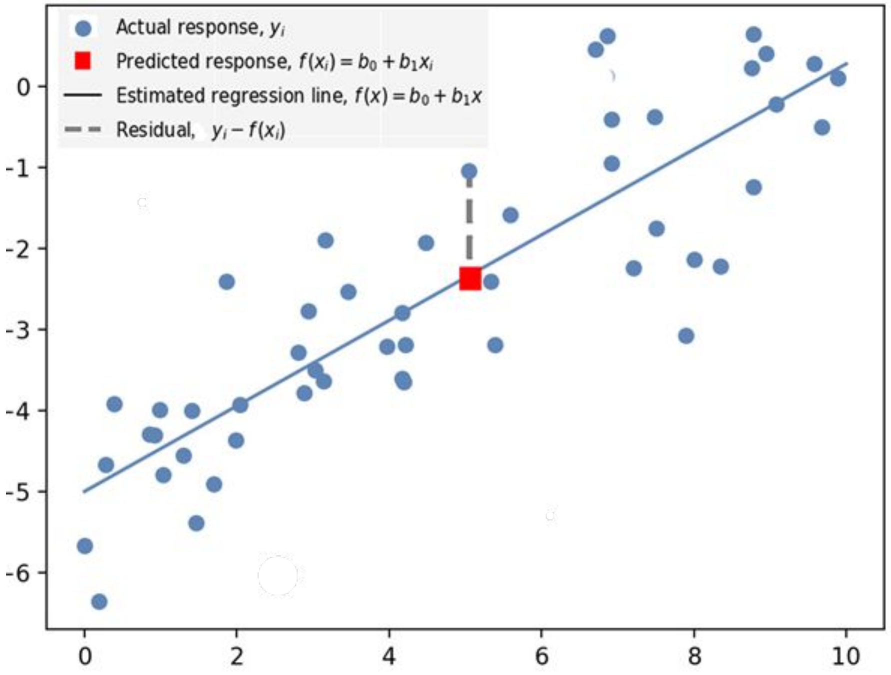
Outputs y are real-valued

Linear Regression As the name suggests, uses a *linear function*:

$$y = w^T x + b$$

$$w^T x := \sum_{d=1}^{D} w_d x_d$$

Fitting Linear Regression



Intuition Find a line that is as *close as possible* to every training data point

The distance from each point to the line is the **residual**

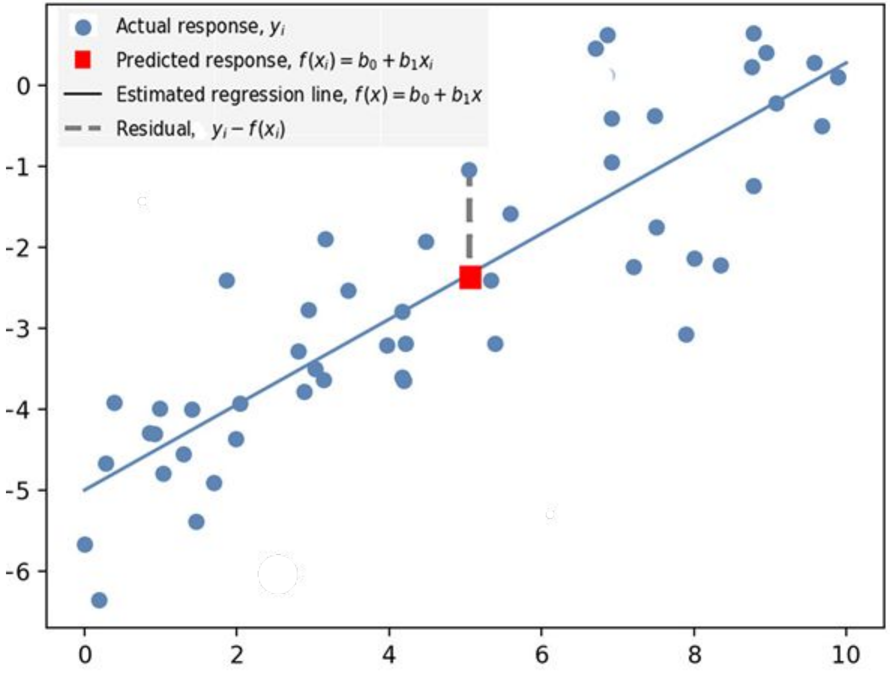
$$y - w^T x$$

Training Output **Prediction**

Let's find w that will minimize the residual!

https://www.activestate.com/resources/quick-reads/how-to-run-linear-regressions-in-python-scikit-learn/

Least Squares Solution



Functional Find a line that minimizes the sum of squared residuals!

