CIS600 Fundamentals of Data and Knowledge Mining

Final Exam

NAME Kevin Martin

SU ID 513828817

Problem 1 (10 pts): Please circle the right variable type for the below attributes.

Celsius temperature values in °C	[nominal / ordinal / interval / ratio]
Weight in pounds	[nominal / ordinal / interval / ratio]
Birth date	[nominal / ordinal / interval / ratio]
SU ID	[nominal / ordinal / interval / ratio]
Income level [high/middle/low income]	[nominal / ordinal / interval / ratio]

Problem 2 (28 pts): Answer the following questions by circling the most likely choice. (2pts for each question)

[TRUE / FALSE] Decision tree algorithm (without pruning) is sensitive to noisy data.

[TRUE / FALSE] Naïve Bayes method is intolerant of irrelevant attributes.

[TRUE / FALSE] Type II errors, instead of type I errors, should be minimized when design a spam email detection algorithm (where spams are considered as the positive examples).

[TRUE / FALSE] Holdout method tends to produce more stable estimate of the model performance accuracy than cross validation method.

[TRUE / FALSE] Convenience sampling is one type of probability sampling method.

[TRUE / FALSE] Artificial neural network can be successfully trained using gradient descent for global optimum.

[TRUE / FALSE] K-means clustering algorithms can find clusters of arbitrary shape.

[TRUE / FALSE] Assume everything else the same, the decision tree induction algorithm produces predictions with higher variance than the Random Forest algorithm.

[TRUE / FALSE] Lift corrects for high confidence in rule X-> Y when item X is bought regularly by customers.

[TRUE / FALSE] The best centroid for minimizing the SSE of a cluster is the mean of the points in the cluster.

[TRUE / FALSE] Apriori principle indicates if an itemset is infrequent, then all of it subsets must also be infrequent.

[TRUE / FALSE] ROC (Receiver Operating Curve) is generated by plotting sensitivity (y-axis) against specificity (x-axis).

[TRUE / FALSE] K Nearest Neighbor is considered as a non-parametric method.

[TRUE / FALSE] Boosting method produces an ensemble of classifiers through random sampling the training data set with replacement.

Problem 3 (10 pts): Explain why decision tree algorithm based on impurity measures such as entropy and Gini index tends to favor attributes with larger number of distinct values. How would you overcome this problem?

A decision tree algorithm based on impurity measures favor attributes with a larger number of distinct values because in either case, the goal is to maximize gain ratio. Attributes with a large number of distinct values will have a large value of root nodes. In order to optimize either criteria, the algorithm will take the greedy approach and try to maximize purity. Purity is defined as how homogeneous the class distributions are, with higher purity/more homogenized classes being preferred. With more root nodes, there are more available options to optimize these splits.

In order to overcome this problem, a penalization could be implemented. Consider an approach where, when deciding where to split at the next level, a higher number of potential braches would now be given some sort of negative impact. This idea looks at each split by looking at all the availale information before making one. The concept of Gain Ratio as a replacement for information gain is one practical implementation of this idea. This can be distilled down into the ratio of Information Gain to Intrinsic Value, or IG/IV. The intrinsic value refers to the potential branch concept, and thus looks at each potential branch more holistically.

Problem 4 (10 pts):

(a) Explain what is the *anti-monotone property* of the support measure and how you can incorporate this property directly into the mining algorithm to effectively prune the exponential search space of candidate itemsets. (5pts)

The anti-monotone property of the support measure is the idea that the support for an itemset never exceeds the support for its subsets. For example, if an item set is frequent, then by the anti-monotone property, its subset must be frequent too. Any item that has the anti-monotone property can be incroprated into the mining algorithm to effectively prune the exponential search space via the Apriori algorithm. The implementation is to first generate all "frequent" (support greater than or equal the minimum support) itemsets with only one item. Then, generate itemsets that are equal to two as all possible combinations of the first (frequent) itemsets. Next, prune the ones which have a support value below the minimum support. Repeat the process for itemsets of length 3, 4, and so on and check the threshold one by one at each step. Now we are left with just the frequent items that follow this rule.

