CSCI 567 Discussion

1 Multiple-choice Questions (30 points)

IMPORTANT: *Select ALL answers that you think are correct. You get 0.5 point for selecting each correct answer and similarly 0.5 point for not selecting each incorrect answer.*

- We will go through the 15 multiple-choice questions from the sample exam.

- For each question,
  - ~2 mins to read and think about the question.
  - Then we will walk through the solution.

(1) Which of the following on machine learning is correct?

(A) Cross-validation is often used to tune the hyper-parameters of a machine learning algorithm.
(B) Overfitting refers to the phenomenon when the training error is low but the test error is high.
(C) One should prevent overfitting by using the test set during training.
(D) Logistic regression is an algorithm for regression tasks.

(1) Which of the following on machine learning is correct?

(A) Cross-validation is often used to tune the hyper-parameters of a machine learning algorithm.
(B) Overfitting refers to the phenomenon when the training error is low but the test error is high.
(C) One should prevent overfitting by using the test set during training.
(D) Logistic regression is an algorithm for regression tasks.

Answer: AB

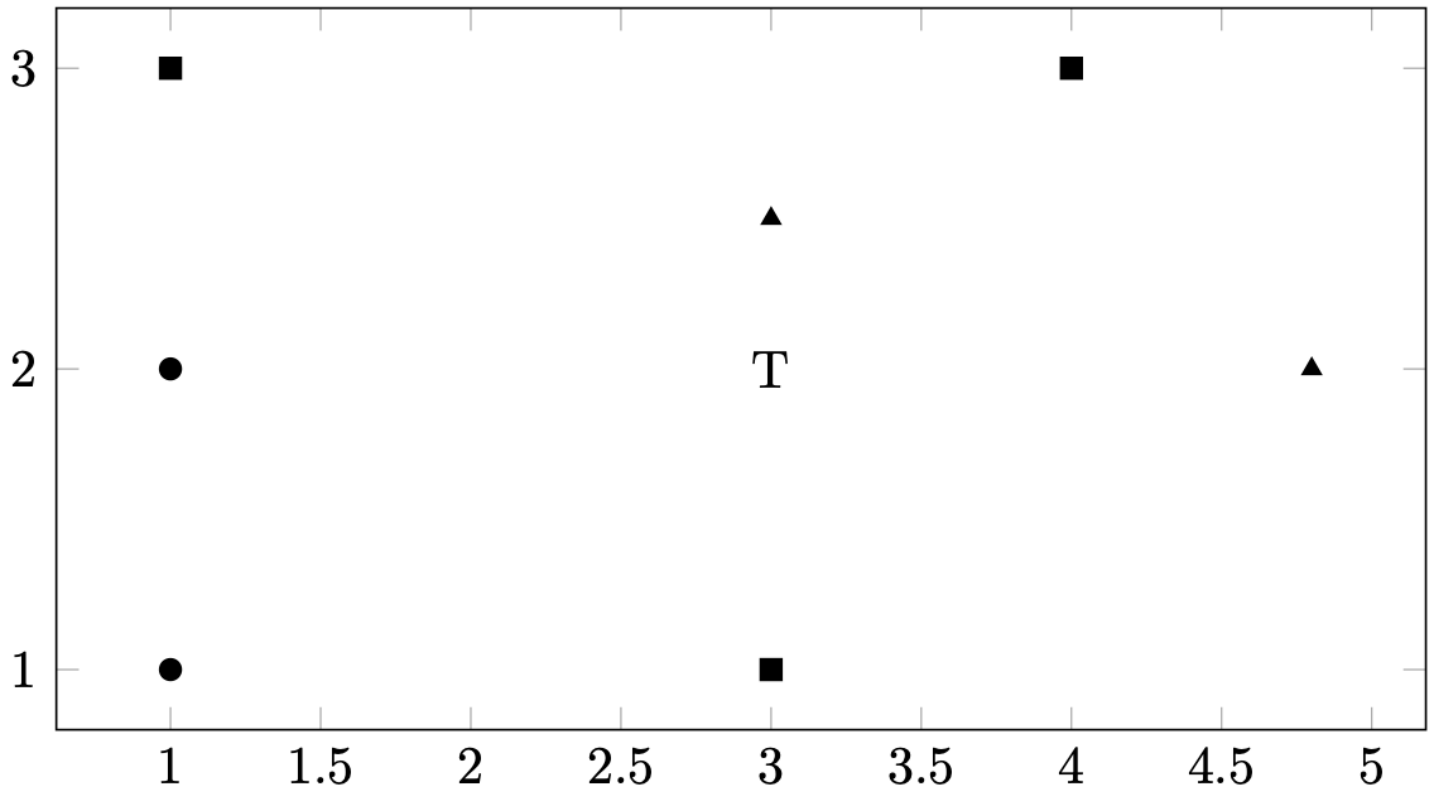
- (C) is **incorrect**:
  - The test set should not be used in training. It is reserved for final evaluation.
  - To prevent overfitting: more training data, regularization, etc.

- (D) is **incorrect**:
  - Logistic regression is used for **classification.**

Overfitting: 09/05 lecture slides, page 43; Logistic regression: 09/12 lecture slides, page 45.

(2) Consider the following two-dimensional dataset with $N = 7$ training points of three classes (triangle, square, and circle), and additionally one test point denoted by the letter T. Which of the following configuration of the K-nearest neighbor algorithm will predict square for the test point?

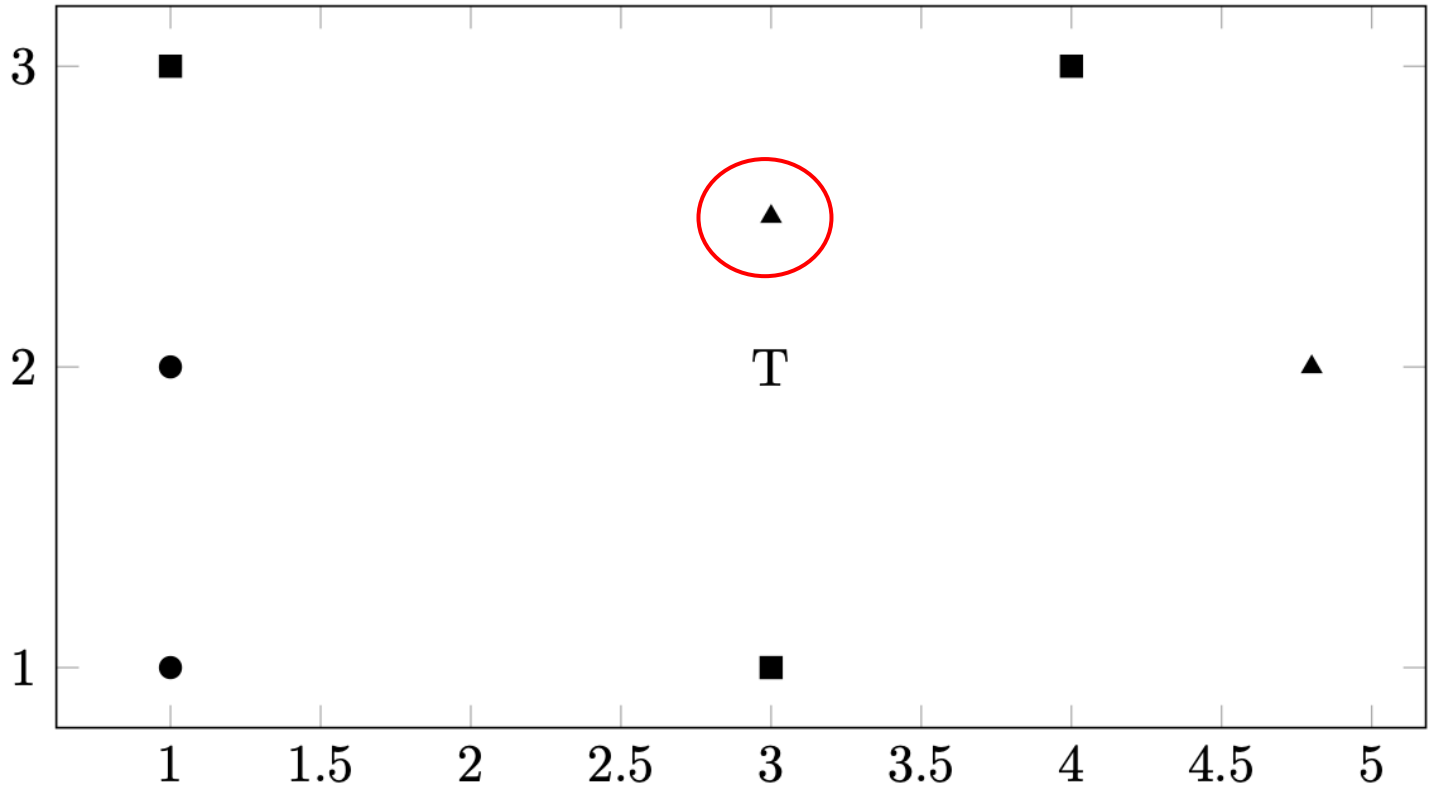
(A) $K = 1$, L2 distance
(B) $K = 3$, L1 distance
(C) $K = 3$, L2 distance.
(D) $K = 7$, any distance.



(2) Consider the following two-dimensional dataset with $N = 7$ training points of three classes (triangle, square, and circle), and additionally one test point denoted by the letter T. Which of the following configuration of the K-nearest neighbor algorithm will predict square for the test point?

~~(A) $K = 1$, L2 distance~~
(B) $K = 3$, L1 distance
(C) $K = 3$, L2 distance.
(D) $K = 7$, any distance.

