6.S966: Exam 1, Spring 2024

Solutions

- This is a closed book exam. One page (8 1/2 in. by 11 in) of notes, front and back, are permitted. Calculators are not permitted.
- The total exam time is 1 hours and 20 minutes.
- The problems are not necessarily in any order of difficulty.
- Record all your answers in the places provided. If you run out of room for an answer, continue on a blank page and mark it clearly.
- If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.
- If you absolutely *have* to ask a question, come to the front.
- **Write your name on every piece of paper.**

Name: ______________________ MIT Email: ______________________

Question	Points	Score
1	15	
2	28	
3	10	
4	27	
5	20	
Total:	100	

Name: ______________________________

Isomorphisms of Multiplication Tables

1. (15 points) Below are three group multiplication tables. Here, we are using various symbols instead of numbers, so it is clear in your answers, which table you are giving answers to.

	♋	♌	♊	♒
♋	♌	♋	♒	♊
♌	♋	♌	♊	♒
♊	♒	♊	♌	♋
♒	♊	♒	♋	♌

Table 1

	♣	♢	♠	♡
♣	♡	♠	♢	♣
♢	♠	♡	♣	♢
♠	♢	♣	♡	♠
♡	♣	♢	♠	♡

Table 2

	●	▲	■	★
●	★	■	●	▲
▲	■	★	▲	●
■	●	▲	■	★
★	▲	●	★	■

Table 3

(a) Give the symbol that corresponds to the identity element for each table

Solution: ♌, ♡, ■

(b) For each symbol, give it's inverse.

Solution:

♋$^{-1}$ → ♋
♌$^{-1}$ → ♌
♊$^{-1}$ → ♊
♒$^{-1}$ → ♒

♣$^{-1}$ → ♣
♢$^{-1}$ → ♢
♠$^{-1}$ → ♠
♡$^{-1}$ → ♡

●$^{-1}$ → ▲
▲$^{-1}$ → ●
■$^{-1}$ → ■
★$^{-1}$ → ★

(c) Two of the three tables represent groups that are isomorphic. Which table represents a group that is not isomorphic to the other two? Explain your reasoning.

Solution: The easiest way to tell is that the order of all the elements of the group in Table 1 and Table 2 is 2 (all elements are their own inverses). This is not the case for Table 3.

Name: ______________________________

Parsing Proofs

2. (28 points) In this problem, we will present a single step of some of the proofs shown in class, exercises, or notes and ask what properties of matrices or groups or lemmas or theorem, allows us to take this step.

   (a) In our proof of Schur's Lemma Part 2, we are trying to prove the properties of matrix M in the equation

$$MD^{(1)}(R) = D^{(2)}(R)M, \tag{1}$$

   for two unitary irreducible representations (irreps) of a group G, $D^{(1)}$ and $D^{(2)}$ and for all $R \in G$. We start by taking the conjugate transpose of this equations.

$$[MD^{(1)}(R)]^\dagger = [D^{(2)}(R)M]^\dagger \tag{2}$$

$$= [D^{(1)}(R)]^\dagger M^\dagger = M^\dagger [D^{(2)}(R)]^\dagger \tag{3}$$

$$= D^{(1)}(R^{-1})M^\dagger = M^\dagger D^{(2)}(R^{-1}), \tag{4}$$

   i. What property of unitary matrices did we use between line (3) and (4)?

   > **Solution:** If the representations are unitary $U^{-1} = U^\dagger = [U^*]^T$, thus $[D(R)]^\dagger = D(R^{-1})$

   ii. We multiply line (4) on the left by M

$$MD^{(1)}(R^{-1})M^\dagger = MM^\dagger D^{(2)}(R^{-1}). \tag{5}$$

   Which line can we use above in lines (1 - 4) to arrive at

$$D^{(2)}(R^{-1})MM^\dagger = MM^\dagger D^{(2)}(R^{-1})? \tag{6}$$

   Explain your reasoning.

   > **Solution:** We can use line (1) to substitute $MD^{(1)}(R^{-1})$ for $D^{(2)}(R^{-1})M$, Because this holds for all $R \in G$.

   iii. Using the lemmas and theorems that you've learned in this class what can you say about the matrix $MM^\dagger$ in line (6)? Explain your reasoning.

