Homework 4: Probabilistic models

Due: Tuesday, March 30th, 2021 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 $\mathcal{D} = \left((x^{(1)}, y^{(1)}), \cdots, (x^{(n)}, y^{(n)})\right)$.

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.

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 $\left\{x \colon x^T w = 0\right\}$. Note that the set will not change if we multiply the weights by some constant c.

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.)

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.

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.

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 minimizes the log-likelihood on the validation set. Plot the log-likelihood for different values of the regularization parameter.

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.

Bayesian Logistic Regression with Gaussian Priors

Let's continue with logistic regression in the Bayesian setting, where we introduce a prior $p(w)$ on $w \in \mathbb{R}^d$.

9. For the same dataset $\mathcal{D}$ described at the beginning of the Section, give an expression for the posterior density $p(w \mid \mathcal{D})$ in terms of the negative log-likelihood function $\text{NLL}_{\mathcal{D}}(\text{w})$ and the prior density $p(w)$ (up to a proportionality constant is fine).

10. Suppose we take a prior on w of the form $w \sim \mathcal{N}(0, \Sigma)$, that is in the Gaussian family. Is this a conjugate prior to the likelihood given by logistic regression?

11. Show that there exist a covariance matrix Σ such that MAP (maximum a posteriori) estimate for w after observing data $\mathcal{D}$ is the same as the minimizer of the regularized logistic regression function defined in Regularized Logistic Regression paragraph above, and give its value. [Hint: Consider minimizing the negative log posterior of w. Also, remember you can drop any terms from the objective function that don't depend on w. You may freely use results of previous problems.]

12. In the Bayesian approach, the prior should reflect your beliefs about the parameters before seeing the data and, in particular, should be independent on the eventual size of your dataset. Imagine choosing a prior distribution $w \sim \mathcal{N}(0, I)$. For a dataset $\mathcal{D}$ of size n, how should you choose λ in our regularized logistic regression objective function so that the ERM is equal to the mode of the posterior distribution of w (i.e. is equal to the MAP estimator).

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 $p(x = T \mid z = T) = 1$.

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

14. 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.

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

16. We additionally obtained a set of clean results $\mathcal{D}_c$ of size N_c, where x is directly observed without the reporter in the middle. Given that there are c_h heads and c_t tails, estimate θ_1 and θ_2 by MLE taking the two data sets into account. Note that the likelihood is $L(\theta_1, \theta_2) = p(\mathcal{D}_r, \mathcal{D}_c \mid \theta_1, \theta_2)$.

17. Since the clean results are expensive, we only have a small number of those and we are worried that we may overfit the data. To mitigate overfitting we can use a prior distribution on θ_1 if available. Let's imagine that an oracle gave use the prior $p(\theta_1) = \text{Beta}(h, t)$. Derive the MAP estimates for θ_1 and θ_2.