Answer: CD



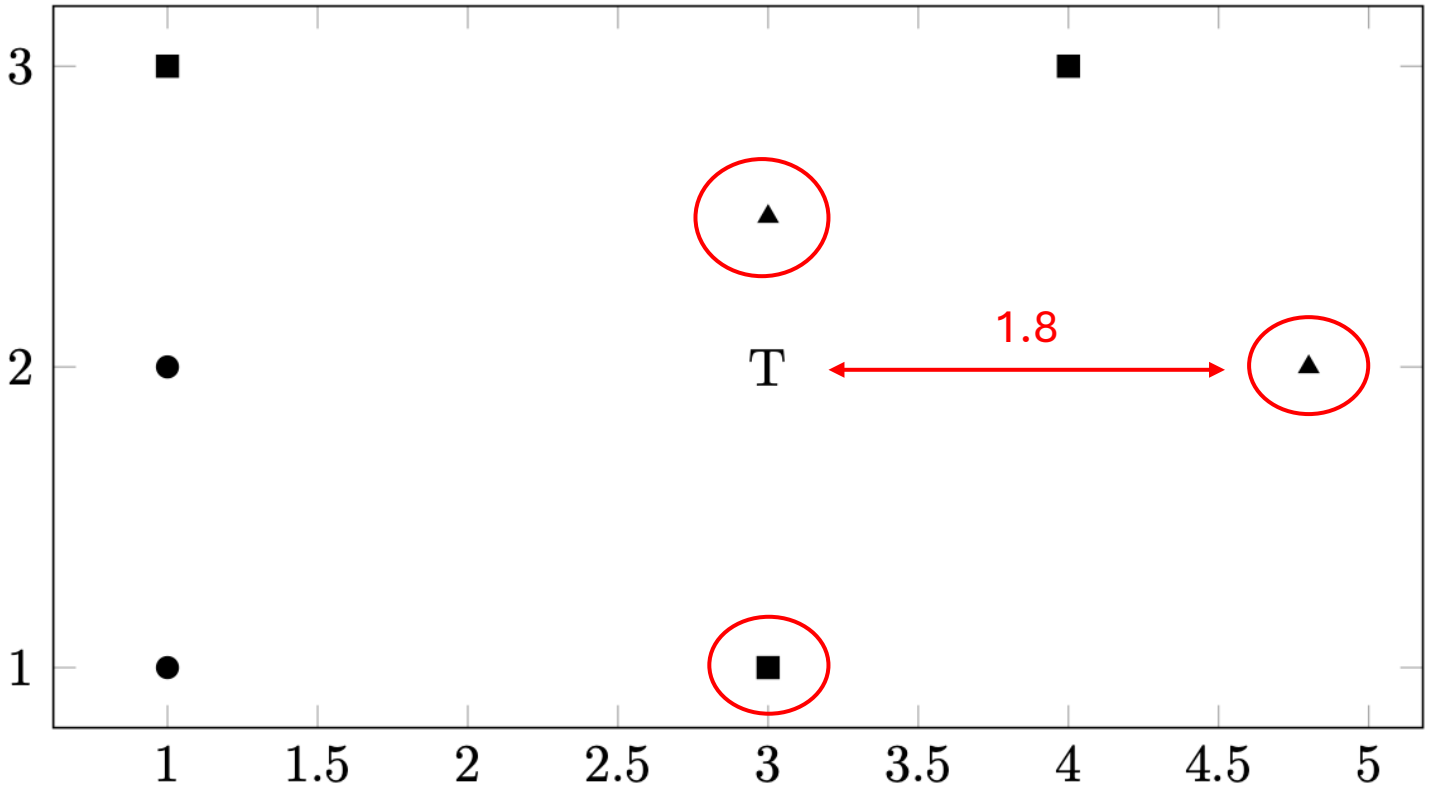
- **(A):** the single nearest neighbor is the ▲, so the prediction is ▲.

KNN: 08/29 lecture slides, page 45.

(2) Consider the following two-dimensional dataset with $N = 7$ training points of three classes (triangle, square, and circle), and additionally one test point denoted by the letter T. Which of the following configuration of the K-nearest neighbor algorithm will predict square for the test point?

~~(A) $K = 1$, L2 distance~~
~~(B) $K = 3$, L1 distance~~
(C) $K = 3$, L2 distance.
(D) $K = 7$, any distance.

Answer: CD



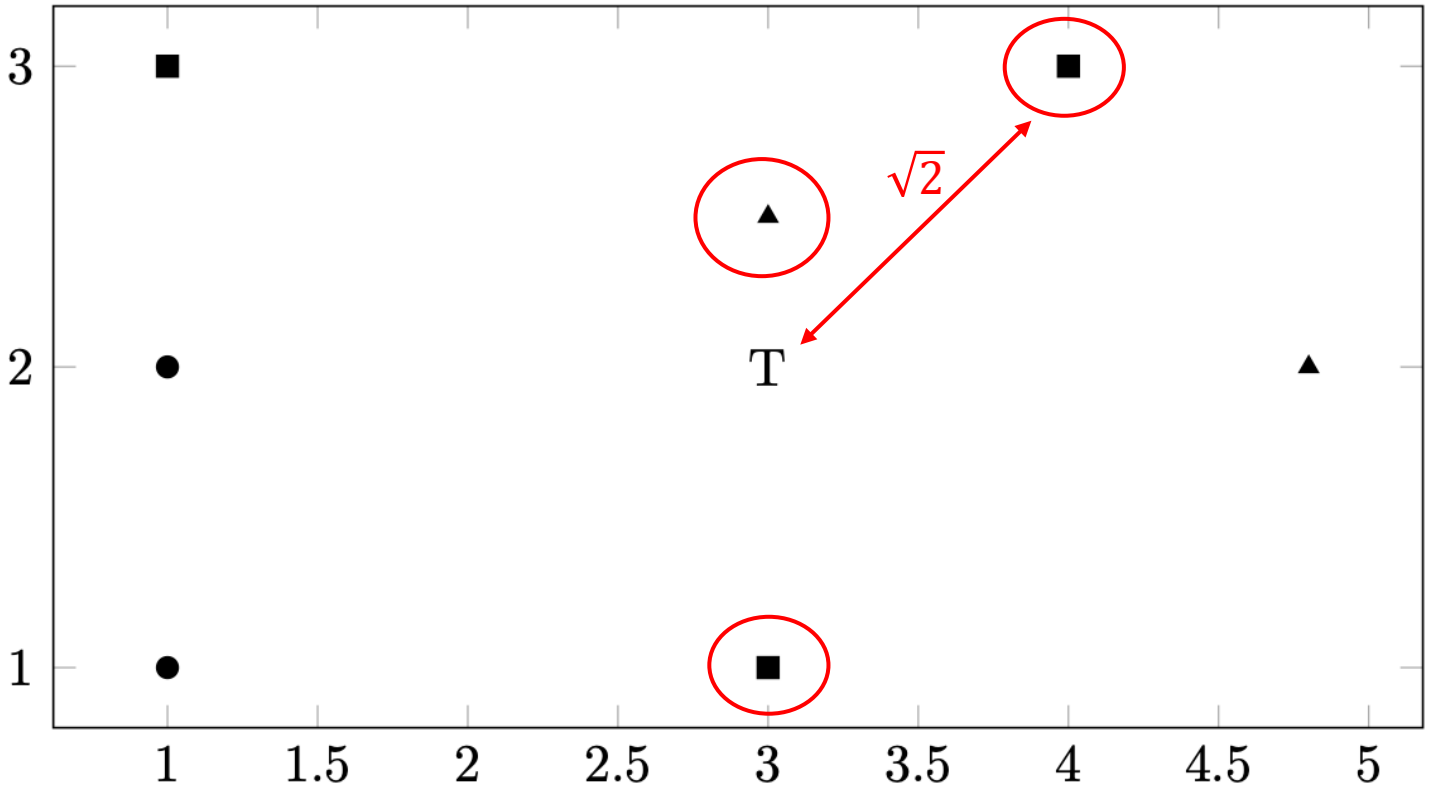
- **(B):** the 3 nearest neighbors are ▲, ■, ▲, the prediction is ▲.

KNN: 08/29 lecture slides, page 45.

(2) Consider the following two-dimensional dataset with $N = 7$ training points of three classes (triangle, square, and circle), and additionally one test point denoted by the letter T. Which of the following configuration of the K-nearest neighbor algorithm will predict square for the test point?

~~(A) $K = 1$, L2 distance~~
~~(B) $K = 3$, L1 distance~~
(C) $K = 3$, L2 distance.
(D) $K = 7$, any distance.

