IB Computer Science

Paper 3 Guide: May I recommend the following?

Summary

1. Practice Paper #1
2. Practice Paper #1 Sample Answers
3. Practice Paper #2
4. Practice Paper #2 Sample Answers
5. Paper 3 Case Study Guide

For any questions, join the CS Classroom Discord server, at https://discord.gg/xKAE9p4D7B

Practice Paper #1

Answer **all** questions.

1)
    a) Identify two types of cloud delivery models. [2]
    b) Identify two measures used to assess the accuracy of recommender systems [2]

2)
    a) Describe the steps that need to be taken to recommend data to a new user in a recommender system that utilizes collaborative filtering with the k-NN algorithm. [4]
    b) Explain why it is important to carefully consider the hyperparameter that will be used with the k-NN algorithm. [4]

3) Distinguish between supervised and unsupervised learning [6]

4) Jungmin and Lijing plan to use the behavioral data they have collected to monetize the site through advertising. Lijing has concerns that this may be unethical.

   To what extent do you agree with Lijing?

Practice Paper #1 Sample Answers

1)

   a) Software as a Service (SaaS), Platform as Service (PaaS), Infrastructure as a Service (IaaS)

   b) Mean Absolute Error (MAE), Root Mean Square Error (RMSE)

2)

   a) Basically, when a new user is entered into a system, they will be asked to enter some data regarding their content preferences. This could either be information on the characteristics of content they prefer, or even specific content they prefer, both of which can be used to construct a user profile with specific features. Using this user profile, we can use a mathematical formula to calculate how different this user profile is from existing user profiles. We will find the k closest user profiles in our databases, and then find the group to which the majority of these user profiles belong. Our new user will become a part of this group and receive the same recommendations as everyone else in the group.

   b) The hyperparameter, k, is crucial to providing new users with relevant recommendations. If the hyperparameter is too small, members of a group will have a much larger influence on recommendations, perhaps leading to content that is overly niche. If the hyperparameter is too large, the range of content recommended may be too broad, and not every member may be happy with recommendations. This is why it's important to find the optimal hyperparameter through trial-and-error.

3) Supervised learning and unsupervised learning both employ very different methodologies and accordingly have very different use cases. Supervised learning employs a set of labeled training data to initially train a machine learning algorithm to place data into a set of predetermined categories. The training data is put through a machine learning algorithm and if the algorithm does not output the label, the algorithm is adjusted until based on the data, the corresponding labels are returned as frequently as possible. In contrast, in unsupervised learning, there is no training process. Unlabeled data is put through the machine learning algorithm, which must then independently differentiate sets of data from each other, placing this data into groups based on common patterns. It is then up to humans to decide the significance of the distinct groups of data points that have been formed. This also means that while there is a clear measure for when a supervised learning algorithm has been successful, in terms of how many times the output label matches the input, there is no real metric for assessing how successful an

unsupervised learning algorithm has been.

In terms of use cases, as a result of the aforementioned characteristics, supervised learning is largely employed in use cases such as predicting house prices based on historical data, or classifying images into different categories, the key being the usage of already existing data to make inferences about new data. In contrast, rather than being used for extrapolation or regression, unsupervised learning may be employed in cases such as fraud detection systems, where it is enough to simply detect atypical data points.

4) I agree with Lijing, but not completely.

I think that if users are given the option to sign some sort of agreement in which it is clearly stated with whom and for what purpose explicit data will be shared, then it could be acceptable. After all, this is data that the user is clearly choosing to share on the NextStar platform, and if they are aware how their data is used, then I think that there is an acceptable level of transparency.

However, I do not think that it is ethical to share implicit behavioral data with an advertiser. This is by definition data and insight that the user does not know is being collected and therefore the user cannot consent for this to be shared. While this can be used to inform recommendation systems, although in my opinion even this is a gray area, the idea of such data being shared with an unknown third-party is a clear violation of both a user's right to privacy and anonymity.

For example, let's say that a person likes to watch adult videos on the NextStar platform. As long as such videos were made legally, this should not be an issue. However, let's say that he and his spouse or partner share a social media account, which is registered under the email address of the NextStar user. All of a sudden, adult-themed adverts pop up in the user's social media feed, leading to very uncomfortable questions from the user's spouse, at the very least.

This is a very possible scenario if data on the user's search history or purchase data is being collected and is clearly not something that would be desired by the user. While it may not be a violation of a user's right to anonymity, it is clearly a violation of that user's right to privacy, especially since presumably that user would not have known that that data was being collected and passed on to an advertiser.

