CS 5350/6350: Machine Learining Fall 2015

Homework 5

Handed out: Nov 20, 2015
Due date: Dec 8, 2015

General Instructions

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free ask the instructor or the TAs questions about the homework.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do *not* need to write it in glory detail. Your assignment should be **no more than 10 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- The homework is due by midnight of the due date. Please submit the homework on Canvas.

1 Naïve Bayes

Consider the Boolean function $f_{TH(3,7)}$. This is a threshold function defined on the 7 dimensional Boolean cube as follows: given an instance x, $f_{TH(3,7)}(x) = 1$ if and only if 3 or more of x's components are 1.

1. [4 points] Show that $f_{TH(3,7)}$ has a linear decision surface over the 7 dimensional Boolean cube.

Solution: I include two approaches, the first a simple proof by example by applying the definition of linear separability which we have become very familiar, the second a derivation of a linear separable expression starting from a substitution of the Naive Bayes assumption $P(\mathbf{x}|y) = \prod_{i=0}^{d} P(x_j|y)$ when calculating if the classifier will predict a value of 1.

(a) We show here linear separability of the form $y = 1$ if $\mathbf{w}^T x + \theta \geq 0$ else $y = 0$. Using the weight vector $\mathbf{w} = [1, 1, 1, 1, 1, 1, 1]^T$ and $\theta = -3$ the inner product of $\mathbf{w}^T x \geq 3$ iff 3 or more $x_j = 1$.

(b) We start by substituting the the Naive Bayes assumption for $P(\mathbf{x}|y)$ into the probability that we are given a label $y = 1$, i.e. $\frac{P(\mathbf{x}|P(f_{TH(3,7)}=1)P(f_{TH(3,7)}=1))}{P(\mathbf{x}|f_{TH(3,7)}=0)P(f_{TH(3,7)}=0)} \geq 1$

$$\frac{P(f_{TH(3,7)}=1)}{P(f_{TH(3,7)}=0)} \cdot \prod_{j=0}^{7} \frac{P(x_j|f_{TH(3,7)}(x)=1)}{P(x_j|f_{TH(3,7)}(x)=0)} \geq 1$$

We now expand the contents of the product with and equivalent expression easier to work with.

$$P(x_j|1) = \frac{P(x_j)P(1|x_j)}{P(1)}$$
$$P(x_j|0) = \frac{P(x_j)P(0|x_j)}{P(0)}$$
$$\rightarrow \frac{P(x_j|1)}{P(x_j|0)} = \frac{P(1|x_j)P(0)}{P(1)P(0|x_j)}$$

The probability $P(f_{TH(3,7)}=1)$ is a combinatorial problem of the probability there will be at least 3 or more $x_j = 1$.

$$\frac{P(f_{TH(3,7)}=1)}{P(f_{TH(3,7)}=0)} =$$
$$\frac{P(f_{TH(3,7)}=1)}{1-P(f_{TH(3,7)}=1)} = \frac{\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}}{1-\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}}$$

With one $x_j = 1$ the probability $P(f_{TH(3,7)}=1)$ is found knowing that at least 2 more x_j must equal 1. Alternatively $P(x_j = 1|f_{TH(3,7)}=0)$ implies at least 5 other x_j have a value of 0.

$$P(f_{TH(3,7)}=1, x_j=1) = \sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^j(0.5)^{6-j} \quad (1)$$

$$P(f_{TH(3,7)}=0, x_j=1) = \sum_{j=6}^{6} \frac{6!}{j!(6-j)!}(0.5)^j(0.5)^{6-j} \quad (2)$$

A similar argument gives us the $P(f_{TH(3,7)}=1, x_j=0)$ and $P(f_{TH(3,7)}=1, x_j=0)$. Respectively, at least 5 x_j are 0 and at least 4 x_j are 0.

$$P(f_{TH(3,7)} = 1, x_j = 0) = \sum_{j=5}^{6} \frac{6!}{j!(6-j)!}(0.5)^j(0.5)^{6-j} \tag{3}$$

$$P(f_{TH(3,7)} = 0, x_j = 0) = \sum_{j=4}^{6} \frac{6!}{j!(6-j)!}(0.5)^j(0.5)^{6-j} \tag{4}$$