Given: $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$

Compute:

$$w^* = \arg\min_w \sum_{i=1}^{m} \left(y^{(i)} - w^T x^{(i)}\right)^2$$

Least squares regression

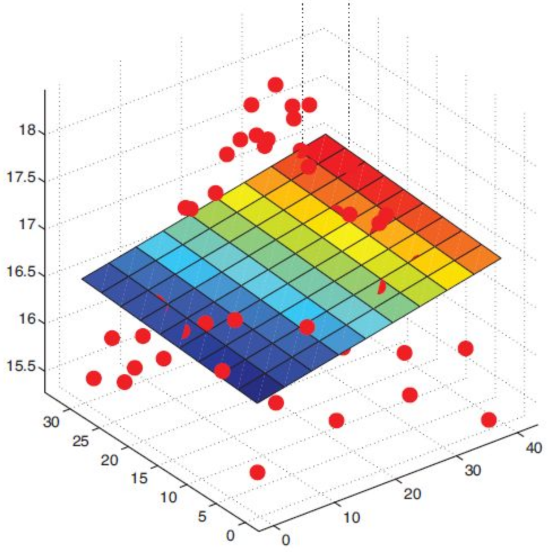
https://www.activestate.com/resources/quick-reads/how-to-run-linear-regressions-in-python-scikit-learn/

Least Squares: Higher Dimensions

Least squares can also be written more compactly,

$$\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}\cdot\boldsymbol{x}}.$$

$$\min_w \sum_{i=1}^N (y^{(i)} - w^T x^{(i)})^2 = \|\mathbf{y} - \mathbf{X}w\|^2$$



Some slightly more advanced linear algebra gives us a solution,

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

Ordinary Least Squares *(OLS)* solution   OLS solution has less residual

Regularized Least Squares

Ordinary least-squares (OLS) estimation (no regularizer),

$$w^{\text{OLS}} = \arg\min_w \sum_{i=1}^{m} (y^{(i)} - w^T x^{(i)})^2$$

L2 norm: $\|w\| = \sqrt{\sum_{d=1}^{D} w_d^2}$

L1 norm: $\|w\|_1 = \sum_{d=1}^{D} |w_d|$

L2-regularized Least-Squares (Ridge)

$$w^{\text{L2}} = \arg\min_w \sum_{i=1}^{m} (y^{(i)} - w^T x^{(i)})^2 + \lambda \|w\|^2$$

Convention: Just saying 'RLS' means L2-RLS

L1-regularized Least-Squares (LASSO) LASSO: Least Absolute Shrinkage and Selection Operator

$$w^{\text{L2}} = \arg\min_w \sum_{i=1}^{m} (y^{(i)} - w^T x^{(i)})^2 + \lambda \|w\|_1$$

Logistic Regression

Model:

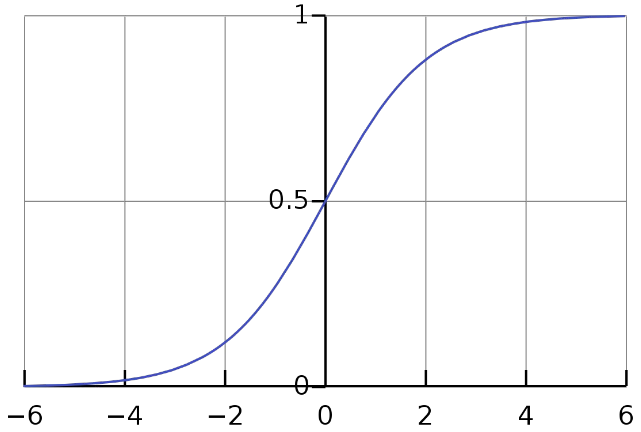
$$y \sim \text{Bernoulli}\big(p = \sigma(w^\top x)\big)$$

Train: compute the MLE $\widehat{w}$

Test: Given test point x^* compute

$$y^* = \arg\max_{v \in \{-1,1\}} p(y = v \mid x^*; \widehat{w})$$

- Equivalent to $y^* = \mathbf{I}\{\widehat{w}^\top x^* \geq 0\}$



Nonlinear Models

Separating Hyperplane

A hyperplane h(x) splits the original d-dimensional space into two half-spaces. If the input dataset is linearly separable:

