Boston University
Department of Electrical and Computer Engineering

ENG EC 414 (Ishwar) Introduction to Machine Learning

HW 6

Issued: Fri 25 Mar 2022 **Due:** 10:55pm Boston time Fri 1 Apr 2022 in Gradescope + Blackboard.
Required reading: Slides on Logistic-Regression & Convex Analysis + your lecture notes & Discussion 8.

Important: Before you proceed, please read the documents pertaining to *Homework formatting and submission guidelines* and the *HW-grading policies* in the Homeworks section of Blackboard, especially guidelines for submitting reports in Gradescope. and code in Blackboard. In particular, for computer assignments *you are prohibited from using any online code or built-in MATLAB functions except as indicated in the problem or skeleton code (when provided).*
In order to receive full credit, all work should be supported by a concise explanation that is clear, relevant, specific, logical, and correct. In particular, for each part, you must clearly outline the key steps and provide proper justification for your calculations.

Note: Problem difficulty = number of coffee cups ☕

Problem 6.1 [30pts, 5 for each part] *(Convex Functions Analytical Exercises)* Which of the following functions are convex, strictly convex, or non-convex on the domains indicated. Proper reasoning is needed to receive full credit.

(a) $f(t) = |t|,\ t \in \mathbb{R}$

(b) $f(t) = t^2,\ t \in \mathbb{R}$

(c) $f(\mathbf{w}) = (\mathbf{a}^\top \mathbf{w} + b)^2,\ \mathbf{w} \in \mathbb{R}^2, \mathbf{a} = (1,1)^\top$

(d) $f(t) = e^{-t^2},\ t \in \mathbb{R}$

(e) ☕ $f(\mathbf{w}) = \log(1 + e^{\mathbf{a}^\top \mathbf{w}}),\ \mathbf{w} \in \mathbb{R}^2, \mathbf{a} = (1,1)^\top$

(f) $f(\mathbf{w}) = \max(0, 1 - \mathbf{a}^\top \mathbf{w}),\ \mathbf{w} \in \mathbb{R}^2, \mathbf{a} = (1,1)^\top$

Problem 6.2 [6pts] *(Logistical Regression Analytical Exercise)* Let

$$\mathcal{D} = \Big\{(x_1 = -3, y_1 = -1), (x_2 = 5, y_2 = +1)\Big\}$$

be a set of pairs of scalar features and their labels for training a binary logistic-loss based classifier with class labels $-1, +1$ and a simplified form of the logistic-loss in which the w and b parameters for class -1 and the b parameter for class $+1$ are all set to zero and only class $+1$ has a free scalar parameter w which needs to be learned from training data. Specifically, let

$$p(y = +1|x, w) = \frac{e^{wx}}{1 + e^{wx}}, \quad p(y = -1|x, w) = \frac{1}{1 + e^{wx}}$$

(a) [2pts] Let $\text{NLL}(w) = -\ln\Big(\prod_{j=1}^{2} p(y_j|x_j, w)\Big)$ be the negative log-likelihood. Compute $\frac{\partial \text{NLL}}{\partial w}(w)$.

(b) [2pts] With initialization $\hat{w}_0 = 0$, compute $\hat{w}_1$, the value of w after one iteration of the gradient descent algorithm for minimizing NLL. Express $\hat{w}_1$ as a function of the step-size $s > 0$.

(c) [2pts] Compute $\hat{w}_{\text{logistic}}$, the maximum-likelihood estimate of w based on $\mathcal{D}$. Connect and discuss what you find with what you learned in the class about the need for regularization.

Problem 6.3 [64pts] *(ℓ_2-regularized multi-class logistic-loss based classifier)*
In this problem we will implement the Stochastic Gradient Descent (SGD) Algorithm to learn the parameters of an ℓ_2-regularized multi-class logistic-loss based classifier. We will use the *Iris* dataset consisting of feature vectors in $\mathbb{R}^4$, capturing various geometrical properties of flowers from the Iris genus, each of which can be in one of 3 classes corresponding to 3 varieties of the Iris plant. The dataset has already been processed and split into a fixed training set and a fixed test set both of which are contained within the provided "dot-mat" file that can be directly loaded into Matlab.
Summary of cost function and algorithm: Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be a training set of n examples consisting of feature vector $\mathbf{x}_j \in \mathbb{R}^d$ and label $y_j \in \{1, \ldots, m\}$ for each j. Let $\mathbf{x}_j^{\text{ext}} = \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix}_{(d+1)\times 1}$ denote the extended representation of the j-th feature vector and let $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m]$ the $(d+1) \times m$ matrix of parameters of the logistic-loss where $\boldsymbol{\theta}_\ell \in \mathbb{R}^{(d+1)}$ for each ℓ. The ℓ_2-regularized logistic loss function is given by:

$$g(\boldsymbol{\Theta}) = f_0(\boldsymbol{\Theta}) + \sum_{j=1}^{n} f_j(\boldsymbol{\Theta})$$

where,

$$f_0(\boldsymbol{\Theta}) = \lambda \left(\sum_{\ell=1}^{m} \|\boldsymbol{\theta}_\ell\|^2 \right), \quad \text{and} \quad f_j(\boldsymbol{\Theta}) = \left[\ln \left(\sum_{\ell=1}^{m} e^{\boldsymbol{\theta}_\ell^\top \mathbf{x}_j^{\text{ext}}} \right) - \sum_{\ell=1}^{m} 1(\ell = y_j)\, \boldsymbol{\theta}_\ell^\top \mathbf{x}_j^{\text{ext}} \right], \quad j = 1, \ldots, n,$$

and their gradients are given by

$$\nabla_{\boldsymbol{\theta}_k} f_0(\boldsymbol{\Theta}) = 2\lambda\, \boldsymbol{\theta}_k, \text{ and } \nabla_{\boldsymbol{\theta}_k} f_j(\boldsymbol{\Theta}) = \Big(p(k|\mathbf{x}_j, \boldsymbol{\Theta}) - 1(k = y_j) \Big)\, \mathbf{x}_j^{\text{ext}}, \quad k = 1, \ldots, m,$$

where

$$p(k|\mathbf{x}_j, \boldsymbol{\Theta}) = \frac{e^{\boldsymbol{\theta}_k^\top \mathbf{x}_j^{\text{ext}}}}{\sum_{\ell=1}^{m} e^{\boldsymbol{\theta}_\ell^\top \mathbf{x}_j^{\text{ext}}}}$$

The pseudocode for implemeting SGD for logistic loss is as follows:

Stochastic Gradient Descent for ℓ_2-regularized multi-class Logistic-Loss

input: Training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
initialize: $\boldsymbol{\Theta} = \mathbf{0}$
for $t = 1, 2, \ldots, t_{\max}$
  choose sample index: j uniformly at random from $\{1, \ldots, n\}$
  compute gradients:
  for $k = 1, \ldots, m$

$$p(k|\mathbf{x}_j, \boldsymbol{\Theta}) = \frac{e^{\boldsymbol{\theta}_k^\top \mathbf{x}_j^{\text{ext}}}}{\sum_{\ell=1}^m e^{\boldsymbol{\theta}_\ell^\top \mathbf{x}_j^{\text{ext}}}}$$

$$\mathbf{v}_k = 2\lambda \boldsymbol{\theta}_k + n\Big(p(k|\mathbf{x}_j, \boldsymbol{\Theta}) - 1(k = y_j)\Big)\ \mathbf{x}_j^{\text{ext}}$$

  end for
  update parameters:
  for $k = 1, \ldots, m$

$$\boldsymbol{\theta}_k \longleftarrow \boldsymbol{\theta}_k - s_t \mathbf{v}_k$$

  end for
end for
output: $\boldsymbol{\Theta}$

Use the following choices for the hyper parameters in all computer experiments for this problem:

$$t_{\max} = 6000, \quad s_t = \frac{0.01}{t}, \quad \lambda = 0.1$$

(a) [13pts] *Data analysis of full dataset (training + test)*: (i)[2pts] Plot the histogram of class labels. (ii)[5pts] Compute the matrix of empirical correlation coefficients for all pairs of features. Recall that the empirical correlation coefficient between a pair of features is equal to their empirical (cross) covariance divided by the square root of the product of their empirical variances. (iii)[6pts] Create 2D scatter plots of the dataset for *all distinct pairs* of features. Discuss your observations.

(b) [6pts] Plot the ℓ_2-regularized logistic loss normalized by the total number of training examples, i.e., $\frac{1}{n} g(\boldsymbol{\Theta})$, against iteration number t, for every 20 iterations, i.e., $t = 20t', t' = 1, 2, 3, \ldots, 300$. In order to avoid taking the logarithm of extremely small numbers, any value of $p(y_j|\mathbf{x}_j, \boldsymbol{\Theta})$ that is smaller than 10^{-10} should be treated as 10^{-10}. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.