Additionally, once the data is passed on to the advertiser, unless there is an explicit agreement in place, they most likely have no obligation to protect the anonymity of users associated with advertising data. Even if they do, a simple case of hacking could lead to this unwanted data being leaked to the public.

In conclusion, while a user agreement regarding the sharing of explicit behavioral data may be acceptable, the collection and sharing of implicit behavioral data is clearly unethical and should be avoided at all costs.

Practice Paper #2

Answer **all** questions.

1)
    a) Identify two collaborative filtering algorithms. [2]
    b) Identify two sources of overfitting. [2]
2)
    a) Distinguish between collaborative and content-based filtering. [4]
    b) Explain why collaborative filtering may be prone to popularity bias. [4]

3) A process called stochastic gradient descent is used with matrix factorization, which is a method of collaborative filtering.

    Explain the role of stochastic gradient descent in matrix factorization. [6]

4) NextStar intends to use IaaS instead of PaaS for delivering their application.

    To what extent do you think this is an economically advantageous choice? [12]

Practice Paper #2 Sample Answers

1)
   a) k-NN, Matrix Factorization
   b) Model is too complex, taking into account irrelevant features, too much times has been spent training the model on a single training data set

2)

   a) Recommendations focused on collaborative filtering are user-centric. More specifically, recommender systems that make use of collaborative filtering seek to group users together based on either previous content they have shown an interest in, or some other aspect of their user profile. These systems then recommend the same content to all users in a given group. In contrast, content-filtering looks at the content an individual user has expressed interest in and then recommends similar content. It makes a recommendation based on the profile of the content an individual user has liked or consumed, rather than the content that similar users have liked or consumed.

   b) Collaborative filtering may be prone to popularity bias because it makes recommendations to users based on what content other users like. If many users like a piece of popular content, then exponentially more and more users will be recommended that piece of content, whether they really like it or not. This is opposed to content-based filtering, where users are more siloed, and content is recommended based on their individual actions on the platform rather than on what other users are consuming.

3) The first step in matrix factorization is to decompose our user-item matrix into a user-feature matrix and an item-feature matrix. The number of missing values in the initial user-item matrix does not impede this process, because through the use of an iterative algorithm, we can aggregate existing data to create these matrices so that the matrices represent a sort of average or approximate general representation of existing data. Once we have these approximated user-feature and item-feature matrices, we want to adjust their values, so that if we were to collapse (dot product) both of these matrices back together, we would get our original user-item matrix, with the initially empty position filled in with approximated values. The stochastic gradient descent process allows us to take the results of the iterative algorithm and adjust them by some amount as dictated by the cost function, so that we will not only get our initial user-item matrix, but also appropriate approximations for those positions in the user-item matrix that were initially empty.

4) I fully agree with NextStar's choice because of the ability to control the use of resources and security with greater granularity using an IaaS platform over PaaS. This can all lead to an economic advantage for NextStar.

This is because of the crucial fact that in a PaaS application, you have control over your application's code, but none of the infrastructure behind it, while in an IaaS application, you control all of your infrastructure and data.

Let's start with the question of the cost of hosting your application and data. When you deploy your code to PaaS, you cannot control what specific resources you want to use or not use. For example, you may want to store your database on your company's servers or even another platform, but your application code on a PaaS application. With an IaaS system, you can pick and choose what resources you want to host in the cloud or on the specific IaaS platform, in order to optimize cost. It may be cheaper for you to host a database in a datacenter, but your application code on an IaaS platform. With IaaS, you can save money with such customization, while PaaS doesn't offer this level of flexibility.

Beyond just the cost question, you can also set up your application in a way to optimize functionality more easily using IaaS. This may be through the use of microservices architecture or load balancing to manage network traffic to your application. Such optimization can lead to a more performant application and a more pleasant user experience, improving the reputation of the company and thereby making the company more money.

By using PaaS, NextStar would also lose out with regards to security, which is highly important for the company and its reputation given the amount of user data it must host. With the PaaS platform, since you have no control over the network infrastructure, both in terms of hardware and software, you are completely reliant on the PaaS company for security. If that company's network or servers get hacked, there's nothing you can do about it. However, with IaaS, you have the ability to manage your own security using your own network setup, firewalls, software etc. Assuming you have competent cyber security specialists, you can ensure that your users' data and privacy is more adequately protected against specific threats your application may face. A cyber attack or loss of data from your platform could economically be a disaster, damaging the reputation of the company.

Therefore, we can say that the greater customizability available through IaaS as well as the greater ability to control the security of data through IaaS make it a more economically advantageous choice for NextStar.