   > **Solution:** Using Schur's Lemma Part 1, we know that $MM^\dagger$ must be constant matrices, i.e. $c\delta_{\mu\nu}$

   (b) In our proof of the Wonderful Orthogonality Theorem we started with the following

$$M = \sum_R D^{(\Gamma_{j'})}(R) X D^{(\Gamma_j)}(R^{-1}) \tag{7}$$

where $D^{(\Gamma_{j'})}$ and $D^{(\Gamma_j)}$ are representations of group G and X is an arbitrary matrix of size $l_{j'} \times l_j$, where $l_{j'}$ and l_j dimensions of the representations, respectively. We then multiplied by $D^{(\Gamma_{j'})}(S)$ on both sides

$$D^{(\Gamma_{j'})}(S)M = \sum_R D^{(\Gamma_{j'})}(S)D^{(\Gamma_{j'})}(R)XD^{(\Gamma_j)}(R^{-1}) \tag{8}$$

$$= \sum_R D^{(\Gamma_{j'})}(SR)XD^{(\Gamma_j)}(R^{-1}S^{-1})D^{(\Gamma_j)}(S) \tag{9}$$

$$= MD^{(\Gamma_j)}(S) \tag{10}$$

i. What property of groups did we use to go from line (8) to line (9)? Explain your reasoning.

> **Solution:** Because every element as an inverse, we are free to insert a copy of the identity and express it as (the representation of) an element acting on its inverse. Furthermore, groups are associative, so $a(be) = (ab)e$, so we are free to regroup our representation.

ii. What theorem did we use to go from line (9) to line (10)? Explain your reasoning.

> **Solution:** The Rearrangement Theorem. M is defined as a sum over the group and since $sG = G$, the sum over R is equivalent whether the arguments are R or SR.

(c) Recall that we defined group correlation as:

$$[f \star \psi](x, k) = \sum_{y \in \mathbb{Z}^2} \sum_{i \in G} f(y, i)\psi(k^{-1}(y - x), k^{-1}i) \tag{11}$$

We proved that this definition (11) satisfies equivalence as follows:

$$[L_g(f) \star \psi](x, k) = \sum_{y \in \mathbb{Z}^2} \sum_{i \in G} f(g^{-1}y, g^{-1}i)\psi(k^{-1}(y - x), k^{-1}i) \tag{12}$$

let $y' = g^{-1}y$ and $i' = g^{-1}i$

$$[L_g(f) \star \psi](x, k) = \sum_{y' \in \mathbb{Z}^2} \sum_{i' \in G} f(y', i')\psi(k^{-1}(gy' - x), k^{-1}gi') \tag{13}$$

$$= \sum_{y' \in \mathbb{Z}^2} \sum_{i' \in G} f(y', i')\psi((g^{-1}k)^{-1}(y' - g^{-1}x), (g^{-1}k)^{-1}i') = L_g([f \star \psi])(x, k) \tag{14}$$

i. What properties have we relied on when when replacing $\sum_i$ with $\sum_{i'}$ in going between (12) and (13) ?

> **Solution:** Rearrangement theorem

ii. What properties / assumptions have we made when replacing $\sum_y$ with $\sum_{y'}$ in going between (12) and (13)? If there's any assumption made here, can you try to defend why it's a reasonable assumption?

> **Solution:** We assume that the lattice $\{gz : z \in \mathcal{Z}^2\} = \mathcal{Z}^2$. Note that this is required also for the definition of group correlation, as f and Ψ are both stored as a grid of values. So we need the action of G to be such that it takes the lattice to itself. We accepted two explanations for this assumption, most other explanations got partial credit:
>
> 1. If the resolution is high enough, we can approximate values that are not on the lattice by values on it.
> 2. g should be an action on $\mathcal{Z}^2$ and that's a property of group action on sets

Name: ______________________________

Sudoku for Group Theorists

3. (10 points) Complete the character tables below using First and Second Wonderful Orthogonal Theorems for Character given below, respectively.

$$\sum_R \chi^{(\Gamma_j)}(R)\chi^{(\Gamma_{j'})}(R^{-1}) = \sum_k N_k \chi^{(\Gamma_j)}(C_k)[\chi^{(\Gamma_{j'})}(C_k)]^* = h\delta_{\Gamma_j\Gamma_{j'}} \tag{15}$$

$$\sum_{\Gamma_i} N_k \chi^{(\Gamma_i)}(C_k)[\chi^{(\Gamma_i)}(C_{k'})]^* = h\delta_{kk'}. \tag{16}$$

Γ are representations of the group, R is an element of the group, h is the size (order) of the group, C_k is a conjugacy class, N_k is the size of a conjugacy class.

(a) Complete character table below where A_1 represents the trivial irrep, and A_2, B_1, and B_2 are 1D irreps.