(c) [6pts] Plot the CCR of the training set against iteration number, for every 20 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend. Recall that the decision rule in Logistic Regression is given by:

$$\hat{y}_j = \arg \max_{\ell=1,\ldots,m} \boldsymbol{\theta}_\ell^\top \mathbf{x}_j^{\text{ext}}$$

and the CCR by

$$\frac{1}{n} \sum_{j=1}^n 1(y_j = \hat{y}_j).$$

(d) [6pts] Plot the CCR of the *test* set against iteration number, for every 20 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.

(e) [6pts] Plot the so-called log-loss of the test set against iteration number, for every 20 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend. The log-loss of the test set is the NLL of the test set normalized by the number of test examples:

$$\text{logloss}(\boldsymbol{\Theta}) = -\frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} \ln p(y_j|\mathbf{x}_j, \boldsymbol{\Theta})$$

where $\mathbf{x}_j$ is a test feature vector and y_j is its ground truth label. In order to avoid taking the logarithm of extremely small numbers, any value of $p(y_j|\mathbf{x}_j, \boldsymbol{\Theta})$ that is smaller than 10^{-10} should be treated as 10^{-10}.

(f) [21pts] *Final values*: After the SGD algorithm terminates, report the final values of the following: (i)[9pts] $\boldsymbol{\Theta}$, (ii)[1pt] the training CCR, (iii)[1pt] the test CCR, (iv)[5pts] training confusion matrix, (v)[5pts] test confusion matrix. Discuss your observations.

(g) [6pts] Consider the six 2-dimensional subsets of 4-dimensional feature space in which two out of the four components are zero, e.g., $\{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 : x_3 = x_4 = 0\}$. In each of these six subsets, indicate the decision regions, i.e., which points will be classified as class 1, which as class 2, and which as class 3. Discuss your observations.

Code-submission via Blackboard: Create one dot-m file. Name it as follows: `<yourBUemailID>_hw6_3.m` for Problem 6.3. All local functions and scripts pertaining to one problem should appear within the single dot-m file for that problem. When run, your scripts should be able to display in the command window whatever you are asked to compute and report in each part. There is no skeleton code provided for this homework. Create your own code. Reach-out to the TA via Piazza and office/discussion hours for questions related to coding. When submitting code, please include a 'Readme' text file in your source code directory describing approximate running times of different parts, any additional comments (as needed) explaining how to use your code, and any dependencies between different parts. Please do not include into the directory, any data files that are already provided. Write your code under the assumption that all data files are in the same directory as your source code.

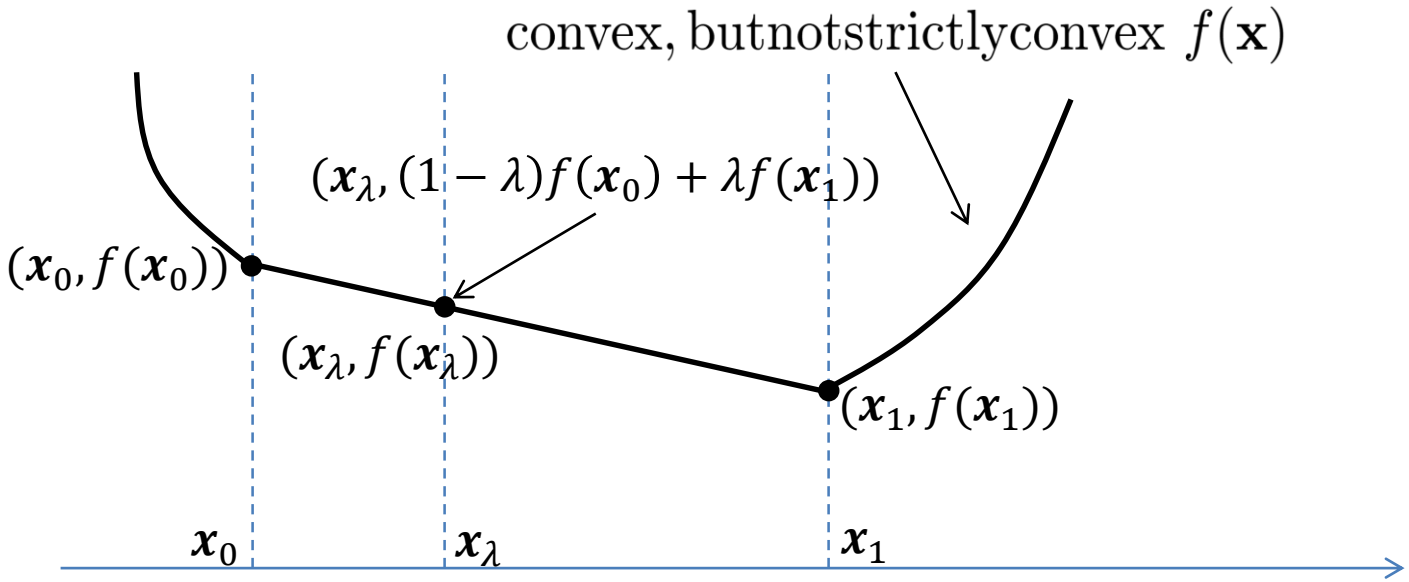
Convex function over a convex set (chord description)

