6.790 Homework 3

Revision: 10/3/24 3:03PM

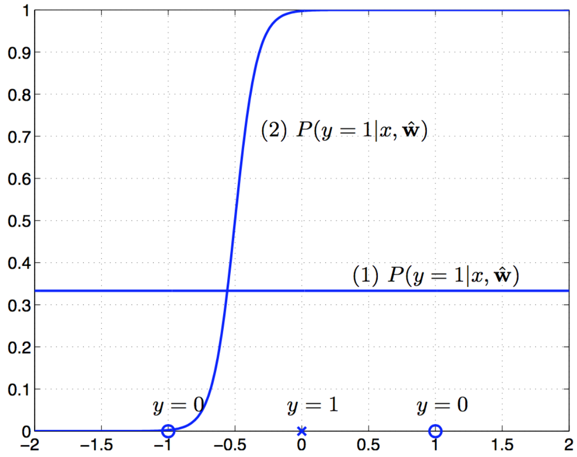
Questions 1 is optional. The remaining problems are required. All the questions are written (don't include running code) and go over modeling choices, decision boundaries, uncertainty estimation, and optimization. There are some hints in the blue boxes that are supposed to help you in your solution, but you can choose to disregard them.

Please hand in your work via Gradescope via the link at https://gradml.mit.edu/info/homeworks/. If you were not added to the course automatically, please use Entry Code R7RGGX to add yourself to Gradescope. Make sure to assign the problems to the corresponding pages in your solution when submitting via Gradescope.

1. Latex is not required, but if you are hand-writing your solutions, please write clearly and carefully. You should include enough work to show how you derived your answers, but you don't have to give careful proofs.
2. Homework is due on Tuesday October 8 at 11PM.
3. Lateness and extension policies are described at https://gradml.mit.edu/info/class_policy/.

Contents

1 Optional warm-up: Logistic regression: basic intuition with one-dimensional data



1.

Consider a simple one dimensional logistic regression model

$$P(y=1 \mid x, w) = \sigma(w_0 + w_1 x)$$

where $\sigma(z) = (1+\exp(-z))^{-1}$ is the logistic function.

The figure above shows two possible conditional distributions $P(y=1 \mid x, w)$, viewed as a function of x, that we can get by changing the parameters w.

Assume we have a data set $\mathcal{D} = \{(-1,0),(0,1),(1,0)\}$.

(a) How many classification errors does hypothesis 1 make?

(b) How many classification errors does hypothesis 2 make?

(c) Which of these two hypotheses assigns a higher likelihood to the data?

(d) If your loss function for predictions was

$$L(g,a) = \begin{cases} 0 & \text{if } g = a \\ 1 & \text{if } g = 1 \text{ and } a = 0 \\ 10 & \text{if } g = 0 \text{ and } a = 1 \end{cases}$$

- What output would you predict for $x=-1$ when conditional (1) is the result of your learning?
- What output would you predict for $x=-1$ when conditional (2) is the result of your learning?

2 Noisy sensor [10 points]

2. You are trying to solve a perceptual recognition problem for a robot in simulation, and you find that you can find a function $f(x) = \theta^T x$ that has the property that if $f(x) > 0$ then the robot can fit through the door it is looking at and if it's less than 0 then the robot can't fit.

   Now you are going to deploy this system in the real world, but find that there are some perturbations in the world that make your prediction rule less accurate than it was. We can model the real-world process as

$$y = \begin{cases} 1 & \text{if } \theta^T x + \epsilon \geqslant 0 \\ 0 & \text{otherwise} \end{cases}$$

   You gather a lot of data and find that the noise ϵ seems to have a distribution with pdf $\frac{e^{-\epsilon}}{(1+e^{-\epsilon})^2}$

   Obviously you can't really apply this predictor in the real world, because you don't know the ϵ value! But, given a new input x, you can compute $f(x)$ and you think the higher that value is, the more sure you ought to be that the actual y value associated with it will be 1.

   Under the assumptions above, what is $p(y = 1 \mid f(x))$? Your final answer can be written in terms of $f(x), \theta, x$ and should not include integrals.

