Homework 1

1. Poisson Regression

In this question we will construct another kind of a commonly used GLM, which is called Poisson Regression. In a GLM, the choice of the exponential family distribution is based on the kind of problem at hand. If we are solving a classification problem, then we use an exponential family distribution with support over discrete classes (such as Bernoulli or Categorical). Similarly, if the output is real valued, we can use Gaussian or Laplace (both are in the exponential family). Sometimes the desired output is to predict counts, for e.g., predicting the number of emails expected in a day, or the number of customers expected to enter a store in the next hour, etc. based on input features (also called covariates). You may recall that a probability distribution with support over integers (i.e. counts) is the Poisson distribution, and it also happens to be in the exponential family.

In the following sub-problems, we will start by showing that the Poisson distribution is in the exponential family, derive the functional form of the hypothesis, derive the update rules for training models, and finally using the provided dataset train a real model and make predictions on the test set.

(a)

Consider the Poisson distribution parameterized by λ:

$$p(y; \lambda) = \frac{e^{-\lambda}\lambda^y}{y!}.$$

(Here y has positive integer values and $y!$ is the factorial of y.) Show that the Poisson distribution is in the exponential family, and clearly state the values for $b(y)$, η, $T(y)$, and $a(\eta)$.

(b)

Consider performing regression using a GLM model with a Poisson response variable. What is the canonical response function for the family? (You may use the fact that a Poisson random variable with parameter λ has mean λ.)

(c)

For a training set $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, n\}$, let the log-likelihood of an example be $\log p(y^{(i)}|x^{(i)}; \theta)$. By taking the derivative of the log-likelihood with respect to θ_j, derive the stochastic gradient descent update rule when using a GLM model with Poisson responses y and the canonical response function.

2. Convexity of Generalized Linear Models

In this question we will explore and show some key properties of Generalized Linear Models, specifically those related to the use of Exponential Family Distributions to model the output.

Most commonly, GLMs are trained by using the negative log-likelihood (NLL) as the loss function. This is broadly equivalent to Maximum Likelihood Estimation (i.e., maximizing the likelihood is the same as minimizing the negative log-likelihood). In this problem, our goal is to show that the NLL loss of a GLM is a convex function w.r.t the model parameters. As a reminder, this is convenient because a convex function is one for which any local minimum is also a global minimum, and there is extensive research on how to optimize various convex loss functions efficiently with various algorithms such as gradient descent or stochastic gradient descent.

To recap, an exponential family distribution is one whose probability density can be represented

$$p(y;\eta) = b(y)\exp\left(\eta^T T(y) - a(\eta)\right),$$

where η is the natural parameter of the distribution. Moreover, in a Generalized Linear Model, η is modeled as $\theta^T x$, where $x \in \mathbb{R}^d$ are the input features of the example, and $\theta \in \mathbb{R}^d$ are learnable parameters. In order to show that the NLL loss is convex for GLMs, we break down the process into sub-parts, and approach them one at a time. Our approach is to show that the second derivative (i.e., Hessian) of the loss w.r.t. the model parameters is Positive Semi-Definite (PSD) at all values of the model parameters. We will also show some nice properties of Exponential Family distributions as intermediate steps.

For the sake of convenience we restrict ourselves to the case where η is a scalar. Assume

$$p(Y|X;\theta) \sim \text{ExponentialFamily}(\eta),$$

where $\eta \in \mathbb{R}$ is a scalar, and $T(y) = y$. This makes the exponential family representation take the form

$$p(y;\eta) = b(y)\exp\left(\eta y - a(\eta)\right).$$

(a)

Derive an expression for the mean of the distribution. Show that $\mathbb{E}[Y;\eta] = \frac{\partial}{\partial\eta}a(\eta)$ (note that $\mathbb{E}[Y;\eta] = \mathbb{E}[Y|X;\theta]$ since $\eta = \theta^T x$). In other words, show that the mean of an exponential family distribution is the first derivative of the log-partition function with respect to the natural parameter.

Hint: Start with observing that

$$\frac{\partial}{\partial\eta}\int p(y;\eta)dy = \int \frac{\partial}{\partial\eta}p(y;\eta)dy.$$

(b)

Next, derive an expression for the variance of the distribution. In particular, show that $\text{Var}(Y) = \frac{\partial^2 a(\eta)}{\partial\eta^2}$ (again, note that $\text{Var}(Y;\eta) = \text{Var}(Y|X;\theta)$). In other words, show that the variance of an exponential family distribution is the second derivative of the log-partition function w.r.t. the natural parameter.

Hint: Building upon the result in the previous sub-problem can simplify the derivation.

(c)

Finally, write out the loss function $\ell(\theta)$, the NLL of the distribution, as a function of θ. Then, calculate the Hessian of the loss w.r.t θ, and show that it is always PSD. This concludes the proof that NLL loss of GLM is convex.

Hint1: Use the chain rule of calculus along with the results of the previous parts to simplify your derivations.

Hint2: Recall that variance of any probability distribution is non-negative.

3. Multivariate Least Squares

So far in class, we have only considered cases where our target variable y is a scalar value. Suppose that instead of trying to predict a single output, we have a training set with multiple outputs for each example:

$$\{(x^{(i)}, y^{(i)}), i = 1, \ldots, m\},\ x^{(i)} \in \mathbb{R}^n,\ y^{(i)} \in \mathbb{R}^p.$$

Thus for each training example, $y^{(i)}$ is vector-valued, with p entries. We wish to use a linear model to predict the outputs, as in least squares, by specifying the parameter matrix Θ in

$$y = \Theta^T x,$$

where $\Theta \in \mathbb{R}^{n\times p}$.