- f is convex over $C \Leftrightarrow$ all chords joining any two points on the graph, never go below the graph:

$$\forall \boldsymbol{x}_0, \boldsymbol{x}_1 \in C, \forall \lambda \in [0,1],$$

$$f(\boldsymbol{x}_\lambda) = f\big((1-\lambda)\boldsymbol{x}_0 + \lambda \boldsymbol{x}_1\big) \leq (1-\lambda) f(\boldsymbol{x}_0) + \lambda f(\boldsymbol{x}_1)$$

- If equality only when $\boldsymbol{x}_0 = \boldsymbol{x}_1$ or $\lambda = 0,1$ then f is strictly convex (no planar segments in graph)



Operations that preserve convexity

- If f is convex, so is αf, for any $\alpha \geq 0$

- If $f_1, f_2, \ldots, f_k$ are each convex, then so is
$$f(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) + \cdots + f_k(\boldsymbol{x})$$
The sum of convex functions is convex

- If $f_1, f_2, \ldots, f_k$ are each convex, then so is
$$f(\boldsymbol{x}) = \max(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$$
The maximum of convex functions is convex

- If f is convex and g is non-decreasing and convex, then $h(\boldsymbol{x}) = g(f(\boldsymbol{x}))$ is convex: a nondecreasing convex function of a convex function is convex

- If $f(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^d$, is convex, so is: $h(\boldsymbol{z}) = f(A\boldsymbol{z} + \boldsymbol{b})$, for any $d \times k$ matrix A, $k \times 1$ vector variable $\boldsymbol{z}$, and $d \times 1$ constant vector $\boldsymbol{b}$: a convex function of an affine map is convex

Example

$f(\underline{x}) = \underline{a}^T\underline{x} + b$ → convex ✓ since f is an affine function of $\underline{x}$

$g(t) = e^t$ → convex because $\frac{d^2g(t)}{dt^2} = e^t > 0 \ \forall t$

e^t

of $\underline{x}$

✓ $g(t) = e^t$ → convex because $\frac{d^2g(t)}{dt^2} = e^t > 0 \ \forall t$

$\frac{dg(t)}{dt} = e^t > 0$

$\Rightarrow$ g is strictly increasing & g is convex

Explanation:

Choose $x_0 = 1$, $x_1 = 5$

$\lambda = \frac{1}{2}$

$g((1-\lambda)x_0 + \lambda x_1) = g\left(\frac{1+5}{2}\right) = g(3)$

$= |3|$

$= 3$

$(1-\lambda)g(x_0) + \lambda g(x_1) = \frac{1}{2}g(1) + \frac{1}{2}g(5)$

$= \frac{1}{2}|1| + \frac{1}{2}|5|$

$= \frac{1+5}{2}$

$= 3$

So $g(x_1) = (1-\lambda)g(x_0) + \lambda g(x_1)$

and $x_0 \neq x_1$ & $\lambda \neq 0$ & $\lambda \neq 1$

$\underline{a} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$(\underline{a}^T \underline{w} + b)^2 = f(\underline{w})$

$\underline{w}_0 \neq \underline{w}_1 \qquad \lambda \neq 0, \ \lambda \neq 1$

e.g. $\lambda = \frac{1}{2}$

$f(\underline{w}_\lambda) = \left(a^T \left(\frac{\underline{w}_0 + \underline{w}_1}{2}\right) + b\right)^2$

$\stackrel{?}{=} \frac{1}{2} f(\underline{w}_0) + \frac{1}{2} f(\underline{w}_1)$