Answer: CD



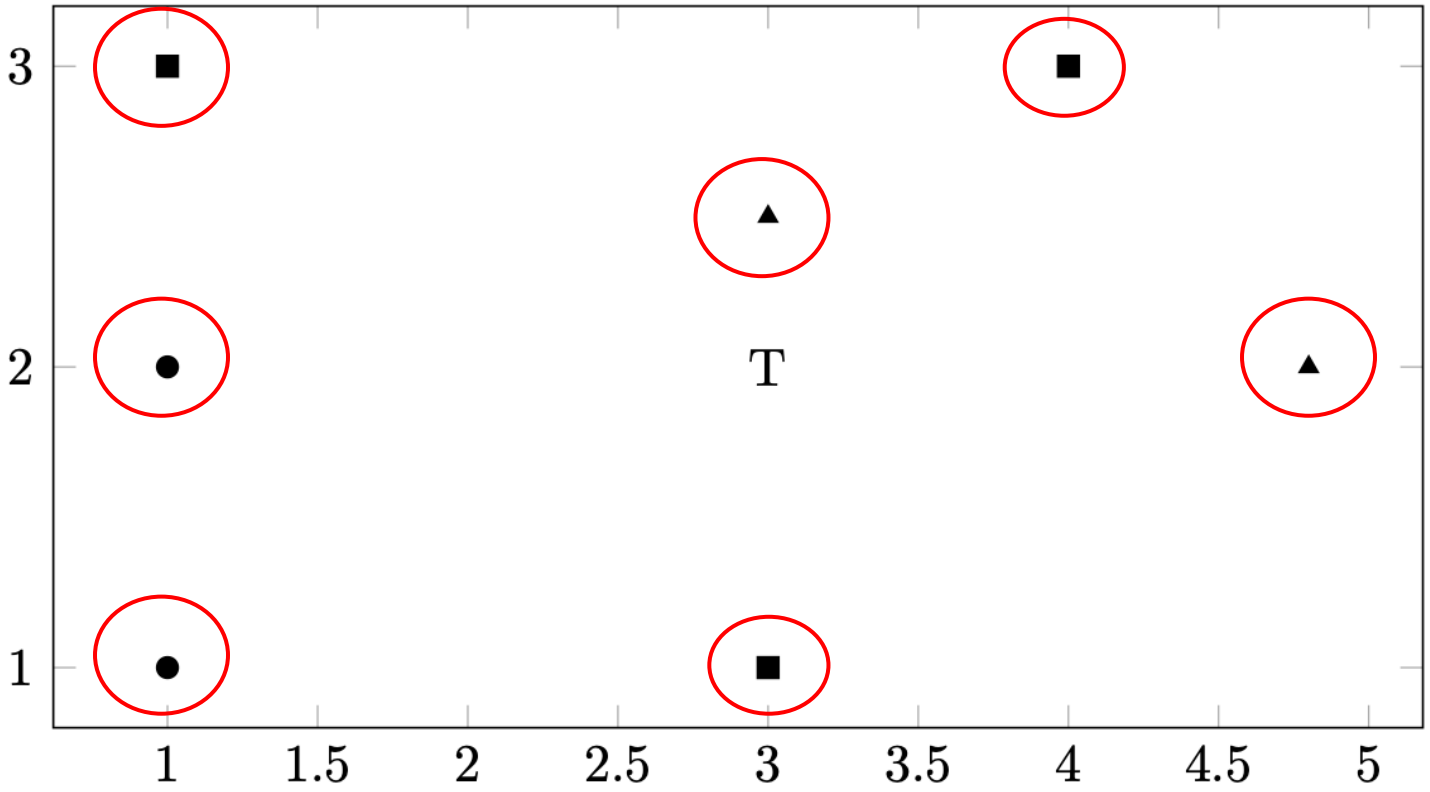
- **(C):** the 3 nearest neighbors are ▲, ■, ■, the prediction is ■.

KNN: 08/29 lecture slides, page 45.

(2) Consider the following two-dimensional dataset with $N = 7$ training points of three classes (triangle, square, and circle), and additionally one test point denoted by the letter T. Which of the following configuration of the K-nearest neighbor algorithm will predict square for the test point?

~~(A) $K = 1$, L2 distance~~
~~(B) $K = 3$, L1 distance~~
(C) $K = 3$, L2 distance.
(D) $K = 7$, any distance.

Answer: CD



- **(D):** all the points are the neighbors: ●, ●, ▲, ▲, ■, ■, ■, so the prediction is ■.

KNN: 08/29 lecture slides, page 45.

(3) Which of the following on linear regression is correct?

(A) The least square solution has a closed-form formula, even if L1 regularization is applied.
(B) The covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible if the number of data points N is smaller than the dimension D.
(C) When the covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible, the Residual Sum of Squares (RSS) objective has infinitely many minimizers.
(D) Linear regression is a parametric method.

(3) Which of the following on linear regression is correct?

(A) The least square solution has a closed-form formula, even if L1 regularization is applied.
(B) The covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible if the number of data points N is smaller than the dimension D.
(C) When the covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible, the Residual Sum of Squares (RSS) objective has infinitely many minimizers.
(D) Linear regression is a parametric method.

Answer: BCD

- **(A)** is incorrect:
  - With L2 regularization (We have worked on this before in Homework 1 Problem 2)

$$w_* = \arg\min_{w \in \mathbb{R}^D} \|Xw - y\|_2^2 + \lambda \|w\|_2^2 \quad \text{or} \quad w'_* = \arg\min_{w \in \mathbb{R}^D} \|Xw - y\|_2^2 + w^T M w \qquad M = \lambda I$$

  - We found its closed form by setting the gradient of $\|Xw - y\|_2^2 + w^T M w$ to 0

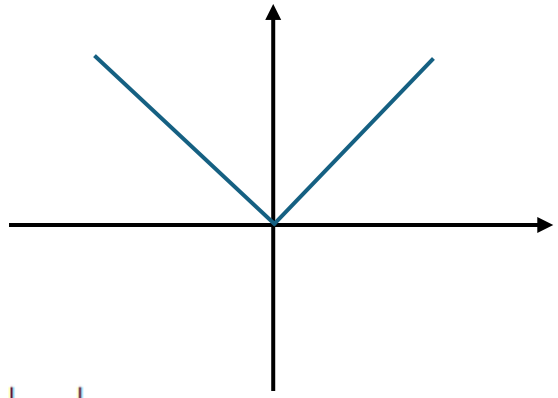
$$w'_* = \left(X^{\mathrm{T}}X + M\right)^{-1} X^{\mathrm{T}} y.$$

Least square solution: 09/05 lecture slides, page 20;

(3) Which of the following on linear regression is correct?

(A) The least square solution has a closed-form formula, even if L1 regularization is applied.
(B) The covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible if the number of data points N is smaller than the dimension D.
(C) When the covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible, the Residual Sum of Squares (RSS) objective has infinitely many minimizers.
(D) Linear regression is a parametric method.

Answer: BCD



- **(A)** is incorrect:
  - With L1 regularization:

$$\min_{w} \|y - Xw\|_2^2 + \lambda\|w\|_1. \qquad \|w\|_1 = \sum_i |w_i|$$

    - The absolute value $|w_i|$ is not differentiable at $w_i = 0$. (L1 is not differentiable at zero)

Least square solution: 09/05 lecture slides, page 20;

(3) Which of the following on linear regression is correct?

(A) The least square solution has a closed-form formula, even if L1 regularization is applied.
(B) The covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible if the number of data points N is smaller than the dimension D.
(C) When the covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible, the Residual Sum of Squares (RSS) objective has infinitely many minimizers.
(D) Linear regression is a parametric method.

Answer: BCD

- **(B)** is correct:

What if $\tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}}$ is not invertible

Why would that happen?

One situation: $N < D + 1$, i.e. not enough data to estimate all parameters.

- Explanation and examples can be found in 09/05 lecture slides, page 30.

(3) Which of the following on linear regression is correct?

(A) The least square solution has a closed-form formula, even if L1 regularization is applied.
(B) The covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible if the number of data points N is smaller than the dimension D.
(C) When the covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible, the Residual Sum of Squares (RSS) objective has infinitely many minimizers.
(D) Linear regression is a parametric method.

Answer: BCD

- **(C)** is correct:

$\mathrm{D} = 1, \mathrm{N} = 2$

sqft	sale price
1000	500K
1000	600K

Any line passing **the average** is a minimizer of RSS.

$$(X^\top X)w = X^\top y$$

$\mathrm{D} = 2, \mathrm{N} = 3$?

sqft	#bedroom	sale price
1000	2	500K
1500	3	700K
2000	4	800K

Again *infinitely many minimizers*.

09/05 lecture slides, page 30.

(3) Which of the following on linear regression is correct?

(A) The least square solution has a closed-form formula, even if L1 regularization is applied.
(B) The covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible if the number of data points N is smaller than the dimension D.
(C) When the covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible, the Residual Sum of Squares (RSS) objective has infinitely many minimizers.
(D) Linear regression is a parametric method.

Answer: BCD

- **(D)** is correct:
  - **Parametric methods**: the size of the model does *not grow* with the size of the training set N.
    - e.g. linear regression, D + 1 parameters, independent of N.

$$\hat{y} = w^{\top}x + b$$

- Parameters are $w \in \mathbb{R}^D$ and b
- No matter how many samples you get, you are only estimating D+1 numbers.

09/05 lecture slides, page 35.

(4) Perceptron algorithm is applying SGD with learning rate $\eta = 1$ to the Perceptron loss, and $\boldsymbol{w}^* = \mathbf{0}$ is clearly a minimizer of the Perceptron loss. Which of the following statements related to these two facts is correct?

(A) Perceptron always converges to $\boldsymbol{w}^* = \mathbf{0}$.
(B) Perceptron converges to $\boldsymbol{w}^* = \mathbf{0}$ when it is initialized at $\mathbf{0}$.
(C) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because the learning rate does not decrease over time.
(D) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because SGD uses a stochastic gradient instead of the exact gradient.

(4) Perceptron algorithm is applying SGD with learning rate $\eta = 1$ to the Perceptron loss, and $\boldsymbol{w}^* = \mathbf{0}$ is clearly a minimizer of the Perceptron loss. Which of the following statements related to these two facts is correct?

(A) Perceptron always converges to $\boldsymbol{w}^* = \mathbf{0}$.
(B) Perceptron converges to $\boldsymbol{w}^* = \mathbf{0}$ when it is initialized at $\mathbf{0}$.
(C) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because the learning rate does not decrease over time.
(D) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because SGD uses a stochastic gradient instead of the exact gradient.

Answer: C

Perceptron algorithm is SGD with $\eta = 1$ applied to perceptron loss:

Repeat:

- Pick a data point $\boldsymbol{x}_n$ uniformly at random
- If $\text{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) \neq y_n$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y_n\boldsymbol{x}_n$$

- **(A) (B)** are incorrect:
  - Even though $w^* = 0$ is a minimizer of the perceptron loss,
  - The algorithm's update rule will move away from zero as soon as it sees data.
  - The update rule keeps adjusting and does not guarantee a return to 0.

Perceptron: 09/12 lecture slides, page 18-19, 37-.

(4) Perceptron algorithm is applying SGD with learning rate $\eta = 1$ to the Perceptron loss, and $\boldsymbol{w}^* = \mathbf{0}$ is clearly a minimizer of the Perceptron loss. Which of the following statements related to these two facts is correct?

