Brown University
DATA 1010
Fall 2019: Practice Midterm III
Samuel S. Watson

Name:

You will have three hours to complete the exam, which consists of 36 questions. Among the first 24 questions, you should only solve problems for standards for which you want to improve your medal from the second exam.

No calculators or other materials are allowed, except the provided reference sheets.

You are responsible for explaining your answer to **every** *question. Your explanations do not have to be any longer than necessary to convince the reader that your answer is correct.*

For questions with a final answer box, please write your answer as clearly as possible and strictly in accordance with the format specified in the problem statement. Do not write anything else in the answer box. Your answers will be grouped by Gradescope's AI, so following these instructions will make the grading process much smoother.

I verify that I have read the instructions and will abide by the rules of the exam: ______________________________

Problem 1 **[STATLEARN]**

Give an example which shows that a simple linear regression model can overfit the data. Give some ideas for how to mitigate the overfitting.

Solution

For simplicity, we'll consider linear regression with no bias term, though the same conclusions would apply if we dropped that assumption. Suppose the $n \times 2$ feature matrix's columns measure the same quantity but with a different rounding rule, so that the columns are not actually identical. In geometric terms, those two columns point in nearly the same direction, but they nevertheless span a plane in $\mathbb{R}^n$.

Performing ordinary least squares linear regression will identify the vector in that plane which is closest to the vector of response values. However, this vector might very well be far from the lines spanned by the two columns. In that case, the regression optimization will leverage the difference between the two columns (which contains no predictive meaning) to obtain a vector which is much closer to the response vector than is meaningfully possible.

There are many possible solutions to this problem. We could penalize large regression coefficients in the loss functional, we could use PCA and throw out components corresponding to small singular values, or we could simply omit one of the two columns.

Problem 2 [LRC]

Show that the Bayes classifier is the classifier which minimizes the misclassification probability.

For simplicity, you may assume the context of a binary classification problem with a discrete sample space.

Solution

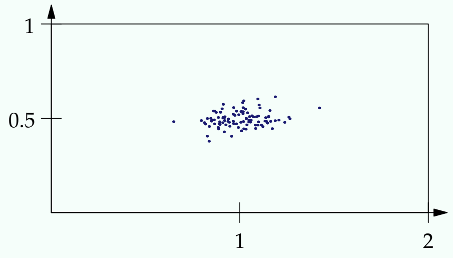
The misclassification probability is a sum of the probability masses at each incorrectly predicted point in the space $\mathcal{X} \times \mathcal{Y}$. For example, if $\mathcal{X} = \{1, 2, 3\}$, $\mathcal{Y} = \{-1, 1\}$, and `h.([1,2,3]) == [-1,+1,-1]`, then the misclassification probability would be the sum of the probability masses at $(1, +1)$, $(2, -1)$, and $(3, +1)$.

Since we can choose predictions for each $x \in \mathcal{X}$ independently, we can minimize the misclassification probability by classifying each x according to whether $(x, +1)$ or $(x, -1)$ has larger mass. In other words, we classify according to whether $p_{+1}f_{+1}(x)$ is larger than $p_{-1}f_{-1}(x)$. Since that is the rule for the Bayes classifier, this shows that the Bayes classification rule does minimize the misclassification probability.

Problem 3 [KDE]

(a) Suppose that $f_\lambda(x, y)$ is the bandwidth-λ kernel density estimator associated with the set of samples shown in the figure (based on the tri-cube weight function).

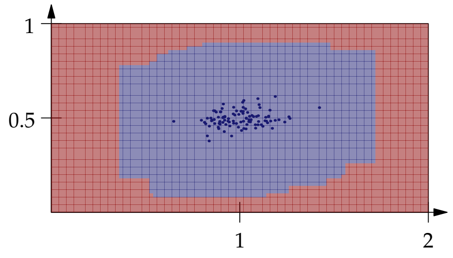
Estimate the value of λ such that the points (x, y) for which $f_\lambda(x, y) = 0$ make up approximately half of the rectangle by area.



