EECS 553 HW 8

Due Friday, Nov. 8, by 11:59 PM Eastern Time

1. Logistic Regression with Label Proportion Shift

Suppose we obtain i.i.d training data $\{(\mathbf{x}_j, y_j)\}_{j=1}^n$ for binary classification (labels $y_j \in \{0, 1\}$), which we use to fit a logistic regression classifier. We ordinarily assume the training data is sampled from the same distribution as the test data to which the classifier is applied. However, in practice there may be a *distributional shift* from training time to the time of model deployment. This is a major concern in the application of ML models, since predicted performance may be overly optimistic when applied to such shifted test data. In this problem you will apply the EM algorithm to adapt a trained logistic classifier to a shift in the label proportions after training.

With Y_j denoting the random variable designating the label of the j-th training feature instance $\mathbf{x}_j$, define the *training prior*: $\pi = P(Y_j = 1)$. Likewise with Y_m denoting the (unobserved) label of an instance $\mathbf{x}_m$ in the test set, assume that the label prior has shifted to $\pi' = P(Y_m = 1)$. We assume that the class-conditional feature distributions $\phi_k(\mathbf{x}) = p(\mathbf{x}|Y = k)$, $k = 0, 1$, are unaffected by the shift. The posterior label distribution $p(y|\mathbf{x}; \pi) = \Pr(Y = y|\mathbf{x}; \pi)$ and feature marginals $p(\mathbf{x}; \pi) = \phi_1(\mathbf{x})\pi + \phi_0(\mathbf{x})(1 - \pi)$ are affected by the shift, and these respectively shift to $p(y|\mathbf{x}; \pi')$ and $p(\mathbf{x}; \pi')$ after the shift. For convenience we define the shorthand $p(y|\mathbf{x}) = p(y|\mathbf{x}; \pi)$, $p'(y|\mathbf{x}) = p(y|\mathbf{x}; \pi')$, $p(\mathbf{x}) = p(\mathbf{x}; \pi)$, $p'(\mathbf{x}) = p(\mathbf{x}; \pi')$. Assume that the logistic classifier has been trained on the training data producing the logistic output (posterior distribution) $p(y|\mathbf{x}) = e^{\mathbf{w}^T\mathbf{x}}/(1 + e^{\mathbf{w}^T\mathbf{x}})$, where $\mathbf{w}$ is the fitted logistic classifier weights.

a. Suppose we knew both the *training prior* π and the shifted prior π' governing the post-training (unobserved) labels. Define π_k, where $\pi_1 = \pi$ and $\pi_0 = 1 - \pi$, and similarly for π'_k. Show that the shifted posterior $p'(y|\mathbf{x})$ can be expressed in terms of π, π' and $p(y|\mathbf{x})$ as:

$$p'(y = k|\mathbf{x}) = \frac{\frac{\pi'_k}{\pi_k} p(y = k|\mathbf{x})}{\frac{\pi'_0}{\pi_0} p(y = 0|\mathbf{x}) + \frac{\pi'_1}{\pi_1} p(y = 1|\mathbf{x})}, \quad k = 0, 1 \tag{1}$$

Hint: First, use the fact that $\phi_k(\mathbf{x})$ is unaffected by the shift and Bayes' rule to show

$$p'(y = k|\mathbf{x}) = f(\boldsymbol{x})\frac{\pi'_k}{\pi_k} p(y = k|\mathbf{x})$$

where $f(\mathbf{x}) = p(\mathbf{x})/p'(\mathbf{x})$. Next, use the fact that $\sum_{k=0}^1 p'(y = k|\mathbf{x}) = 1$ to get a formula for $f(\mathbf{x})$ in terms of the other quantities.

Note that if we knew π and π', this update tells us how to change the posterior $p(y|\mathbf{x})$ of our classifier due to the shift. The prior π is easily estimated from training data as the MLE $\frac{1}{n}\sum_{j=1}^n y_j$, and we will henceforth assume that π is this estimate. In the following parts, we'll derive an EM algorithm for estimating the shifted prior π' from test data by treating the test labels as hidden data. In the end, we'll use our estimate in place of the true prior π' in equation (1) to update the classifier's predictions.

b. Let the unlabeled test data consist of m i.i.d. samples $\mathbb{X} = \{\mathbf{x}_i\}_{i=1}^m$, which are acquired *after* the label proportion shift $\pi \rightarrow \pi'$. Let the hidden variables be the set of true test labels $\mathbb{Z} = \{y_i\}_{i=1}^m$. Write down an explicit form for the incomplete data log likelihood function $l(\pi') = \log p(\mathbb{X}; \pi')$ for π' and an explicit form for the complete data log likelihood function $\log p(\mathbb{X}, \mathbb{Z}; \pi')$.

