Homework 4: Probabilistic models

Due: Wednesday, March 22, 2023 at 11:59PM EST

Instructions: Your answers to the questions below, including plots and mathematical work, should be submitted as a single PDF file. It's preferred that you write your answers using software that typesets mathematics (e.g.LaTeX, LyX, or MathJax via iPython), though if you need to you may scan handwritten work. You may find the minted package convenient for including source code in your LaTeX document. If you are using LyX, then the listings package tends to work better.

---

1 Logistic Regression

Consider a binary classification setting with input space $\mathcal{X} = \mathbb{R}^d$, outcome space $\mathcal{Y}_\pm = \{-1, 1\}$, and a dataset $\mathcal{D} = \left((x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\right)$.

Equivalence of ERM and probabilistic approaches

In the lecture we derived logistic regression using the Bernoulli response distribution. In this problem you will show that it is equivalent to ERM with logistic loss.

ERM with logistic loss.

Consider a linear scoring function in the space $\mathcal{F}_{\text{score}} = \left\{x \mapsto x^T w \mid w \in \mathbb{R}^d\right\}$. A simple way to make predictions (similar to what we've seen with the perceptron algorithm) is to predict $\hat{y} = 1$ if $x^T w > 0$, or $\hat{y} = \text{sign}(x^T w)$. Accordingly, we consider margin-based loss functions that relate the loss with the margin, $yx^T w$. A positive margin means that $x^T w$ has the same sign as y, i.e. a correct prediction. Specifically, let's consider the **logistic loss** function $\ell_{\text{logistic}}(y, w) = \log\left(1 + \exp(-yw^T x)\right)$. This is a margin-based loss function that you have now encountered several times. Given the logistic loss, we can now minimize the empirical risk on our dataset $\mathcal{D}$ to obtain an estimate of the parameters, $\hat{w}$.

MLE with a Bernoulli response distribution and the logistic link function.

As discussed in the lecture, given that $p(y = 1 \mid x; w) = 1/(1+\exp(-x^T w))$, we can estimate w by maximizing the likelihood, or equivalently, minimizing the negative log-likelihood ($\text{NLL}_{\mathcal{D}}(\text{w})$ in short) of the data.

1. Show that the two approaches are equivalent, i.e. they will produce the same solution for w.

   - first re-write the erm approach
     - so recall that in ERM the empirical risk of a function is given by
     - $$\hat{R}_n = \frac{1}{n}\Sigma_{i=1}^n \ell(f(x^i), y^i)$$
     - so in the case of logistic loss we can see that our optimal weight vector will be given by

       $$\hat{w}_{erm} = argmin_{w \in \mathbb{R}^d} \frac{1}{n}\Sigma_{i=1}^n \ell_{log}(f(x^i), y^i) = argmin_{w \in \mathbb{R}^d} \frac{1}{n}\Sigma_{i=1}^n log(1 + e^{-y^i w^t x^i})$$

- as shown in homework 2 question 26 this expression can be re-written as $\hat{w}_{erm} = argmin_{w\in\mathbb{R}^d}\frac{1}{n}\Sigma_{i=1}^n \ell_{log}(f(x^i), y^i) = argmin_{w\in\mathbb{R}^d}\frac{1}{n}\Sigma_{i=1}^n log(1+e^{-y^iw^tx^i}) = argmin_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(1+e^{-w^tx^i}) + (1-y^i)log(1+e^{w^tx^i})$
- further we know that maximising the inverse of a function is equiv lent to minimizing it. that is $\hat{w}_{erm} = argmin_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(1+e^{-w^tx^i})+(1-y^i)log(1+e^{w^tx^i}) \iff argmax_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(\frac{1}{log(1+e^{-w^tx^i})}) + (1-y^i)\frac{1}{log(1+e^{w^tx^i})})$
- and further as the log is a monotonic transformation we can say $\hat{w}_{erm} = argmin_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(1+e^{-w^tx^i}) + (1-y^i)log(1+e^{w^tx^i}) \iff \iff argmax_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(\frac{1}{log(1+e^{-w^tx^i})})+(1-y^i)\frac{1}{log(1+e^{w^tx^i})}) \iff argmax_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(\frac{1}{1+e^{-w^tx^i}})+(1-y^i)log(\frac{1}{1+e^{w^tx^i}})$
- notice further that if we define the sigmoid function as $f(x) = \frac{1}{1+e^{-x}}$ we can express our problem as $\hat{w}_{erm} = argmax_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(\frac{1}{1+e^{-w^tx^i}}) + (1-y^i)log(\frac{1}{1+e^{w^tx^i}}) = argmax_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)f(w^tx^i) + (1-y^i)f(-w^tx^i)$
- further note the following equality $1-f(x) = 1-\frac{1}{1+e^{-x}} = \frac{e^{-x}}{1+e^{-x}} = \frac{1}{1+e^{x}} = f(-x)$
- so substituting this back into our erm function yields our final result for erm

