CS7641 Problem Set
Fall 2024

Instructions

This problem set is not a part of your final grade but rather a means to help you with the final exam. If all of the problems are attempted and your grade is around the cutoff, we will round up to the higher letter grade. You will need to attempt each problem and submit your solutions on Canvas. We will verify work is submitted at the end of the term. After the deadline, we will provide solutions for you to compare your answers. We plan to hold two Office Hours, one for each part of the problem set before the final.

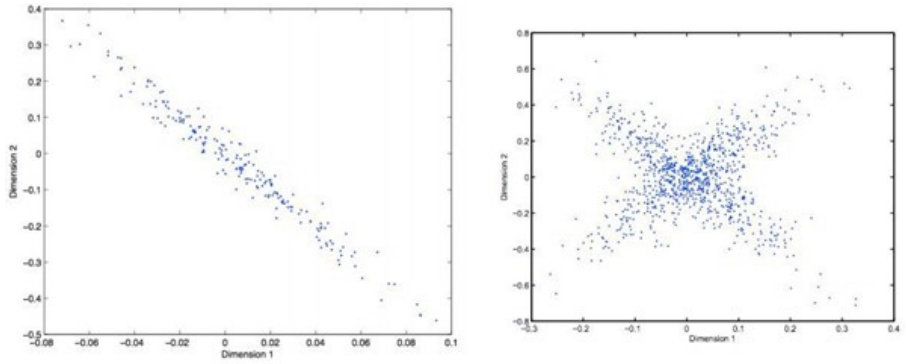
Part One

1. Where we are doing supervised learning, we have mostly assumed a deterministic function. Imagine instead a world where we are trying to capture a non-deterministic function. In this case, we might see training pairs where the x value appears several times, but with different y values. For example, we might use attributes of humans to the probability that they have had chicken pox. In that case, we might see the same kind of person many times but only sometimes they may have had chicken pox. We would like to build a learning algorithm that will compute the probability that a person has chicken pox. So, given a set of training data where each instance is mapped to 1 for true or 0 for false:
   (a) Derive the proper error function to use for finding the ML hypothesis using Bayes' Rule. You should go through a similar process as the one used to derive least squared error in the lessons.
   (b) Compare and contrast your result to the rule we derived for a deterministic function perturbed by zero-mean gaussian noise. What would a normal neural network using sum of squared errors do with these data? What if the data consisted of x, y pairs where y was an estimate of the probability instead of 0s and 1s?
2. Design a two-input perceptron that implements the boolean function $A \wedge \neg B$. Design a two-layer network of perceptrons that implements $A \oplus B$ (where $\oplus$ is XOR).
3. Derive the perceptron training rule and gradient descent training rule for a single unit with output o, where $o = w_0 + w_1 x_1 + w_1 x_1^2 + \ldots + w_n x_n + w_n x_n^2$. What are the advantages of using gradient descent training rule for training neural networks over the perceptron training rule?
4. Explain how one can use Decision Trees to perform regression? Show that when the error function is squared error that the expected value at any leaf is the mean. Take the Boston Housing dataset (`http://lib.stat.cmu.edu/datasets/boston`) and use Decision Trees to perform regression.
5. Suggest a lazy version of the eager decision tree learning algorithm ID3. What are the advantages and disadvantages of your lazy algorithm compared to the original eager algorithm?
6. Imagine you had a learning problem with an instance space of points on the plane and a target function that you knew took the form of a line on the plane where all points on one side of the line are positive and all those on the other are negative. If you were constrained to only use decision tree or nearest-neighbor learning, which would you use? Why?
7. Give the VC dimension of these hypothesis spaces, briefly explaining your answers:
   (a) An origin-centered circle (2D)
   (b) An origin-centered sphere (3D)

8. You have to communicate a signal in a language that has 3 symbols A, B and C. The probability of observing A is 50% while that of observing B and C is 25% each. Design an appropriate encoding for this language. What is the entropy of this signal in bits?

