CS 136 - 2022s - Checkpoint3 Submission

Student Names: Alexander Lobo and Nate Davis

Collaboration Statement:

Turning in this assignment indicates you have abided by the course Collaboration Policy:

`www.cs.tufts.edu/comp/136/2022s/index.html#collaboration-policy`

Total hours spent: 10

Contents

Recap of your dataset, your model, and the issues you hope to address with your upgrade

Our dataset contains 6,598 confirmations (orientations/rotations) of 102 molecules, all of which have been classified in terms of smell as musk or non-musk by human experts. If a molecule has been deemed a musk, all of its confirmations are listed so, and the same is true for non-musks. The features of our dataset comprise 166 measures of intramolecular distance.

We are using a logistic regression model with a sigmoid link function and multivariate Gaussian prior on the weight vector to predict whether a given confirmation is musk or non-musk. Our learning method is first-order stochastic gradient descent

The upgrades we propose are:

1. changing our learning method to second-order stochastic gradient descent and
2. changing our prior to a spike and slab prior.

The main problem we seek to address with these upgrades is the time it takes to train our model. Incorporating information from the second derivative of the loss function, as occurs in second-order gradient descent, should allow the model to converge more quickly. When looking at the model coefficients learned from our dataset, it appears that many lie close to zero while many others are orders of magnitude larger. We believe that the spike and slab prior, which tends to split weights into these two categories, might also help the model to converge more quickly.

Detailed description of your upgrade

Overview

The purpose of this study is to explore how our model performs when upgrading our learning method from a first order to a second order gradient descent algorithm to solve the weight vector MAP estimate for logistic regression. As stated previously in Checkpoint 2, considering the issues we had with training convergence and prediction accuracy when testing our model for this checkpoint, we have actually decided to expand our scope to two upgrades.

Upgrades:

1. Upgrade gradient descent algorithm to obtain the MAP estimate for Bayesian logistic regression from a first-order stochastic gradient descent to a second-second order stochastic gradient descent
2. Upgrade the prior distribution from a ”trivial” prior to a ”spike-and-slab” prior

Considering both of these upgrades, we propose the following hypotheses for how we expect our model to perform.

Hypotheses:

1. We hypothesize that the time and iterations needed to compute the weight vectors will decrease after upgrading from a first order to second order gradient descent.
2. We hypothesize that our multivariate logistic regression model with both first-order and second-order stochastic gradient descent will converge in fewer iterations when we use a spike-and-slab prior.
3. We hypothesize that our multivariate logistic regression model with second-order stochastic gradient descent will converge in fewer iterations, but with ultimately similar accuracy, when we use different step sizes for misclassifying the two output classes, as our output classes are highly imbalanced.
    - Since we were already able to test this hypothesis with first-order gradient descent with positive results, we expect the same to occur with second-order.

Upgrade 1: First to Second-Order Gradient Descent

Description

In Bayesian logistic regression, the Posterior distribution is a useful probability distribution that may be used to predict the probability of certain weight vectors given then occurrence of specific data. It is a powerful tool that allows us to utilize information from the Likelihood (probability of data given weight vector), the Prior (a priori knowledge about the weight vector), the Evidence (marginal probability of the data). One goal of Bayesian reasoning for logistic regression is to find the most probable weight vector that maximizes the probability density of the Posterior, which is known as the MAP estimate. By taking the negative of the Posterior, we can convert our optimization problem from a maximization to a minimization problem by convention. However, unlike with linear regression, the MAP estimate of logistic regression is not known to have an analytical solution. Therefore we depend on optimization methods like gradient descent to solve

$$\hat{\mathbf{w}}_{\text{MAP}} = \arg\min_{\mathbf{w}\in\mathbb{R}^M} -\ln p(\mathbf{w}|\mathbf{t}) \tag{1}$$

where M is feature dimension and $-\ln p(\mathbf{w}|\mathbf{t})$ is the negative of the posterior distribution.

We then have two options to choose from: first-order and second-order gradient descent. First order gradient descent uses a first-order derivative of the Posterior to make a step change in the weight vector, while second-order gradient descent uses a second-order derivative.

This upgrade falls under **Option 3** (Changing the inference method) out of those provided.

Relevant Equations

Probability Distributions

To recapitulate, The log posterior distribution is given by the following formula:

$$\ln p(\mathbf{w}|\mathbf{t}) = \underbrace{\sum_{n=1}^{N} \ln p(t_n|\mathbf{w})}_{\text{log Likelihood}} + \underbrace{\ln p(\mathbf{w}|\alpha)}_{\text{log Prior}} - \underbrace{\ln p(\mathbf{t})}_{\text{log Evidence}} \tag{2}$$

where the log Likelihood is given by

$$\begin{aligned}\sum_{n=1}^{N} \ln p(t_n|\mathbf{w}) &= \sum_{n=1}^{N} \ln \mathrm{Bern}(t_n|\sigma(\mathbf{w}^\top \phi(x_n)) &(3)\\ &= \sum_{n=1}^{N} \ln[y_n^{t_n}(1-y_n)^{1-t_n}] &(4)\\ &= \sum_{n=1}^{N} t_n \ln(y_n) + (1-t_n)\ln(1-y_n) &(5)\end{aligned}$$

