CPSC 340, 2017W2: SOLUTIONS TO MIDTERM EXAM

We are providing solutions because supervised learning is easier than unsupervised learning, and so we think having solutions available can help you learn. However, the solution file is meant for you alone and we do not give permission to share these solution files with anyone. Both distributing solution files to other people or using solution files provided to you by other people are considered academic misconduct. Please see UBC's policy on this topic if you are not familiar with it:
http://www.calendar.ubc.ca/vancouver/index.cfm?tree=3,54,111,959
http://www.calendar.ubc.ca/vancouver/index.cfm?tree=3,54,111,960

Question 1. (8 points)

Answer the questions below using 1-2 short sentences.

2 pts (a) What are two differences between KNN and k-means? (No more than two, please.)

Solution: Some possible answers: KNN supervised, k-means unsupervised; KNN non-parametric, k-means parametric; k-means sensitive to initialization, KNN not.

2 pts (b) Is it possible to have a machine learning model that makes predictions in $O(1)$ time? If yes, give an example; if no, explain why not.

Solution: Yes: decision tree, random forest.

2 pts (c) You're working on a machine learning problem and decide you need more data. You collect twice as much training data but end up with the same validation error for your parametric model. Are you likely experiencing underfitting or overfitting? Briefly justify your answer.

Solution: Underfitting; more training data should reduce approximation error but it doesn't seem like it's getting reduced.

2 pts (d) What is an advantage and a disadvantage of using more folds with cross-validation?

Solution: Advantage: better estimate of test error. Disadvantage: slower.

Question 2. (6 points)

Answer the questions below using 1-2 short sentences.

2 pts (a) Assume you have a classifier that takes $O(nd)$ time to train and $O(td)$ time to predict on t examples, like naive Bayes. If you are testing p possible values of a hyper-parameter, what is the cost of choosing the best value of the hyper-parameter using k-fold cross-validation? Express the result in terms of n, d, t, k, and p (if these affect the runtime).

Solution: $O(ndkp)$.

2 pts (b) Consider the "consistent nearest neighbour" classifier: it runs our usual KNN classifier but instead of viewing k as a hyper-parameter it always sets $k = \lceil \log(n) \rceil$ (the logarithm

of n rounded up to the nearest integer). Would call this a parametric classifier or a non-parametric classifier? Briefly justify your answer.

> **Solution:** Non-parametric, still need to store X which is $O(nd)$.

2 pts (c) Consider the "condensed nearest neighbour" classifier: at training time it chooses the c "best" training examples (where c is a hyper-parameter), and at test time uses the usual KNN prediction but based only on these c training examples. Would call this a parametric classifier or a non-parametric classifier? Briefly justify your answer.

> **Solution:** Parametric, only need to store c examples which is $O(cd)$.

Question 3. **(6 points)**

Answer the questions below using 1-2 short sentences.

2 pts (a) Does it make sense to do k-means clustering with $k > n$? Briefly justify your answer.

> **Solution:** No, that would be more than one cluster per example but clustering is about grouping examples.

2 pts (b) Does it make sense to do k-means clustering with $k > d$? Briefly justify your answer.

> **Solution:** Sure, e.g. the example we discussed in lectures, with 4 clusters in 2 dimensions.

2 pts (c) In k-means we can often obtain a much better clustering by using a large number of random initializations of the initial means. In DBSCAN (density-based clustering), we could randomize the order of the training examples that we test for new clusters. Is it generally a good idea to run DBSCAN with a large number of different random orderings? Briefly justify your answer.

> **Solution:** No, this would only change the boundary points so it wouldn't be a good use of time.

Question 4. **(6 points)**

Answer the questions below using 1-2 short sentences.

2 pts (a) Name one advantage and one disadvantage of using gradient descent instead of the normal equations to fit a least squares linear regression model.

> **Solution:** Advantages: can be faster for large d, can specify the desired accuracy of the solution. Disadvantages: need to set step size, slower and less precise for many reasonable problem sizes.

2 pts (b) When we do regression with a polynomial basis, how does the degree of the polynomial p affect the two parts of the fundamental trade-off?

Solution: As p goes up, the training error goes down and approximation error goes up.

2 pts (c) Construct a matrix X where the least squares solution would not be unique.

Solution: One possibility is a repeated column.

Question 5. **(6 points)**

Loss functions.

2 pts (a) Describe a situation where using a linear regression model with the squared error could give very misleading results.

Solution: Outliers. Or the actual relationship is non-linear.