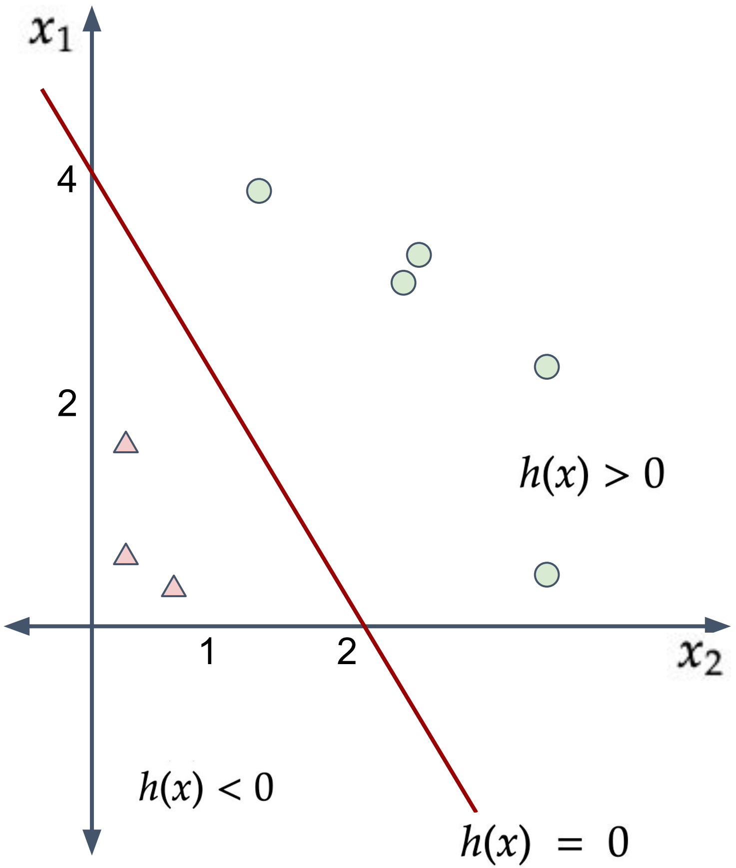
$$y = \begin{cases} +1 & \text{if } h(\mathbf{x}) > 0 \\ -1 & \text{if } h(\mathbf{x}) < 0 \end{cases}$$

Example:

$$h(x) = x_1 + 2x_2 - 4$$

Q: label for (0, 3)?

A: +1



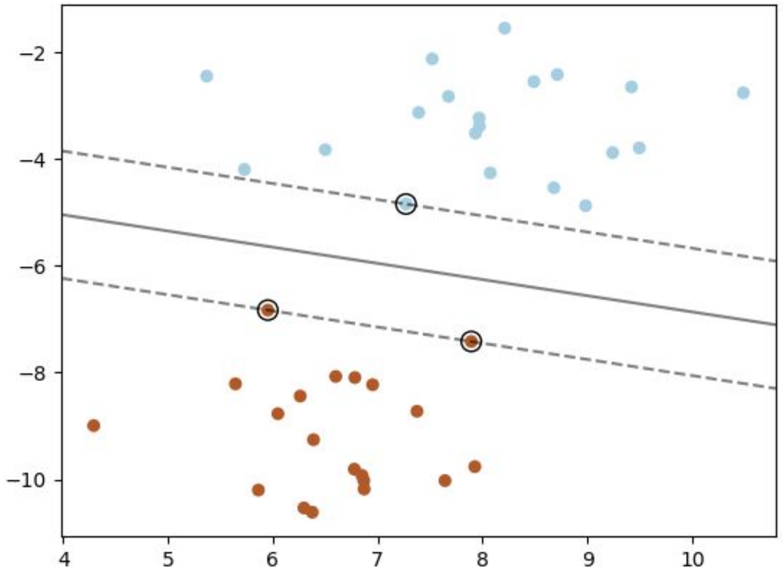
Margin and Support Vectors

Over all the n points, the ***margin*** of the linear classifier is the minimum distance of a point from the separating hyperplane:

$$\delta^* = \min_{\mathbf{x}_i} \left\{ \frac{y_i(\mathbf{w}^T \mathbf{x}_i + b)}{||\mathbf{w}||} \right\}$$

All the points that achieve this minimum distance are called ***support vectors***.

$$\delta^* = \frac{y^*(\mathbf{w}^T \mathbf{x}^* + b)}{||\mathbf{w}||}$$



Max-Margin Classifier (Linear Separable Case)

For training data $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, a classifier $f(x) = w^\top x + b$ with 0 train error will satisfy

$$y^{(i)} f(x^{(i)}) = y^{(i)}(w^\top x^{(i)} + b) > 0$$

↓ negative margin when misclassifying it!