(a)

The cost function for this case is

$$J(\Theta) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{p} \left((\Theta^T x^{(i)})_j - y_j^{(i)} \right)^2 .$$

Write $J(\Theta)$ in matrix-vector notation (i.e., without using any summations). [**Hint**: Start with the $m \times n$ design matrix

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix}$$

and the $m \times p$ target matrix

$$Y = \begin{bmatrix} (y^{(1)})^T \\ (y^{(2)})^T \\ \vdots \\ (y^{(m)})^T \end{bmatrix}$$

and then work out how to express $J(\Theta)$ in terms of these matrices.]

(b)

Find the closed form solution for Θ which minimizes $J(\Theta)$. This is the equivalent to the normal equations for the multivariate case.

(c)

Suppose instead of considering the multivariate vectors $y^{(i)}$ all at once, we instead compute each variable $y_j^{(i)}$ separately for each $j = 1, \ldots, p$. In this case, we have p individual linear models, of the form

$$y_j^{(i)} = \theta_j^T x^{(i)}, \; j = 1, \ldots, p.$$

(So here, each $\theta_j \in \mathbb{R}^n$.) How do the parameters from these p independent least squares problems compare to the multivariate solution?

4. Incomplete, Positive-Only Labels

In this problem we will consider training binary classifiers in situations where we do not have full access to the labels. In particular, we consider a scenario, which is not too infrequent in real life, where we have labels only for a subset of the positive examples. All the negative examples and the rest of the positive examples are unlabelled.

We formalize the scenario as follows. Let $\{(x^{(i)}, t^{(i)})\}_{i=1}^n$ be a standard dataset of i.i.d distributed examples. Here $x^{(i)}$'s are the inputs/features and $t^{(i)}$'s are the labels. Now consider the situation where $t^{(i)}$'s are not observed by us. Instead, we only observe the labels of some of the positive examples. Concretely, we assume that we observe $y^{(i)}$'s that are generated by

$$\forall x, \quad p(y^{(i)} = 1 \mid t^{(i)} = 1, x^{(i)} = x) = \alpha,$$

$$\forall x, \quad p(y^{(i)} = 0 \mid t^{(i)} = 1, x^{(i)} = x) = 1 - \alpha,$$

$$\forall x, \quad p(y^{(i)} = 1 \mid t^{(i)} = 0, x^{(i)} = x) = 0,$$

$$\forall x, \quad p(y^{(i)} = 0 \mid t^{(i)} = 0, x^{(i)} = x) = 1,$$

where $\alpha \in (0, 1)$ is some unknown scalar. In other words, if the unobserved "true" label $t^{(i)}$ is 1, then with α chance we observe a label $y^{(i)} = 1$. On the other hand, if the unobserved "true" label $t^{(i)} = 0$, then we always observe the label $y^{(i)} = 0$.

Our final goal in the problem is to construct a binary classifier h of the true label t, with only access to the partial label y. In other words, we want to construct h such that $h(x^{(i)}) \approx p(t^{(i)} = 1 \mid x^{(i)})$ as closely as possible, using only x and y.

In the following sub-questions we will attempt to solve the problem with only partial observations. That is, we only have access to $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, and will try to predict $p(t^{(i)} = 1 \mid x^{(i)})$.

(a) Warm-up with Bayes rule

Show that under our assumptions, for any i,

$$p(t^{(i)} = 1 \mid y^{(i)} = 1, x^{(i)}) = 1$$

That is, observing a positive partial label $y^{(i)} = 1$ tells us for sure the hidden true label is 1. Use Bayes rule to derive this (an informal explanation will not earn credit).

(b)

Show that for any example, the probability that the true label $t^{(i)}$ is positive is $1/\alpha$ times the probability that the partial label is positive. That is, show that

$$p(t^{(i)} = 1 \mid x^{(i)}) = \frac{1}{\alpha} \cdot p(y^{(i)} = 1 \mid x^{(i)})$$

Note that the equation above suggests that if we know the value of α, then we can convert a function $h(\cdot)$ that approximately predicts the probability $h(x^{(i)}) \approx p(y^{(i)} = 1 \mid x^{(i)})$ into a function that approximately predicts $p(t^{(i)} = 1 \mid x^{(i)})$ by multiplying the factor $1/\alpha$.

(c) Estimating α

The solution to estimate $p(t^{(i)} \mid x^{(i)})$ outlined in the previous sub-question requires the knowledge of α which we don't have. Now we will design a way to estimate α based on the function $h(\cdot)$ that approximately predicts $p(y^{(i)} = 1 \mid x^{(i)})$.

To simplify the analysis, let's assume that we have magically obtained a function $h(x)$ that perfectly predicts the value of $p(y^{(i)} = 1 \mid x^{(i)})$, that is, $h(x^{(i)}) = p(y^{(i)} = 1 \mid x^{(i)})$.

We make the crucial assumption that $p(t^{(i)} = 1 \mid x^{(i)}) \in \{0, 1\}$. This assumption means that the process of generating the "true" label $t^{(i)}$ is a noise-free process. This assumption is not very unreasonable to make. Note, we are NOT assuming that the observed label $y^{(i)}$ is noise-free, which would be an unreasonable assumption!

Now we will show that:

$$\alpha = \mathbb{E}[h(x^{(i)}) \mid y^{(i)} = 1]$$

To show this, prove that $h(x^{(i)}) = \alpha$ when $y^{(i)} = 1$, and $h(x^{(i)}) = 0$ when $y^{(i)} = 0$.

(The above result motivates the following algorithm to estimate α by estimating the RHS of the equation above using samples: Let V_+ be the set of labeled (and hence positive) examples in the validation set V, given by $V_+ = \{x^{(i)} \in V \mid y^{(i)} = 1\}$. Then we use $\alpha \approx \frac{1}{|V_+|} \sum_{x^{(i)} \in V_+} h(x^{(i)})$.)