$\frac{1}{2}\left(a^T w_0 + b\right)^2 + \frac{1}{2}\left(a^T w_1 + b\right)^2$

if $w_0 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $\underline{w}_1 = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$ ... $\underline{a}^T \underline{w}_0 = a^T \underline{w}_1 = 0$

$\underline{w}_0 \neq \underline{w}_1$ $\qquad \underline{w}_0, \underline{w}_1 \perp \underline{a}$

$\left(\frac{a^T w_0 + a^T w_1}{2} + b\right)^2 = (0 + b)^2 = b^2 = 1/2\, b^2 + \frac{1}{2} b^2$

$b^2 = b^2$

Strictly convex means a function is curved

||

vice versa

if a function has a flat line (linear flat face) it is no

$g(t) = 0 \quad \forall t$

$g((1-\lambda)x_0 + \lambda x_1) = 0$

$(1-\lambda)\, g(x_0) + \lambda g(x_1) = 0$

So $g((1-\lambda)x_0 + \lambda x_1) \leq (1-\lambda)\, g(x_0) + \lambda g(x_1)$

$0 \leq 0$ ✓

Problem 6.1 [30pts, 5 for each part] *(Convex Functions Analytical Exercises)* Which of the following functions are convex, strictly convex, or non-convex on the domains indicated. Proper reasoning is needed to receive full credit.

(a) $f(t) = |t|,\ t \in \mathbb{R}$

(b) $f(t) = t^2,\ t \in \mathbb{R}$

(c) $f(\mathbf{w}) = (\mathbf{a}^\top \mathbf{w} + b)^2,\ \mathbf{w} \in \mathbb{R}^2, \mathbf{a} = (1,1)^\top$

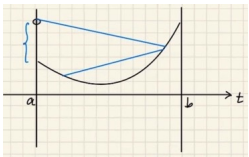
(d) $f(t) = e^{-t^2},\ t \in \mathbb{R}$

(e) $f(\mathbf{w}) = \log(1 + e^{\mathbf{a}^\top \mathbf{w}}),\ \mathbf{w} \in \mathbb{R}^2, \mathbf{a} = (1,1)^\top$

(f) $f(\mathbf{w}) = \max(0, 1 - \mathbf{a}^\top \mathbf{w}),\ \mathbf{w} \in \mathbb{R}^2, \mathbf{a} = (1,1)^\top$

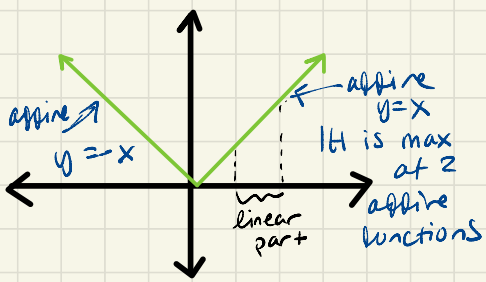
convexed

→ No jumps in convex except at the boundary b



→ Bar has to be above graph all time

a) $f(t) = |t|,\ t \in \mathbb{R}$



convex

g is not strictly convex over $[0,\infty]$ and over $(-\infty, 0]$

Explanation:

$x_0 = 1,\ x_1 = 5,\ \lambda = 1/2$

$f(x_\lambda) = f((1-\lambda)x_0 + \lambda x_1) \le (1-\lambda)f(x_0) + \lambda f(x_1)$ if $\lambda \neq \{0, 1\}$

$g((1-\lambda)x_0 + \lambda x_1) = g\left(\frac{1+5}{2}\right) = g(3)$

$= |3|$

$= 3$

$(1-\lambda)g(x_0) + \lambda g(x_1) = 1/2\, g(1) + 1/2\, g(5)$

$= 1/2 + 1/2(5) = |3| = 3$

$3 = 3$

Convex

b) $f(t) = t^2,\ t \in \mathbb{R}$

$f'(t) = 2t$

$f''(t) = 2 > 0$

$f(x_\lambda) = f((1-\lambda)x_0 + \lambda x_1) \le (1-\lambda)f(x_0) + \lambda f(x_1)$ if $\lambda \neq \{0, 1\}$

$x_0 = 1,\ x_1 = 2,\ \lambda = 2$

$f((1-2) + 2) = f(1) = 1$

$(1-2)f(1) + 2f(2) = -1 + 2(4) = 3$

$1 \neq 3$

Strictly convex

c) $f(w) = (a^T w + b)^2$, $w \in \mathbb{R}^2$, $a = (1,1)^T$

Proved in b) that $f(t) = t^2$ is strictly convex.