2 pts (b) Consider the loss function $f(r) = \sum_{i=1}^{n} \max\{r_i, -2r_i\}$. Write down a version of this loss function that is smoothed with the log-sum-exp.

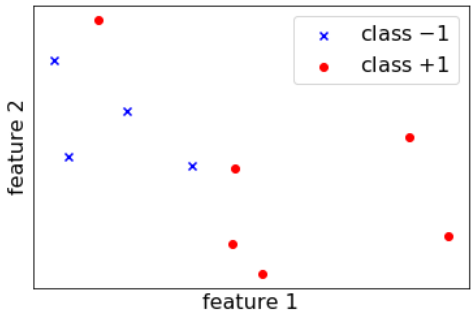
Solution: $f(r) = \sum_{i=1}^{n} \log(\exp(r_i) + \exp(-2r_i))$

2 pts (c) The Huber loss has a hyperparameter, δ, which controls where you switch from a parabola to a constant slope. What could go wrong if δ is set to an extremely large (far from zero) value?

Solution: You're no longer robust to outliers because all the the data values would land in the parabolic part.

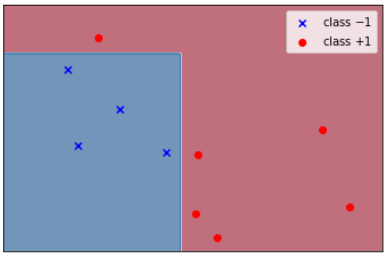
Question 6. **(6 points)**

Consider the binary classification training data set shown below. The two classes are denoted with o's and $\times$'s.



3 pts (a) What is the minimum depth of decision tree needed to get zero training error on this data set? Briefly explain your reasoning. Feel free to draw on the plot if it helps.

Solution: Depth=2. For example, first we slice off the red points on the right, then we slice the left half vertically to separate out the remaining two red points. See below:



3 pts (b) What is the maximum value of k such that KNN gets zero training error on this data set? Assume ties are broken by voting for class +1 (the circles). Briefly explain your reasoning. Feel free to draw on the plot if it helps.

Solution: $k = 1$. Even $k = 2$ already gets training errors, and then so does any larger value of k.

Question 7. **(6 points)**

Consider the following one-dimensional data set:

$$X = \begin{bmatrix} -3 \\ 4 \\ -1 \\ 3 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

2 pts (a) Write down the Z matrix for using a polynomial basis with $p = 2$ (quadratic) on this data set. Use the same standard format/notation that we used in the lectures and assignments.

Solution: $Z = \begin{bmatrix} 1 & -3 & 9 \\ 1 & 4 & 16 \\ 1 & -1 & 1 \\ 1 & 3 & 9 \end{bmatrix}$

2 pts (b) Let's say the weights come out to be $v = \begin{bmatrix} -1 \\ 3 \\ 0 \end{bmatrix}$. What is the model's prediction for $\tilde{x} = 2$?

Solution: $2^0 \times (-1) + 2^1 \times 3 + 2^2 \times (0) = -1 + 6 - 0 = 5$.

2 pts (c) If we instead used $p = 1$ on this dataset, would we expect the weights to be $\begin{bmatrix} -1 \\ 3 \end{bmatrix}$ (i.e. the first two elements of our previous v), or something different? Briefly justify your answer.

Solution: Yes. This wouldn't be the case in general for most v vectors, but because $v_2 = 0$, it means the quadratic term isn't being used and so what we have is already the best linear fit.

Question 8. **(6 points)**

3 pts (a) Consider the following objective, which considers a weighted worst-case error with a penalty on the absolute value of the weights,

$$f(w) = \max_{i \in \{1,2,\ldots,n\}} \{v_i |w^T x_i - y_i|\} + \lambda \sum_{j=1}^{d} |w_j|,$$

where λ is a non-negative scalar. Re-write this objective function in matrix and norm notation. You can use V as a diagonal matrix with the elements v_i along the diagonal.

Solution: $f(w) = \|V(Xw - y)\|_\infty + \lambda \|w\|_1$.

3 pts (b) Consider the L2-regularized *tilted* least squares objective,

$$f(w) = \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \frac{\lambda}{2} \sum_{j=1}^{d} w_j^2 + \sum_{j=1}^{d} v_j w_j,$$

where the λ is a non-negative scalar and the v_j are real-valued "tilting" variables. Write down a linear system whose solution minimizes this (convex and quadratic) objective function. You can use v as a vector containing the v_j values.

Solution:

$$(X^T X + \lambda I) w = (X^T y - v).$$