   Hint: it might be helpful (but not strictly necessary) to look at the logistic distribution in order to avoid complicated integral calculations.

3 How wrong? [15 points]

3. You are the chief data scientist at Yoyodyne. Your job is to predict the lifetime, in days, of your new Dynamo fan product. You have run 200 dynamo fans to exhaustion in different situations, drawn from the same environmental distribution. For each you have collected information about the humidity, temperature, particulate density, and energy quality of the location it was used in, as well as the working lifetime of the fan in days. You divide the data into a training set of size 180 and a validation set of size 20, and use 6.790 methods on the training set (without peeking at the validation set...even once!) to come up an awesome hypothesis h.

   (a) (5 points) You evaluate h on your validation set, using squared loss, and end up with a mean loss of $M_n = 50$ with a standard deviation of $\hat{\sigma} = 10$.

   You are completely convinced that you know the standard deviation and that the true $\sigma = 10$*, as well.*

   Provide a bound on the probability that the true risk of h is greater than 60.

   Feel free to use the following application of Chebychev's inequality:

   Proposition: Let $X_1, \ldots, X_n$ be iid with finite mean μ and variance σ^2, and let $M_n = (X_1 + \ldots + X_n)/n$. Then

$$P(|M_n - \mu| \geqslant \epsilon) \leqslant \frac{\sigma^2}{n\epsilon^2}$$

(b) (5 points) Now, the marketing department wants to make a web site that allows customers to input the parameters of their intended deployment, and then outputs the result of applying h to that situation to predict the Dynamo's lifetime. They call to ask you what kind of claim it can make about the reliability your predictions!

In particular, they want to say that, 99 times out of 100, the prediction is within some δ of the true lifetime. Assuming that *the standard deviation of the distribution of the prediction error* is $\sigma_{\text{err}} = 10$, what is the smallest value of δ you can responsibly claim?

Is the *standard deviation of the prediction error* the same as the *standard error* (described in lecture 7)? Explain.

(c) (5 points) Thinking back, you wonder what would have happened if you had changed the percentage of your data you used for training vs testing. Qualitatively, how would you expect your answers to the previous two questions to change, if you had used 50% of your data for training and 50% for validation, how would this change the following outcomes compared to your original setup, where 90% of the data was used for training and 10% for validation?

(d) (0 points) (Optional but interesting if you like statstics!) Equation 2 of the paper *Multivariate Chebyshev Inequality with Estimated Mean and Variance* by Stellato, Van Parys, and Goulart, `https://mitsloan.mit.edu/shared/ods/documents?PublicationDocumentID=4759` contains a bound (due to Saw et al) on the error of using the sample mean as a prediction for the $n+1$st value, *as a function of the sample mean, sample standard deviation, and sample size*. This is pretty cool! Play with it and see if you can get an answer to part 2 this way. If you do this, you can skip the rest of this question.

4 Why not regression? [10 points]

4. Jan is trying to do classification: $y^{(i)} \in \{0, 1\}$ and $x^{(i)} \in \mathbb{R}$. But he slept through all of the classification lectures, so he decides to solve classification using regression. That is, he ignores the fact that $y^{(i)}$ is binary, and fits a linear regression function via least squares. The resulting regression function is:

$$\hat{y} = f(x; w) = w_0 + x w_1$$

Jan uses the decision rule: label 1 if $f(x; w) > 1/2$; and label 0 otherwise.

Suppose the training data is linearly separable. Is Jan's decision rule (with associated regression function) guaranteed to classify the training data without error?

1. Yes. Provide a short argument:
2. No. Provide a counterexample:

5 Modeling the distribution versus training a classifier [15 points]

5. (Murphy 10.3)

Suppose we train the following binary classifiers via maximum likelihood.

(a) GaussI: A generative classifier, where the class conditional densities are Gaussian, with both covariance matrices set to I (identity matrix), i.e., $p(\mathbf{x} \mid y = c) = \mathcal{N}(\mathbf{x} \mid \mu_c, I)$. We assume $p(y)$ is uniform.