Solution:

	e	C_2 (z)	$\sigma_v(xz)$	$\sigma_v(yz)$
A_1	1	1	1	1
A_2	1	1	-1	-1
B_1	1	-1	1	-1
B_2	1	-1	-1	1

(b) Cyclic groups are only irreducible over complex numbers (rather than real numbers), but have all 1D irreps. Complete the character table below for the cyclic group C_5 where A_1 is the trivial irrep and A_2 and A_3 are (complex) 1D representations. $\omega = e^{i2\pi/5}$, $\omega^2 = e^{i4\pi/5}$, etc. Hint: $1 + \omega + \omega^2 + \omega^3 + \omega^4 = 0$

Solution:

	E	C_5	$(C_5)^2$	$(C_5)^3$	$(C_5)^4$
A_1	1	1	1	1	1
A_2	1	ω	ω^2	ω^3	ω^4
A_3	1	ω^4	ω^3	ω^2	ω
A_4	1	ω^2	ω^4	ω	ω^3
A_5	1	ω^3	ω	ω^4	ω^2

Interpreting Outputs

4. (27 points) In the following questions, we will present you code snippets using the functions you have coded in the exercises and ask you to interpret what the outputs means. You may assume the following has been imported. The Docstrings for functions used in this problem are available at near the end of your exam booklet, before the Work space pages.

```
import numpy as np
from symm4ml import groups, linalg, rep, vis
import torch
```

(a) In this problem, we will chain together several code snippets to determine properties of two representations of the group D_4, the rotation and mirror symmetries of a square.

i. Consider the following code snippet.

```
1  subset_D4 = np.array([
2      # mirror across y axis
3      [[-1., 0.],
4       [0., 1]],
5      # Four-fold (90 degree) rotation counterclockwise
6      [[np.cos(np.pi / 2), -np.sin(np.pi / 2)],
7       [np.sin(np.pi / 2), np.cos(np.pi / 2)]],
8  ])
9  D4 = groups.generate_group(subset_D4)
10 D4_table = groups.make_multiplication_table(D4)
11 D4_reg_rep = rep.regular_representation(D4_table)
```

Describe what is happening in lines 1-9.

> **Solution:** We are using a subset of elements of D_4 (represented as 2D rotations and mirrors) to generate the entire group of D_4.

ii. Describe how the (left) regular representation `D4_reg_rep` is constructed from `D4_table` in line (10-11).

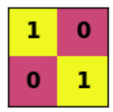
> **Solution:** The left regular representation is constructed from the multiplication table by first permuting the columns of the multiplication table to be the inverses of the original columns. Then the representation of each element is simply constructed by inserting 1s where the entry matches the element we are constructing the representation for or 0s otherwise.

iii. What are the dimensions (shape) of the representation of `D4` and `D4_reg_reg`? Explain your reasoning.

> **Solution:** The shape of `D4` is [8, 2, 2] because there are 8 elements of D_4 and we used 2×2 matrices to generate the group. The regular representation of D_4, i.e. `D4_reg_rep` has shape [8, 8, 8] because the regular representations acts on a vector space the same size as the group since we define it using the group multiplication tables.

iv. Now, we continue with the following code snippet.

```
12 M = linalg.infer_change_of_basis(D4, D4)
13 norm_M = M / np.max(np.abs(M))
14 norm_M[np.isclose(norm_M, -0.0)] = +0.0
15
16 vis.plot_image_values(
17     norm_M, norm_M, vmax=1, vmin=-1, colormap='plasma',
18     decimal_places=0, figsize=(1, 1), size=12,
19     fontcolor=['white', 'black'], fontcolor_cutoff=-0.1
20 );
21 >>
```



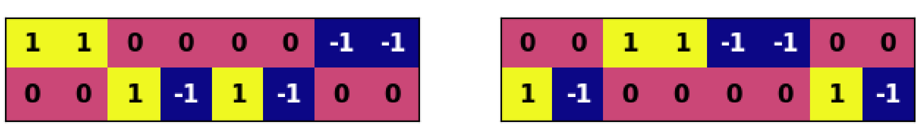
In line 13-14, we are simply normalizing the matrix by its absolute maximum value and replacing `-0`s with `+0`s, so it's nicer to visualize using the code in lines 16-20. You can assume there are no rounding errors.

Describe what is happening in line 12 and explain the significance of `linalg.infer_change_of_basis` returning a single matrix equal to a constant matrix (identity multiplied by a constant). What does this tell us about the representation of D_4 as 2D rotations and mirrors? Explain your reasoning.