(b) Consider a set of points $\{(x_i, y_i)\}_{i=1}^n$ in $\mathbb{R}^2$ and a positive value of λ. Suppose that the vertical line $x = a$ passes through the three sample points (x_1, y_1), (x_2, y_2), and (x_3, y_3), and that no other sample points have an x value within λ of a. Find the value of $r_\lambda(a)$ (where r_λ is the Nadaraya-Watson estimator associated with the samples).

Solution

(a) If we choose a λ value of $\frac{1}{2}$, the union of the side-length-2λ squares centered at all the samples would fill up nearly the whole rectangle. I would estimate $\lambda \approx 0.3$ to fill up half the rectangle. (Checking it by computer reveals the answer to be about 0.302.)



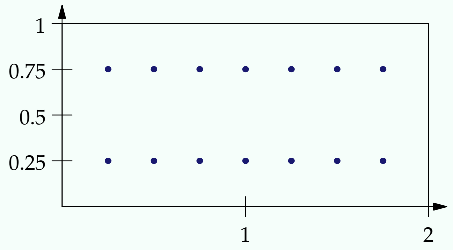
(b) Integrating the kernel density estimator along a vertical line results in a weighted average of the y-values of the centers of the intersecting squares, with weight given by the kernel function evaluated at the horizontal distance to the square center. Since the vertical line passes through the center of each square, the weights are equal, so the result is $\frac{1}{3}(y_1 + y_2 + y_3)$.

Final answer:

$\frac{1}{3}(y_1 + y_2 + y_3)$

Problem 4 [LINREG]

Find the residual sum of squares for the line of best fit for the samples shown.



Solution

The line which minimizes the residual sum of squares is $y = 0.5$, because for any function $r(x)$, the sum of squared residuals along each vertical line through a pair of sample points is $(0.75 - r(x))^2 + (r(x) - 0.25)^2$, which can be no smaller than its value when $r(x) = 0.5$.

Therefore, the minimum RSS is $14(0.25)^2 = \boxed{7/8}$.

Final answer:

$$\frac{7}{8}$$

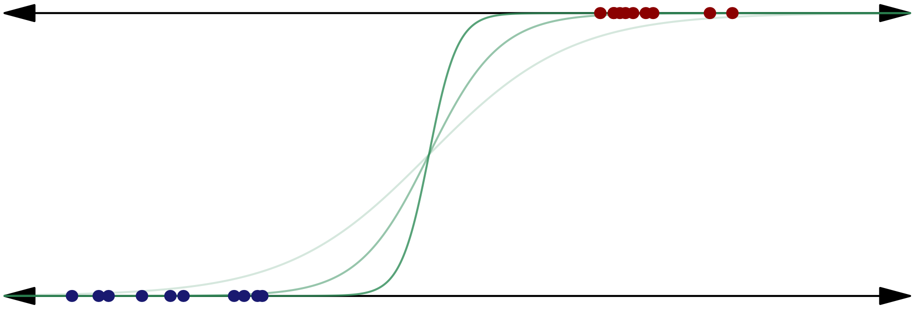
Problem 5 **[LOGIST]**

Consider a binary classification problem for which there exists a hyperplane separating the classes. What goes wrong if you try to apply logistic regression?

Solution

The problem is that the logistic regression optimization problem is unbounded. Given any logistic function $\mathbf{x} \mapsto \sigma(\boldsymbol{\beta} \cdot \mathbf{x} + \alpha)$ whose decision boundary separates the two classes, we can improve it by merely scaling up $\boldsymbol{\beta}$ and α. This will preserve the decision boundary while enhancing the confidence of all predictions (which are already correct). Therefore, the loss function can always be decreased, and no minimum exists.

The figure below shows a 1D classification problem with separable classes. Higher opacity corresponds to scaled-up α and $\boldsymbol{\beta}$ values.

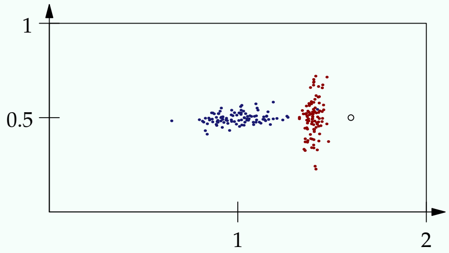


Problem 6 [QDA]

Select which of the two statements is correct (given that one of them is correct), and explain why the two classifiers behave differently.