(A) Perceptron always converges to $\boldsymbol{w}^* = \mathbf{0}$.
(B) Perceptron converges to $\boldsymbol{w}^* = \mathbf{0}$ when it is initialized at $\mathbf{0}$.
(C) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because the learning rate does not decrease over time.
(D) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because SGD uses a stochastic gradient instead of the exact gradient.

Answer: C

Perceptron algorithm is SGD with $\eta = 1$ applied to perceptron loss:

Repeat:

- Pick a data point $\boldsymbol{x}_n$ uniformly at random
- If $\text{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) \neq y_n$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y_n \boldsymbol{x}_n$$

- **(C)** is correct:
  - The learning rate is a constant value 1 and it doesn't decrease.
  - This constant-step size SGD keeps bouncing and doesn't settle at the minimizer.

Perceptron: 09/12 lecture slides, page 18-19, 37-.

(4) Perceptron algorithm is applying SGD with learning rate $\eta = 1$ to the Perceptron loss, and $\boldsymbol{w}^* = \mathbf{0}$ is clearly a minimizer of the Perceptron loss. Which of the following statements related to these two facts is correct?

(A) Perceptron always converges to $\boldsymbol{w}^* = \mathbf{0}$.
(B) Perceptron converges to $\boldsymbol{w}^* = \mathbf{0}$ when it is initialized at $\mathbf{0}$.
(C) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because the learning rate does not decrease over time.
(D) Perceptron does not converge to $\boldsymbol{w}^* = \mathbf{0}$ because SGD uses a stochastic gradient instead of the exact gradient.

Answer: C

Perceptron algorithm is SGD with $\eta = 1$ applied to perceptron loss:

Repeat:

- Pick a data point $\boldsymbol{x}_n$ uniformly at random
- If $\text{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n) \neq y_n$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y_n \boldsymbol{x}_n$$

- **(D)** is incorrect:
  - The primary cause is the **constant learning rate**.

Perceptron: 09/12 lecture slides, page 18-19, 37-.

(5) Which of the following statements is correct when running SGD to minimize a non-convex function?

(A) SGD with random initialization can still get stuck at saddle points.
(B) SGD with random initialization escapes all saddle points with high probability.
(C) As long as we run SGD long enough, it will find a local minimum.
(D) As long as we run SGD long enough, it will find a global minimum.

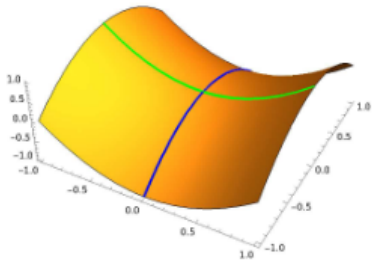
(5) Which of the following statements is correct when running SGD to minimize a non-convex function?

(A) SGD with random initialization can still get stuck at saddle points.
(B) SGD with random initialization escapes all saddle points with high probability.
(C) As long as we run SGD long enough, it will find a local minimum.
(D) As long as we run SGD long enough, it will find a global minimum.

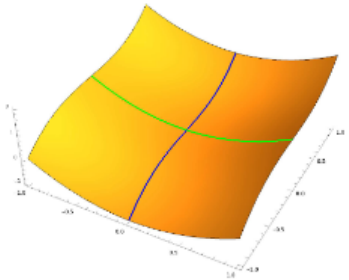
Answer: A

- **(A)** is correct:
  - SGD does not guarantee an escape from all saddle points.
  - It may get stuck at flat saddle points.
  - It may hover near saddle points for a long time before it eventually escapes.
- **(B)** is incorrect:

- for nonconvex objectives, can get stuck at local minimizers or "bad" saddle points (random initialization escapes "good" saddle points)



"good" saddle points



"bad" saddle points

(5) Which of the following statements is correct when running SGD to minimize a non-convex function?

(A) SGD with random initialization can still get stuck at saddle points.
(B) SGD with random initialization escapes all saddle points with high probability.
(C) As long as we run SGD long enough, it will find a local minimum.
(D) As long as we run SGD long enough, it will find a global minimum.

Answer: A

- **(C)** is incorrect:
  - It is possible that SGD can get stuck at some saddle points.
  - The statement is not necessarily correct.

- **(D)** is incorrect:
  - SGD converges to a stationary point that is not necessarily the global minimum.

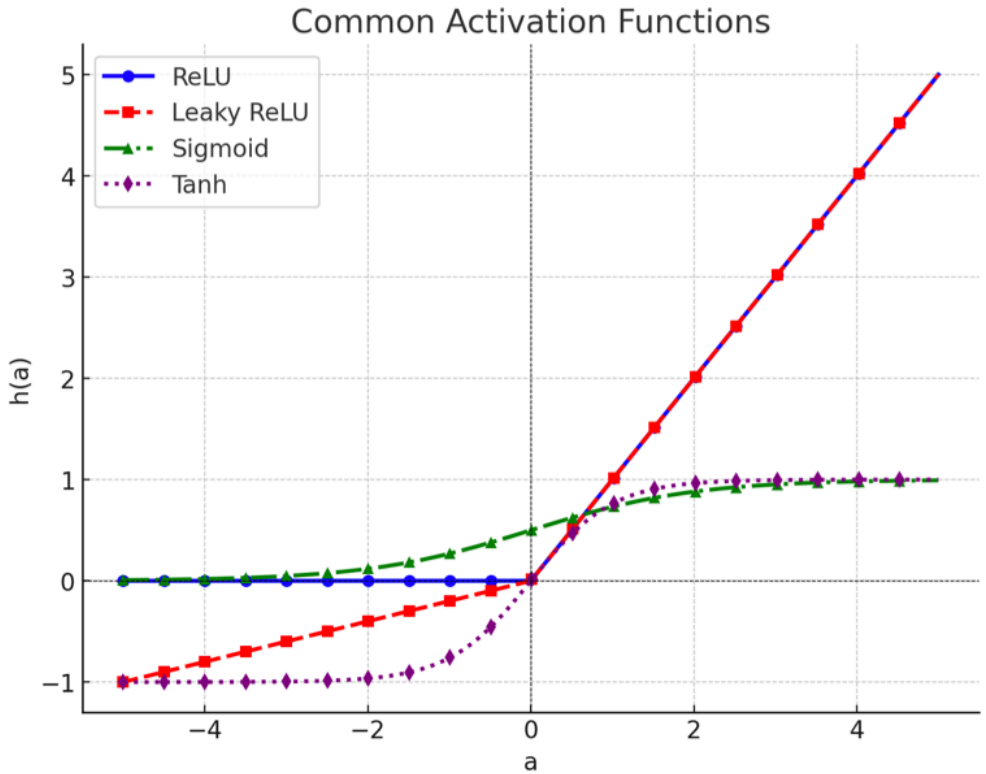
(6) Which of the following activation functions has a vanishing derivative when its input becomes too large (either too positive or too negative)? You might find the plot below useful.

(A) ReLU: $h(a) = \max\{a, 0\}$

(B) Leaky ReLU: $h(a) = \begin{cases} a & \text{if } a \geq 0 \\ 0.2a & \text{else} \end{cases}$

(C) Sigmoid: $h(a) = \frac{1}{1+e^{-a}}$

(D) TahH: $h(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$



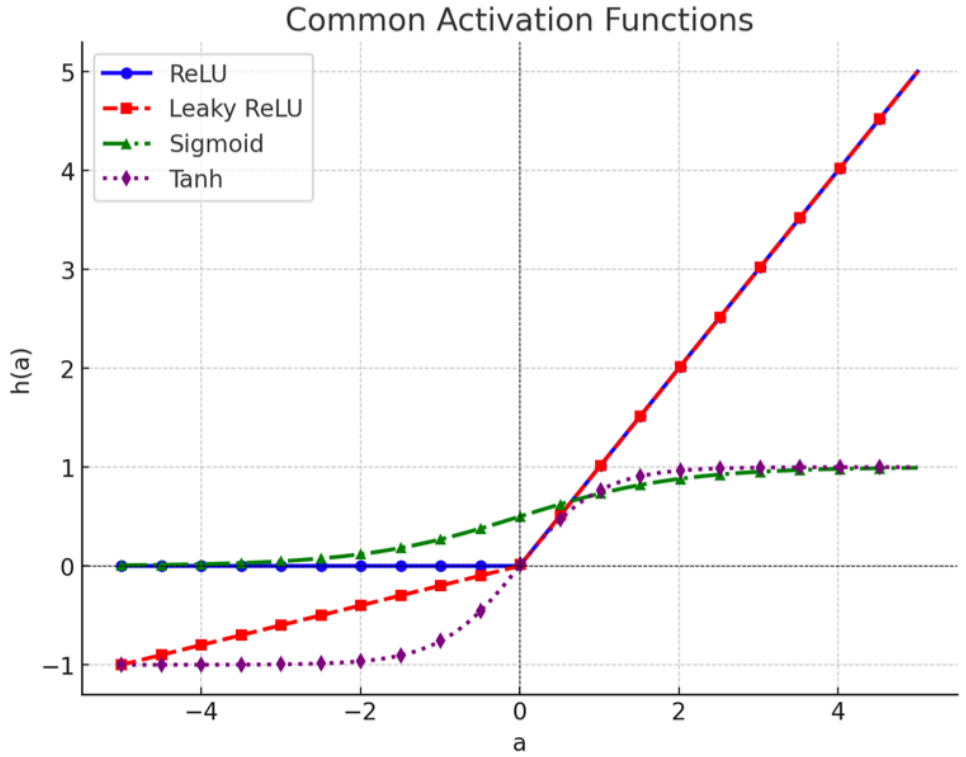
(6) Which of the following activation functions has a vanishing derivative when its input becomes too large (either too positive or too negative)? You might find the plot below useful.

