Homework #5, EECS 553, F26. Due **Fri. Oct. 23**, by 11:00PM

Problems

1. [6] **Imbalanced data**

As mentioned in class, the risk associated with the 0-1 loss function can look unrealistically favorable when there is class imbalance. An alternate approach is to design (and assess) a classifier using the **balanced accuracy**, or equivalently by minimizing the **balanced error**:

$$\mathrm{BE}(\boldsymbol{\theta}) = \frac{1}{K} \sum_{k=1}^{K} \mathsf{P}\{f(\boldsymbol{x}; \boldsymbol{\theta}) \neq k \mid \mathsf{Y} = k\} .$$

(a) [3] Define a loss function $L(y, \hat{y})$ for classification for which the corresponding risk is the balanced error shown above. You may assume $\pi_k \neq 0$ for all k.

(b) [3] Describe how to train a multinomial logistic regression method in a way that is more compatible with balanced error.

2. [16] **Logistic regression for handwritten digit recognition**

The logistic regression demo shown in class used PCA features and a quasi-Newton method from an optimization library. In this problem, you will apply logistic regression for binary classification directly to the original pixel-space images using Newton's method as the optimizer.

The first step is to download the above demo code (or the corresponding python template) and change it to use digits 4 and 9, which are more challenging than digits 1 and 7. (The classification accuracy using $K = 2$ PCA components is very poor.)

(a) [10] Use a Gen AI tool of your choice to implement Newton's method and then debug/refine/rework that code until you believe you have a correct implementation of Newton's method for seeking a minimizer of the regularized negative log-likelihood $J(\boldsymbol{\theta})$ derived in a previous problem. (HW4#3)

- Let the initial iterate $\boldsymbol{\theta}_0$ be the zero vector by default.
- As a termination criterion, return $\boldsymbol{\theta}_{t+1}$ as soon as $|J(\boldsymbol{\theta}_{t+1}) - J(\boldsymbol{\theta}_t)|/J(\boldsymbol{\theta}_t) \leq \epsilon \triangleq 10^{-6}$.
- Even though you are allowed to use GenAI here, ultimately you must understand this Newton method.

Your submitted file should be named `logreg_newton.jl` or `logreg_newton.py` and should contain the following function:

```
"""
    θ = logreq_newton(Xp, Xm, λ; maxiter, tol)

Logistic regression binary classifier design using the ERM cost function
``J(θ) = (1/n) \\sum_{i=1}^n φ(y_i ⟨θ, [1;x_i]⟩) + (λ/2) ‖ θ ‖_2^2``
where `θ = [b; w]`
and ``φ(t) = log(1 + exp(-t))``.

Runs Newton's method until `maxiter` iterations are reached
or until the relative cost function change is ≤ tol.

In:
- `Xp` `d x n1` matrix of training data for class +1
- `Xm` `d x n2` matrix of training data for class -1
- `lambda` regularization parameter λ ≥ 0

Option:
- `maxiter` maximum number of iterations; default 1000
- `tol` cost function tolerance; default 1e-6
- `θ0` initial weight vector; default zeros(1+size(Xp,1))

Out:
- `θ` length-`d+1` vector of weights such that `sign(⟨θ,[1;x]⟩)` is the classifier
"""
```

```
function logreg_newton(
    Xp::AbstractMatrix,
    Xm::AbstractMatrix,
    lambda::Real;
    maxiter::Int = 1000,
    tol::Real = 1e-6,
    θ0::AbstractVector = zeros(1+size(Xp,1)),
)
```

Submit your solution to autograder.io
Python version:

```
from typing import Optional
def logreg_newton(
    Xp: np.ndarray,
    Xm: np.ndarray,
    lambda_val: float,
    maxiter: int = 1000,
    tol: float = 1e-6,
    theta0: Optional[np.ndarray] = None,
) -> np.ndarray:
```

(b) [3] Append code to the demo that calls your Newton method using the original (pixel-space) MNIST images in the variable `dtrain`, rather than the PCA components in `Xtrain`.
- Set $\lambda = 10^{-3}$.
- After training your classifier, apply it to the test data in `dtest1`, and show histograms of the discriminant values like the in the demo.
- Report the test error and the number of iterations run,

(c) [3] Submit to gradescope a screenshot of the code you appended to the demo. Your code is expected to follow best coding practices, including comments, indentation, and descriptive variable names.

---

3. [12] **Constrained optimization**

Given a vector $\boldsymbol{x} \in \mathbb{R}^n$ such that $x_1 > x_2 \geq x_3 \cdots \geq x_{n-1} > x_n$, in this problem you will solve the following optimization problem using the KKT conditions:

$$\begin{aligned} \underset{\boldsymbol{p} \in \mathbb{R}^n}{\arg\min} \quad & \left(\sum_{i=1}^n x_i p_i \right)^2 - \sum_{i=1}^n x_i^2 p_i \\ \text{s.t.} \quad & \sum_{i=1}^n p_i = 1 \\ & p_i \geq 0, \quad \forall i = 1, \cdots n. \end{aligned} \tag{1}$$

These constraints are often called the **probability simplex**, and this problem has the following probabilistic interpretation. Given a **categorical distribution** that takes values in $\{x_1, \cdots, x_n\}$ whose probabilities are given by $p_1, \cdots, p_n$ respectively, then the cost function above computes the negative of variance of the categorical random variable, *i.e.*, $-\mathsf{Var}\{\mathsf{X}\} = \mathsf{E}^2[\mathsf{X}] - \mathsf{E}[\mathsf{X}^2]$. Hence the problem is equivalent to finding the probability distribution that maximizes the variance.