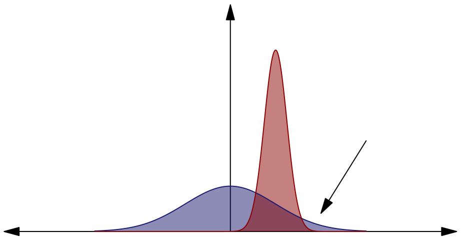
(a) The point marked with a hollow circle is classified as blue by a QDA classifier and red by a kernel density classifier.

(b) The point marked with a hollow circle is classified as red by a QDA classifier and blue by a kernel density classifier.



Solution

(a) is correct. The kernel density estimator will assign more red mass to the location of the hollow point, since there are many more nearby red points. However, QDA assumes that the distributions are multivariate Gaussian, which means that the horizontal component of blue distribution is a Gaussian with smaller mean but larger variance than the horizontal component of the red distribution. It can therefore be larger at the hollow point (as shown in the figure).

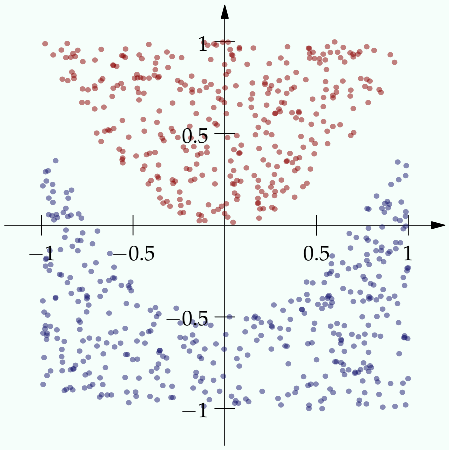


Final answer:

(a)

Problem 7 [SVM]

Find a map from the plane to some other Euclidean space such that hard-margin SVM could, after applying the map, be used for classification problem shown in the figure.



Solution

We could map (x, y) *down* in dimension to $y - x^2 \in \mathbb{R}$. Then all of the red samples would lie to the right of the origin, and all of the blue samples would lie to the left.

Alternatively, we could apply the map $(x, y) \mapsto (x, y, y - x^2)$, which retains the original information in the data but also permits a separating plane (any plane of the form $z = c$, where c is between $-\frac{1}{2}$ and 0).

Final answer:

$$(x, y) \mapsto (x, y, y - x^2)$$

Problem 8 [DECTREE]

Are decision tree classifiers scale sensitive? In other words, if a feature is scaled by the same constant factor for all observations, do we end up with a different trained decision tree classifier?

Solution

No, decision trees are not sensitive to scaling. Each step of the decision tree training process considers only how the observations split along each feature axis, and sets of real numbers have the property that their splits are the same regardless of scaling (e.g., if all adult giraffe heights are greater than all adult mouse heights, then that remains true whether heights are measured in inches or centimeters).

Problem 9 [ENSEMBLE]

For each of the following assertions, determine whether it is true or false.

(a) If the individual models which make up an ensemble classifier do not have very high accuracy, then the ensemble classifier will also have pretty low accuracy.

(b) An ensembled regression function always has lower loss than its constitutent models individually.

(c) The constituent models must be independent for ensembling to work.

(d) Bagging is analogous to a referendum, while gradient boosting is like getting better over time incrementally, with each step taken to make the best improvement we can given our constraints.

Solution

(a) **False**. The individual models can be weak learners and still aggregate to a very strong learner. For example, suppose that each of 1000 independent models can solve a binary classification problem with probability 55%. Then by the central limit theorem, the probability that the majority of their votes will go to the correct class is very high.