Combining all of these allows for an inequality which when true implies the Boolean cube is linearly separable:

$$\frac{\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}}{1-\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}} \cdot \prod_{i=0}^{7} \frac{(\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6)^{x_j}(1-(\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6)^{1-x_j}}{(0.5)^{6\cdot x_j}.5^{6\cdot(1-x_j)}} \geq 1$$

To simplify we collect like terms and then take the log of the inequality.

$$\frac{\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}}{1-\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}} \cdot \prod_{j=0}^{7} (\frac{\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6}{0.5^6} \frac{1-0.5^6}{1-\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6})^{x_j}$$

$$\rightarrow log(\frac{\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}}{1-\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}}) + \sum_{0}^{7} x_j \frac{\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6}{0.5^6} \frac{1-0.5^6}{1-\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6}$$

The first term is constant, $b = log(\frac{\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}}{1-\sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}})$ and the product sum is an inner product with $\mathbf{w}$.

$$s.t.\ w_j = \frac{\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6}{0.5^6} \frac{1-0.5^6}{1-\sum_{j=2}^{6} \frac{6!}{j!(6-j)!}(0.5)^6}$$

$$b + \mathbf{x} \cdot \mathbf{w}^t > 0$$

2. [7 points] Assume that you are given data sampled according to the uniform distribution over the Boolean cube $\{0,1\}^7$ and labeled according to $f_{TH(3,7)}$. Use naïve Bayes to learn a hypothesis that predicts these labels. What is the hypothesis generated by

the naïve Bayes algorithm? (You do not have to implement the algorithm here. You may assume that you have seen all the data required to get accurate estimates of the probabilities).

Solution:
We know $h(\mathbf{x}) = arg \max_{y\in\{0,1\}} P(y)\prod_{i=1}^{n} P(x_j|y)$ where $\mathbf{x}$ is the Boolean cube of 7-Dimensions and $y = f_{TH(3,7)}(\mathbf{x})$.

$$h(\mathbf{x}) = arg \max_{y\in\{0,1\}} P(y)\prod_{j=1}^{n} P(x_j|y)$$
$$= arg \max_{y\in\{0,1\}} (\ P(1)\cdot P(x_1|1)\cdots P(x_7|1),\ P(1)\cdot P(x_1|0)\cdots P(x_7|0)\ )$$

Recall from the previous problem the values of $P(0)$ where for at least 5 $x_j = 0$ and $P(1)$ for 3 or more $x_j = 1$

$$P(1) = \sum_{j=3}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}$$
$$P(0) = \sum_{j=5}^{7} \frac{7!}{j!(7-j)!}(.5)^j(.5)^{7-j}$$

Computing this we find $P(1) = \frac{99}{128}$ and $P(0) = \frac{29}{128}$
Now we use the expression for $P(x_i|0) = \frac{P(x_j)P(0|x_j)}{P(0)}$ and $P(x_i|1) = \frac{P(x_j)P(1|x_j)}{P(1)}$ as done previously. And solve for the equations $(1-4)$

$$P(1|x_j = 1) = \frac{57}{64}$$
$$P(1|x_j = 0) = \frac{21}{32}$$
$$P(0|x_j = 1) = \frac{7}{64}$$
$$P(0|x_j = 0) = \frac{11}{32}$$

substituting the now known values into $P(x_j|0) = \frac{P(x_j)P(0|x_j)}{P(0)}$ and $P(x_i|1) = \frac{P(x_j)P(1|x_j)}{P(1)}$

$$P(x_j|0) = \begin{cases} \frac{1/2 \cdot 7/64}{29/128}, & \text{if } x_j = 1 \\ \frac{1/2 \cdot 11/32}{29/128}, & \text{if } x_j = 0 \end{cases} \quad P(x_j|1) = \begin{cases} \frac{1/2 \cdot 57/64}{99/128}, & \text{if } x_j = 1 \\ \frac{1/2 \cdot 21/32}{99/128}, & \text{if } x_j = 0 \end{cases}$$

Our hypothesis then, given some $\mathbf{x}$, chooses the maximum of the previously listed probabilities.

$$h(\mathbf{x}) = arg \max_{y \in \{0,1\}} \frac{29 + 70y}{128} \prod_{j=1}^{7} \frac{42 + 15x_j}{99} y - \frac{22 - 15x_j}{29}(y - 1) \tag{5}$$

3. [4 points] Show that the hypothesis produced in the previous question does not represent this function.