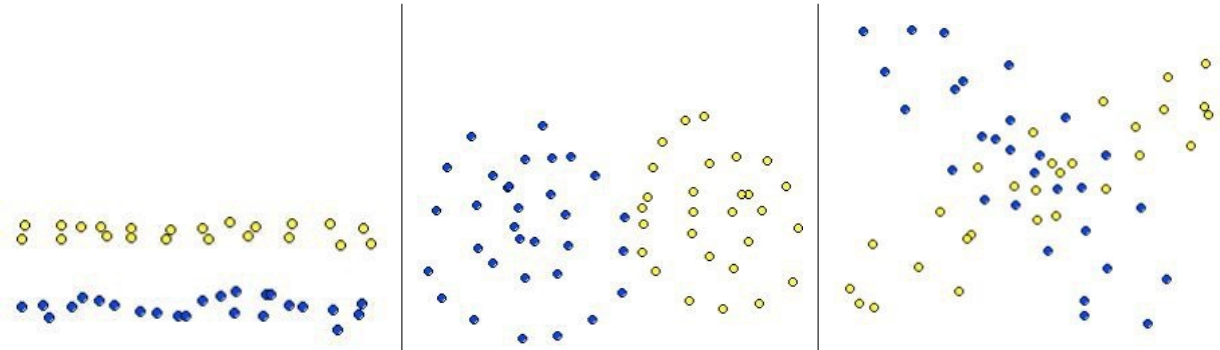
9. Show that the K-means procedure can be viewed as a special case of the EM algorithm applied to an appropriate mixture of Gaussian densities model.

10. Plot the direction of the first and second PCA components in the figures given.



11. Which clustering method(s) is most likely to produce the following results at $k = 2$? Choose the most likely method(s) and briefly explain why it/they will work better where others will not in at most 3 sentences.

    - Hierarchical clustering with single link
    - Hierarchical clustering with complete link
    - Hierarchical clustering with average link
    - K-means
    - EM



(a) (b) (c)

12. You receive the following letter -

> Dear Friend,
> Some time ago, I bought this old house, but found it to be haunted by ghostly sardonic laughter. As a result it is hardly habitable. There is hope, however, for by actual testing I have found that this haunting is subject to certain laws, obscure but infallible, and that the laughter can be affected by my playing the organ or burning incense.
> In each minute, the laughter occurs or not, it shows no degree. What it will do during the ensuing minute depends, in the following exact way, on what has been happening during the preceding minute:
> Whenever there is laughter, it will continue in the succeeding minute unless I play the organ, in which case it will stop. But continuing to play the organ does not keep the house quiet. I notice, however, that whenever I burn incense when the house is quiet and do not play the organ it remains quiet for the next minute.
> At this minute of writing, the laughter is going on. Please tell me what manipulations of incense and organ I should make to get that house quiet, and to keep it so.
> Sincerely,
> At Wit's End

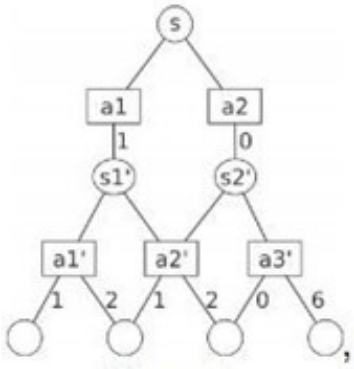
(a) Formulate this problem as an MDP. (For the sake of uniformity, formulate it as a continuing discounted problem, with $\gamma = 0.9$. Let the reward be +1 on any transition into the silent state, and -1 on any transition into the laughing state.) Explicitly give the state set, action sets, state transition, and reward function.

(b) Start with policy $\pi(\text{laughing}) = \pi(\text{silent}) = (\text{incense, no organ})$. Perform a couple of steps of policy iteration (by hand!) until you find an optimal policy. (Clearly show and label each step. If you are taking a lot of iteration, stop and reconsider your formulation!)

(c) Do a couple of steps of value iteration as well.

(d) What are the resulting optimal state-action values for all state-action pairs?

(e) What is your advice to "At Wit's End"?



13. Use the Bellman equation to calculate $Q(s, a_1)$ and $Q(s, a_2)$ for the scenario shown in the figure. Consider two different policies:

- Total exploration: All actions are chosen with equal probability.
- Greedy exploitation: The agent always chooses the best action.

Note that the rewards/next states are stochastic for the actions a'_1, a'_2 and a'_3. Assume that the probabilities for the outcome of these actions are all equal. Assume that reward gathering/decision making stops at the empty circles at the bottom.

14. Consider the following simple grid-world problem. (Actions are N, S, E, W and are deterministic.) Our goal is to maximize the following reward:

S	2	3
4	5	6
7	8	G

- 10 for the transition from state 6 to G

- 10 for the transition from state 8 to G
- 0 for all other transitions

(a) Draw the Markov Decision Process associated to the system.

(b) Compute the value function for each state for iteration 0, 1, 2 and 3 with $\gamma = 0.8$.

15. Find a Nash Equilibrium in each case. The rows denote strategies for Player 1 and columns denote strategies for Player 2.