(a) [3] Explain why, to solve this problem, it suffices to find a point satisfying the KKT conditions.

(b) [3] For parts (b) and (c), we solve this particular case:

$$\begin{aligned} \underset{\boldsymbol{p} \in \mathbb{R}^2}{\arg\min} \quad & (2p_1 + 3p_2)^2 - 4p_1 - 9p_2 \\ \text{s.t.} \quad & p_1 + p_2 = 1 \\ & p_1 \geq 0 \\ & p_2 \geq 0 \end{aligned} \tag{2}$$

Write down the Lagrangian and KKT conditions of (2).

(c) [3] Solve the primal problem (2) using the KKT conditions.

Hint. For each inequality constraint, there are two cases: either the constraint is satisfied with equality, or the corresponding dual variables must be zero (complimentary slackness). Because there are two inequality constraints, consider all four cases.

(d) [0] (Ungraded: do not submit) Determine the Lagrangian dual function and write down the dual optimization problem. Then solve the original problem by first solving the dual problem and then inferring the primal solution from the dual solution.

(e) [3] Now consider the general problem (1). Find a solution to the problem using the KKT conditions.
Hint. Based on your answer in part (c) and the probabilistic interpretation, you may "guess" the solution first, and show it's indeed the solution by verifying that the KKT conditions hold. Though true, you are not required to show all the other cases violate one of the KKT conditions.

---

4. [6] **Optimal soft-margin hyperplane**

Let $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ denote the solution to the **soft-margin hyperplane quadratic program** defined as

$$\begin{aligned} \underset{\boldsymbol{w},\, b,\, \boldsymbol{\xi}}{\arg\min}\ & \frac{1}{2}\left\|\boldsymbol{w}\right\|_2^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \\ \text{s.t. } & y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1 - \xi_i, \quad \forall i, \\ & \xi_i \geq 0, \quad \forall i. \end{aligned}$$

(a) [3] Show that if training point $\boldsymbol{x}_i$ is misclassified by the optimal soft-margin hyperplane classifier, then $\xi_i^* \geq 1$. Conclude that $\frac{1}{n}\sum_i \xi_i^*$ upper bounds the training error. Hence, the OSM objective balances margin maximization with minimizing a bound on the training error.
(The training error is defined as the number of $\boldsymbol{x}_i$ misclassified$/n$.)

(b) [3] Show that if $\xi_i^* > 0$, then ξ_i^* is proportional to the distance from $\boldsymbol{x}_i$ to the margin hyperplane associated with class y_i, that is, the set $\left\{\boldsymbol{x} \,:\, (\boldsymbol{w}^*)^\top \boldsymbol{x} + b^* = y_i\right\}$, and give the constant of proportionality.

(c) [0] (Ungraded: do not submit.) Consider the optimization problem

$$\begin{aligned} \underset{\boldsymbol{w},\, b,\, \boldsymbol{\xi}}{\arg\min}\ & \frac{1}{2}\left\|\boldsymbol{w}\right\|_2^2 + \frac{D}{n}\sum_{i=1}^{n} \xi_i \\ \text{s.t. } & y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq \kappa - \xi_i, \quad \forall i = 1, \ldots, n \\ & \xi_i \geq 0, \quad \forall i = 1, \ldots, n, \end{aligned}$$

where $D > 0$, $\kappa > 0$. Show that the above quadratic program yields the same classifier as the optimal soft-margin classifier for a certain parameter C, and express this C in terms of D and κ.

---

5. [15] **Kernel logistic regression**

For parts **(a)-(d)** of this problem, you may use a GenAI tool of your choice to help you. It is up to you to decide how you will use the GenAI tool, but keep in mind that you will not have access to such tools on the exams. Save the chat transcript(s), as you will need to submit them as part of **(e)**.

This problem asks you to kernelize the logistic regression method by kernelizing the Newton-Raphson algorithm for logistic regression. Because kernelizing corresponds to a high dimensional feature space, we will consider the regularized form of logistic regression.

To simplify the problem somewhat, we absorb the offset term into the first element of $\boldsymbol{x}$, so that the class probability model is

$$\eta(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^\top \boldsymbol{x})} = \sigma(\boldsymbol{\theta}^\top \boldsymbol{x}), \quad \sigma(t) = \frac{1}{1 + \mathrm{e}^{-t}}.$$

Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ be training data for binary classification, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. Consider the cost function

$$J(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n} \left[y_i \log \sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i) + (1 - y_i) \log\left(1 - \sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i)\right)\right] + \frac{\lambda}{2}\left\|\boldsymbol{\theta}\right\|_2^2,$$

where $\lambda > 0$ is the regularization parameter.

