Name+Email ID Jon-Michael Hoang | hoang339

CSCI 5521: Machine Learning Fundamentals (Spring 2022)[1]

Final Exam

Due on Gradescope by 12pm, May 11

Instructions:

- The final exam has 5+1 questions, 100+2 points, on 8 pages, including one extra credit problem worth 2 points.
- Please **write your name & ID on this cover page**.
- **For full credit, show how you arrive at your answers**.

1. (**30 points**) In I-IV, select the correct option(s) (it is not necessary to explain).

I. Select all the option(s) that are **loss functions**:

(a) Cross entropy (b) Maximum likelihood (c) Squared error (d) 0-1 loss (e) Hinge loss

II. Select all the option(s) that are **activation functions**:

(a) Sigmoid (b) ReLu (c) Hyperbolic tangent (d) Softmax (e) Entropy

III. Select all the option(s) that are true when describing **kernel methods**:

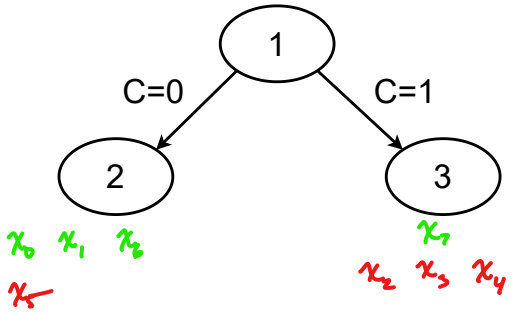
(a) Kernel methods are designed to reduce overfitting.
(b) Kernel methods only work with Support Vector Machines (SVMs).
(c) Common kernels include polynomial, Gaussian, and Lagrange functions.
(d) Kernel methods are designed to map data into better representational space.
(e) Linear SVMs always work better than kernel SVMs.

IV. Select all the option(s) that are true about **overfitting**:

(a) A bigger number of hidden nodes in a Multilayer Perceptron helps reduce overfitting with the same amount of data.
(b) A smaller number of layers in a Multilayer Perceptron helps reduce overfitting with the same amount of data.
(c) Pruning helps control overfitting in decision trees.
(d) Random forest reduces overfitting compared with decision tree.
(e) Data augmentation (e.g., creating copies of image data that are rotated or scaled versions of the original ones) helps reduce overfitting with the same amount of original data.

[1]Instructor: Catherine Qi Zhao. TA: Prithvi Raj Botcha, Shi Chen, Suzie Hoops, James Yang, Yifeng Zhang. Email: csci5521.s2022@gmail.com

2. (**10 points**) Given the decision tree in the figure below, the node 1 was split using feature C. Now suppose we wish to split node 3. What is the feature that you will be using to split? Show your work.



A	B	C	Class
0	0	0	0
0	1	0	0
1	1	1	1
0	1	1	1
1	1	1	1
1	1	0	1
0	0	0	0
0	1	1	0

B=0 B=1

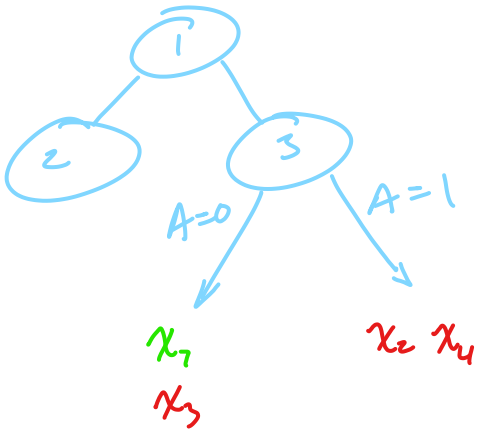
χ_7

χ_2 χ_3 χ_4

$Entropy_{3B}$:

$-\left[0 + \frac{4}{4}\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right)\right]$