Solution: Consider the counter example $\mathbf{x} = [1, 1, 0, 0, 0, 0, 0]$. The calculated hypothesis will choose the maximum value considering $y = 0$ and $y = 1$ for the product of each x_j

$$h([1, 1, 0, 0, 0, 0, 0]) = arg\ max \begin{cases} 29/128 \cdot (7/64)^2 \cdot (22/29)^5, & \text{for } y = 0 \\ 99/128 \cdot (57/99)^2 \cdot (42/99)^5, & \text{for } y = 1 \end{cases}$$

Which is incorrect since the maximum value corresponds to y=1. The hypothesis does not represent $f_{(3,7)}(\mathbf{x})$

4. [5 points] Are the naïve Bayes assumptions satisfied by $f_{TH(3,7)}$? Justify your answer.

Solution: The counter example of the previous question showed our hypothesis did not represent $f_{TH(3,7)}$ due to the independence of attribute values leading to a false labeling. Namely the outcome of $f_{TH(3,7)}$ is determined by the conditional dependence between all x_j.

2 EM Algorithm

There are two grocery stores in the neighborhood of the U: Smith's and Trader Joe's. Each store has n checkout lanes. The number of customers for each lane per unit time, say one day, is distributed according to Poisson distribution with parameter λ. That is, for the i'th checkout lane,

$$P(\#\text{ of customers for lane } i = x_i|\lambda) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

with parameters λ_S and λ_T for Smith's and Trader Joe's, respectively.

1. [10 points] Given a record of customer counts $(x_1, \cdots, x_n)$, where x_i denotes the number of customers went through lane i, what is the most likely value of λ?

Solution: λ is the average number of customers throughout the day. The total number of persons through all lanes is $\sum_{i=1}^{n} x_i$. The average, and λ, is then $\frac{1}{n}\sum_{i=1}^{n} x_i$. More precisely:

$$\begin{aligned} P(x_1, \cdots x_n|\lambda) &= \prod_{i=1}^{n} P(x_i|\lambda) \\ &= \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} \\ &= \frac{\lambda^{\sum_i^n x_i} e^{-n\lambda}}{\prod_{i=1}^{n} x_i!} \end{aligned}$$

The log likelihood for lambda is proportional to the joint probability distribution, i.e. the probability that x_i belongs to Smiths or Trader Joes, which simplifies our probability to terms which depend on λ.

$$P(x_1, \cdots x_n|\lambda) = \lambda^{\sum_i^n x_i} e^{-n\lambda}$$

We now take the log

$$\begin{aligned} log(P(x_1, \cdots x_n|\lambda)) &= log(\lambda^{\sum_i^n x_i} e^{-n\lambda}) \\ &= \sum_{i=1}^{n} x_i log(\lambda) - \lambda n \end{aligned}$$

Taking the derivative with respect to lambda and solving for the maximum:

$$\begin{aligned} dP/d\lambda &= \sum_{i=1}^{n} x_i \frac{1}{\lambda} - n \\ &= 0 \\ \rightarrow \lambda &= \frac{\sum_{i=1}^{n} x_i}{n} \end{aligned}$$

2. [10 points] Assume now that you are given a collection of m records $\{(x_{j1}, \cdots x_{jn})\}$, where $j = 1, \cdots, m$. You do not know which record is from Smith's and which is from Trader Joe's. Assume that the probability of a record is from the Smith's is η. In other words, it means that the probability that a record is from Trader Joe's is $1 - \eta$. Explain the generative model that governs the generation of this data collection. In doing so, name the parameters that are required in order to fully specify the model.