We are free to initialize iterative algorithms any way we choose. Let us initialize $\boldsymbol{\theta}_0$ so that $\boldsymbol{\theta}_0 = \boldsymbol{X}^\top \boldsymbol{\alpha}_0$ for some vector $\boldsymbol{\alpha}_0 \in \mathbb{R}^n$. In other words, $\boldsymbol{\theta}_0$ is a linear combination of data points. For example, we could initialize $\boldsymbol{\theta}_0$ to be the first data point $\boldsymbol{x}_1$, in which case $\boldsymbol{\alpha}_0 = [1\ 0\ 0\ \cdots\ 0]^\top$.

(a) [3] Verify that the Newton-Raphson update has the following form:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + (\boldsymbol{X}^\top \boldsymbol{W}_t \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} (\boldsymbol{X}^\top \boldsymbol{v}_t - \lambda \boldsymbol{\theta}_t),$$

where $\boldsymbol{X}$ is an $n \times d$ matrix whose ith row is $\boldsymbol{x}_i^\top$, $\boldsymbol{W}_t$ is a diagonal matrix with ith diagonal entry equal to $\frac{1}{n}\sigma(\boldsymbol{\theta}_t^\top \boldsymbol{x}_i)(1 - \sigma(\boldsymbol{\theta}_t^\top \boldsymbol{x}_i))$, and $\boldsymbol{v}_t \in \mathbb{R}^n$ has ith entry given by $\frac{1}{n}\left(y_i - \sigma(\boldsymbol{\theta}_t^\top \boldsymbol{x}_i)\right)$.

(b) [3] The **matrix inversion lemma** (aka Woodbury identity) states that

$$(\boldsymbol{P} + \boldsymbol{Q}\boldsymbol{R}\boldsymbol{S})^{-1} = \boldsymbol{P}^{-1} - \boldsymbol{P}^{-1}\boldsymbol{Q}(\boldsymbol{R}^{-1} + \boldsymbol{S}\boldsymbol{P}^{-1}\boldsymbol{Q})^{-1}\boldsymbol{S}\boldsymbol{P}^{-1},$$

provided all of the matrix operations are well defined. Rewrite the logistic regression iterative update step by applying the matrix inversion lemma to $(\boldsymbol{X}^\top \boldsymbol{W}_t \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}$.

(c) [3] Consider an arbitrary step in the iterative procedure. Show that if $\boldsymbol{\theta}_0 = \boldsymbol{X}^\top \boldsymbol{\alpha}_0$ for some $\boldsymbol{\alpha}_0 \in \mathbb{R}^n$, then $\boldsymbol{\theta}_{t+1} = \boldsymbol{X}^\top \boldsymbol{\alpha}_{t+1}$ for some $\boldsymbol{\alpha}_{t+1}$. Give an update formula for $\boldsymbol{\alpha}_{t+1}$ in terms of $\boldsymbol{\alpha}_t$ and the other quantities.

(d) [3] Explain how to implement kernel logistic regression. That is, show how to incorporate inner product kernels into the iterative parameter update, and also how to evaluate the final classifier.

(e) [3] If you did not use GenAI on the above parts (a)-(d), choose a subproblem now to enter into a GenAI tool of your choice and get its answer. In a few sentences, describe how the GenAI solution compares to yours. Did GenAI get the same answer as you, in a similar way? Is there anything in the GenAI solution that gives you more insight? Did you gain anything by waiting until the end to look at a GenAI solution? Submit your chat transcript(s) to gradescope.
If you used GenAI already, in a few sentences, describe how you used GenAI to help you solve this problem. Did you just give it the problem and copy the answers? Did you try the problem yourself and use it when you got stuck? Did you use it to help get started and finish the problem yourself? Something else? Write a few more sentences describing whether you feel using a GenAI tool made solving this problem easier or more difficult. Do you feel like you understand the problem and your solution? If this same problem appeared on an exam, do you think you would be able to solve it without GenAI? Submit your chat transcript(s) on gradescope.

---

6. [3] Mid-semester feedback

Please complete the mid-semester course evaluation for EECS 553 before the deadline given by UM (Oct. 22 in F26). When you complete the evaluation, the system will give you a confirmation page that shows your name. Submit a screenshot of that confirmation (with your name) to gradescope to earn credit for this problem. Your evaluation is very important to us for improving the course. Your evaluation is entirely anonymous.

Please share your views on course AI policies, including ideas for how to make HW problems with AI components most helpful.

---

7. [5] Make your own problem

Make a clearly stated problem that could be used on an EECS 553 exam based on the topics covered by the due date of this HW set, and provide your own solution to the problem.

Follow all the instructions from HW2 except create your problem in `hw5.tex` using the Overleaf link on EECS 553 Canvas home page.