and y_n is given by

$$\begin{aligned}y_n = p(\mathcal{C}_1|\phi(x_n)) &= \sigma(\mathbf{w}^\top \phi(x_n)) &(6)\\ &= \sigma(\mathbf{w}^\top x_n) &(7)\end{aligned}$$

since we are not using a kernel function. The log Prior (at the moment) is given by

$$\begin{aligned}\ln p(\mathbf{w}|\alpha) &= \ln \mathcal{N}(\mathbf{w}|\vec{0}, \alpha^{-1} I_{166}) &(8)\\ &= \ln\left[\frac{\alpha}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\alpha \mathbf{w}^\top \mathbf{w}\right\}\right] &(9)\\ &= \ln \alpha - \frac{1}{2}\ln 2\pi - \frac{1}{2}\alpha \mathbf{w}^\top \mathbf{w} &(10)\end{aligned}$$

Now, substituting equations (5) and (9) into (2), we see that the log Posterior is now given by

$$\ln p(\mathbf{w}|\mathbf{t}) = \sum_{n=1}^{N} t_n \ln(y_n) + (1-t_n)\ln(1-y_n) + \ln \alpha - \frac{1}{2}\ln 2\pi - \frac{1}{2}\alpha \mathbf{w}^\top \mathbf{w} - \ln p(\mathbf{t}) \quad (11)$$

First-Order Stochastic Gradient Descent

Now, since our objective is to find the MAP estimate for $\mathbf{w}$, we can write our loss equation as the following, grouping terms that are constant.

$$\ln \mathcal{L}(\mathbf{w}) = -\ln p(\mathbf{w}|\mathbf{t}) = -\left(\sum_{n=1}^{N} t_n \ln(y_n) + (1-t_n)\ln(1-y_n)\right) + \frac{1}{2}\alpha \mathbf{w}^\top \mathbf{w} + \cancel{constant} \tag{12}$$

which can be further simplified by removing terms that are constant with respect to $\mathbf{w}$ and replacing the terms for y_n given equation (7)

$$\ln \mathcal{L}(\mathbf{w}) = -\left(\sum_{n=1}^{N} t_n \ln(\sigma(\mathbf{w}^\top x_n)) + (1-t_n)\ln(1-\sigma(\mathbf{w}^\top x_n))\right) + \frac{1}{2}\alpha \mathbf{w}^\top \mathbf{w} \tag{13}$$

Now, with using Bishop equation (4.91), we can take the gradient of the log loss with respect to $\mathbf{w}$ to obtain

$$\nabla_{\mathbf{w}} \ln \mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \sigma(\mathbf{w}^\top x_n) - t_n)x_n + \alpha \mathbf{w} \tag{14}$$

Using first-order gradient descent descent, we can now use the gradient to make a step change given some step-size η and iterate until we converge to a solution.

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta \left[\sum_{n=1}^{N} \sigma(\mathbf{w}^\top x_n) - t_n)x_n + \alpha \mathbf{w}^t\right] \tag{15}$$

However, since we are interested in conducting stochastic gradient descent, each step update will use a single paired example $(x^{(i)}, t^{(i)})$ to make an update:

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta \left[\left(\sigma(\mathbf{w}^t \cdot x^{(i)}) - y^{(i)}\right) x^{(i)} + \alpha \mathbf{w}^t\right] \tag{16}$$

$$c^{t+1} \leftarrow c^t - \eta(\sigma(\mathbf{w}^t \cdot x^{(i)}) - y^{(i)}) \tag{17}$$

After each update, the log loss given by equation (14) is computed to measure if the loss continues to decrease as wight updates are made.

Second-Order Stochastic Gradient Descent

In order to perform second-order stochastic gradient descent, we must take the second derivative of the log loss function in order to obtain the Hessian matrix. We can consider the first-order gradient vector $g(\mathbf{w}) \in \mathbb{R}^M$ to be a function of $\mathbf{w}$

$$\begin{aligned} g(\mathbf{w}) = \nabla_{\mathbf{w}} \ln \mathcal{L} &= \sum_{n=1}^{N} (\sigma(\mathbf{w}^\top x_n) - t_n) x_n + \alpha \mathbf{w} \qquad (18) \\ &= \mathbf{x}^\top (\sigma(\mathbf{x}\mathbf{w}) - t_n) + \alpha \mathbf{w} \qquad (19) \end{aligned}$$

Then, we can consider the second-order gradient Hessian matrix $H(\mathbf{w}) \in \mathbb{R}^{M \times M}$ to be

$$H(\mathbf{w}) = \nabla_{\mathbf{w}} \nabla_{\mathbf{w}} = \mathbf{x}^\top R(\mathbf{w}) \mathbf{x} + \alpha I_M \qquad (20)$$

where $R(\mathbf{w}) \in \mathbb{R}^{N \times N}$ is defined as

$$R(\mathbf{w}) = \begin{bmatrix} \sigma(\mathbf{w}^\top x_n)\sigma(-\mathbf{w}^\top x_n) & & \\ & \ddots & \\ & & \sigma(\mathbf{w}^\top x_n)\sigma(-\mathbf{w}^\top x_n) \end{bmatrix} \qquad (21)$$

a diagonal matrix with all 0 off-diagonal entries. Now each update in the weight vector is