-
	A	B
A	2,1	0,0
B	0,0	1,2

-
	A	B
A	2,1	1,2
B	1,2	2,1

-
	L	R
T	2,2	0,0
B	0,0	1,1

Part Two

The following are example questions from a previous final. The questions will be multiple choice, multiple answer. To received full credit, you need to give an explanation for each answer. You will not need to give explanation on the final, however this will help solidify knowledge and challenge assumptions. Simply circling the answers will not count for the EC.

Supervised Learning: Decision Trees

1. Why might pruning be applied to a decision tree?
   - To simplify the model and improve interpretability
   - To ensure the tree is balanced
   - To reduce overfitting
   - To always achieve the best accuracy
   - To remove branches that provide little to no predictive power
   - To increase tree depth

Supervised Learning: Regression & Classification

2. Which statements are true regarding cross-validation in regression analysis?
   - Cross-validation eliminates the need for hyperparameter tuning in regression models.
   - Cross-validation is only applicable to regression tasks and not classification.
   - Cross-validation involves training the model on the entire dataset without any validation.
   - Cross-validation guarantees that the model will perform well on unseen data.
   - In k-fold cross-validation, the dataset is divided into k subsets, and each subset is used as a validation set.
   - Cross-validation helps in estimating the generalization performance of a regression model.

Supervised Learning: Neural Networks

3. In the context of training neural networks, which of the following are challenges that can arise?
   - Direct interpretability
   - Overfitting
   - Inefficient support vector calculations
   - Getting stuck in local minima
   - Collinearity among features
   - Vanishing gradient problem

Supervised Learning: Instance-Based Learning

4. Which of the following are characteristics of instance-based learning?
   - It always uses a probabilistic model for predictions.
   - It's often sensitive to irrelevant or redundant features.
   - It is always faster than model-based learning.
   - New instances are classified based on similarity measures.
   - Training is typically computationally intensive.
   - The model "memorizes" the training instances.

Supervised Learning: Ensemble Methods

5. Why might one use ensemble learning techniques?

   - To handle missing values in the data
   - To prevent overfitting by leveraging diversity
   - To combine multiple models' strengths and mitigate individual weaknesses
   - To speed up training times for large datasets
   - To reduce variance and improve generalization
   - To provide a more interpretive model

Supervised Learning: Kernel Support Vector Machines (SVMs)

6. What are the primary reasons for using kernel methods in SVMs?

   - To handle missing values in the data
   - To find a hyperplane that maximizes the margin between classes in the transformed space
   - To decrease the number of support vectors in the model
   - To tackle non-linearly separable data using SVMs
   - To reduce the computational complexity of the SVM algorithm
   - To map input data into a higher-dimensional space

Supervised Learning: Computational Learning Theory

7. Which of the following are essential components of the PAC (Probably Approximately Correct) learning framework?

   - A fixed set of features to represent all possible inputs
   - A confidence parameter representing the probability that a hypothesis will perform worse than the error measure
   - A hypothesis space from which hypotheses are drawn
   - An error measure representing the probability that a hypothesis will misclassify a randomly drawn instance
   - A specific learning algorithm, such as a neural network or SVM
   - A sample complexity determining the number of examples required to achieve a certain error and confidence level

Supervised Learning: VC Dimensions

8. In computational learning theory, the Vapnik-Chervonenkis (VC) dimension is a critical concept. Which of the following statements about VC dimension are accurate?

   - The VC dimension for all linear classifiers in a 2D space is 1
   - It is always equal to the number of features in the dataset
   - A high VC dimension can be an indicator of a model's potential to overfit
   - It measures the capacity or complexity of a hypothesis class
   - Lower VC dimension always guarantees better model performance on new data
   - It is the largest number of points that can be shattered by the hypothesis class

Supervised Learning: Bayesian Learning

9. In the context of Bayesian learning, which statements about the likelihood are correct?

   - A higher likelihood always indicates a more probable hypothesis.
   - The likelihood represents the probability of observing the data given a specific hypothesis.
   - It's always uniform across all hypotheses.
   - Bayes' theorem uses the likelihood to weigh the evidence provided by the data.
   - It is the same as the prior probability for a hypothesis.
   - It quantifies how well a hypothesis explains the observed data.

Supervised Learning: Bayesian Inference

10. Why is Bayesian inference considered a principled way of updating beliefs?

    - It guarantees that the posterior distribution is always unimodal.
    - It follows the rules of probability theory to adjust beliefs in light of new data.
    - It disregards prior beliefs and focuses only on new data.
    - It provides a full probability distribution (posterior) over the variables of interest, not just point estimates.
    - It allows for the integration of prior knowledge with observed data.
    - It solely relies on the likelihood, making computations straightforward.