Solution: We are given a set of j data sets ranging from 1 to n consisting of customers for a lane $\{(x_{1,1}, \cdots x_{j,n})\} \cdots \{(x_{j1}, \cdots x_{jn})\}$. For the purpose of this problem $m \in (0, j)$ and $i \in (0, n)$. Our generative model is

$$h_m(x) = arg \max_{y \in (S,T)} P(y) \prod_{j=1}^{m} \sum_{i=1}^{n} P(y|x)$$
$$= arg \max_{y \in (S,T)} P(y) \prod_{j=1}^{m} \sum_{i=1}^{n} \frac{P(x|y)}{P(x)}$$

We are given a binomial probability distribution that some record is from Smiths $P(y = S) = \eta$ and from Trader Joes $P(y = T) = 1 - \eta$. This binomial distribution is then our prior for calculating the maximum likelihood and it is important to note that this prior is uniform over a single record, i.e. for $k \in (1, n)$ $P(x_{m,i}) = P(x_{m,i+k}) = \eta$ or $1 - \eta$ if $y = S$ or $y = 1 - \eta$ respectively. This uniformity simplifies our generative model in that there is no dependence on $P(x)$.

$$h_m(x) = arg \max_{y \in (S,T)} \prod_{j=1}^{m} \sum_{i=1}^{n} P(y)P(x|y)$$

The probability of x given y is given by the Poisson distribution using λ_S and λ_T given by equation (6). The poisson is therefore $P(x|y)$. During learning λ_S and λ_T are initialized to λ_o. λ_S are then λ_T updated **only if** the $h_m(x)$ generating function is maximized for $y = S$ or $y = T$ respectively.

$$h_m(x) = arg \max_{y \in (S,T)} \begin{cases} \eta \prod_{j=1}^{m} \frac{\lambda_S^{\sum_{i=1}^{n} x_{j,i}} \ e^{-\lambda_S}}{\prod_{i=1}^{n} x_i!}, & \text{if } y = S \\ (1 - \eta) \prod_{j=1}^{m} \frac{\lambda_T^{\sum_{i=1}^{n} x_{j,i}} e^{-\lambda_T}}{\prod_{i=1}^{n} x_i!}, & \text{if } y = T \end{cases}$$

Each summation in our hypothesis is unique to Trader Joes and Smiths. The set of needed parameters is then $\lambda_o, \lambda_T, \lambda_S,$ and η

3. [10 points] Assume that you are given the parameters of the model described above. How would you use it to cluster records to two groups, the Smith's and the Trader Joe's?

Solution: Recall $h_m(x)$ from the previous question our generative model. Since our model, the maximum likelihood is dependent on the labeling, meaning proportional to the probability of some $x_{j,n}$ belonging to the Smiths data or Trader Joes.

For this reason we can simplify our expression.

$$h_m(x) = arg \max_{y \in (S,T)} \begin{cases} \eta \prod_{j=1}^{m} \lambda_S^{\sum_{i=1}^{n} x_{j,i}} \, e^{-\lambda_S}, & \text{if } y = S \\ (1-\eta) \prod_{j=1}^{m} \lambda_T^{\sum_{i=1}^{n} x_{j,i}} \, e^{-\lambda_T} \ , & \text{if } y = T \end{cases}$$

From this generative method we can bin records as belonging to Smiths or Trader Joes, given some $x_{j,n}$, by calculating $h(x_{j,n})$ and labeling $x_{j,n}$ with the label corresponding to the maximum value. For the label chosen, S or T, its respective λ_S or λ_T to incorporate the current record. To consolidate consider $y = 1$ if the max is S and $y = 0$ if the max is S.

$$h_m(x) = arg \max_{y \in (S,T)} =$$
$$\eta \prod_{j=1}^{m} \lambda_S^{\sum_{i=1}^{n} x_{j,i}} \, e^{-\lambda_S} \cdot y - (y-1)(1-\eta) \prod_{j=1}^{m} \lambda_T^{\sum_{i=1}^{n} x_{j,i}} \, e^{-\lambda_T}$$

4. [10 points] Given the collection of m records without labels of which store they came from, derive the update rule of the EM algorithm. Show all of your work.