> **Solution:** We are using the Kronecker sum to see whether there exists a change of basis between the 2D representation of D_4 consisting 2D rotations and mirrors itself. Because `linalg.infer_change_of_basis` returns only a constant matrix, by Schur's Lemma, we know it is irreducible.

v. Finally, we finish with the following code snippet.

```
22 cob = linalg.infer_change_of_basis(D4, D4_reg_rep)
23 norm_cob = cob/np.max(np.abs(cob))
24 norm_cob[np.isclose(norm_cob, -0.0)] = +0.0
25
26 vis.plot_image_values(
27     norm_cob, norm_cob, vmax=1, vmin=-1, colormap='plasma',
28     decimal_places=0, figsize=(8, 2), size=12,
29     fontcolor=['white', 'black'], fontcolor_cutoff=-0.1
30 );
31 >>
```



Again line 23-24, we are simply normalizing the matrix by its absolute maximum value and replacing `-0`s with `+0`s, so it's easier to visualize using the code in lines 26-30. You can assume there are no rounding errors.

Describe what is happening in line 22. Why does `linalg.infer_change_of_basis(D4, D4_reg_rep)` return two matrices? What does this tell us about `D4_reg_rep` given what we know about `D4` from part (iv)? How does this align with what we know about the properties of the regular representation from lecture?

Solution: We are using the Kronecker sum to see whether there exists a change of basis between the 2D representation of D_4 consisting 2D rotations and mirrors and the regular representation of D_4.
The regular representation contains a l_j copies of each irrep j, where l_j is the dimension of irrep j, thus there are two copies of the 2D irrep of in the regular representation, so two change of basis options, one for each.

(b) Consider the following code snippet and output.

```
linalg.nullspace(np.array([[1, 0, 0], [0, 0, 0], [0, 0, 0]]))
>> (array([[0., 1., 0.],
        [0., 0., 1.]]),
 array([[0., 0., 0.],
        [0., 1., 0.],
        [0., 0., 1.]]))
```

i. What does `linalg.nullspace` do and how are the first output `array([[0., 1., 0.], [0., 0., 1.]])` and the second output `array([[0., 0., 0.], [0., 1., 0.], [0., 0., 1.]])` related?

> **Solution:** `linalg.nullspace` finds vectors and a projector that span the space of solutions $Ax = 0$.

ii. Is the first output unique (e.g. are there any other possible outputs that would also be correct)? Explain your reasoning.

> **Solution:** The first outputs are not unique. In the case of a single vector, we can flip it's sign. Additionally, if we have more that one vector, any rotation in the subspace are valid spanning vectors of the null space.

iii. Is the second output unique? Explain your reasoning.

> **Solution:** The second outputs are unique as it is the projection matrix onto the entire subspace. It is invariant under rotations within the subspace.

Name: ________________________________

This Side Up

5. (20 points) After graduation, you join a startup building next generation photo digitizers. Your first task is to implement a feature that automatically rotates square photos so that they are in the right direction regardless of how users input the photo. You wonder if this task is well-suited for group convolution.

   (a) Assume the scanner scans a single side of square photos. What group should you use to build your Group Convolutional Neural Network? You can use its shorthand name or describe what elements are in the group. Feel free to draw a diagram. Explain your reasoning.

   > **Solution:** C_4, because we can't invert physical pictures.
   >
   > The intention of the problem is that the images were placed so that there sides are parallel to the scanner (or that the scanner could find the edges of the image, but not properly orient content), but since it might not have been clear we also accepted $SO(2)$
   >
   > Note that the group convolution we studied in class does not work on infinite groups.

   (b) Recall that group convolution was defined to be equivariant under under the action $L_g(f) = f \circ g^{-1}$, i.e., $L_g(f) \star \psi = L_g(f \star \psi)$. If f was a function over $\mathbb{Z}^2$ such that $f(x)$ is the value of the pixel at x, then $[L_g(f)](x) = f(D_{2d}(g^{-1})x)$, where D_{2d} is a 2-dimensional representation of group $\mathcal{G}$. If $D_{2d}(g)$ was a matrix that rotated by 90° <u>clockwise</u>, how does the image of the function represented by $L_g(f)$ relate to that represented by f?.

   Hint: If you do an example, define f and apply $l_g(f)$, before drawing images.