≈ 0.811



χ_7 χ_3

χ_2 χ_4

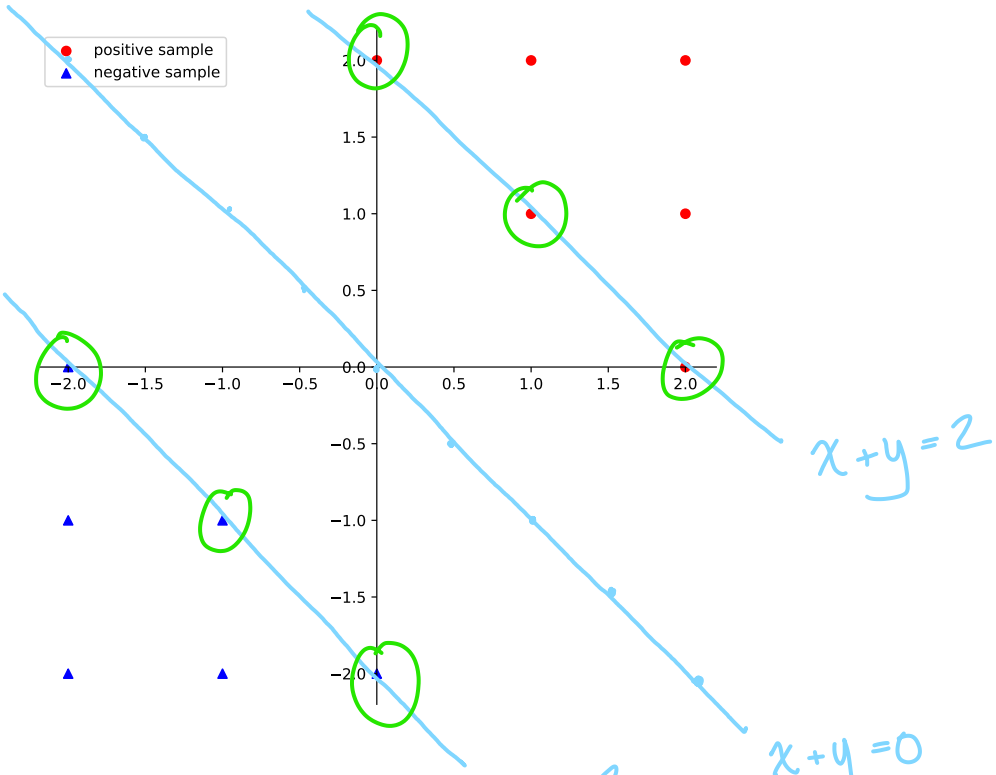
$Entropy_{3A}$:

$-\left[\frac{2}{4}\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) + \frac{2}{4}\left(0 + \frac{2}{2}\log_2\frac{2}{2}\right)\right]$

$= 0.500$

∴ Since $Entropy_{3B} > Entropy_{3A}$ ($\approx 0.811 > 0.500$), we would split node 3 by feature A

3. (**15 points**) Suppose we are training a linear SVM on a tiny dataset of 12 points shown in the figure below. Samples with positive labels are (1, 1), (2, 2), (1,2), (2,1), (2,0), (0,2) (denoted as red dots) and samples with negative labels are (-1, -1), (-2, -2), (-1, -2), (-2,-1), (-2,0), (0,-2) (denoted as blue triangles).



(a) Draw the maximum-margin hyperplane.

(b) Circle the support vectors.

(c) Pick one positive and one negative sample, and calculate their distances to the hyperplane.

• : $\frac{|1+1|}{\sqrt{1^2+1^2}} = \frac{2}{\sqrt{2}}$ ▲ : $\frac{|-1+(-1)|}{\sqrt{1^2+1^2}} = \frac{2}{\sqrt{2}}$

