

## Homework 5: SGD for Multiclass Linear SVM

**Due:** Wednesday, April 5, 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 Bayesian Modeling

#### Bayesian Logistic Regression with Gaussian Priors

This question analyzes logistic regression in the Bayesian setting, where we introduce a prior  $p(w)$  on  $w \in \mathbb{R}^d$ . Consider a binary classification setting with input space  $\mathcal{X} = \mathbb{R}^d$ , outcome space  $\mathcal{Y}_{\pm} = \{-1, 1\}$ , and a dataset  $\mathcal{D} = ((x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)}))$ .

1. Give an expression for the posterior density  $p(w | \mathcal{D})$  in terms of the negative log-likelihood function  $\text{NLL}_{\mathcal{D}}(w)$  and the prior density  $p(w)$  (up to a proportionality constant is fine).
2. 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?
3. Show that there exist a covariance matrix  $\Sigma$  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.]
4. 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  $\lambda$  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).

#### Coin Flipping with Partial Observability

**This is continuing your analysis done in HW4, you may use the results you obtained in HW4.**

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

- 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  $\theta_1$  and  $\theta_2$  by MLE taking the two data sets into account. Note that the likelihood is  $L(\theta_1, \theta_2) = p(\mathcal{D}_r, \mathcal{D}_c \mid \theta_1, \theta_2)$ .
- 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  $\theta_1$  if available. Let's imagine that an oracle gave use the prior  $p(\theta_1) = \text{Beta}(h, t)$ . Derive the MAP estimates for  $\theta_1$  and  $\theta_2$ .

## 2 Derivation for multi-class modeling

Suppose our output space and our action space are given as follows:  $\mathcal{Y} = \mathcal{A} = \{1, \dots, k\}$ . Given a non-negative class-sensitive loss function  $\Delta : \mathcal{Y} \times \mathcal{A} \rightarrow [0, \infty)$  and a class-sensitive feature mapping  $\Psi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$ . Our prediction function  $f : \mathcal{X} \rightarrow \mathcal{Y}$  is given by

$$f_w(x) = \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle.$$

For training data  $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ , let  $J(w)$  be the  $\ell_2$ -regularized empirical risk function for the multiclass hinge loss. We can write this as

$$J(w) = \lambda \|w\|^2 + \frac{1}{n} \sum_{i=1}^n \max_{y \in \mathcal{Y}} [\Delta(y_i, y) + \langle w, \Psi(x_i, y) - \Psi(x_i, y_i) \rangle]$$

for some  $\lambda > 0$ .

- Show that  $J(w)$  is a convex function of  $w$ . You may use any of the rules about convex functions described in our notes on Convex Optimization, in previous assignments, or in the Boyd and Vandenberghe book, though you should cite the general facts you are using. [Hint: If  $f_1, \dots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}$  are convex, then their pointwise maximum  $f(x) = \max \{f_1(x), \dots, f_m(x)\}$  is also convex.]
- Since  $J(w)$  is convex, it has a subgradient at every point. Give an expression for a subgradient of  $J(w)$ . You may use any standard results about subgradients, including the result from an earlier homework about subgradients of the pointwise maxima of functions. (Hint: It may be helpful to refer to  $\hat{y}_i = \arg \max_{y \in \mathcal{Y}} [\Delta(y_i, y) + \langle w, \Psi(x_i, y) - \Psi(x_i, y_i) \rangle]$ .)
- Give an expression for the stochastic subgradient based on the point  $(x_i, y_i)$ .
- Give an expression for a minibatch subgradient, based on the points  $(x_i, y_i), \dots, (x_{i+m-1}, y_{i+m-1})$ .

### (Optional) Hinge Loss is a Special Case of Generalized Hinge Loss

Let  $\mathcal{Y} = \{-1, 1\}$ . Let  $\Delta(y, \hat{y}) = \mathbb{1}_{y \neq \hat{y}}$ . If  $g(x)$  is the score function in our binary classification setting, then define our compatibility function as

$$\begin{aligned} h(x, 1) &= g(x)/2 \\ h(x, -1) &= -g(x)/2. \end{aligned}$$

- Show that for this choice of  $h$ , the multiclass hinge loss reduces to hinge loss:

$$\ell(h, (x, y)) = \max_{y' \in \mathcal{Y}} [\Delta(y, y') + h(x, y') - h(x, y)] = \max \{0, 1 - yg(x)\}$$

### 3 Implementation

In this problem we will work on a simple three-class classification example. The data is generated and plotted for you in the skeleton code.

#### One-vs-All (also known as One-vs-Rest)

First we will implement one-vs-all multiclass classification. Our approach will assume we have a binary base classifier that returns a score, and we will predict the class that has the highest score.

12. Complete the methods `fit`, `decision_function` and `predict` from `OneVsAllClassifier` in the skeleton code. Following the `OneVsAllClassifier` code is a cell that extracts the results of the fit and plots the decision region. You can have a look at it first to make sure you understand how the class will be used.
13. Include the results of the test cell in your submission.

#### Multiclass SVM

In this question, we will implement stochastic subgradient descent for the linear multiclass SVM, as described in class and in this problem set. We will use the class-sensitive feature mapping approach with the “multivector construction”, as described in the multiclass lecture.

14. Complete the function `featureMap` in the skeleton code.
15. Complete the function `sgd`.
16. Complete the methods `subgradient`, `decision_function` and `predict` from the class `MulticlassSVM`.
17. Following the multiclass SVM implementation, we have included another block of test code. Make sure to include the results from these tests in your assignment, along with your code.