Paper 3 Case Study Guide

- **NextStar (Hypothetical Company used in Case Study)**
    - To be created by two friends, Jungmin and Lijing
    - Application to allow users to discover new artists (painters, actors, sculptors, comedians, screenwriters, singers, and filmmakers)
    - Users can upload, and rate content, which will eventually be fed into a recommendation system
    - Free to join, income from advertising
    - Plan to use IaaS (Infrastructure as a Service)
- **Cloud Computing**
    - **Software as a Service (SaaS)**
        - Application that can be accessed through your browser, instead of locally on your computer
        - Examples: Netflix, Amazon, Google Docs
    - **Platform as a Service (PaaS)**
        - Provides services that allow web applications to run including storage, configuration, and networking capabilities
        - Meant to be an easy, all-in-one solution for deploying web applications
        - **Examples**: Amazon Web Services, Heroku, Google App Engine, Microsoft Azure
    - **Infrastructure as a Service (IaaS)**
        - Provides individual resources required to run an application, on-demand
        - Can create web servers, database servers, virtual networking devices, etc. at the click of a button
        - Allows for more granular access to specific resources necessary to run a web application than PaaS, where everything is created and taken care of automatically
        - Examples: Amazon Web Service, Google Compute Engine Digital Ocean
- **Machine Learning Paradigms**
    - **Supervised Learning**
        1) Machine learning algorithm is provided with labeled data to train it - that is, the data has some descriptor(s) attached to it
            a) For example, an algorithm being trained to distinguish different animals may have images of cats, dogs, and birds fed into it, with the respective labels
            b) The label represents the desired output of the algorithm
            c) This initial set of data, used to help the algorithm learn to categorize different data points is called the **training set**.
        2) The algorithm breaks down data into features.

    a) Each input is distilled into a set of features that describes that particular image, text, etc, called a **feature vector**. The algorithm makes decisions based on these features.
    b) For example, the algorithm will eventually be able to recognize the combination of pixels in an image that describes each respective animal.

3) Feed the inputs and the labels into the algorithm and based on whether the output matches the initial label, adjust how each feature is evaluated.
    a) The algorithm may adjust the expected values of certain features and/or weigh certain features higher than others, based on whether the resulting output more consistently matches the input labels.
    b) The algorithm basically learns the "pattern" that corresponds to each specified category of input, by adjusting its feature expectations based on the the inputs and their labels
4) Once the algorithm has been trained using the training set, its accuracy is validated using a **testing set**.
    a) The training set usually corresponds to 80% of the initial data and the testing set corresponds to the other 20%.

- **Examples of Usage:** Market Forecasting (Extrapolation), Identity Fraud Detection

Supervised Learning

Advantages	Disadvantages
Simpler to understand	Training is very computationally heavy
You can create testing/training data to create specific boundaries for classification (you have full control over what the machine is learning and how)	Prone to overfitting (See section "Overfitting")
After training is complete, don't need to hold training data in memory (which can take up a lot of space)	Cannot give you unknown insights from data, like unsupervised learning
Can pre-specify possible outputs of algorithm (you have control over possible algorithm outputs)	Need a large number of examples of every possible category, otherwise algorithm will not predict correctly
Able to leverage past data to make decisions	Limited by training data

- **Unsupervised Learning**
    1) The machine learning algorithm is initially given a set of data WITHOUT labels.
    2) The algorithm autonomously breaks the data set into groups or categories based on patterns or characteristics that it detects in the data
    - **Examples of Usage:** Object Recognition, Identifying Customer Segments, Spam Detection

Unsupervised Learning

Advantages	Disadvantages
Can pick out patterns, features or insights unknown to the user (algorithm output not limited as with supervised learning)	Results can be less accurate than with supervised learning
Easier and less work to obtain unlabeled data than labeled data	More work for humans, who have to make sense of patterns or groups produced by algorithm
Can determine to what extent similarities exist within a data set	Cannot be used for classification problems, because algorithm has no idea what to classify groups or patterns of data by
	Sometimes output can be useless, as there is no metric to confirm their benefit, unlike in supervised learning, where target outputs can be used to determine effectiveness of model

- **Reinforcement Learning**
    - Similar to supervised learning, but algorithm learns in a real-time simulation
    - In supervised learning, there is an “answer key” in the form of labels in the training dataset, but in reinforcement learning, the algorithm tries to take different actions, for which it then receives a quantitative reward or punishment.
    - **Example**: A reinforcement learning algorithm wants to learn to win at a board game. Certain squares are more desirable than others and will result in a reward, while less desirable squares will result in a punishment. The algorithm will play the game thousands of times until it finds the optimal path to win the game, with the greatest reward and the least punishment. It’s essentially trial-and-error.
    - **Examples of Usage**: Playing chess, a robot navigating a forest

