

Midterm Examination

Tuesday October 28, 2pm to 3:20pm

Your name:

Instructions: Look through the whole exam and answer the questions that you find easiest first. Answer each question in the space below the question, using the backs of the pages for extra space as necessary. If necessary, you may make assumptions that are reasonable, and that do not make a question trivial. If you do make an assumption, state it clearly. This exam is open-book. You may use a calculator.

(Question 1) [24 points] UCSD has about 26,000 students. The following numbers are approximately true, according to Dr. John Sexton from Counseling and Psychological Services at UCSD. Every year, 1.4% of all students attempt suicide, and there are about 2 actual suicides. Students who have attempted suicide are 543 times more likely to commit suicide successfully in the next year.

(a) [2 points] What percentage of all suicide attempts are successful?

Number of attempts is $0.014 \cdot 26000 = 364$. Success rate is $2/364 = 0.55\%$.

(b) [6 points] On average, how many successful suicides per year are by students who did *not* previously attempt suicide?

Not enough information is given to treat this question from a temporal perspective. So, we will answer it just in terms of percentages. The result will still be useful for deciding where to allocate resources for suicide prevention. Write $x = p(\text{success}|\neg\text{attempt})$.

$$p(\text{success}|\text{attempt}) = 543x$$

$$p(\text{success}) = 2/26000$$

$$\begin{aligned} p(\text{success}) &= p(\text{success}|\text{attempt})p(\text{attempt}) + p(\text{success}|\neg\text{attempt})p(\neg\text{attempt}) \\ &= 543x \cdot 0.014 + x \cdot 0.986 \end{aligned}$$

$$2/26000 = x(543 \cdot 0.014 + 0.986)$$

$$x = \frac{2/26000}{543 \cdot 0.014 + 0.986} = 8.957 \cdot 10^{-6}$$

$$p(\text{success} \wedge \neg\text{attempt}) = x \cdot p(\neg\text{attempt}) = 8.831 \cdot 10^{-6}$$

$$\text{count}(\text{success} \wedge \neg\text{attempt}) = 26000p(\text{success} \wedge \neg\text{attempt}) = 0.230$$

On average, among the two suicides per year, only 0.23 are by students who did not previously make an attempt. So, it might be sensible to focus prevention efforts on the 364 students who have made previous attempts.

Now, suppose we have a database with detailed information about each UCSD student, with two labels for each person: whether or not s/he actually committed suicide, and whether or not s/he attempted suicide.

(c) [4 points] Explain why it would or would *not* be useful to train a classifier to distinguish the actual suicides from all other students.

It would not be useful because there are too few examples of the “actual suicide” class. The classifier would overfit the particular training examples of this class. For example, if both happened to be of the same gender, the classifier would predict that the probability of successful suicide was zero (or at least much lower) for the other gender.

(d) [4 points] Suppose you train a classifier to distinguish students who attempt suicide from all other students. Suppose the accuracy of this classifier, measured via cross-validation, is 95%. What can you say about the usefulness of this classifier?

This classifier has lower accuracy than 98.6% as achieved by the classifier that predicts no students ever attempt suicide. However, this classifier may still be useful if it has high precision and/or high recall.

(e) [4 points] Explain why a perceptron classifier is likely to be more useful than a 1-nearest neighbor classifier for part (d), even if both classifiers have identical confusion matrices.

With identical confusion matrices, both classifiers have the same accuracy, precision, and recall. However, the perceptron classifier can provide more information: the dot-product score $\bar{w} \cdot \bar{x}$ gives a ranking of students from most-likely to least-likely to attempt suicide. Hence, it can be used to focus resources on the students who are at greatest risk. The 1NN classifier doesn't provide a useful ranking of test examples.

The perceptron will be faster on test examples, but this is not a major benefit, assuming that the NN classifier is fast enough to be usable, which it would be with a training set of cardinality 26,000.

Another benefit of the perceptron classifier is that for each feature x_j , the weight w_j provides some information about the predictiveness of this feature. A nearest-neighbor classifier provides no information about the relative importance of different features.

(f) [4 points] Explain why a logistic regression classifier is likely to be more useful than a perceptron classifier for part (d), even if both classifiers have identical confusion matrices.

The logistic classifier will provide calibrated estimates of the probability of a suicide attempt for each student. These numbers are more useful for making decisions than having just a ranking, or just scores that are not interpretable as probabilities.

For each feature x_j , the logistic regression weight w_j is interpretable as a log-odds contribution. However, these weights (and also perceptron weights) only have meaning in combination, not in isolation.

(Question 2) [24 points] Consider the nonlinear prediction function

$$\hat{y}(\bar{x}) = f\left(\alpha + \sum_{j=1}^d \beta_j x_j\right)$$

where d is the dimensionality of an example \bar{x} . Suppose that the training objective s to be minimized is a sum of per-example errors:

$$s = \sum_{i=1}^n e(\hat{y}(\bar{x}_i), y_i)$$

where n is the number of training examples. Your goal is to derive a stochastic gradient training algorithm for this general prediction model.