Solution: We must find a linearly separated form of our previous classifier governed by the decision boundary

$$given \ \frac{P(x_j|y=S)}{P(x_j|y=T)} = \frac{P(x_j|y=S)P(T)}{P(x_j|y=T)P(S)}$$
$$\frac{P(y=S)}{P(y=T)} \cdot \prod_{j=1}^{m} \sum_{i=1}^{n} \frac{P(x_{j,i}|y=S)P(T)}{P(x_{j,i}|y=T)P(S)} > 1$$

Where $\frac{P(y=S)}{P(y=T)} = \frac{\eta}{1-\eta}$ and $P(x_j|y=S)$, $P(x_j|y=T)$ are given by the Poisson distribution $\frac{\lambda_S^{x_{j,i}} e^{-\lambda_S}}{x_i!}$, $\frac{\lambda_T^{x_{j,i}} e^{-\lambda_T}}{x_i!}$ respectively. Substituting

$$\frac{\eta}{(1-\eta)} \prod_{j=1}^{m} \frac{\lambda_S^{\sum_{i=1}^{n} x_{j,i}} e^{-\lambda_S}}{\lambda_T^{\sum_{i=1}^{n} x_{j,i}} e^{-\lambda_T}} \frac{1-\eta}{\eta} > 1$$

Taking the log to simplify.

$$log(\frac{\eta}{1-\eta}) + \sum_{j=1}^{m} (\sum_{i=1}^{n} x_{j,i} log(\lambda_S) - log\lambda_S - (\sum_{i=1}^{n} x_{j,i} log\lambda_T - log\lambda_T + log(1-\eta) - log\eta)) > 0$$

collecting like terms we are able to see the form of our update vector

$$(m-1)(log(1-\eta)-log\eta))+m(log\lambda_T-log\lambda_S)+\sum_{j=1}^{m}\sum_{i=1}^{n}x_{j,i}(log(\lambda_S-log\lambda_T)>0$$

Here we see the form $b+\mathbf{w}^T\mathbf{x}\geq 0$ where $\mathbf{w}^T=log\lambda_S$ or $log\lambda_T$ where recall λ_S and λ_T are the mean of the current record vector under question **if** the current record vector is from Smiths or Trader Joes. If not then there is no value which prevents their difference from being zero. Lastly, we have $b=log(\frac{\eta}{1-\eta})+m(log\lambda_T-log\lambda_S))$.

3 Experiment

We looked maximum a posteriori learning of the logistic regression classifier in class. In particular, we showed that learning the classifier is equivalent to the following optimization problem:

$$\min_{\mathbf{w}}\Big\{\sum_{i=1}^{m}\log(1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i))+\frac{1}{\sigma^2}\mathbf{w}^T\mathbf{w}\Big\}$$

In this question, you will derive the stochastic gradient descent algorithm for the logistic regression classifier, and also implement it with cross-validation. Detailed instructions on cross-validation procedure can be found in homework 1, and instructions on SGD can be found in homework 4.

1. [5 points] What is the derivative of the function $\log(1+\exp(-y_i\mathbf{w}^T\mathbf{x}_i))$ with respect to the weight vector?

Solution:

$$\begin{aligned}\frac{d}{d\mathbf{w}}(log(\ 1+e^{-y_i\mathbf{w}^T\mathbf{x_i}}))) &= \\ &=\frac{1}{1+e^{-y_i\mathbf{w}^T\mathbf{x_i}}}\cdot(-y_i\mathbf{x_i}) \\ &=\frac{-y_i\mathbf{x_i}}{1+e^{-y_i\mathbf{w}^T\mathbf{x_i}}}\end{aligned}$$

By the logistic regression model this is equivalent to an alternative representation which will prove useful for cases of overflow during the experiment.

$$=\frac{e^{-y_i\mathbf{w}^T\mathbf{x_i}}}{1+e^{-y_i\mathbf{w}^T\mathbf{x_i}}}\cdot(-y_i\mathbf{x_i})$$

2. [5 points] The inner most step in the SGD algorithm is the gradient update where we use a single example instead of the entire dataset to compute the gradient. Write down the objective where the entire dataset is composed of a single example, say $(\mathbf{x}_i, y_i)$. Derive the gradient of this objective with respect to the weight vector.