(b) GaussX: as for GaussI, but the covariance matrices are unconstrained, i.e., $p(\mathbf{x} \mid y = c) = \mathcal{N}(\mathbf{x} \mid \mu_c, \Sigma_c)$.

(c) LinLog: A logistic regression model with linear features.

(d) QuadLog: A logistic regression model, using linear and quadratic features (i.e., polynomial basis function expansion of degree 2).

After training we compute the performance of each model M on the training set as follows:

$$L(M) = \frac{1}{n} \sum_{i=1}^{n} \log p(y^{(i)} \mid \mathbf{x}^{(i)}, \hat{\theta}, M)$$

Note that this is the conditional log-likelihood $p(y \mid \mathbf{x}, \hat{\theta}, M)$ and not joint log-likelihood $p(y, \mathbf{x} \mid \hat{\theta}, M)$. We now want to compare the performance of each model. We will write $L(M) \leqslant L(M')$ if model M must have lower or equal log likelihood on the training set than M', for any training set (in other words, M is worse than M', at least as far as training set logprob is concerned). For each of the following model pairs, state whether $L(M) \leqslant L(M'), L(M) \geqslant L(M')$, or whether no such statement can be made (i.e., M might sometimes be better than M' and sometimes worse); also, for each question, briefly (1-2 sentences) explain why.

(a) (3 points) GaussI, LinLog

(b) (3 points) GaussX, QuadLog

(c) (3 points) LinLog, QuadLog

(d) (3 points) GaussI, QuadLog

(e) (3 points) Now suppose we measure performance in terms of the average misclassification rate on the training set:

$$R(M) = \frac{1}{n} \sum_{i=1}^{n} I(y^{(i)} \neq \hat{\mathbf{y}}(\mathbf{x}^{(i)}))$$

where $\hat{y}(x^{(i)})$ is the predicted y for $x^{(i)}$. Is it true in general that $L(M) > L(M')$ implies that $R(M) < R(M')$? Explain why or why not.

6 Naive Bayes Classification [15 points]

Consider a K-class classification problem where the input vector $x = (x_1, x_2, \ldots, x_M)$ consists of M binary components, where each $x_i \in \{0, 1\}$, and the output y is a one-hot vector of length K, encoding the class label.

Consider the following generative model:

- The probability of y is given by $p(y = k)$ for each class k, and $\sum_{k=1}^{K} p(y = k) = 1$.

- The features x_i are conditionally independent given the class y, so the joint probability can be factorized as:

$$p(x, y) = p(y) \prod_{i=1}^{M} p(x_i \mid y)$$

This represents an example of the naive Bayes model. Assuming you have access to labeled training data, we can use Maximum Likelihood Estimation (MLE) to estimate the parameters:

- The prior probability estimate is $\hat{p}(y = k) = \frac{N_k}{N}$, where N_k is the number of training examples that belong to class k and N is the total number of training examples.
- The MLE estimate for the conditional probability is $\hat{p}(x_i = 1 \mid y = k) = \frac{N_{ki}}{N_k}$, where N_{ki} is the number of examples in class k where $x_i = 1$, and N_k is the number of examples in class k.

(a) (5 points) Suppose we are given a new binary feature vector $x = (x_1, x_2, \ldots, x_M)$, and we want to predict the class label. The prediction is made using the posterior probability $p(y = k \mid x)$. Show that the posterior probability can be written in the form:

$$p(y = k \mid x) = \frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)}$$

where

$$a_k = \log p(x, y = k)$$

(b) (10 points) Show that a_k is a linear function of the components of x.

7 Optimizing logistic regression [15 points]

6. In this problem, suppose you were hypothetically provided with a classification dataset `dataset.csv` for binary classification. Suppose the file `dataset.csv` contains a matrix of size 1672×65, where the last column of the matrix is the class label. Thus, the data matrix X is of dimension $n \times d$, where $n = 1672$ and $d = 64$ (*Remark*: The digits are 8×8 pixel images that have been turned into vectors of length 64), and the label vector $y \in \{0, 1\}^n$ is the last (hence, 65^{th}) column.