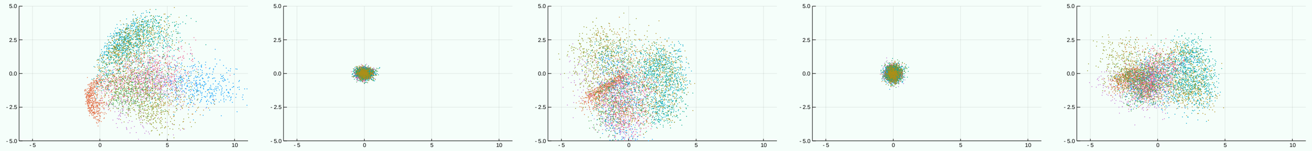
(b) **False**. This is typically true, but it doesn't have to be. For a simple (if trivial) counterexample, consider ensembling 100 instances of the exact same model. The predictions of the ensemble will be the same as the predictions of the individual models, so the loss will be the same.

(c) **False**. Low correlation is sufficient.

(d) **True**. Both analogies are solid. The voters are like the individual models in a bagged ensemble, while gradient boosting can be thought of as gradient descent with steps taken by fitting a model to the desired movement, rather than stepping freely in direction opposite the gradient (as one would do in ordinary gradient descent).

Problem 10 [DIMRED]

The first graph below shows the dot product of each of the first 5000 (de-meaned) vectors in the MNIST training set with the first principal component and the second principal component. The remaining graphs are similar, but using different pairs of principal components. One uses the second and third, one uses the second and tenth, one uses the 80th and 81st, and one uses the 80th and 120th. Figure out which is which.



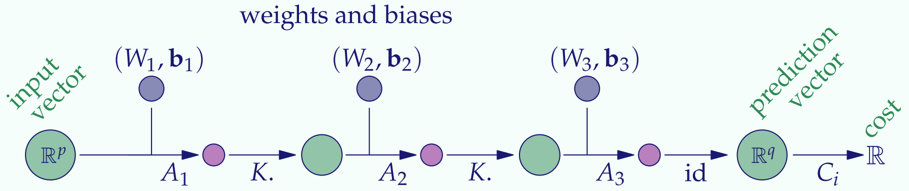
Solution

The first one shows the first two principal components, as stated in the problem. The key idea for identifying the remaining four is that the principal components are arranged in decreasing order of variation. This means that the sum of squared lengths of the vectors obtained by projecting the points onto the first principal component is larger than for the second principal component, which in turn is larger than for the third, and so on.

So, the middle and last figures show $(2, 3)$ and $(2, 10)$, which we can distinguish by the amount of vertical variation. The remaining two show $(80, 120)$ and $(80, 81)$, respectively, which we can again distinguish by the amount of vertical variation.

Problem 11 [NN]

Suppose that weights and biases have been chosen for the neural network shown, and that a vector has been forward propagated through the network. Suppose that the vectors recorded at the purple nodes are $[1, -4, 2]$, $[6, 3]$, and $[9, 7, -4, -1, 5]$.



(a) What is the architecture of this neural net?

(b) What vector is recorded at the second green node (the one between A_1 and A_2)?

(c) Now suppose that we are in the midst of the backpropagation process, and we have just determined that the derivative of the cost with respect to the vector in the second purple node is equal to $[-3, -4]'$. Calculate the derivative of the cost with respect to the matrix W_2.

Solution

(a) The architecture is $[p, 3, 2, 5]$, since those are the dimensions of the Euclidean spaces in the green nodes in the diagram.

(b) We apply $K.$ to $[1, -4, 2]$, and we get $[1, 0, 2]$.

(c) We learned that this derivative is equal to the outer product of the *gradient* matrix stored in the following purple node and the original vector stored in the previous green node. So we get that the derivative of cost with respect to W_2 is

$$\begin{bmatrix} -3 \\ -4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} -3 & 0 & -6 \\ -4 & 0 & -8 \end{bmatrix}$$

Problem 12 **[FREQBAYES]**

(a) Explain why conjugate priors are an exclusively Bayesian statistics topic (in other words, explain why they are not useful/meaningful in the frequentist framework).

(b) Outline a strategy for computing a Bayesian point estimate, supposing that our prior distribution is not a conjugate prior for the problem at hand.

Solution

(a) Frequentist statistics does not treat the model parameters as random, so in particular they do not have prior distributions. Therefore, it is not meaningful to discuss whether the prior and posterior distributions belong to the same family.

(b) We sample from the posterior distribution using Markov Chain Monte Carlo and then take an average of many such observations from that distribution.