(b) Unlike the support measure, confidence does not have any monotone property. Please explain how does confidence-based pruning work and provide a proof why it works. (5pts)

Confidence does not show this same property because the confidence for X→Y (X implies Y) can be greater than, less than, or equal to the confidence for anotehr rule X'→Y' (where X' and Y' are subsets of X and Y). Confidence-based pruning works based on the following situation:

Let Y be an itemset and X a subset of Y. if a rule X → Y (X implies Y) does **not** satisfy the confidence threshold, than any rule X'→ Y-X', where X' is a subset of X, must **not** satisfy the confidence threshold as well. Consider two situations, one where X'→Y-X' and another where X → Y – X, again where X' is a subset of X. The confidence rules C(Y)/C(X') (where C represents the count of) and C(Y)/C(X). Because X' is a subset of X, C(X') ≥ C(X). Therefore, the first situation cannot have a higher confidence than the second.

Problem 5 (12 pts):

We will build a naïve Bayes classifier based on the below weather dataset in order to determine whether to play golf or not. There are four categorical attributes (outlook, temperature, humidity, windy) and one binary target (play).

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

(a) For a day when it is sunny, hot, windy with high humidity, please use naïve Bayes to predict if we should play golf or not. (10pts)

Based on the results below, we should **not** play golf tomorrow (final answer in yellow):

	yes	no	yes	no
sunny	2	3	22.22%	60.00%
overcast	4	0	44.44%	0.00%
rainy	3	2	33.33%	40.00%
	9	5		
hot	2	2	22.22%	40.00%
mild	4	2	44.44%	40.00%
cool	3	1	33.33%	20.00%
	9	5		
high	3	4	33.33%	80.00%
normal	6	1	66.67%	20.00%
	9	5		
TRUE	3	3	33.33%	60.00%
FALSE	6	2	66.67%	40.00%
	9	5		
yes	9	5	64.29%	35.71%
no				
	14	14		

	sunny	hot	high	TRUE	Total
P(yes)	22.22%	22.22%	33.33%	33.33%	0.0054869684
P(no)	60.00%	40.00%	80.00%	60.00%	0.1152

	Conditions	Overall	
P(yes)	0.0054869684	64.29%	0.0035273369
P(no)	0.1152	35.71%	**0.0411428571**

(b) Does the prediction agree with the classification provided in the training data set? (2pts)

The prediction based on Naive Bayes does agree with the classification provided in the training set. The matching record (highlighted yellow) agrees that in those conditions, we should not play golf.

Problem 6 (10 pts):

Use the same training dataset (as in Problem 5) for golf playing and calculate the better splitting attribute (between OUTLOOK and HUMIDITY) to use as the first level attribute in constructing decision tree with entropy.

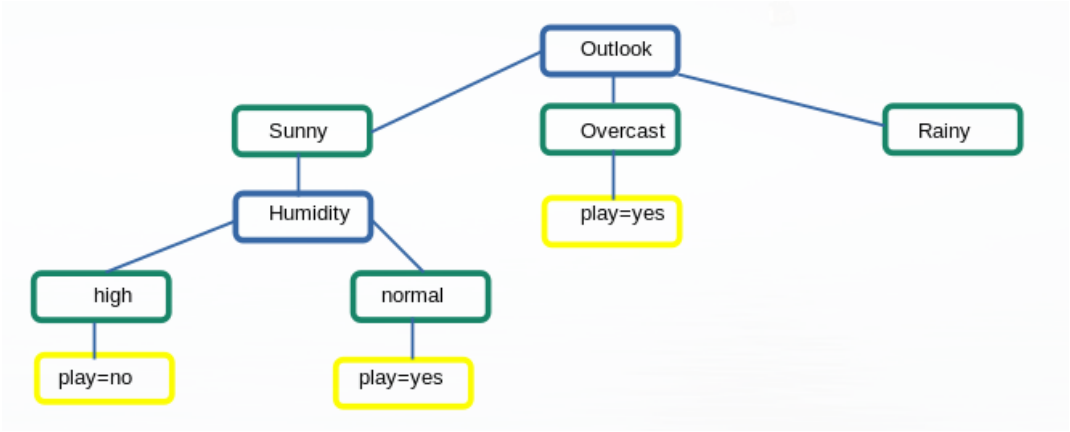
outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

First, we will calculate the entropies for play, outlook, and humidity. Then we will calculate the information gain for outlook and play, and choose the higher one as the root (in this case, it was outlook). Next, to build the decision tree, we can resort the above table and remove the unused columns to help identify the optimal splits. We can see that, in all cases of an overcast day, the decision to play is yes, so we can add that leaf. Finally, the split for humidity on a sunny day gives us a homogenous split, which would be the optimal choice in building a decision tree, so we can add the yes and no leafs to complete the tree.