Since $f(w) = (a^T w + b)^2$ and $a^T w + b$ is an affine function which is convex. Then $f(w) = (a^T w + b)^2$ is convex.

$\perp$ to a (dot prod. = zero)

let $w_0 = \begin{vmatrix} 1 \\ -1 \end{vmatrix}$, $w_1 = \begin{vmatrix} 2 \\ -2 \end{vmatrix}$, $\lambda = 1/2$

$f(x_\lambda) = f((1-\lambda)x_0 + \lambda x_1) \leq (1-\lambda) f(x_0) + \lambda f(x_1)$
if $\lambda \neq \{0, 1\}$

$f(\underline{w}) = \left(a^T\left(\frac{w_0 + w_1}{2}\right) + b\right)^2 \leq 1/2\,(a^T w_0 + b^2) + 1/2\,(a^T w_1 + b^2)$

$f(\underline{w}) = b^2 \leq 1/2\, b^2 + 1/2\, b^2$

$f(\underline{w}) = b^2 = b^2$

Convex

d) $f(t) = e^{-t^2}$, $t \in \mathbb{R}$

let $f(x) = e^t$, $g(x) = -t^2$

$f''(x) = e^t \leftarrow$ convex

$g'(x) = -2t$

$g''(x) = -2 \not> 0$

Non-convex

e) $f(w) = \log(1+e^{a^T w})$, $w \in \mathbb{R}$, $a = (1,1)^T$

$g(\underline{w}) = a^T \underline{w}$ ← function of t

$f(x) = \log x \quad g(x) = 1 + e^{a^T \underline{w}}$

$f'(x) = \frac{1}{x \ln(10)}$

$f''(x) = -\frac{1}{x^2 \ln(10)} \not> 0$

Non-convex

f) $f(w) = \max(0, 1 - \underline{a}^T \underline{w}) \quad w \in \mathbb{R}^2 \quad a = (1,1)^T$

$f(t) = 0 \ \forall t$

$x_0 = 1$, $x_1 = 2$, $\lambda = 1/2$

$g((1-\lambda)x_0 + \lambda x_1) = 0$

$(1-\lambda) g(x_0) + \lambda g(x_1) = 0$

$0 \leq 0$, so convex

$g(t) = 1 - \underline{a}^T \underline{w}$, convex

The maximum of convex functions is convex

Convex

Problem 6.2 [6pts] *(Logistical Regression Analytical Exercise)* Let

$$\mathcal{D} = \Big\{(x_1 = -3, y_1 = -1), (x_2 = 5, y_2 = +1)\Big\}$$

be a set of pairs of scalar features and their labels for training a binary logistic-loss based classifier with class labels $-1, +1$ and a simplified form of the logistic-loss in which the w and b parameters for class -1 and the b parameter for class $+1$ are all set to zero and only class $+1$ has a free scalar parameter w which needs to be learned from training data. Specifically, let

$$p(y = +1|x, w) = \frac{e^{wx}}{1 + e^{wx}}, \quad p(y = -1|x, w) = \frac{1}{1 + e^{wx}}$$

(a) [2pts] Let $\text{NLL}(w) = -\ln\Big(\prod_{j=1}^{2} p(y_j|x_j, w)\Big)$ be the negative log-likelihood. Compute $\frac{\partial \text{NLL}}{\partial w}(w)$.

(b) [2pts] With initialization $\hat{w}_0 = 0$, compute $\hat{w}_1$, the value of w after one iteration of the gradient descent algorithm for minimizing NLL. Express $\hat{w}_1$ as a function of the step-size $s > 0$.

(c) [2pts] Compute $\hat{w}_{\text{logistic}}$, the maximum-likelihood estimate of w based on $\mathcal{D}$. Connect and discuss what you find with what you learned in the class about the need for regularization.