c. Derive the E-step of the EM algorithm, i.e., determine $Q(\pi'; \pi'^{(t)}) = \mathbb{E}[\log p(\mathbb{X}, \mathbb{Z}; \pi')|\mathbb{X}, \pi'^{(t)}]$ where $\pi'^{(t)}$ is the EM estimate of π' at iteration t. Now derive the M-step to update $\pi'^{(t)} \rightarrow \pi'^{(t+1)}$. This requires solving the optimization $\pi'^{(t+1)} = \arg\max_{\pi'} \quad Q(\pi', \pi'^{(t)})$. Show that the final M-step update reduces to

$$\pi'^{(t+1)} = \frac{1}{m} \sum_{i=1}^{m} p'^{(t)}(y = 1|\mathbf{x}_i)$$

where $p'^{(t)}(y = 1|\mathbf{x}_i)$ is obtained from (1) after replacing π'_k with $\pi_k'^{(t)}$ and $\mathbf{x}$ with $\mathbf{x}_i$.

d. Implement the adjusted LR for label shift on Fashion MNIST. Skeleton code is provided to load the data and fit the initial logistic classifier to the training data. Implement the M-step update for $\pi'^{(t+1)}$ using the result of parts (a) and (e), which will adjust the initial logistic regression logistic mapping $p(y|\mathbf{x})$ to the mapping $p'(y|\mathbf{x})$ that accounts for the shifted proportions.

Report the final test accuracy of **(i) the adjusted LR using** $p'(y|\mathbf{x})$ and **(ii) the unadjusted LR using** $p(y|\mathbf{x})$ (you will need to use the true labels of the test data to compute the accuracy).

Now use the true test labels to derive a maximum likelihood estimate of π' and adjust the original classifier according to (1) in part a. **Compare the accuracy of this "clairvoyant" logistic classifier to the performance of the classifier you implement with EM**.

Additional submission instructions:

1. Attach your codes for respective parts to your write-up (copy-and-paste or screenshot) and submit it to Gradescope.

2. Submit your code as a zip file to Canvas. Only zip the following file: `hw6_p4.py`.

2. Linear Regression with Laplacian Likelihood (5 points)

Consider linear regression in the nonBayesian setting. Assume that the likelihood is Laplacian, that is,

$$y_i = \boldsymbol{w}^T \boldsymbol{x}_i + \epsilon_i,$$

where ϵ_i are iid and Laplacian distributed. Show that maximum likelihood estimation of $\boldsymbol{w}$ is equivalent empirical risk minimization with a certain loss.

3. Bayesian Optimization (5 points each)

In this problem, you will explore the use of GPs for finding the maximum value of a black box function (meaning we can only evaluate it, we can't find numerical gradients) that is expensive to evaluate. A prime example is the cross-validation accuracy estimate when doing model selection for some machine learning algorithm. In this problem, to keep things simple, we will take our function to be maximized to be

$$f(x) = \sin(2\pi x) + \sin(11\pi x), x \in [0, 1].$$

Suppose we know the values of this function at 0.2, 0.35, 0.5, 0.6, 0.75. In this problem you will apply Gaussian Processes to optimize this function, viewing it as a black box function. Use a Gaussian kernel with parameters equal to the default parameter values $\sigma_e = 10^{-10}$, $\sigma_k = 0.9$, $\gamma = 1/18$ (See the lecture slides on GP regression for the definition of the Gaussian kernel in terms of these parameters.)

(a) Turn in a single plot that shows the true function, the posterior mean function, the posterior uncertainty envelopes $\mu(x) \pm 2\sigma(x)$ (95% confidence bands), and the five points at which the function value is known. Include a legend. Here $\mu(x)$ and $\sigma^2(x)$ are the mean and variance of $f(x)$, conditioned on the observed function values so far.

(b) The Upper Confidence Bound approach is to select the point x such that $\mu(x) + c\sigma(x)$ is maximal, where c is a tuning parameter. Using the UCB approach, determine the next value of x at which to evaluate the function. Use $c = 2$.

(c) The Probability of Improvement approach selects the point x such that the probability that $f(x)$ exceeds $f(x^*)$, where x^* is the point with highest function value so far, is maximal. Since the distribution of $f(x)$ is $\mathcal{N}(\mu(x), \sigma^2(x))$, you can calculate the PI in terms of the CDF of this Gaussian distribution. Using the PI approach, determine the next value of x at which to evaluate the function.

(d) The Expected Improvement approach is similar to PI, except it selects x to maximize $\mathbb{E}[\max(0, f(x) - f(x^*)]$. Once again you'll need to determine a formula which may be expressed in terms of a Gaussian CDF and/or pdf. Using the EI approach, determine the next value of x at which to evaluate the function.

(e) Now run each of the three approaches to convergence and report the estimated maximum function value obtained by each approach.

Additional submission instructions:

1. Attach your codes for respective parts to your write-up (copy-and-paste or screen-shot) and submit it to Gradescope.
2. Submit your code as a zip file to Canvas. Only zip the following file: hw8 p2.py.