Math 7243 Machine Learning - Homework 3

For programming questions, you can only use numpy library.

Question 1. Softmax regression Recall the setup of logistic regression: We assume that the posterior probability is of the form

$$p(Y = 1|\vec{x}) = \frac{1}{1 + e^{-\beta^T \vec{x}}}$$

This assumes that $Y|X$ is a Bernoulli random variable. We now turn to the case where $Y|X$ is a multinomial random variable over K outcomes. This is called softmax regression, because the posterior probability is of the form

$$p(Y = k|\vec{x}) = \frac{e^{\beta_k^T \vec{x}}}{\sum_{j=1}^{K} e^{\beta_j^T \vec{x}}}$$

which is called the softmax function. Assume we have observed data $D = \{\vec{x}^{(i)}, y^{(i)}\}_{i=1}^N$. Our goal is to learn the weight $\beta_1, ..., \beta_K$.

(1) Find the negative log likelihood of the data $l(\beta_1, ..., \beta_K) = -\log L(\beta_1, ..., \beta_K) = -\log P(Y|X)$

(2) We want to minimize the negative log likelihood. To combat overfitting, we put a regularizer on the objective function. Find the **gradient** w.r.t. β_k of the regularized objective

$$l(\beta_1, ..., \beta_K) + \lambda \sum_{k=1}^{K} ||\beta_k||^2$$

(3) State the gradient updates for both batch gradient descent and stochastic gradient descent.

Question 2. - Linear Discriminant Analysis: Consider the categorical learning problem consisting of a data set with two labels:

Label 1:

X_1	3.81	0.23	3.05	0.68	2.67
X_2	-0.55	3.37	3.53	1.84	2.74

Label 2:

X_1	-2.04	-0.72	-2.46	-3.51	-2.05
X_2	-1.25	-3.35	-1.31	0.13	-2.82

a) For each label above, the data follow a multivariate normal distribution Normal(μ_i, Σ) where the covariance Σ is the same for both label 1 and for label 2. Fit a pair of Guassian discriminant functions to the labels by computing the covariances, means, and proportions of datapoints as discussed in the Linear Discriminant Analysis section. You may use a computer, but you should **not** use an LDA solver. You should report the values for μ_i and Σ.

b) Give the **formula for the line** forming the discretion boundary.

c) (bonus) Try the **QDA** method for this question and obtain an quadratic boundary.

Question 3. later

Question 4. later