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta H(\mathbf{w}^t)^{-1} g(\mathbf{w}^t) \qquad (22)$$

But, once again, we are interested in stochastic gradient descent, so the weight update will be updated using a single paired example $(x^{(i)}, t^{(i)})$ to make an update:

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta \underbrace{\left[r_{(i)}(\mathbf{w}^{\mathbf{t}})(x_{(i)} \otimes x_{(i)}) + \alpha\right]^{-1}}_{H_{(i)}(\mathbf{w}^t)^{-1}} \underbrace{\left[\left(\sigma(\mathbf{w}^t \cdot x^{(i)}) - y^{(i)}\right) x^{(i)} + \alpha \mathbf{w}^t\right]}_{g_{(i)}(\mathbf{w}^t)} \qquad (23)$$

where $r_{(i)}(\mathbf{w})$ is the i-th diagonal element in $R(\mathbf{w})$:

$$r_{(i)}(\mathbf{w}) = \sigma(\mathbf{w}^t \cdot x_{(i)})\sigma(-\mathbf{w}^t \cdot x_{(i)}) \tag{24}$$

Implementation

To implement this upgrade, we will need to make the following changes to our model:

1. Add an attribute to our initialization to specify the gradient descent method that should be used (e.g. `solver=`$\{$`'fo','so'`$\}$)
2. Create an `if` statement in the `fit()` method to update the weights accordingly
3. Use the second-order gradient descent formula and implement it in the `fit()` method

At the moment, our first-order stochastic gradient descent algorithm checks the log loss function defined in literature for convergence. Instead, for both first and second-order, we will need to make sure that we evaluate equation (13) as our loss function.

Benefit(s)

It is widely accepted –and stated in the lecture notes– that second-order gradient descent is the ”gold standard form MAP estimation”. However, unlike linear regression it is not able to converge to the optimal solution in a single step. Regardless, it is expected that that implementing a second order gradient descent for logistic regression MAP estimation greatly reduces the number of iterations needed to converge.

We think that this will not only help reduce the high number of iterations observed with first-order SGD, but also possibly help improve the prediction accuracy as well. First-order SGD is notorious for having high variance loss convergence, making it easier to prematurely end the convergence loop or to overshoot the posterior maxima.

Upgrade 2: Changing Prior

Description

A typical prior used in logistic regression is a 0 mean prior with a small precision. The spike-and-slab prior [1], on the other hand, is a Gaussian mixture model (GMM) is used to combine two MVNs as a single prior. In this case, both MVNs will be 0 mean, but one with high precision, and the other with low precision. The MVN with lower precision is allocated a larger weight ($\pi_1 = 0.8$), while the other is allocated a lower weight ($\pi_2 = 0.2$).

This upgrade falls under **Option 1** (Changing the prior) out of those provided.

Relevant Equations

Probability Distributions

The spike and slab approach changes our log prior as follows,

$$\ln p(\mathbf{w}|\alpha_1, \alpha_2, \pi_1, \pi_2) = \ln(\pi_1 \mathcal{N}(\mathbf{w}|\vec{0}, \alpha_1^{-1} I_{166}) + \pi_2 \mathcal{N}(\mathbf{w}|\vec{0}, \alpha_2^{-1} I_{166})), \qquad (25)$$

where $\pi_1 + \pi_2 = 1$.

Implementation

To implement this upgrade, we will need to make the following changes to our model:

- Add an attribute to our initialization to specify the prior that should be used (e.g. `prior=`{`'trivial'`,`'sas'`})
- Create an `if` statement in the `fit()` method to update the weights accordingly
- Enable use of autograd for first- and second-order gradient descent in the `fit()` method

As mentioned above, since we periodically evaluate the loss associated with the model while fitting it, we will need to make sure we do so using equation (25).

Benefit(s)

We expect that, with 166 features, some features should conceivably have little to no weight while others will bear the brunt of prediction. This prediction is gleaned from our data exploration in Checkpoint 1, where we observed that many features were highly positively

[1] `https://en.wikipedia.org/wiki/Spike-and-slab_regression`

and negatively correlated with each other. Hence, we expect that a spike-and-slab prior will be able to randomly explore higher magnitude weights for some features, while assuming most to be close to 0 mean. It is also claimed that ”in statistics, spike-and-slab regression is a Bayesian variable selection technique that is particularly useful when the number of possible predictors is larger than the number of observations”. Although it does not seem that we have more predictors (features) than we do examples, it should be noted that many examples in our dataset are replicate molecules with different confirmations. Therefore, by number of unique molecules, we really only have 102 examples, which means that the benefit quoted may apply to our dataset. At the moment, our model with a trivial prior is converging on a weight vector solution that is yielding poor prediction accuracy. Perhaps a spike-and-slab prior will help address this problem.

Questions/Issues you want advice on

We believe we have already addressed all questions we have in office hours. Thank you for the help!