Unsupervised Learning: Randomized Optimization

11. What are key characteristics of randomized optimization algorithms?

    - They always converge faster than deterministic algorithms.
    - They often involve probabilistic decisions instead of deterministic ones.
    - They are deterministic and predictable.
    - They can escape local optima by introducing randomness.
    - They guarantee to find the global optimum in a finite number of steps.
    - They use random processes to search through the solution space.

Unsupervised Learning: Clustering

12. In K-Means clustering, what are important considerations to ensure effective clustering?

    - Ensuring that all features have unequal importance.
    - Considering the impact of outliers on the clustering.
    - Proper initialization of centroids.
    - Choosing a small dataset.
    - Selecting the appropriate number of clusters (K).
    - Scaling of features so that one feature doesn't dominate the distance calculations.

Unsupervised Learning: Feature Selection

13. What are some challenges or considerations when performing feature selection?

    - Feature selection is only applicable to supervised learning tasks.
    - The potential increase in bias if important features are removed.
    - Feature selection always guarantees an improvement in model accuracy.
    - The risk of discarding features that might be relevant in combination with others.
    - The computational cost of evaluating numerous feature subsets, especially with wrapper methods.
    - The need to balance between model simplicity and the retention of informative features.

Unsupervised Learning: Feature Transformation

14. How does Independent Component Analysis (ICA) differ from PCA in terms of feature transformation and extraction?

    - ICA reduces the dimensionality of data by projecting it onto linearly dependent axes.
    - ICA is designed to find components that are statistically independent, not just uncorrelated as in PCA.
    - ICA is a supervised learning technique, unlike PCA which is unsupervised.
    - PCA components are orthogonal, while ICA components are not necessarily orthogonal.
    - ICA focuses on maximizing variance like PCA.
    - ICA is often used for applications such as blind source separation, whereas PCA is typically used for dimensionality reduction.

Unsupervised Learning: Information Theory

15. Which of the following statements are true about information theory?

    - Entropy captures the amount of information contained in a random variable.
    - A low probability event carries more information whereas a high probability event carries less information.
    - Higher the randomness or entropy of a symbol, smaller the size of the message containing that symbol.
    - Joint entropy is a measure of randomness contained in two variables together.
    - Mutual information is a measure of the reduction of randomness of a variable, given knowledge of another variable.
    - Mutual information is a particular case of KL divergence because minimizing KL divergence between two distributions maximizes the information shared between them, resulting in information gain between the two distributions.

Markov Decision Processes

16. In the context of Markov Decision Processes (MDPs), what is the Bellman Equation used for?

    - To estimate the immediate rewards from each action.
    - To directly calculate the optimal policy.
    - To compute the transition probabilities between states.
    - To relate the value of a state to the values of its successor states.
    - To find the expected return starting from a state and acting according to a given policy.
    - To provide a recursive decomposition of the value function.

Reinforcement Learning

17. What are the key differences between value iteration and policy iteration?

    - Value iteration focuses on finding the optimal value function first, then deriving the optimal policy.
    - Policy iteration involves alternating between policy evaluation and policy improvement steps.
    - Policy iteration always converges faster than value iteration.
    - Value iteration requires an explicit model of the environment (transition probabilities and rewards), while policy iteration is a model-free algorithm.
    - Value iteration always converges faster than policy iteration.
    - Policy iteration often has a higher computational cost per iteration compared to value iteration.

Game Theory

18. What is Nash Equilibrium in the context of game theory and reinforcement learning?

    - A situation in which no player can gain by unilaterally changing their strategy, assuming other players keep their strategies unchanged.
    - Nash Equilibrium can be applied in multi-agent reinforcement learning to predict stable outcomes.
    - Nash Equilibrium can exist in pure or mixed strategies.
    - It is a state where all players have complete information about the others' strategies.
    - It is a strategy that guarantees the highest possible reward for all players.
    - It occurs only in deterministic games.

Game Theory Continued

19. The differences between zero-sum stochastic games and general-sum stochastic games are:

    - Value iteration converges in zero-sum stochastic games.
    - Value iteration converges in general-sum stochastic games.
    - Nash-Q is used for zero-sum stochastic games.
    - Side payments or repeated stochastic games can be used to improve limitations of general-sum games.
    - minimax-Q is used for zero-sum stochastic games.
    - There is a unique solution to Q* for general-sum games.

Version Control

- 10/14/24 - TJL updated and posted for Fall 2024.