Solution: Using the $J(\mathbf{w})$ given and the previously solved derivative we can find the minimum weight vector by solving for $\mathbf{w}$ when $\frac{dJ(\mathbf{w})}{d\mathbf{w}}$.
$\frac{dJ(\mathbf{w})}{d\mathbf{w}} = \sum_{i=1}^{m} \frac{-y_i \mathbf{x_i}}{1+e^{y_i \mathbf{w}^T \mathbf{x_i}}} + \frac{d}{d\mathbf{w}}(\frac{1}{\sigma^2}\mathbf{w}^T\mathbf{w})$
However $m = 1$ since the data point in consideration *"is"* the entire data set.
$\frac{dJ(\mathbf{w})}{d\mathbf{w}} = \frac{-y_i \mathbf{x_i}^T}{1+e^{y_i \mathbf{w}^T \mathbf{x_i}}} + \frac{2\mathbf{w}^T}{\sigma^2}$
Calculating the w given $\frac{dJ(\mathbf{w})}{d\mathbf{w}} = 0$ is equivalent iteratively reducing w by our loss function.

3. [10 points] Write down the pseudo code for the stochastic gradient algorithm using the gradient from previous part.

Given data set $S = [[x_{1,1}, \cdots, x_{1,n}], \cdots, [x_{j,1}, x_{j,n}]]$
Initialize $\mathbf{w} = [0, \cdots, 0]$ of size n
$(\mathbf{x_i}, y) \leftarrow$ random(S)
$\nabla(\mathbf{w^{t-1}}) \leftarrow \min_{\mathbf{w}}\{\sum_{i=1}^{m} \frac{-y_i \mathbf{x_i}}{1+e^{y_i \mathbf{w}^T \mathbf{x_i}}} + \frac{2\mathbf{w}^T}{\sigma^2}\}$
$\mathbf{w^t} \leftarrow \mathbf{w^{t-1}} - r \cdot \nabla(\mathbf{w^{t-1}})$
return $\mathbf{w^t}$

4. [20 points] Implement your pseudo code as a training algorithm with cross-validation on the σ parameter. This parameter basically helps trade off between generalizability and model fit.

Solution:

r, σ, homework 4, update derive (gradiant) per example, negative log likilihood at the end of thetraining set. not on each iteration. avoid explotion with approximations $\frac{1}{1+e^{-z}}$ is deriv. aproximate on overflow with $\frac{e^{-z}}{1+e^{z}}$

Use the two `astro` data sets (original and scaled) from homework 4 to train and evaluate the learner. In your writeup, please report on the accuracy of your system, what value of sigma you chose based on cross validation, how many epochs you chose to run SGD, and a plot of the NEGATIVE log likelihood after each epoch of SGD.

As mentioned in previous homeworks, you may use any programming language for your implementation. Upload your code along with a script so the TAs can run your solution in the CADE environment.

4 HAPPY HOLIDAYS Extra Credit

25pts You've seen stochastic gradient decent applied to logistic regression. Now we ask why this is a viable strategy for optimizing this objective. Prove that SGD will find the optimal value for this function. This can be done by demonstrating that the objective is convex. There are many ways to prove this. One of the most straightforward ways to show this is demonstrate that the Hessian is positive-semidefninite.

1. [5 ppoins] Find the Gradient of:

$$\sum_{i=1}^{m} \log(1 + \exp(-y_i w^T x_i)) + \frac{1}{\sigma^2} w^T w$$

2. [5 points] Find the Hessian of (a).

3. [10 points] prove that the Hessian is positive semidefinite.

4. [5 points] Why does this prove that we are GAURANTEED to have within $\epsilon > 0$ of the right answer?