(A) ReLU: $h(a) = \max\{a, 0\}$

(B) Leaky ReLU: $h(a) = \begin{cases} a & \text{if } a \geq 0 \\ 0.2a & \text{else} \end{cases}$

(C) Sigmoid: $h(a) = \frac{1}{1+e^{-a}}$

(D) TahH: $h(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$



Answer: ACD

- **Vanishing gradient problem**: during training, the gradients (used to update weights) become extremely small.
- **(A)**: derivative = 1 if a>0; derivative = 0 if a<0. For large negative inputs, derivative is 0.
- **(B)**: derivative = 1 if a>0; derivative = 0.2 if a<0.
- **(C)**: derivative is h(a) (1- h(a)).
  - For very large positive inputs, h(a) → 1, so derivative → 0.
  - For very large negative inputs, h(a) → 0, so derivative → 0.
- **(D)**: derivative is $1 - h(a)^2$, for very large positive (negative) inputs, $h(a)^2$ → 1, derivative → 0.

(7) Which of the following about neural nets is correct?

(A) A fully-connected neural net with a million neurons can approximate any continuous function.
(B) Data augmentation is useful for preventing overfitting when training a nerual net.
(C) Momentum and adaptive learning rate are useful for speeding up the training of a neural net.
(D) One should keep running Backpropagation until the training error goes down to 0.

(7) Which of the following about neural nets is correct?

(A) A fully-connected neural net with a million neurons can approximate any continuous function.
(B) Data augmentation is useful for preventing overfitting when training a nerual net.
(C) Momentum and adaptive learning rate are useful for speeding up the training of a neural net.
(D) One should keep running Backpropagation until the training error goes down to 0.

Answer: BC

- **(A)**: incorrect.

Universal approximation theorem (Cybenko, 89; Hornik, 91):

A feedforward neural net with a single hidden layer can approximate any continuous functions.

It might need a huge number of neurons though, and *depth helps!*

The theorem says you can approximate with such a neural net with certain size, but does not say there is a fixed-size that can approximate all.

- **(B)**: correct.

Overfitting is very likely since neural nets are too powerful.

Methods to overcome overfitting:

- data augmentation
- regularization
- dropout
- early stopping
- ...

09/26 lecture slides, page 14; page 24-25

(7) Which of the following about neural nets is correct?

(A) A fully-connected neural net with a million neurons can approximate any continuous function.
(B) Data augmentation is useful for preventing overfitting when training a nerual net.
(C) Momentum and adaptive learning rate are useful for speeding up the training of a neural net.
(D) One should keep running Backpropagation until the training error goes down to 0.

Answer: BC

- **(C)**: correct.

Important tricks to optimize neural nets

Many important tricks on top on Backprop

- **mini-batch**: randomly sample a batch of examples to form a stochastic gradient (common batch size: 32, 64, 128, etc.)
- **batch normalization**: normalize the inputs of each neuron over the mini-batch (to zero-mean and one-variance; *c.f.* Lec 1)
- **adaptive learning rate**: scale the learning rate of each parameter based on some moving average of the magnitude of the gradients
- **momentum**: make use of previous gradients (taking inspiration from physics)

09/26 lecture slides, page 22.

(7) Which of the following about neural nets is correct?

(A) A fully-connected neural net with a million neurons can approximate any continuous function.
(B) Data augmentation is useful for preventing overfitting when training a nerual net.
(C) Momentum and adaptive learning rate are useful for speeding up the training of a neural net.
(D) One should keep running Backpropagation until the training error goes down to 0.

Answer: BC

- **(D)**: incorrect.
  - Can lead to overfitting.
  - The goal is **not to** get perfect training accuracy, but to **generalize** well on unseen data.

(8) Suppose that a convolution layer takes a 4×6 image with 3 channels as input and outputs a $3 \times 4 \times 10$ volume. Which of the following is a possible configuration of this layer?

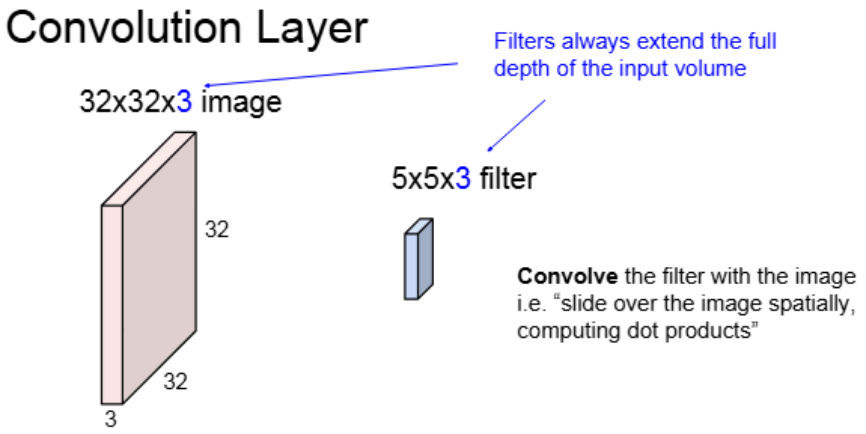
(A) Ten 2×3 filters with depth 10, stride 1, and no zero-padding.
(B) Ten 2×2 filters with depth 3, stride 2, and 1 pixel of zero-padding.
(C) Ten 2×2 filters with depth 3, stride 2, and 2 pixels of zero-padding.
(D) One 2×3 filter with depth 10, stride 1, and no zero-padding.

(8) Suppose that a convolution layer takes a 4×6 image with 3 channels as input and outputs a $3 \times 4 \times 10$ volume. Which of the following is a possible configuration of this layer?

(A) Ten 2×3 filters with depth 10, stride 1, and no zero-padding.
(B) Ten 2×2 filters with depth 3, stride 2, and 1 pixel of zero-padding.
(C) Ten 2×2 filters with depth 3, stride 2, and 2 pixels of zero-padding.
(D) One 2×3 filter with depth 10, stride 1, and no zero-padding.

Answer: B

- **Input: 4 x 6 x 3, output: 3 x 4 x 10**
- **(A)**: incorrect, filter depth = 10, which does not match the 3 channel input.



Input: a volume of size $W_1 \times H_1 \times D_1$

Hyperparameters:

- K filters of size $F \times F$
- stride S
- amount of zero padding P (for one side)

Output: a volume of size $W_2 \times H_2 \times D_2$ where

- $W_2 = (W_1 + 2P - F)/S + 1$
- $H_2 = (H_1 + 2P - F)/S + 1$
- $D_2 = K$

#parameters: $(F \times F \times D_1 + 1) \times K$ weights

Common setting: $F = 3, S = P = 1$

09/26 lecture slides, page 43.

(8) Suppose that a convolution layer takes a 4×6 image with 3 channels as input and outputs a $3 \times 4 \times 10$ volume. Which of the following is a possible configuration of this layer?

(A) Ten 2×3 filters with depth 10, stride 1, and no zero-padding.
(B) Ten 2×2 filters with depth 3, stride 2, and 1 pixel of zero-padding.
(C) Ten 2×2 filters with depth 3, stride 2, and 2 pixels of zero-padding.
(D) One 2×3 filter with depth 10, stride 1, and no zero-padding.

Answer: B

- **Input: 4 x 6 x 3, output: 3 x 4 x 10.**
- **(B)**: Correct.
  - Number of filters = 10, which matches the output depth D_{out} 10.
  - Filter depth = 3, which matches the 3 channel input D_{in}.
  - $W_{out} = \frac{4+2*1-2}{2} + 1 = 3$, which matches the output W.
  - $H_{out} = \frac{6+2*1-2}{2} + 1 = 4$, which matches the output H.

Input: a volume of size $W_1 \times H_1 \times D_1$

Hyperparameters:

- K filters of size $F \times F$
- stride S
- amount of zero padding P (for one side)

Output: a volume of size $W_2 \times H_2 \times D_2$ where

- $W_2 = (W_1 + 2P - F)/S + 1$
- $H_2 = (H_1 + 2P - F)/S + 1$
- $D_2 = K$

#parameters: $(F \times F \times D_1 + 1) \times K$ weights

Common setting: $F = 3, S = P = 1$

09/26 lecture slides, page 43.

(8) Suppose that a convolution layer takes a 4×6 image with 3 channels as input and outputs a $3 \times 4 \times 10$ volume. Which of the following is a possible configuration of this layer?

(A) Ten 2×3 filters with depth 10, stride 1, and no zero-padding.
(B) Ten 2×2 filters with depth 3, stride 2, and 1 pixel of zero-padding.
(C) Ten 2×2 filters with depth 3, stride 2, and 2 pixels of zero-padding.
(D) One 2×3 filter with depth 10, stride 1, and no zero-padding.

Answer: B

- **Input: 4 x 6 x 3, output: 3 x 4 x 10.**
- **(C)**: incorrect.
  - $W_{out} = \frac{4+2*2-2}{2} + 1 = 4$, which does not match the output W.
  - $H_{out} = \frac{6+2*2-2}{2} + 1 = 5$, which does not match the output H.
- **(D)**: incorrect.
  - Number of filters = 1, which does not match the 10 output depth.

Input: a volume of size $W_1 \times H_1 \times D_1$

Hyperparameters:

- K filters of size $F \times F$
- stride S
- amount of zero padding P (for one side)

Output: a volume of size $W_2 \times H_2 \times D_2$ where

- $W_2 = (W_1 + 2P - F)/S + 1$
- $H_2 = (H_1 + 2P - F)/S + 1$
- $D_2 = K$

#parameters: $(F \times F \times D_1 + 1) \times K$ weights

Common setting: $F = 3, S = P = 1$

09/26 lecture slides, page 43.

(9) How many parameters do we need to learn for the following network structure? An $8 \times 8 \times 3$ image input, followed by a convolution layer with 8 filters of size 3×3 (stride 1 and 1 pixel of zero-padding), then another convolution layer with 4 filters of size 2×2 (stride 2 and no zero-padding), and finally an average pooling layer with a 2×2 filter (stride 2 and no zero-padding). (Note: the depth of all filters are not explicitly spelled out, and we assume no bias/intercept terms used.)

