

CPSC 340 Midterm (Fall 2016)

Name:

Student Number:

Please enter your information above, turn off cellphones, space yourselves out throughout the room, and wait until the official start of the exam to begin. You can raise your hand to ask a question, and please look up occasionally in case there are clarifications written on the projector (the time will also be written on the projector). You are welcome to (quietly) leave early if you finish early, and we will ask everyone to stop at 2:55.

The midterm consists of 5 questions, and they will all be equally weighted in the marking scheme. Note that some question have multiple parts, written as **(a)** and **(b)**. All parts are equally weighted. *Clearly show where you are answering each part*, and make sure to check that you answered all parts before handing in your midterm.

Good luck!

1
<hr/>
2(a)
<hr/>
2(b)
<hr/>
3(a)
<hr/>
3(b)
<hr/>
4(a)
<hr/>
4(b)
<hr/>
5(a)
<hr/>
5(b)
<hr/>
Total

1 Cross-Validation

Consider a supervised **regression** problem where we have **60** training examples and 10 features, where the features are stored in a **60** by 10 matrix X and the labels are stored in a **60** by 1 vector y . The examples are given to you **sorted** by y_i values. As in the assignments, assume that you have a ‘model’ function that depends on a parameter ‘ k ’ with the following interface:

- `model = train(X,y,k);` % Train model on $\{X, y\}$ with parameter k
- `yhat = predict(model,Xhat);` % Predict using the model on $Xhat$.

Assume that k can be either 1, 2, or 3.

Give pseudo-code describing how to choose k using *3-fold* cross-validation.

2 Predictions for Linear Regression and K-Means

Consider the dataset below, which has 5 training examples and 2 features:

$$X = \begin{bmatrix} 2 & 1 \\ 10 & 4 \\ 10 & 9 \\ 5 & 8 \\ 8 & 10 \end{bmatrix}, \quad y = \begin{bmatrix} 8.0 \\ 6.5 \\ 4.0 \\ 7.5 \\ 2.0 \end{bmatrix}.$$

Suppose that you have the following test example:

$$\hat{x} = [9 \quad 4].$$

(a) Suppose we use a linear regression model with coefficients given by

$$w = \begin{bmatrix} 2/3 \\ -1/4 \end{bmatrix}.$$

What prediction \hat{y} would we make for the test example?

(b) Suppose we fit an unsupervised k-means model to this dataset and obtained the following means:

$$W = \begin{bmatrix} 9.0 & 9.5 \\ 3.0 & 5.0 \\ 9.5 & 4.0 \end{bmatrix}$$

Which cluster would the test example \hat{x} get assigned to?

3 Fundamental Trade-Off for CNN and Laplace Smoothing

On the assignment, you implemented a **condensed nearest neighbours** (CNN) classifier as a faster alternative to k-nearest neighbours. In your CNN method, you went through the training set *once* and stored examples that were misclassified. Consider a variation of CNN where you go through the training examples k times, each time adding even more examples that are misclassified.

So $k = 0$ is the usual CNN method, $k = 1$ will go through the dataset again and have more training examples in the subset than $k = 0$, and $k = 2$ will have more training examples than $k = 1$.

(a) Briefly explain how k would affect the two parts of the fundamental trade-off, and why it would have this effect on each part.

In the most basic naive Bayes model, we estimate our conditional probabilities from the data using

$$p(x_{ij} = c | y_i = 1) = \frac{(\text{number of times } x_{ij} = c \text{ and } y_i = 1)}{(\text{number of times } y_i = 1)}.$$

This can cause a problem if we've never seen a training example where $x_{ij} = c$ and $y_i = 1$, since the probability will be 0. The standard solution to this problem is to estimate the conditional probabilities with **Laplace smoothing**,

$$p(x_{ij} = c | y_i = 1) = \frac{(\text{number of times } x_{ij} = c \text{ and } y_i = 1) + \beta}{(\text{number of times } y_i = 1) + \beta k},$$

where k is the number of values that c can take and β is a positive constant.

(b) Briefly explain how β would affect the two parts of the fundamental trade-off, and why it would have this effect on each part.

4 Runtime of Random Trees

In the assignment, you implemented a variant on decision trees called a *random tree*. In a *random tree*, each time we fit a decision stump we only consider \sqrt{d} random features as candidates for the split variable. Following our usual convention, we'll use:

- n as the number of training examples.
- d as the number of features..
- m as the depth of the decision tree.
- t as the number of test examples.

(a) What is the cost of fitting a random tree of depth m to our training data in terms of the above four quantities? (Briefly justify your answer.)

(b) What is the cost of prediction on t test examples using a random tree in terms of the above four quantities? (Briefly justify your answer.)

5 Weighted Regression with Weighted Regularization

Consider a variation on L2-regularized least squares where we have a **positive** weight z_i for each training example i and a positive variable-specific regularization weight λ_j for each feature j . In this case our objective function can be written as

$$f(w) = \frac{1}{2}(Xw - y)^T Z(Xw - y) + \frac{1}{2}w^T \Lambda w,$$

where Z is an n by n diagonal matrix with the z_i values along its diagonal and Λ is a d by d matrix with the λ_j values along its diagonal.

(a) Write finding a minimizer w of this (convex) function as a system of linear equations.

Consider a variant where to gain robustness we use the absolute error and where we also regularize by a weighted sum of absolute values (instead of squaring the w_j),

$$f(w) = \sum_{i=1}^n z_i |w^T x_i - y_i| + \sum_{j=1}^d \lambda_j |w_j|.$$

(b) Express this function in terms of vectors, matrices, and norms (there should be no summations in the final answer, and you can use Z and Λ as defined in the previous part).