We will use this data with regularized logistic regression that has the objective function:

$$R(w) := \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell_i(w)}_{\ell(w)} + \underbrace{\frac{\lambda}{2} \|w\|_2^2}_{r(w)}, \tag{1}$$

where $\ell_i(w) := \log(1 + e^{-y_i w^\top x^i})$ and $r(w)$ is the regularizer. **Note:** (1) expects $y_i \in \{-1, 1\}$. Moreover, we have designed the dataset to be linearly separable, so actually a loss value close to 0 should be achievable empirically (which corresponds to 100% training accuracy) when $\lambda = 0$ is used.

In this question, we will recall gradient descent (GD) and stochastic gradient descent (SGD) for optimizing the regularized loss function (1) with the choice $\lambda > 0$.

(a) (5 points) Provide an expression for the gradient of the objective function, i.e., an expression for $\nabla R(w)$.

(b) (5 points) In SGD, we use a stochastic gradient $g(w)$ instead of $\nabla R(w)$. We compute $g(w)$ by selecting a batch of b data points $\{x^{i_1}, \ldots, x^{i_b}\}$, where each i_j is sampled uniformly with replacement from $\{1, 2, \ldots, n\}$, and each x^{i_j} is the corresponding row from the data matrix X. Provide an expression for the resulting (mini-batch) stochastic gradient .

(c) (5 points) Show that the gradient you derived above is unbiased, i.e., $\mathbb{E}[g(w)] = \nabla R(w)$, where the expectation is computed over the randomness of the batch.

8 Softmax [20 points]

An approach to multi-class classification is to use a generalized version of the logistic model. Let $x = [x_1, x_2, \ldots, x_d]$ be an input vector, and suppose we would like to classify into k classes; that is, the output y can take a value in $1, \ldots, k$. The softmax generalization of the logistic model uses $k(d+1)$ parameters $\theta = (\theta_{ij}), i = 1, \ldots, k, j = 0, \ldots, d$, which define the following k intermediate values:

$$\begin{aligned} z_1 &= \theta_{10} + \sum_j \theta_{1j} x_j \\ &\ldots \\ z_i &= \theta_{i0} + \sum_j \theta_{ij} x_j \\ &\ldots \\ z_k &= \theta_{k0} + \sum_j \theta_{kj} x_j \end{aligned}$$

The classification probabilities under the softmax model are:

$$\Pr(y = i \mid x; \theta) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad .$$

(a) (5 points) Show that when $k = 2$ the softmax model reduces to the logistic regression model. That is, show how both give rise to the same classification probabilities $\Pr(y \mid x)$. Do this by constructing an explicit transformation between the parameters: for any given set of $2(d+1)$ softmax parameters, show an equivalent set of $(d+1)$ logistic parameters.

> Hint: you can write the posterior of the logistic model given its parameter θ', and find the relationship between θ' and θ

(b) (5 points) What type of decision boundary is possible for a softmax model? (e.g. linear, quadratic etc.)

(c) (10 points) A stochastic gradient ascent learning rule for softmax is given by:

$$\theta_{ij} \leftarrow \theta_{ij} + \alpha \sum_t \frac{\partial}{\partial \theta_{ij}} \log \Pr(y^t \mid x^t; \theta) \ \ ,$$

where (x^t, y^t) are the training examples. We would like to rewrite this rule as a delta rule. In a delta rule the update is specified as a function of the difference between the target and the prediction. In our case, our target for each example will actually be a vector $y^t = (y^t_1, \ldots, y^t_k)$ where $y^t_i = 1$ if $y^t = i$ and 0 otherwise.
Our prediction will be a corresponding vector of probabilities:

$$\hat{y}^t = (\Pr(y = 1 \mid x^t; \theta), \ldots, \Pr(y = k \mid x^t; \theta))$$

Calculate the derivative above (i.e. $\frac{\partial}{\partial \theta_{ij}} \log \Pr(y^t \mid x^t; \theta)$) and rewrite the update rule as a function of $y - \hat{y}$

Hint: it might be helpful to calculate $\frac{\partial z_i}{\partial \theta ij}$

Hint: it might be helpful to consider two cases: $y = i$ and $y \neq i$