The distance for ($x^{(i)}$, $y^{(i)}$) to separating hyperplane

$$\frac{y^{(i)}(w^\top x^{(i)} + b)}{\|w\|}$$

The margin of a classifier $f(x)$ is

$$\min_i \frac{y^{(i)}(w^\top x^{(i)} + b)}{\|w\|}$$

Find f that maximize margin

$$\arg\max_{w,b} \min_i \frac{y^{(i)}(w^\top x^{(i)} + b)}{\|w\|}$$

Canonical Hyperplane

Way to solve this issue:

- Choose the scalar s such that the absolute distance of a ***support vector*** from the hyperplane is 1.

$$sy^*(\mathbf{w}^T\mathbf{x}^* + b) = 1$$

$$s = \frac{1}{y^*(\mathbf{w}^T\mathbf{x}^* + b)} = \frac{1}{y^* h(\mathbf{x}^*)}$$

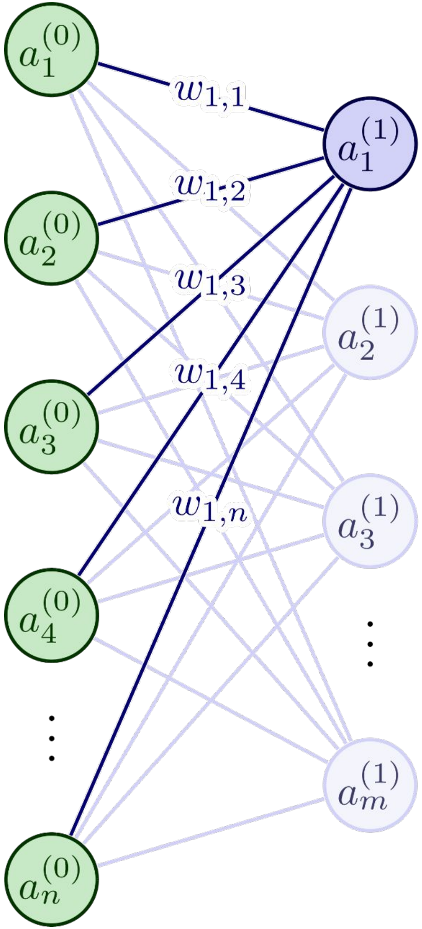
$$y_i\ (\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \text{ for all points } \mathbf{x}_i \in \mathbf{D}$$

$$\arg\max_{w,b}\min_i \frac{\overbrace{\boxed{y^{(i)}(w^\top x^{(i)} + b)}}^{1}}{\|w\|}$$

Margin: $\delta^* = \dfrac{1}{\|\mathbf{w}\|}$

Max margin: $h^* = \arg\max_h \{\delta_h^*\} = \arg\max_{\mathbf{w},b} \left\{ \dfrac{1}{\|\mathbf{w}\|} \right\}$

NN: Feedforward Procedure



$$= \sigma\left(w_{1,0}a_0^{(0)} + w_{1,1}a_1^{(0)} + \ldots + w_{1,n}a_n^{(0)} + b_1^{(0)}\right)$$