* $g(x) = w^T x$ & dist = $\frac{|g(x)|}{\|w\|}$

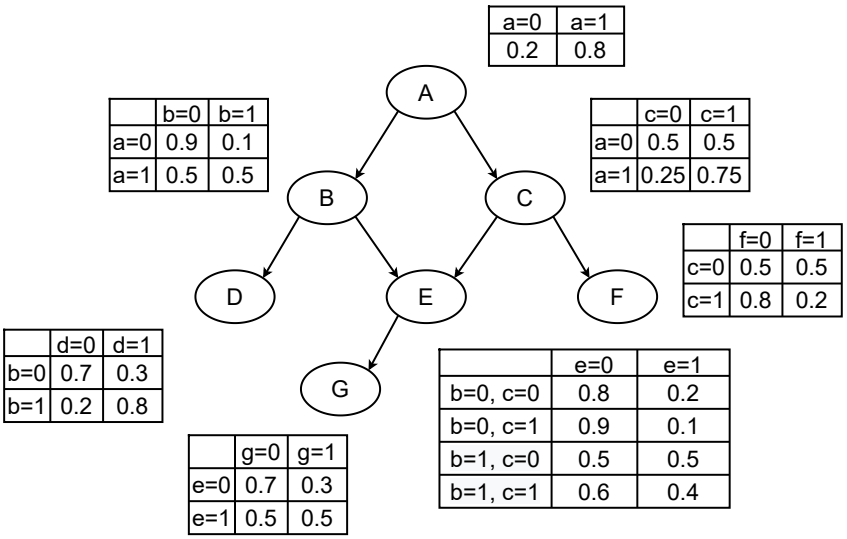
(d) If a new sample (-1, 1) comes as a negative sample on top of the original 12 points, select all the option(s) that are true:

i. The decision boundary would change. → bc it's literally on the hyperplane & will flip the boundary

ii. Kernel SVM would be a good option with the new data.

→ No, bc the new data's still linearly separable.

4. (**25 points**) Consider the Bayesian Network below:



a=0	a=1
0.2	0.8

	b=0	b=1
a=0	0.9	0.1
a=1	0.5	0.5

	c=0	c=1
a=0	0.5	0.5
a=1	0.25	0.75

	f=0	f=1
c=0	0.5	0.5
c=1	0.8	0.2

	d=0	d=1
b=0	0.7	0.3
b=1	0.2	0.8

	e=0	e=1
b=0, c=0	0.8	0.2
b=0, c=1	0.9	0.1
b=1, c=0	0.5	0.5
b=1, c=1	0.6	0.4

	g=0	g=1
e=0	0.7	0.3
e=1	0.5	0.5

Note: The numerical values of the probabilities are for part (e). You do not need to use them for (a)-(d).

(a) Find the joint probability $P(A, B, C, D, E, F, G)$ as the product of conditional probabilities, according to the graphical model given above.

$$P(A,B,C,D,E,F,G) = P(A)P(B|A)P(C|A)P(D|B)P(E|B,C)P(F|C)P(G|E)$$

(b) List all conditional independence given the graph.

$$P(A)P(B|A)P(C|A,B)P(D|A,B,C)P(E|A,B,C,D)P(F|A,B,C,D,E)P(G|A,B,C,D,E,F)$$ ← Chain Rule

- $C \perp\!\!\!\perp B \mid A$
- $D \perp\!\!\!\perp \{A, C\} \mid B$
- $E \perp\!\!\!\perp \{A, D\} \mid \{B, C\}$
- $F \perp\!\!\!\perp \{A, B, D, E\} \mid C$
- $G \perp\!\!\!\perp \{A, B, C, D, F\} \mid E$

(c) Show how to find the conditional probability $P(A|C)$.

$$P(A|C) = \frac{P(C|A)P(A)}{P(C)}$$ Baye's Theorem

$$= \frac{P(C|A)P(A)}{P(C|A)P(A) + P(C|\neg A)P(\neg A)}$$

(d) Show how to find the marginal probability $P(A, D)$.