		No	Yes	Entropy	Probability	Total
	E(play)	35.71%	64.29%	**0.9402859587**		
				Entropy	Probability	Total
E(outlook)	sunny	40.00%	60.00%	0.9709505945	0.3467680694	
	overcast	0	44.44%		0	
	rainy	60.00%	40.00%	0.9709505945	0.3467680694	**0.6935361389**
E(humidity)	high	42.86%	57.14%	0.985228136	0.492614068	
	normal	85.71%	14.29%	0.5916727786	0.2958363893	**0.7884504573**

G(Play, Outlook)	**0.2467498198**
G(Play, humidity)	**0.1518355014**

overcast	high	yes
overcast	normal	yes
overcast	high	yes
overcast	normal	yes
rainy	high	yes
rainy	normal	yes
rainy	normal	no
rainy	normal	yes
rainy	high	no
sunny	high	no
sunny	high	no
sunny	high	no
sunny	normal	yes
sunny	normal	yes



Problem 7 (10 pts):

A dataset of 800 cases was partitioned into a training set of 650 cases and a validation set of 150 cases. Classifier *A* predicts class label for the validation case based on the closest training case (a.k.a.1 Nearest Neighbor classifier). Classifier *A* has a misclassification error rate of 4% on the validation data. It was later discovered that the partitioning had been done incorrectly so that 80 cases from the training data set had been accidentally duplicated and had overwritten 80 cases in the validation dataset. What is the true misclassification error rate for the validation data?

If the classifier had a misclassification error rate of 4% on 150 validation cases, that means it misclassified (either false positive or false negative, does not matter) 6 records (4% of 150). If 80 of those records had accidentally been duplicated from the training set, then that means the classifier had seen them before. So we know that it could not have missed any of those 80. When we remove the 80, we now have a true validation set of only 70 records. We also still have the same amount of misclassifications, 6. Therfore, the classifier missed 6 out 70, for a true misclassification error rate of 8.57%.

Problem 8 (10 pts):

Please provide at least **FIVE** hyper-parameters we discussed in the class for a deep learning algorithm and discuss in details how each one of them impacts the performance of the model.

1) Optimization Method – Gradient Descent Learning Rate: the learning rate is the rate at which a network will update its parameters. To improve performance, you would want a lower learning rate. However, this comes at the expense of speed. Set the learning rate too high and the model may miss the ideal state. The optimal state needs to be conveged upon, too slow and the model may never converge (descend too slowly), too fast and it may overshoot (descend too quickly).

2) Number of Hidden Layers: Depending on the data itself will help answer. If the data is linearly seprable, then a single-layerd model (a linera model) wil be sufficient. However if the data is convex, hidden layers will be needed to derive an appropriate model. If the data has continuous mapping form only one space to only one other, one hidden layer should be sufficient. Anything more complex would require 2+ layers. Often times adding more layers does not produce any measurable gains. So if the data requires more than 1 hidden layer, I would argue for a trial-and-error approach. The number of layers is also influenced by the number of nodes in each layer, discussed separately.

3) Number of Nodes in Each Layer: Optimal number of nodes requires striking a balance. Too few nodes in each hidden layer can cause underfitting, and too many can cause overfitting. Additionally, more nodes increases computation time and may not yield appropriate gains for the computational effort. In short, add more nodes within reason until the desired overfitting/underfitting dynamic is achieived.

4) Loss Function: The error function that measures the amount of loss for the model. It is used so the weights can be updated to reduce the loss on each evaluation/pass. The choice of loss function will depend on the dataset. A regression loss function is appropriate for continuous datasets and the desired output is a real-value quantity (such as predicting stock prices). This will most likely have a linear activation function, and an example would be mean squared error. If the desired output is a classification (either binary or multi-class), the appropriate loss function would be cross-entropy/logarithmic loss.

5) Epochs: The number of times the entire dataset is passed through the network. If an optimization technique such as gradient descent is used and the weights are updated at each pass, then multiple passes will be required to find the optimal weights. The choice of epochs is another balance situation: too few can result in underfitting, while too many can reduce in overfitting. Also more epochs equate to more computational resources required. This last point is more apparent in epochs than other hyperparameters because the entire dataset is being reviewed each time, so the computational penalty for each epoch can be quite large. The best choice is to balance the number of epochs with both the batch size and the iterations so that an appropriate model can be found without resulting in an overfitting situation.