Reinforcement Learning

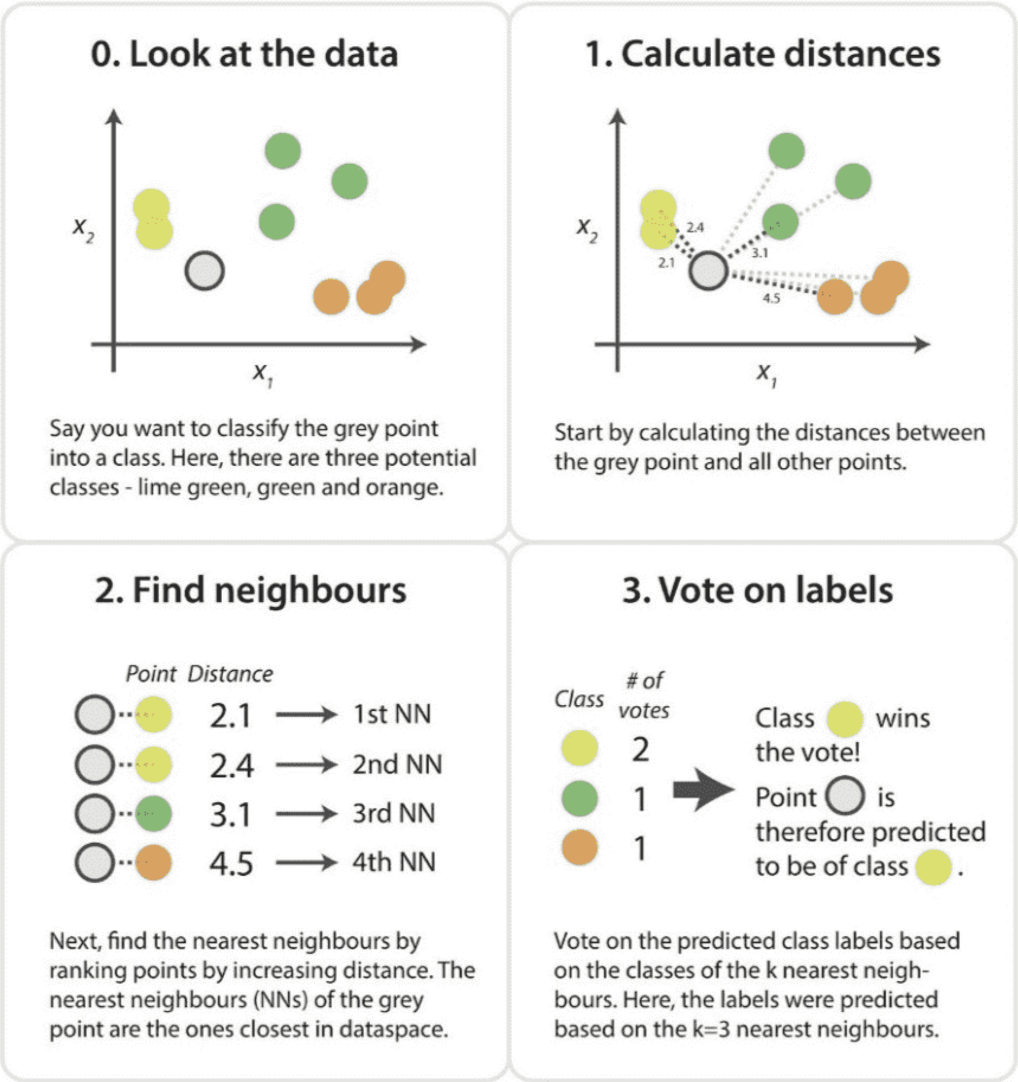
Advantages	Disadvantages
Can learn even in the absence of training data, from experience	Requires a lot of data and computational resources
Similar to “trial-and-error” method by which humans learn	Defining rewards in order to “teach” algorithm is difficult
Model can self-correct errors that occurred during the training process through experience, unlike supervised learning	
Can be useful when the only way to collect information about an environment is to interact with it	

- <u>**Recommender Systems**</u>
    - **Collaborative Filtering**
        - Recommendations are made based on content purchased, ranked, watched or clicked by users who have similar profiles to a given user
        - Prone to **popularity bias** - when popular content is unduly favored and overly recommended to users
        - Depends on the presence of other users
    - **Content-based Filtering**
        - Recommendations are made based on the characteristics of content already purchased, ranked, clicked or watched by a given user
        - Utilizes content profile rather than other user profiles