$P(A,D) \Rightarrow \sum_B P(A,B,D) = \sum_B P(A)P(B|A)P(D|B)$

$= \boxed{P(A)\sum_B P(B|A)P(D|B)}$

(e) Using the conditional probability distribution (CPD) tables in the figure, find:

i. $P(a = 1|c = 0)$

$$\frac{P(C=0|A=1)P(A=1)}{P(C=0|A=1)P(A=1) + P(C=0|\neg A=1)P(\neg A=1)} = \frac{0.25 * 0.8}{(0.25*0.8)+(0.5*0.2)}$$

$= \boxed{\frac{2}{3} \approx 0.\overline{6}7}$

ii. $P(a = 1, d = 0)$

$$P(A=1)\sum_B P(B|A=1)P(D=0|B) = P(A=1)\big(P(B=0|A=1)P(D=0|B=0) + P(B=1|A=1)P(D=0|B=1)\big)$$

$= 0.8 * (0.5 * 0.7 + 0.5 * 0.2)$

$= \boxed{\frac{9}{25} \approx 0.36}$

iii. $P(a = 1, b = 0, c = 0, d = 0, e = 0, f = 0, g = 1)$

$P(A=1)P(B=0|A=1)P(C=0|A=1)P(D=0|B=0)P(E=0|B=0,C=0)P(F=0|C=0)P(G=1|E=0)$

$= 0.8 * 0.5 * 0.25 * 0.7 * 0.8 * 0.5 * 0.3$

A B C D E F G

$= \boxed{0.0084 \approx \frac{21}{2500}}$

iv. $P(b = 0, c = 0, e = 0, f = 0, g = 1|a = 1, d = 0)$

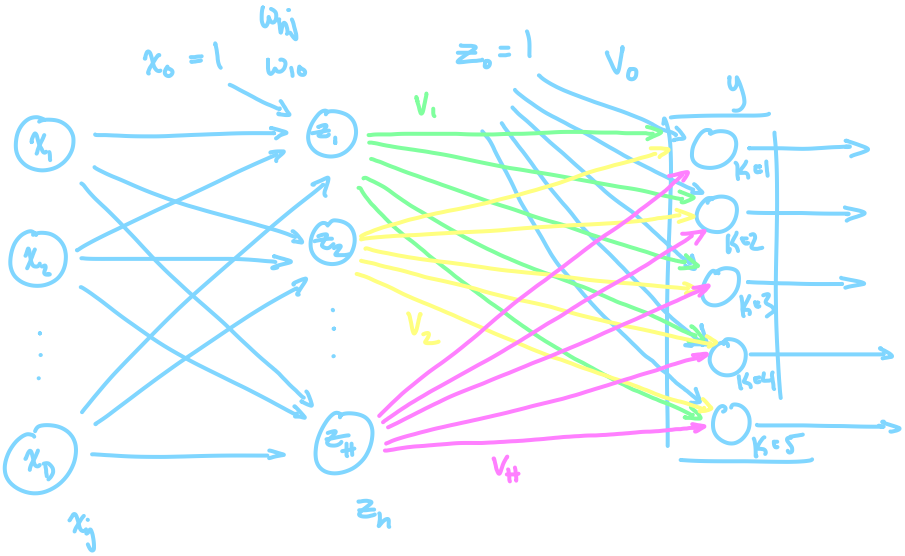
$$\frac{P(A=1,B=0,C=0,D=0,E=0,F=0,G=1)}{P(A=1,D=0)} = \frac{0.0084}{0.36} = \frac{\frac{21}{2500}}{\frac{9}{25}}$$

$\approx \boxed{0.2\overline{33} = \frac{7}{300}}$

5. (**20 points**) Consider a Multi-layer Perceptron (MLP) for multi-class classification of K=5 categories with 5 output units, where each hidden unit uses a hyperbolic tangent function such that $z_h^t = \tanh(\sum_{j=1}^{D} w_{hj} x_j^t + w_{h0})$. The output unit uses a softmax activation function such that $y_i^t = \frac{\exp(\sum_h v_{ih} z_h^t + v_{i0})}{\sum_j \exp(\sum_h v_{jh} z_h^t + v_{j0})}$. The error function is given below:

$$E(W, V|X) = -\underbrace{\sum_{t=1}^{N}\sum_{i=1}^{K} r_i^t \log y_i^t}_{E_1} + \underbrace{\frac{\lambda}{2}\sum_{h=1}^{H} ||w_h||_2^2}_{E_2} + \underbrace{\frac{\sigma}{2}\sum_{k=1}^{K} ||v_k||_2^2}_{E_3} \quad (1)$$

(a) Draw the Multi-layer Perceptron showing: input values $x_0 ... x_D$, output of the hidden units $z_0 ... z_H$, weights W and V, and the outputs.