(A) 144 (B) 344 (C) 348 (D) 360

(9) How many parameters do we need to learn for the following network structure? An $8 \times 8 \times 3$ image input, followed by a convolution layer with 8 filters of size 3×3 (stride 1 and 1 pixel of zero-padding), then another convolution layer with 4 filters of size 2×2 (stride 2 and no zero-padding), and finally an average pooling layer with a 2×2 filter (stride 2 and no zero-padding). (Note: the depth of all filters are not explicitly spelled out, and we assume no bias/intercept terms used.)

(A) 144 (B) 344 (C) 348 (D) 360

Answer: B

- Convolution layer 1:
  - Parameters = (3 * 3 * 3) * 8 = 216.
  - There is no "+1" since we assume no bias/intercept terms used.
- Convolution layer 2:
  - Parameters = (2 * 2 * 8) * 4 = 128
  - Previous convolution layer output D is 8.
- Average pooling layer:
  - Parameter = 0.
- 216 + 128 + 0 = 344.

Input: a volume of size $W_1 \times H_1 \times D_1$

Hyperparameters:

- K filters of size $F \times F$
- stride S
- amount of zero padding P (for one side)

Output: a volume of size $W_2 \times H_2 \times D_2$ where

- $W_2 = (W_1 + 2P - F)/S + 1$
- $H_2 = (H_1 + 2P - F)/S + 1$
- $D_2 = K$

#parameters: $(F \times F \times D_1 + 1) \times K$ weights

Common setting: $F = 3, S = P = 1$

Pooling

Similar to a filter, except

- depth is always 1
- different operations: average, L2-norm, max
- no parameters to be learned

Pooling: 09/26 lecture slides, page 45.

(9) How many parameters do we need to learn for the following network structure? An $8 \times 8 \times 3$ image input, followed by a convolution layer with 8 filters of size 3×3 (stride 1 and 1 pixel of zero-padding), then another convolution layer with 4 filters of size 2×2 (stride 2 and no zero-padding), and finally an average pooling layer with a 2×2 filter (stride 2 and no zero-padding). (Note: the depth of all filters are not explicitly spelled out, and we assume no bias/intercept terms used.)

(10) What is the final output dimension of the last question?

(A) $4 \times 4 \times 1$ (B) $4 \times 4 \times 4$ (C) $2 \times 2 \times 1$ (D) $2 \times 2 \times 4$

(9) How many parameters do we need to learn for the following network structure? An $8 \times 8 \times 3$ image input, followed by a convolution layer with 8 filters of size 3×3 (stride 1 and 1 pixel of zero-padding), then another convolution layer with 4 filters of size 2×2 (stride 2 and no zero-padding), and finally an average pooling layer with a 2×2 filter (stride 2 and no zero-padding). (Note: the depth of all filters are not explicitly spelled out, and we assume no bias/intercept terms used.)

(10) What is the final output dimension of the last question? **Answer: D**

(A) $4 \times 4 \times 1$ (B) $4 \times 4 \times 4$ (C) $2 \times 2 \times 1$ (D) $2 \times 2 \times 4$

- Convolution layer 1:
  - $W_{c1} = \frac{8+2*1-3}{1} + 1 = 8$
  - $H_{c1} = \frac{8+2*1-3}{1} + 1 = 8$
  - $D_{c1} = 8$

(8,8,8)

- Convolution layer 2:
  - $W_{c2} = \frac{8+2*0-2}{2} + 1 = 4$
  - $H_{c2} = \frac{8+2*0-2}{2} + 1 = 4$
  - $D_{c2} = 4$

(4,4,4)

- Average pooling layer:
  - $W_p = \frac{4+2*0-2}{2} + 1 = 2, H_p = \frac{4+2*0-2}{2} + 1 = 2, \boldsymbol{D_p = 4}$
  - Pooling does not change the depth.

(2,2,4)

Input: a volume of size $W_1 \times H_1 \times D_1$

Hyperparameters:

- K filters of size $F \times F$
- stride S
- amount of zero padding P (for one side)

Output: a volume of size $W_2 \times H_2 \times D_2$ where

- $W_2 = (W_1 + 2P - F)/S + 1$
- $H_2 = (H_1 + 2P - F)/S + 1$
- $D_2 = K$

#parameters: $(F \times F \times D_1 + 1) \times K$ weights

Common setting: $F = 3, S = P = 1$

(11) Suppose that k_1 and k_2 are two kernel functions with $\boldsymbol{\phi}_1 : \mathbb{R}^D \to \mathbb{R}^M$ and $\boldsymbol{\phi}_2 : \mathbb{R}^D \to \mathbb{R}^M$ being the corresponding feature maps. Which of the following is the corresponding feature map $\boldsymbol{\phi}$ for the product of k_1 and k_2 (which we know is also a kernel function)?

(A) $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}_1(\boldsymbol{x})\boldsymbol{\phi}_2(\boldsymbol{x})^\top \in \mathbb{R}^{M \times M}$
(B) $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}_1(\boldsymbol{x})^\top \boldsymbol{\phi}_2(\boldsymbol{x}) \in \mathbb{R}$
(C) $\boldsymbol{\phi}(\boldsymbol{x}) = (\boldsymbol{\phi}_1(\boldsymbol{x}), \boldsymbol{\phi}_2(\boldsymbol{x})) \in \mathbb{R}^{2M}$
(D) $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}_1(\boldsymbol{x}) \circ \boldsymbol{\phi}_2(\boldsymbol{x}) \in \mathbb{R}^M$ (element-wise product)

(11) Suppose that k_1 and k_2 are two kernel functions with $\phi_1 : \mathbb{R}^D \to \mathbb{R}^M$ and $\phi_2 : \mathbb{R}^D \to \mathbb{R}^M$ being the corresponding feature maps. Which of the following is the corresponding feature map ϕ for the product of k_1 and k_2 (which we know is also a kernel function)?

(A) $\phi(x) = \phi_1(x)\phi_2(x)^\top \in \mathbb{R}^{M \times M}$
(B) $\phi(x) = \phi_1(x)^\top \phi_2(x) \in \mathbb{R}$
(C) $\phi(x) = (\phi_1(x), \phi_2(x)) \in \mathbb{R}^{2M}$
(D) $\phi(x) = \phi_1(x) \circ \phi_2(x) \in \mathbb{R}^M$ (element-wise product)

Answer: A

Kernel functions

Definition: a function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is called a *kernel function* if there exists a function $\phi : \mathbb{R}^D \to \mathbb{R}^M$ so that for any $x, x' \in \mathbb{R}^D$,

$$k(x, x') = \phi(x)^T \phi(x')$$

Can be seen as a kind of similarity measure.

Kernel functions: 09/19 lecture slides, page 39.

(11) Suppose that k_1 and k_2 are two kernel functions with $\boldsymbol{\phi}_1 : \mathbb{R}^D \to \mathbb{R}^M$ and $\boldsymbol{\phi}_2 : \mathbb{R}^D \to \mathbb{R}^M$ being the corresponding feature maps. Which of the following is the corresponding feature map $\boldsymbol{\phi}$ for the product of k_1 and k_2 (which we know is also a kernel function)?

(A) $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}_1(\boldsymbol{x})\boldsymbol{\phi}_2(\boldsymbol{x})^\top \in \mathbb{R}^{M \times M}$
(B) $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}_1(\boldsymbol{x})^\top \boldsymbol{\phi}_2(\boldsymbol{x}) \in \mathbb{R}$
(C) $\boldsymbol{\phi}(\boldsymbol{x}) = (\boldsymbol{\phi}_1(\boldsymbol{x}), \boldsymbol{\phi}_2(\boldsymbol{x})) \in \mathbb{R}^{2M}$
(D) $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}_1(\boldsymbol{x}) \circ \boldsymbol{\phi}_2(\boldsymbol{x}) \in \mathbb{R}^M$ (element-wise product)

Answer: A

$$k_1(x, x')k_2(x, x') = \phi_1(x)^\top \phi_1(x')\, \phi_2(x)^\top \phi_2(x'). \qquad \text{By definition}$$

$$= \phi_1(x)^\top \big(\phi_1(x')\phi_2(x')^\top\big)\phi_2(x). \qquad s = \mathrm{tr}(s). \text{ where s is a scalar}$$

1 x M M x 1 1 x M M x 1

$$= \mathrm{tr}\big(\phi_1(x)^\top \phi_1(x')\phi_2(x')^\top \phi_2(x)\big). \qquad \text{Now we can use } \mathrm{tr}(AB) = \mathrm{tr}(BA).$$

$$= \mathrm{tr}\big(\phi_2(x)\phi_1(x)^\top \phi_1(x')\phi_2(x')^\top\big). \qquad \langle A, B\rangle = \mathrm{tr}(A^\top B) = \sum_{i,j} A_{ij}B_{ij}.$$

$$= \langle \phi_1(x)\phi_2(x)^\top,\ \phi_1(x')\phi_2(x')^\top \rangle. \qquad k(x, x') = \langle \phi(x), \phi(x')\rangle.$$

$$= \phi(x)^\top \phi(x')$$

Put A into a kernel function gives: $\langle \phi_1(x)\phi_2(x)^\top,\ \phi_1(x')\phi_2(x')^\top \rangle$

Kernel functions: 09/19 lecture slides, page 39.

(12) Which of the following on SVM is correct? In case you need a reminder, the primal formulation of SVM is $\min_{\boldsymbol{w},b,\{\xi_n\}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$ subject to $1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n$ and $\xi_n \geq 0$ for all n, and the dual formulation can be found in Problem 4.1.

(A) The larger the hyper-parameter C, the smaller the amount of L2 regularization.
(B) The larger the hyper-parameter C, the larger the amount of L2 regularization.
(C) To handle multiclass classification, one can use the one-versus-one reduction together with SVM.
(D) The α coefficients obtained by SVM satisfy $\sum_{n:y_n=+1} \alpha_n = \sum_{n:y_n=-1} \alpha_n$.