$$\hat{w}_{erm} = argmax_{w\in\mathbb{R}^d}\frac{1}{2n}\Sigma_{i=1}^n(1+y^i)log(f(x)) + (1-y^i)log(1-f(x))$$

- now we can re-writ the maximum likelihood approach.
  - if we assume that our data is iid, and that each individual example is a Bernoulli random viable we can write the likelihood of our data set as $\mathcal{L}(D) = P(y^1, y^2...y^n|x^1...x^n, , w) = P(y^1|x^1, w)P(y^2|y^1, x^1, x^2, w)... = \Pi_{i=1}^n P(y^i|x^i, w)$
  - then our log likelihood of the data set is $\ell(d) = log(\mathcal{L}(D) = \Sigma_{i=1}^n P(y^i|x^i, w)$
  - in the binary classification case with $y \in \{-1, 1\}$this becomes $\ell(d) = \frac{1}{2}\Sigma_{i=1}^n(1+y^i)P(y^i=1|x^i, w) + (1-y^i)P(y^i=-1|x^i, w)$
  - under the assumptions of logistic regression $P(y=1|x, w) = f(w^tx)$ where f as defined above is the sigmoid function thus we have $\ell(d) = \frac{1}{2}\Sigma_{i=1}^n(1+y^i)f(w^tx^n) + (1-y^i)(1-f(w^tx^n)$
  - so the weight vector this would yield would be $\hat{w}_{mle} = argmax_{w\in\mathbb{R}^d}\frac{1}{2}\Sigma_{i=1}^n(1+y^i)log(\frac{1}{1+e^{-w^tx^i}}) + (1-y^i)log(\frac{1}{1+e^{w^tx^i}})$
- so it is clear that the $\hat{w}_{erm}$ and $\hat{w}_{mle}$ only differ by a constant factor of $\frac{1}{n}$ which will not effect the arg max and thus $\hat{w}_{erm} = \hat{w}_{mle}$
- now we can solve for $\hat{w} = \hat{w}_{erm} = \hat{w}_{mle}$
  - given

$$\hat{w} = argmax_{w\in\mathbb{R}^d}\frac{1}{2}\Sigma_{i=1}^n(1+y^i)log(\frac{1}{1+e^{-w^tx^i}}) + (1-y^i)log(\frac{1}{1+e^{w^tx^i}})$$

  - we can find the gradient with respect to w as $\nabla_w(\hat{w}) = \frac{1}{2}\Sigma_{i=1}^n[\frac{y^i+1}{f(w^tx^i)} - \frac{1-y^i}{f(w^tx^i}]\frac{\partial f(w^tx^i)}{\partial w} = \frac{1}{2}\Sigma_{i=1}^n[\frac{y^i+1}{f(w^tx^i)} - \frac{1-y^i}{f(w^tx^i}](f(w^tx^i)(1-f(w^tx^i))(x^i) = \frac{1}{2}\Sigma_{i=1}^n[\frac{(y^n+1)(1-f(w^tx^n)-f(w^tx^n)(1-y^n)}{f(w^tx^n)(1-f(w^tx^n)}](f(w^tx^n)(1-f(w^tx^n))x^i = \frac{1}{2}\Sigma_{i=1}^n[y^n+1-2f(w^tx^n)] = \Sigma_{i=1}^n[\frac{y^n+1}{2} - f(w^tx^n)]$

Linearly Separable Data

In this problem, we will investigate the behavior of MLE for logistic regression when the data is linearly separable.

2. Show that the decision boundary of logistic regression is given by $\{x \colon x^T w = 0\}$. Note that the set will not change if we multiply the weights by some constant c.