   > **Solution:**
   >
   > ○ 90° rotation counter-clockwise
   >
   > √ 90° **rotation clockwise**
   >
   > Explain your answer with a simple example (like a single dot on a corner) or a proof:
   >
   > **Example:** Let $f\{-1, 0, 1\}^2 \to \{0, 1\}$ let $f(1, 1) = 1$ and $f(x, y) = 0$ otherwise. Note that $L_g f(1, -1) = f(g^{-1}(1, -1)) = f(1, 1) = 1$ So then then the location of the 1 changed by 90 degrees clockwise
   >
   > **Intuition:**
   >
   > You might remember when studying trigonometry that $\sin 2x$ is actually a contracted version of $\sin x$. So when you are multiplying the input to a function by something, the effect on the graph is actually equal to the inverse.
   >
   > **Proof:**
   >
   > Another way to think about it is, if $f(a) = y$, $l_g f(ga) = f(g^{-1}ga) = f(a) = y$. So the location of y changed by an action of g, when g^{-1} operated on the input! Which is what we hoped writing an example would help you see.

   Relating this to lecture: We defined L_g when f has an additional regular representation dimension. That is f has arguments $(x, i) \in \mathbb{Z} \times \mathcal{G}$. $L_g(f) = f(D_{2d}(g^{-1})x, g^{-1}i)$. This has the effect of permuting the elements in the group dimension on top of the effect of the action on the spacial dimension (this question is only asking you to identify the effect on the spacial dimension), since we repeat the given image across the regular representation dimension in the input before passing it to the first layer. The action you identified is the way to act on the input of the network (as permuting a repeated value has no effect).

(c) You build a model by purely stacking group convolutions, point-wise non-linearities and pooling operations, as discussed in class. The second convolution layer has 4 input channels and 4 output channels with kernel height 3 and kernel width 3.

How many parameters does the second layer have?

Hint: You might want to look at the doc string for the following, as it takes the filter parameters as input: `group_conv.image2D_group_convolution_filter_bank`.

Solution:
$C_{\text{in}} \times C_{\text{out}} \times H \times W \times |G| = 144 \times |C_2| = 576$
Common errors:

1. Using the size of the output of `filter_bank` instead of the input. This ignores the weight sharing accross the regrep dimension that gives us equivariance! and gives an answer that's a factor of $|G|$ more than the actual answer
2. Giving the number of parameters of a regular convolotion. This ignores the fact that we have $|G|$ convnets in a group convnet. (and we get equivariance by properly shuffling which ones we apply depending on reg_rep dimension)

(d) In your model from part (c), the last layer outputs a single channel with a regular representation dimension equal to the size of the group you chose in Part (a), and spatial dimensions $H = W = 1$, i.e. the output has shape `[1, |G|, 1, 1]`. Finally, we apply a soft-max function ($\frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$) to this output to convert the output values to probabilities over the group elements (the vectorspace of the regular representation).

In training, you use cross entropy loss between the output of the model, and a one hot encoding of a group element (assume the content of the images is not symmetric for this task to be meaningful). For a properly oriented images you use a one hot encoding of the identity e as a label.

What element should you use as a label for an image that resulted from applying L_g to a properly oriented image, where both images are defined to be functions of pixel coordinates? Look at part (b) to remember what this means.

Which of the following should you use a one hot encoding of in the loss, and why?

Solution:

√ g

◯ g^{-1}

◯ Either works.

◯ Something else.

The model should predict e when given an image in proper orientation (at least according to our labels above). When given an image that's rotated by g, that is L_g was applied on it.

Our label was the function $f(e) = 1$, and $f(g) = 0$ otherwise. We should act on it by L_g, hence in a similar reasoning to part b), $L_g(f)(g) = 1$, and everything else 0. So it should be labeled by g, If some of them were labeled by $g^{-1} \neq g$ then those samples would hurt the model's ability to learn.

To avoid double penalizing people, if people understood that only one works due to equivariance, but derived the wrong answer due to impromer understanding of the effect of function composotion (similar to part b). We gave full credit on this part.

Hint: If we predict either, we can recover the transformation we need to apply to the image. So the questions is concerning what can group convolution learn as a target, not what group element directly represents the transformation we want to apply to our image.

(e) You are sharing your finding with your team, and of them says that you should augment the data by rotating the images and updating the labels (the one hot encodings). Do you think that would help improve performance? Why?

> **Solution:**
> No, the gradient would look the same due to weight sharing. (Alternatively, augmenting the data does not add more information, since they are labeled according to equivariance rules, and the model is equivariant).

(f) What do you think the model would predict when given an blank photo?

> **Solution:** Note that the since the image is blank, it does not change under L_g for any g. By equivariance, this must hold for the output as well. This makes all entries of the output equal, and since the last layer is a softmax, should should add up to 1. They must all be $\frac{1}{|G|}$.