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \alpha_n y_n = 0 \quad \text{and} \quad \alpha_n \geq 0, \quad \forall\, n$$

(12) Which of the following on SVM is correct? In case you need a reminder, the primal formulation of SVM is $\min_{\boldsymbol{w},b,\{\xi_n\}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$ subject to $1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n$ and $\xi_n \geq 0$ for all n, and the dual formulation can be found in Problem 4.1.

(A) The larger the hyper-parameter C, the smaller the amount of L2 regularization.
(B) The larger the hyper-parameter C, the larger the amount of L2 regularization.
(C) To handle multiclass classification, one can use the one-versus-one reduction together with SVM.
(D) The α coefficients obtained by SVM satisfy $\sum_{n:y_n=+1} \alpha_n = \sum_{n:y_n=-1} \alpha_n$.

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \alpha_n y_n = 0 \quad \text{and} \quad \alpha_n \geq 0, \quad \forall\, n$$

Answer: ACD

- **(A):** correct.
  - A larger C means that we focus more on correctly classifying examples.
  - The optimizer would rather get weaker L2 regularization to achieve that.
- **(B):** incorrect.

One-versus-one: 9/19 lecture slides, page 23

(12) Which of the following on SVM is correct? In case you need a reminder, the primal formulation of SVM is $\min_{\boldsymbol{w},b,\{\xi_n\}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$ subject to $1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n$ and $\xi_n \geq 0$ for all n, and the dual formulation can be found in Problem 4.1.

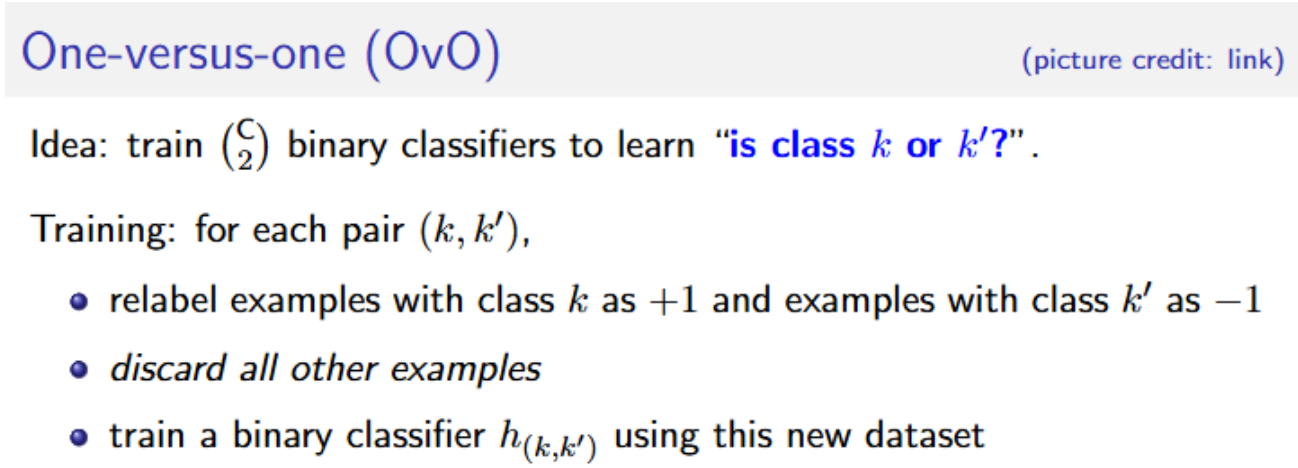
(A) The larger the hyper-parameter C, the smaller the amount of L2 regularization.
(B) The larger the hyper-parameter C, the larger the amount of L2 regularization.
(C) To handle multiclass classification, one can use the one-versus-one reduction together with SVM.
(D) The α coefficients obtained by SVM satisfy $\sum_{n:y_n=+1} \alpha_n = \sum_{n:y_n=-1} \alpha_n$.

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \alpha_n y_n = 0 \quad \text{and} \quad \alpha_n \geq 0, \quad \forall\, n$$

Answer: ACD

- **(C):** correct.
  - For each pair of classes, train a SVM to separate these two classes.



One-versus-one: 9/19 lecture slides, page 23

(12) Which of the following on SVM is correct? In case you need a reminder, the primal formulation of SVM is $\min_{\boldsymbol{w},b,\{\xi_n\}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$ subject to $1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n$ and $\xi_n \geq 0$ for all n, and the dual formulation can be found in Problem 4.1.

(A) The larger the hyper-parameter C, the smaller the amount of L2 regularization.
(B) The larger the hyper-parameter C, the larger the amount of L2 regularization.
(C) To handle multiclass classification, one can use the one-versus-one reduction together with SVM.
(D) The α coefficients obtained by SVM satisfy $\sum_{n:y_n=+1} \alpha_n = \sum_{n:y_n=-1} \alpha_n$.

Answer: ACD

$$\max_{\alpha_1,\ldots,\alpha_N} \quad \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \alpha_n y_n = 0 \quad \text{and} \quad \alpha_n \geq 0, \quad \forall\, n$$

- **(D):** correct.

$$\sum_{n=1}^{N} \alpha_n y_n = 0$$

$$\sum_{n=1}^{N} y_n \alpha_n = \sum_{y_n=+1} (+1)\alpha_n + \sum_{y_n=-1} (-1)\alpha_n.$$

$$= \sum_{y_n=+1} \alpha_n - \sum_{y_n=-1} \alpha_n = 0$$

$$\sum_{y_n=+1} \alpha_n = \sum_{y_n=-1} \alpha_n.$$

(13) Which of the following about decision trees is correct?

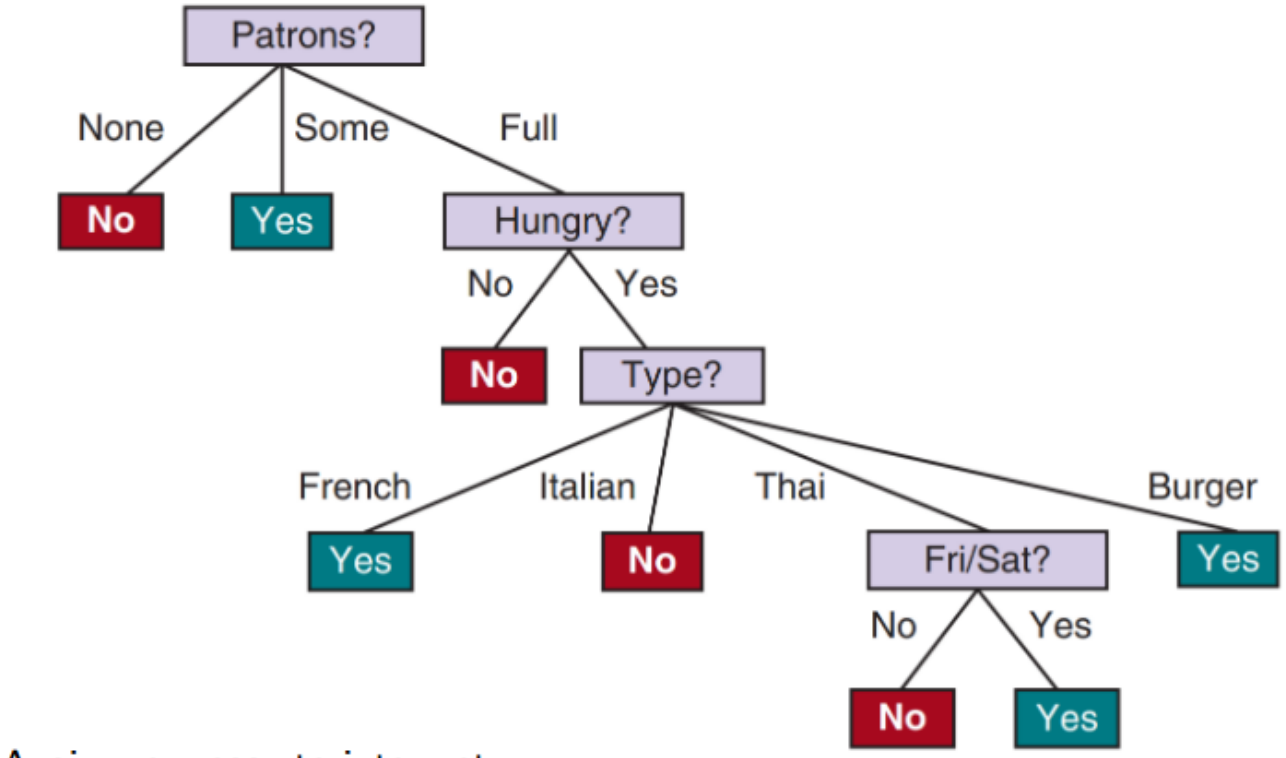
(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Shannon entropy can be used to measure the uncertainty of a node when building a decision tree.
(D) Random forest is a random ensemble of decision trees.

(13) Which of the following about decision trees is correct?

(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Shannon entropy can be used to measure the uncertainty of a node when building a decision tree.
(D) Random forest is a random ensemble of decision trees.

Answer: ABCD

- **(A):** correct.



Again, very easy to interpret.