   - our decision boundary is the set of points such that we are equally likely to predict either class
   - this can thus be expressed in terms of the log odds between the two classes $log\frac{P(y=1|x,w)}{P(y=-1|x,w)} = log(\frac{P(y=1|x,\theta)}{1-P(y=1|x,\theta)}) = log(\frac{\frac{1}{1+e^{-w^tx}}}{1-\frac{1}{1+e^{w^tx}}}) = log(\frac{\frac{1}{1+e^{-w^tx}}}{\frac{e^{-w^tx}}{1+e^{w^tx}}}) = log(\frac{1}{e^{-w^tx}}) = log(1)-log(e^{-w^tx}) = 0-(-w^txlog(e)) = w^tx$
   - and we know that our decision boundary is the space such that the odds between the two classes is 1, or in other words the log odds between the two classes are zero. we showed above that the $log\frac{P(y=1|x,w)}{P(y=-1|x,w)} = log(\frac{P(y=1|x,\theta)}{1-P(y=1|x,\theta)}) = w^tx = x^tw$ setting this equal to zero we see that the decision boundary is given by $\{x : x^tw = 0\}$ in other words it s a hyperplane that is perpendicular to our weight vector w.

3. Suppose the data is linearly separable and by gradient descent/ascent we have reached a decision boundary defined by $\hat{w}$ where all examples are classified correctly. Show that we can always increase the likelihood of the data by multiplying a scalar c on $\hat{w}$, which means that MLE is not well-defined in this case. (Hint: You can show this by taking the derivative of $L(c\hat{w})$ with respect to c, where L is the likelihood function.)

   - first note that the likelihood of our data in this case is given by $\ell(D,w) = \frac{1}{2}\Sigma_{i=1}^n((1+y^i)log(P(y^i = 1|x_i = 1,w)) + ((1-y^i)log(P(y^i = -1|x_i = 1,w))) = \frac{1}{2}\Sigma_{i=1}^n((1+y^i)log(f(w^tx^i)) + (1-y^i)log(1-f(w^tx^i))$
   - so we can write the likelihood of our data set and weight vector times some constant c as $\ell(D,cw) = \frac{1}{2}\Sigma_{i=1}^n((1+y^i)log(f(cw^tx^i)) + (1-y^i)log(1-f(cw^tx^i))$
   - taking the derivative with respect to c we see that $\frac{\partial\ell}{\partial c} = \frac{1}{2}\Sigma_{i=1}^n[\frac{y^i+1}{f(cw^tx^i)} + \frac{1-y^i}{1-f(cw^tx^i)}]f(cw^tx^i)(1-f(xw^tx^i))(w^tx^i) = \Sigma_{i=1}^n[\frac{y^n+1}{2} - f(cw^tx^i)]w^tx^i = \Sigma_{i=1}^n[\frac{y^n+1}{2} - \frac{1}{1+e^{-cw^tx^n}}]w^tx^i$
   - first off as we know the data is lineally separable if w is a separating hyperplane $w^tx > 0$ will always holds when $y^i = 1$ and $w^tx < 0$ will always hold when $y = -1$
   - this tells us that $\frac{\partial\ell}{\partial c} = \Sigma_{i=1}^n[\frac{y^n+1}{2} - \frac{1}{1+e^{-cw^tx^n}}]w^tx^i$ will always be increasing in c.
   - so we can keep geometrically the same hyperplane but arbitrarily increase our likelihood of seeing the data by increasing the constant on it c which multiples our weight vector w.

Regularized Logistic Regression

As we've shown in above, when the data is linearly separable, MLE for logistic regression may end up with weights with very large magnitudes. Such a function is prone to overfitting. In this part, we will apply regularization to fix the problem.