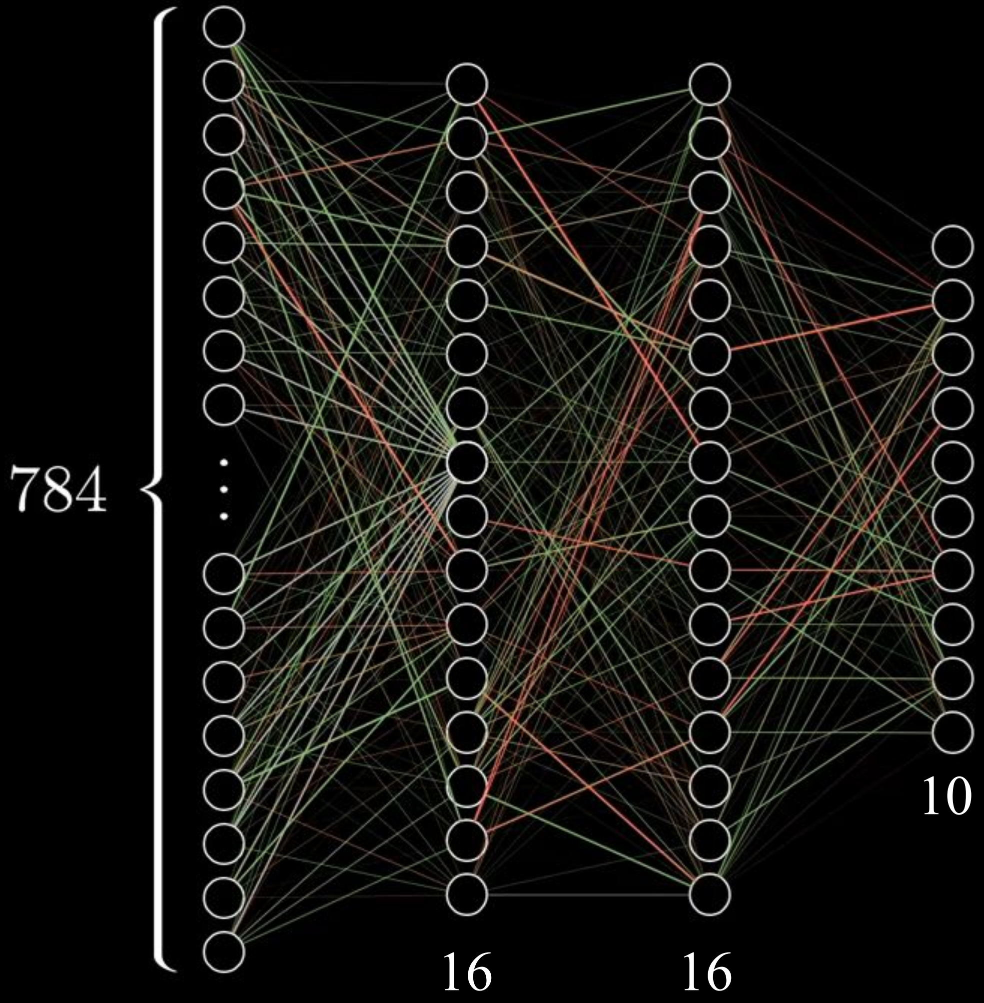
$$= \sigma\left(\sum_{i=1}^{n} w_{1,i}a_i^{(0)} + b_1^{(0)}\right)$$

$$\begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} = \sigma\left[\begin{pmatrix} w_{1,0} & w_{1,1} & \ldots & w_{1,n} \\ w_{2,0} & w_{2,1} & \ldots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,0} & w_{m,1} & \ldots & w_{m,n} \end{pmatrix}\begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \\ \vdots \\ a_n^{(0)} \end{pmatrix} + \begin{pmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_m^{(0)} \end{pmatrix}\right]$$

$$a^{(1)} = \sigma\left(\mathbf{W}^{(0)}a^{(0)} + \mathbf{b}^{(0)}\right)$$

Number of parameters in this example: $m \times n + m$

[Source : 3Blue1Brown : https://www.youtube.com/watch?v=aircAruvnKk]



$$784\times16+16\times16+16\times10$$
weights

$$16+16+10$$
biases

13,002

Each parameter has some impact on the output…need to train all these parameters simultaneously to have a good prediction accuracy

Clustering

k-means clustering

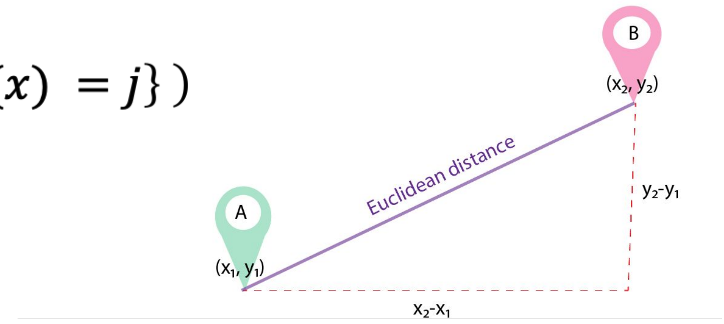
Input: k: num. of clusters, $S = \{x_1, \dots, x_n\}$

[Initialize] Pick $c_1, \dots, c_k$ as randomly selected points from S (see next slides for alternatives)

For t=1,2,…,max_iter

- ***[Assignments]*** $\forall x \in S, \quad a_t(x) = \arg\min_{j \in [k]} \|x - c_j\|_2^2$
- If $t \neq 1 \quad \text{AND} \quad a_t(x) = a_{t-1}(x), \forall x \in S$
  - break
- ***[Centroids]*** $\forall j \in [k], \quad c_j \leftarrow \text{average}(\{x \in S: a_t(x) = j\})$

Output: $c_1, \dots, c_k$ and $\{a_t(x_i)\}_{i \in [n]}$



Thank you!