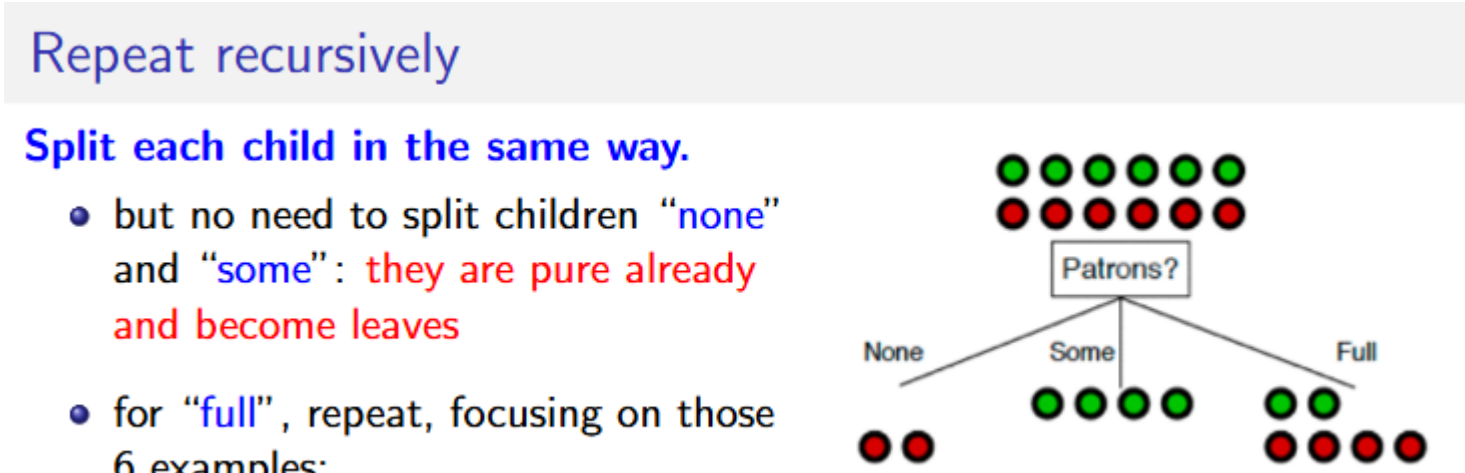
10/03 lecture slides, page 45

(13) Which of the following about decision trees is correct?

(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Shannon entropy can be used to measure the uncertainty of a node when building a decision tree.
(D) Random forest is a random ensemble of decision trees.

Answer: ABCD

- **(B):** correct.
  - The process of building a decision tree (e.g., ID3) involves recursively splitting nodes based on chosen features until a stopping criterion is met.



10/03 lecture slides, page 44

(13) Which of the following about decision trees is correct?

(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Shannon entropy can be used to measure the uncertainty of a node when building a decision tree.
(D) Random forest is a random ensemble of decision trees.

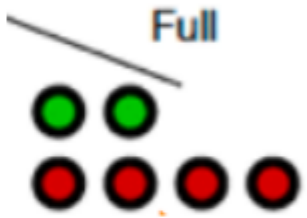
Answer: ABCD

- **(C):** correct.

Measure of uncertainty of a node

It should be **a function of the distribution of classes**

- e.g. a node with 2 positive and 4 negative examples can be summarized by a distribution P with $P(Y = +1) = 1/3$ and $P(Y = -1) = 2/3$



One classic uncertainty measure of a distribution is its *(Shannon) entropy*:

$$H(P) = -\sum_{k=1}^{C} P(Y = k) \log P(Y = k)$$

10/03 lecture slides, page 38

(13) Which of the following about decision trees is correct?

(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Shannon entropy can be used to measure the uncertainty of a node when building a decision tree.
(D) Random forest is a random ensemble of decision trees.

Answer: ABCD

- **(D):** correct.

Random forest is an **ensemble** of trees:

- each tree is built using *a bootstrapped dataset* (that is, a set of points randomly sampled from the training set with replacement)

(14) Consider a binary dataset with 50 positive examples and 50 negative examples. Decision stump $\mathcal{T}_1$ splits this dataset into two children where the left one has 20 positive examples and 40 negative examples, while another decision stump $\mathcal{T}_2$ results in a left child with 25 positive examples and 25 negative examples. Which of the following is correct? (Recall that entropy is defined as $H(P) = -\sum_{k=1}^{C} P(Y = k) \log P(Y = k)$.)

(A) The entropy of the left child of $\mathcal{T}_1$ is $\frac{1}{3} \log 3 + \frac{2}{3} \log \frac{3}{2}$.
(B) The entropy of the right child of $\mathcal{T}_1$ is $\frac{1}{4} \log 4 + \frac{3}{4} \log \frac{4}{3}$.
(C) The entropy of either child of $\mathcal{T}_2$ is $\frac{1}{2} \log 2 + \frac{1}{2} \log 2$.
(D) Based on conditional entropy, $\mathcal{T}_1$ is a better split than $\mathcal{T}_2$.

(14) Consider a binary dataset with 50 positive examples and 50 negative examples. Decision stump $\mathcal{T}_1$ splits this dataset into two children where the left one has 20 positive examples and 40 negative examples, while another decision stump $\mathcal{T}_2$ results in a left child with 25 positive examples and 25 negative examples. Which of the following is correct? (Recall that entropy is defined as $H(P) = -\sum_{k=1}^{C} P(Y = k) \log P(Y = k)$.)

(A) The entropy of the left child of $\mathcal{T}_1$ is $\frac{1}{3}\log 3 + \frac{2}{3}\log\frac{3}{2}$.
(B) The entropy of the right child of $\mathcal{T}_1$ is $\frac{1}{4}\log 4 + \frac{3}{4}\log\frac{4}{3}$.
(C) The entropy of either child of $\mathcal{T}_2$ is $\frac{1}{2}\log 2 + \frac{1}{2}\log 2$.
(D) Based on conditional entropy, $\mathcal{T}_1$ is a better split than $\mathcal{T}_2$.

Answer: ABCD

- **(A):** correct.
  - Left child node: 20 +, 40 -, => p(+) = 20/60 = 1/3, p(-) = 40/60 = 2/3.

$$H_L = -\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right) = \frac{1}{3}\log 3 + \frac{2}{3}\log\frac{3}{2}.$$

- **(B)**: correct.
  - Right child node (the rest of examples): 30 +, 10 -, => p(+) = 3/4, p(-) = 1/4.

$$H_R = -\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) = \frac{1}{4}\log 4 + \frac{3}{4}\log\frac{4}{3}.$$

(14) Consider a binary dataset with 50 positive examples and 50 negative examples. Decision stump $\mathcal{T}_1$ splits this dataset into two children where the left one has 20 positive examples and 40 negative examples, while another decision stump $\mathcal{T}_2$ results in a left child with 25 positive examples and 25 negative examples. Which of the following is correct? (Recall that entropy is defined as $H(P) = -\sum_{k=1}^{C} P(Y=k) \log P(Y=k)$.)

(A) The entropy of the left child of $\mathcal{T}_1$ is $\frac{1}{3}\log 3 + \frac{2}{3}\log\frac{3}{2}$.
(B) The entropy of the right child of $\mathcal{T}_1$ is $\frac{1}{4}\log 4 + \frac{3}{4}\log\frac{4}{3}$.
(C) The entropy of either child of $\mathcal{T}_2$ is $\frac{1}{2}\log 2 + \frac{1}{2}\log 2$.
(D) Based on conditional entropy, $\mathcal{T}_1$ is a better split than $\mathcal{T}_2$.

measured by the **conditional entropy**:

$$H(Y \mid A) = \sum_a P(A=a) H(Y \mid A=a) = \sum_a P(A=a) \left(-\sum_{k=1}^{C} P(Y \mid A=a) \log P(Y \mid A=a) \right)$$

Pick the feature that leads to the smallest conditional entropy.

Answer: ABCD

- **(C):** correct.
  - Each child node: 25 +, 25 -, => p(+) = p(-) = 1/2.

$$H = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) = \frac{1}{2}\log 2 + \frac{1}{2}\log 2$$

- **(D):** correct.
  - T1:

$$0.6\, H_L + 0.4\, H_R = 0.6(0.6365) + 0.4(0.5623) = 0.607.$$

  - T2:

$$0.5 \cdot H + 0.5 \cdot H = H = \log 2 \quad = 0.693 > 0.607$$

Conditional entropy: 10/03 lecture slides, page 2

(15) Which of the following about boosting is correct?

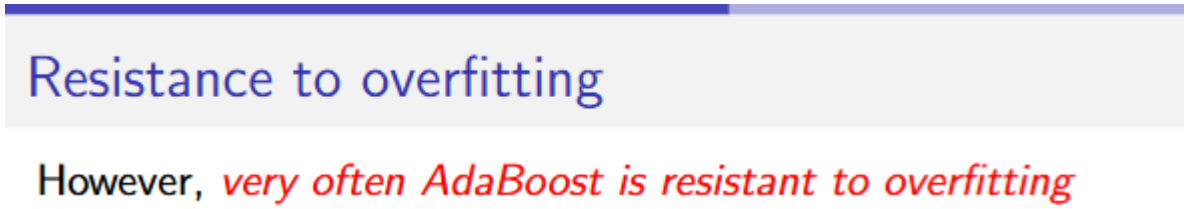
(A) Boosting is guaranteed to achieve zero training error.
(B) AdaBoost is often resistant to overfitting.
(C) AdaBoost never overfits.
(D) The idea of boosting is to repeatedly reweight the examples so that “difficult” ones get more attention.

(15) Which of the following about boosting is correct?

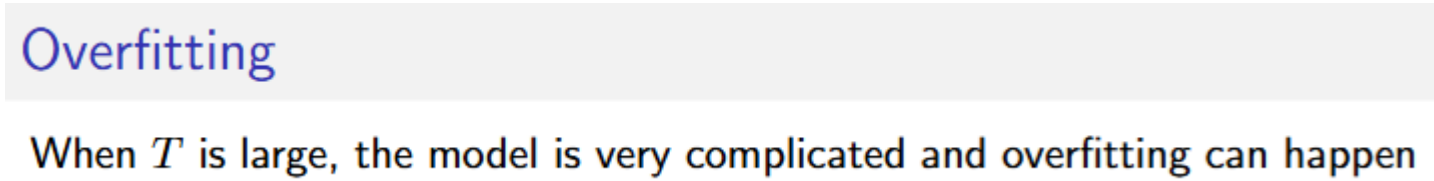
(A) Boosting is guaranteed to achieve zero training error.
(B) AdaBoost is often resistant to overfitting.
(C) AdaBoost never overfits.
(D) The idea of boosting is to repeatedly reweight the examples so that "difficult" ones get more attention.

Answer: BD

- **(A):** incorrect. It is not guaranteed.
- **(B):** correct.



- **(C):** incorrect.



- **(D):** correct.

10/03 lecture slides, page 58, page 59