The ℓ_2 regularized logistic regression objective function can be defined as

$$\begin{aligned} J_{\text{logistic}}(w) &= \hat{R}_n(w) + \lambda\|w\|^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-y^{(i)}w^T x^{(i)}\right)\right) + \lambda\|w\|^2. \end{aligned}$$

4. Prove that the objective function $J_{\text{logistic}}(w)$ is convex. You may use any facts mentioned in the convex optimization notes
   - first off we know that the sum of two convex functions are convex.
   - so let us define $J(w) = f(w) + g(w)$ and prove that both $f(W)$ and $g(w)$ are convex separately
   - first lets look as $f(w) = \frac{1}{n}\Sigma_{i=1}^{n} log(1 + e^{-y^i w^t x^i}))$
     - working form inside out we know that e^x is convex over R
     - we know that the function h(x)=1 is convex thus $1 + e^{-y^i w^t x^i}$ is convex.
     - then we know $1 + e^{-y^i w^t x^i} \geq 1$ since clearly the the smallest value e^x can take is 0
     - further as we know log(x) is convex over $\mathbb{R} > 0$ it must be the case that $log(1 + e^{-y^i w^t x^i}))$ is convex
     - finally we know that scaling and summing convex functions makes them remain convex thus we have shown $f(w) = \frac{1}{n}\Sigma_{i=1}^{n} log(1 + e^{-y^i w^t x^i}))$ is convex
   - next lets look at $g(w) = \lambda||w||^2$
     - to show a function f is convex it must be the case that $\forall\theta \in [0,1]\forall x, y \in dom f$ we must have $f(\theta x + (1-\theta)y) \leq \theta(f(x)) + (1-\theta)(f(y))$
     - so consider and arbitrary $\theta \in [0,1], x, y \in$dom(f).
     - we can see that $g(\theta x + (1-)y) = \lambda||\theta x + (1-\theta)y||^2 \leq \lambda(||\theta x|| + ||(1-\theta)y||^2$ (by the triangle inequality) $= \lambda\theta||x||^2 + \lambda(1-\theta)||y||^2 = \theta g(x) + (1-\theta)g(y)$ proving that g(x) is indeed convex

   .

5. Complete the `f_objective` function in the skeleton code, which computes the objective function for $J_{\text{logistic}}(w)$. (Hint: you may get numerical overflow when computing the exponential literally, e.g. try e^{1000} in Numpy. Make sure to read about the log-sum-exp trick and use the numpy function *logaddexp* to get accurate calculations and to prevent overflow.
   - first note that in the case of our data $y \in \{0, 1\}$
   - recall from part 1, that this approach (ie using ERM) will yield the same sollution as MLE
   - thus we can work with the MLE objective of the from $j(w) = \frac{-1}{m}\Sigma_{i=1}^{n}(y^i)log(\frac{1}{1+e^{-w^t x^i}}) + (1-y^i)log(\frac{1}{1+e^{w^t x^i}}) + \lambda||w||^2$
   - we can further simplify this objective$j(w) = \frac{-1}{m}\Sigma_{i=1}^{n}(y^i)log(\frac{1}{1+e^{-w^t x^i}}) + (1-y^i)log(\frac{1}{1+e^{w^t x^i}}) + \lambda||w||^2 = \frac{-1}{m}\Sigma_{i=1}^{n}(y^i)[log(1) - log(1+e^{-w^t x^i})] + (1-y^i)[log(1) - log(1+e^{w^t x^i})) + \lambda||w||^2 = \Sigma_{i=1}^{n}\lambda||w||^2 - (y^i)[log(1+e^{-w^t x^i})] - (1-y^i)[log(1+e^{w^t x^i})]) = \Sigma_{i=1}^{n}\lambda||w||^2 - (y^i)[log(e^0 + e^{-w^t x^i})] - (1-y^i)[log(e^0 + e^{w^t x^i})])$

-

6. Complete the `fit_logistic_regression_function` in the skeleton code using the `minimize` function from `scipy.optimize`. Use this function to train a model on the provided data. Make sure to take the appropriate preprocessing steps, such as standardizing the data and adding a column for the bias term.

7. Find the ℓ_2 regularization parameter that maximizes the log-likelihood on the validation set. Plot the log-likelihood for different values of the regularization parameter.