(a) [2 points] Explain how to rewrite the model to eliminate α as a separate parameter.

Let $\alpha = \beta_0$ and let the zeroth feature value $x_0 = 1$ for every training and test example.

(b) [4 points] Explain the e and f functions that make least-squares linear regression a special case of this model.

$f(x)$ is the identity function $f(x) = x$ and $e(a, b)$ is the squared-difference function $e(a, b) = (a - b)^2$.

(c) [4 points] Is logistic regression a special case of this model? Justify your answer.

Yes, $f(x)$ is the sigmoid function $f(x) = 1/(1 + e^{-x})$ and $e(a, b)$ is negative log conditional likelihood: $e(a, b) = -\log a$ if $b = 1$ and $e(a, b) = -\log(1 - a)$ if $b = 0$.

(d) [4 points] Obtain the partial derivative for one training example and one parameter,

$$\frac{\partial}{\partial \beta_j} e(\hat{y}(\bar{x}), y).$$

Note that y does not depend on β_j but $\hat{y}(\bar{x})$ does. The partial derivative is

$$\begin{aligned} \frac{\partial}{\partial \beta_j} e(\hat{y}(\bar{x}), y) &= e'(\hat{y}(\bar{x}), y) \frac{\partial}{\partial \beta_j} \hat{y}(\bar{x}) \\ &= e'(\hat{y}(\bar{x}), y) f'\left(\sum_{j=0}^d \beta_j x_j\right) \frac{\partial}{\partial \beta_j} \sum_{j=0}^d \beta_j x_j \\ &= e'(\hat{y}(\bar{x}), y) f'\left(\sum_{j=0}^d \beta_j x_j\right) x_j \end{aligned}$$

where f' is the derivative of f and e' is the partial derivative of e with respect to its first argument.

(e) [4 points] Obtain the special case of part (d) for least-squares linear regression.

The result is

$$e'(\hat{y}(\bar{x}), y) f'\left(\sum_{j=0}^d \beta_j x_j\right) x_j = 2(\hat{y}(\bar{x}) - y) \cdot 1 \cdot x_j$$

(f) [6 points] Explain the design decisions you must make in order to turn part (d) into a fully-specified stochastic gradient training method.

Since the goal is minimization, we do stochastic gradient descent. The update rule for each parameter, for each training example, is

$$\beta_j := \beta_j - \lambda \frac{\partial}{\partial \beta_j} e(\hat{y}(\bar{x}), y).$$

Minimally, we need to specify (a) the learning rate λ , (b) a termination criterion, and (c) how to initialize the β vector. Additionally, we should specify how to normalize each dimension of the training data, and how to reduce the learning rate as training progresses.

(Question 3) [30 points] For each statement below, clearly write “True” if it is mostly true, or “False” if it is mostly false. Then in the space below, write one or two sentences explaining why or how the statement is true or false. The maximum score for each answer is three points.

1. For use with a nearest neighbor classifier, Euclidean distance and squared Euclidean distance are equivalent.

True. They lead to the same ranking of neighbors, with the same ties. (One possible non-equivalence is if distances are summed or averaged in order to break ties.)

2. The 3-nearest neighbor classifier is always more accurate than the 2-nearest neighbor classifier.

False. Two of the three nearest neighbors could be wrong, while tie-breaking makes the 2-nearest neighbor classifier right.

3. With enough training data, the error of a nearest neighbor classifier always goes down to zero.

False. Some classes could overlap, in which case the Bayes' error rate is non-zero. Then no classifier can achieve zero error rate.

4. The perceptron algorithm updates the current linear separator if and only if the current training example is misclassified.

True. This is part of the statement of the algorithm.

5. If the training set is *finite* and linearly separable, then the perceptron convergence theorem says that the perceptron algorithm will learn a correct linear separator in finite time.

True. If the training set is finite and linearly separable, then the real values R and δ needed by the convergence theorem always exist.

6. If the training set is *infinite* and linearly separable, then the perceptron convergence theorem says that the perceptron algorithm will learn a correct linear separator in finite time.

False. In this case, the norm of training examples may be unbounded, and then R does not exist. And/or, the minimum positive separation δ may not exist.

7. Cross-validation can reveal overfitting.

True. If accuracy on the training folds is higher than on the test folds, then overfitting is revealed.

8. Overfitting is a danger when learning a classifier, but not when doing unsupervised learning.

False. Informally, overfitting means finding patterns in the data that are spurious because they are true in the training data but false in the test data. It is certainly possible to find spurious patterns in unlabeled data.

9. With k -fold cross-validation, larger k is always better.

False. Larger k is slower, and may fail to reveal an issue that would be revealed by smaller k . Remember the scenario where every training example is duplicated.

10. Bernoulli and Gaussian distributions are both probability density functions (pdfs).

False. A Gaussian is a pdf but a Bernoulli is a pmf (probability mass function).