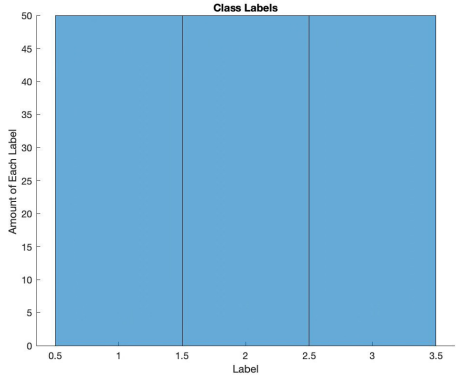
a) $\text{NLL}(w) = -\ln\left(\prod_{j=1}^{2} p(y_j|x_j, w)\right)$

Find $\frac{\partial \text{NLL}}{\partial w}(w)$

6.3

(a) [13pts] *Data analysis of full dataset (training + test)*: (i)[2pts] Plot the histogram of class labels. (ii)[5pts] Compute the matrix of empirical correlation coefficients for all pairs of features. Recall that the empirical correlation coefficient between a pair of features is equal to their empirical (cross) covariance divided by the square root of the product of their empirical variances. (iii)[6pts] Create 2D scatter plots of the dataset for *all distinct pairs* of features. Discuss your observations.

(i)

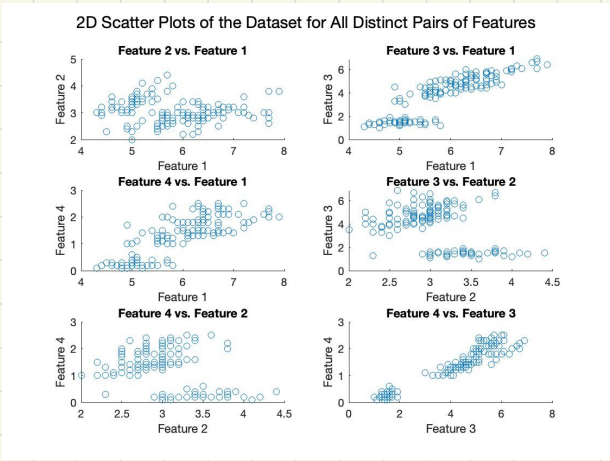


Equal amount of labels for each type of label.

(ii)

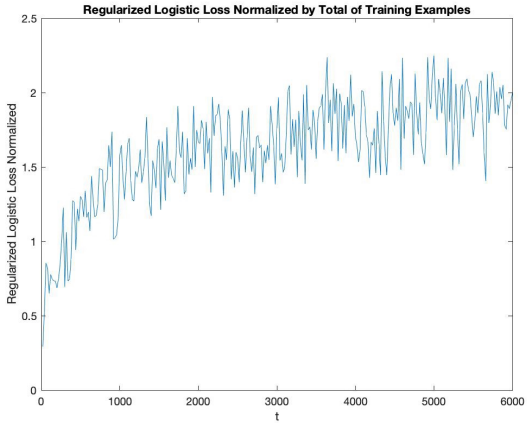
```
Empirical Correlation Coefficients between features 1 and 2:
    1.0000   -0.1094
   -0.1094    1.0000

Empirical Correlation Coefficients between features 1 and 3:
    1.0000    0.8718
    0.8718    1.0000

Empirical Correlation Coefficients between features 1 and 4:
    1.0000    0.8180
    0.8180    1.0000

Empirical Correlation Coefficients between features 2 and 3:
    1.0000   -0.4205
   -0.4205    1.0000

Empirical Correlation Coefficients between features 2 and 4:
    1.0000   -0.3565
   -0.3565    1.0000

Empirical Correlation Coefficients between features 3 and 4:
    1.0000    0.9628
    0.9628    1.0000
```

(iii)



Strong correlation noted between 4 and 3 also between 4 and 1

(b) [6pts] Plot the ℓ_2-regularized logistic loss normalized by the total number of training examples, i.e., $\frac{1}{n}g(\boldsymbol{\Theta})$, against iteration number t, for every 20 iterations, i.e., $t = 20t', t' = 1, 2, 3, \ldots, 300$. In order to avoid taking the logarithm of extremely small numbers, any value of $p(y_j|\mathbf{x}_j, \boldsymbol{\Theta})$ that is smaller than 10^{-10} should be treated as 10^{-10}. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.



Short term: values are fluctuating greatly.

Long term: the values are increasing for regularized logistic loss with every 20th iteration of t.