   -
   - homework/homework_4/immages/q7_1.png
   -

- homework/homework_4/immages/q7_2.png
- homework/homework_4/immages/q7_3.png

- i found optimal $\lambda = 0.028571514285714 29$

8. [Optional] It seems reasonable to interpret the prediction $f(x) = \phi(w^T x) = 1/(1 + e^{-w^T x})$ as the probability that $y = 1$, for a randomly drawn pair (x, y). Since we only have a finite sample (and we are regularizing, which will bias things a bit) there is a question of how well "calibrated" our predicted probabilities are. Roughly speaking, we say $f(x)$ is well calibrated if we look at all examples (x, y) for which $f(x) \approx 0.7$ and we find that close to 70% of those examples have $y = 1$, as predicted... and then we repeat that for all predicted probabilities in $(0, 1)$. To see how well-calibrated our predicted probabilities are, break the predictions on the validation set into groups based on the predicted probability (you can play with the size of the groups to get a result you think is informative). For each group, examine the percentage of positive labels. You can make a table or graph. Summarize the results. You may get some ideas and references from scikit-learn's discussion.

    - homework/homework_4/immages/question_8_1.JPG
    -

homework/homework_4/immages/q_8_2output.png

-

2 Coin Flipping with Partial Observability

Consider flipping a biased coin where $p(z = H \mid \theta_1) = \theta_1$. However, we cannot directly observe the result z. Instead, someone reports the result to us, which we denotey by x. Further, there is a chance that the result is reported incorrectly *if it's a head*. Specifically, we have $p(x = H \mid z = H, \theta_2) = \theta_2$ and $p(x = T \mid z = T) = 1$.

9. Show that $p(x = H \mid \theta_1, \theta_2) = \theta_1\theta_2$.

   - and assume that all x_i are conditionally independent given z
   - we know that $z \in \{H, T\}$ so by the law of total probability $P(x = H|\theta_1, \theta_2) = P(x = H, z = H|\theta_1, \theta_2) + P(x = H, z = T|\theta_1, \theta_2) = P(z = H|\theta_1, \theta_2)P(X = h|z = H, \theta_1, \theta_2) + P(z = T|\theta_1, \theta_2)P(x = H|z = t, \theta_1, \theta_2)$
   - we know that $P(x = T|Z = t) = 1 \Rightarrow P(X = H|z = T) = 0$
   - further we know z does not depends on θ_2 so $P(z = h|\theta_1, \theta_2) = P(z = h|\theta_1)$
   - and finally we know that $P(x = 1|\theta_1, \theta_2, Z = H) = P(x = 1|\theta_2, z = h) = \theta_2$ since the value of z is already set
   - so thus we can see $P(x = H|\theta_1, \theta_2) = \theta_1\theta_2$

10. Given a set of reported results $\mathcal{D}_r$ of size N_r, where the number of heads is n_h and the number of tails is n_t, what is the likelihood of $\mathcal{D}_r$ as a function of θ_1 and θ_2.

- here i am again assuming that x_i is conditionally independent of z_i
- it is clear that $N_r = n_h + n_t$
- we can think of each reported coin flip z_i as a Bernoulli with parameters (θ_1, θ_2)
- we know that the sum of conditionally independent Bernoulli is binomial
- so we can say $\mathcal{L}(\theta_1, \theta_2, n_h) = \Pi_{i=1}^{n} P(x_i|\theta_1, \theta_2) = (P(x_i = h|\theta_1, \theta_2))^{n_h} (P(x_i = T|\theta_1, \theta_2))^{N_r - n_h} = (\theta_1\theta_2)^{n_h} (P(x_i = T, z_i = H|\theta_1, \theta_2) + P(x_i = T, z_i = T|\theta_1, \theta_2))^{N_r - n_h} = (\theta_1\theta_2)^{n_h} (P(z_i = H|_1, \theta_2) P(x_i = T|z_i = H\theta_1, \theta_2) + P(z_i = T|\theta_1, \theta_2) P(x_i = T|z_i = T\theta_1, \theta_2))^{N_r - n_h} = (\theta_1\theta_2)^{n_h} [(1 - \theta_1) + \theta_1(1 - \theta_2)]^{N_r - n_h} = (\theta_1\theta_2)^{n_h} (1 - \theta_1\theta_2)^{N_R - n_h}$

11. Can we estimate θ_1 and θ_2 using MLE? Explain your judgment.

    - i don't think we can estimate θ_1, θ_2 it using mle
    - we are only able to observe a binary outcome representing what we are told is the outcome but not the true outcome
    - so it is not possible for us to distinguish from our data set, what source of uncertainty comes from the coin flip θ_1 or from if we will be told θ_2
    - i think we could maximise the likelihood of $\theta_1\theta_2$