(b) Derive the Forward Step equation.

$$z_h^t = \tanh\left(\sum_{j=1}^{D} w_{hj} x_j^t + w_{h0}\right)$$

$$y_i^t = \text{softmax}\left(\frac{\exp\left(\sum_h v_{ih} z_h^t + v_{i0}\right)}{\sum_j \exp\left(\sum_h v_{jh} z_h^t + v_{j0}\right)}\right)$$

(c) Derive the Backward Step equation for w_{hj}.
Hint:

- $\tanh'(x) = 1 - \tanh^2(x)$
- Given the softmax function $f(\alpha_i) = \frac{exp(\alpha_i)}{\sum_j exp(\alpha_j)}$, then $\frac{\partial f(\alpha_i)}{\partial \alpha_j} = f(\alpha_i)(\delta_{ij} - f(\alpha_j))$, in which δ_{ij} is an indicator function, such that $\delta_{ij} = 1$ if $i = j$, and 0 otherwise.

$$\frac{\partial E_1}{\partial w_{hj}} = \frac{\partial E_1}{\partial y_i^t} \frac{\partial y_i^t}{\partial z_h^t} \frac{\partial z_h^t}{\partial w_{hj}} = -\sum_t^N \sum_i^K \left(\frac{r_i^t}{y_i^t} * \left(y_i^t (1 - y_j^t)\right) * \left(1 - \tanh^2\left(\sum_{j=1}^{D} w_{hj} x_j^t + w_{h0}\right)\right)\right)$$

for $i = j$

$$= -\sum_t^N \sum_i^K \left(\frac{r_i^t}{y_i^t} * \left(-y_j^t y_i^t\right) * \left(1 - \tanh^2\left(\sum_{j=1}^{D} w_{hj} x_j^t + w_{h0}\right)\right)\right)$$

for $i \neq j$

$$\frac{\partial E_2}{\partial w_{hj}} = \boxed{\lambda w_{hj}}$$

$$\frac{\partial E_3}{\partial w_{hj}} = \boxed{0}$$

$$\therefore \Delta w_{hj} = -\eta \frac{\partial E}{\partial w_{hj}} = -\left[-\sum_t^N \sum_i^K \frac{r_i^t}{y_i^t} * \left(y_i^t (\delta_{ij} - y_j^t)\right) * \left(1 - \tanh^2\left(\sum_j w_{hj} x_j^t + w_{h0}\right)\right) + \lambda w_{hj}\right]$$

* converging $i=j$ & $i \neq j$ assumptions into δ_{ij} bc I have no more space

6. (**2 points, extra credit**) Charlotte is a machine learning scientist in a self-driving company. The company is developing an obstacle detection algorithm. Charlotte's team has access to over 1,000,000 images each with label, and the data are in the form of raw images without any features extracted.

   (a) Suggest a machine learning method she could use, and explain your suggestion.

A machine learning method that she could use would be the Graphical Model. The reason being so is because the relationships of the pixels between themselves could be modeled as a graph -- with similar and/or related pixels having stronger values between each other, holding as well for the opposite asusmption. Since each image has a graph, the system could learn off of those like features and given labels, it would be able to produce appropriate outputs for each image.

   (b) What if only 1,000 out of the 1,000,000 images have labels? Suggest a method and explain your suggestion.

A method would be to develop some sort of layered, graphical network. Similar with answer (a), we add the extra layer of building graphs from the 1,000 images with labels to learn and predict the other 999,000 images to then be able to produce appropriate outputs during the validation stage. By just learning the 1,000 images and not incorporating the other 999,000, we risk underfitting. But by also using those 999k during training and validation(?) stages, we could increase the probability of a successful probability in the final stages.