- <u>**Collaborative Filtering Algorithms**</u>
    - **k-Nearest Neighbor**
        - Stores all available cases (i.e. user profiles) and groups new cases with those that are most similar to existing cases (i.e. new users are grouped with those whose profiles are the most similar)
        - The same recommendations could be made to all the cases (i.e. user profiles) in the same group/cluster
        - “k” refers to the number of most similar cases to take into consideration for deciding which group new cases should be a part of
        - If k=5, then we look at the 5 cases that are the most similar. Whatever group that the majority of these cases belongs to will be the group of our new case
        - For example, let’s say that we have a new Netflix user with their own unique profile and all Netflix profiles are placed into distinctive groups

based on their user profiles. If k = 4. We would look at the 4 most similar user profiles and put our new user in the group to which the majority of these 4 belonged.

K-Nearest Neighbor



0. Look at the data

Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances

Start by calculating the distances between the grey point and all other points.

2. Find neighbours

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

Class ● wins the vote!

Point ○ is therefore predicted to be of class ●.

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

- **Choosing 'k'**
    - **hypertuning** - process of choosing 'k'
    - **hyperparameter** - refers to any value that is used to control the machine learning process (like 'k')
    - No single method to choose 'k' - chosen through trial-and-error

Smaller values of ‘k’	Larger values of ‘k’
Neighbors have greater influence on result, which can be a problem if there are atypical data points	Very computationally expensive, because of the greater number of distance calculations involved between the new data point and prospective neighbors
	Leads to **oversmoothing** - a new point may be in a group that isn’t a specific enough match to make good recommendations

- **Matrix Factorization**
    - Watch the video [here](#).
    - User-item Interaction Matrix - Contains ratings of multiple users for multiple items
    - Item-feature Matrix - Displays extent to which each item matches the descriptors or features that describe the item being recommended
    - User-feature Matrix - Displays extent to which each feature is relevant to each user

    1) Decompose user-item interaction matrix into item-feature matrix and user-feature matrix according to the desired number of features.
    2) Use a **cost function** (in the **stochastic gradient descent** process) to adjust item-feature matrix and user-feature matrix values until their dot product can reliably produce something close to the user-iterm interaction matrix (given the available values)
    3) With the new user-feature and item-feature matrices, produce a new user-item interaction matrix that gives values for all users and items. These are predictions for how each user would rate each item.

    *All of the above operations require basic linear algebra.

- **<u>Training Recommender Systems</u>**
    - Can be trained using train/test splits
    - 80% of existing data is used to train recommender system (**training set**)
    - 20% used to test effectiveness of recommender system (**testing set**)

- **<u>Overfitting</u>** [Supervised Learning]
    - Phenomenon where the model created by your algorithm fits the training dataset too closely and can’t account for cases even slightly different

- Your model (basically a mathematical function) predicts the outcomes of your testing data extremely accurately, but doesn't work on any other set of testing data

- **Causes**
    - Can occur when the model generated by the algorithm is too complex and takes into account features that are very specific to the training data, but overall irrelevant
    - Can also control when the model trains for too long on one specific set of the training data

- **Example:** It's like if you were a footballer and trained specifically to play against one team of 10 specific players in one type of weather and at one time of day.

- **Recommender System Accuracy**
    - **Terms**
        - **Precision** - fraction of relevant instances among the retrieved instances
        - **Recall** - a measure of completeness - fraction of relevant instances that were retrieved
        - **Example**: When a search engine returns 30 pages, only 20 of which are relevant, while failing to return 40 additional relevant pages, its precision is 20/30 = 2/3
        - **F-measure** (F-score) - Balances precision and recall
            - F-Measure = (2 * Precision * Recall) / (Precision + Recall)
    - **Methods**
        - **Mean Absolute Error (MAE)**
            - measures the average magnitude of errors in a prediction
            - The average of the **absolute difference** between predicted and actual values
            - Better if there are many outliers (atypical data points)
        - **Root Mean Square (RMSE)**
            - square root of the average of **squared differences** between prediction and actual observation

- **Social and Ethical Concerns**
    - **Explicit Behavioral Data** - refers to data gathered from users' submitted data, such as when a user rates a video clip, enters their preference, or searches for an item
    - **Implicit Behavioral Data** - refers to data that the user is not aware is being collected, i.e. click data, purchase data, or key logging
    - Right to Anonymity

- Right to Privacy