(c) [6pts] Plot the CCR of the training set against iteration number, for every 20 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend. Recall that the decision rule in Logistic Regression is given by:

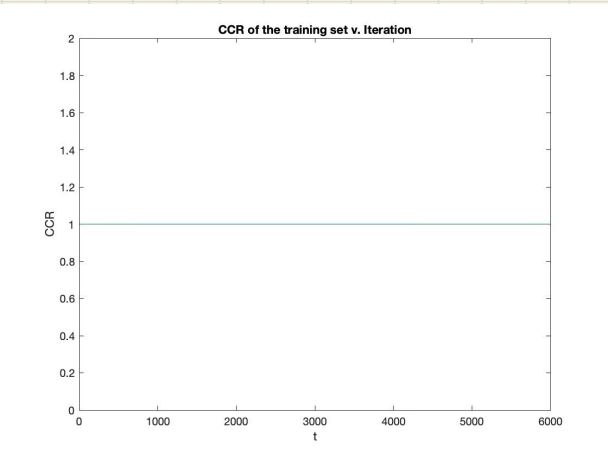
$$\hat{y}_j = \arg\max_{\ell=1,\ldots,m} \boldsymbol{\theta}_\ell^T \mathbf{x}_j^{\text{ext}}$$

and the CCR by

$$\frac{1}{n}\sum_{j=1}^{n} 1(y_j = \hat{y}_j).$$

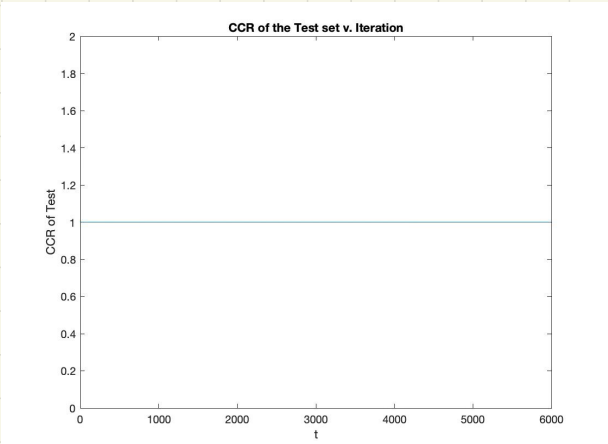
(d) [6pts] Plot the CCR of the *test* set against iteration number, for every 20 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.

c)



Flat line. understand should have a logistic increase. Can not figure out what's wrong.

d)



Same response as c)!

(e) [6pts] Plot the so-called log-loss of the test set against iteration number, for every 20 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend. The log-loss of the test set is the NLL of the test set normalized by the number of test examples:

$$\text{logloss}(\boldsymbol{\Theta}) = -\frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} \ln p(y_j|\mathbf{x}_j, \boldsymbol{\Theta})$$

where $\mathbf{x}_j$ is a test feature vector and y_j is its ground truth label. In order to avoid taking the logarithm of extremely small numbers, any value of $p(y_j|\mathbf{x}_j, \boldsymbol{\Theta})$ that is smaller than 10^{-10} should be treated as 10^{-10}.

(f) [21pts] *Final values*: After the SGD algorithm terminates, report the final values of the following: (i)[9pts] $\boldsymbol{\Theta}$, (ii)[1pt] the training CCR, (iii)[1pt] the test CCR, (iv)[5pts] training confusion matrix, (v)[5pts] test confusion matrix. Discuss your observations.

(g) [6pts] Consider the six 2-dimensional subsets of 4-dimensional feature space in which two out of the four components are zero, e.g., $\{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 : x_3 = x_4 = 0\}$. In each of these six subsets, indicate the decision regions, i.e., which points will be classified as class 1, which as class 2, and which as class 3. Discuss your observations.

e) Not able to get function working. Should have gotten a logistic plot where the value has a deep drop and then settles as t increases.

f) (i) on matlab, hard to copy over.

(ii) – (v)

```
Final training CCR
    1


Final test CCR
    1


Training Confusion matrix
    0     0     0     0
   31     4     0     0
   35     0     0     0
   35     0     0     0


Test Confusion matrix
    0     0     0     0
   31     4     0     0
   35     0     0     0
   35     0     0     0
```

There is obvious errors in my code. The correct classification rate is 1, even though the confusion matrix explains otherwise. My algorithm is inaccurate.

(g) [6pts] Consider the six 2-dimensional subsets of 4-dimensional feature space in which two out of the four components are zero, e.g., $\{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 : x_3 = x_4 = 0\}$. In each of these six subsets, indicate the decision regions, i.e., which points will be classified as class 1, which as class 2, and which as class 3. Discuss your observations.

Subset 1) class 1

Subset 2) class 1

Subset 3) class 2

Subset 4) class 2

Subset 5) class 3

Subset 6) class 3

Since 2 components are empty, each class will be equally distributed in the six subsets, each only having one decision region.