# Assigment 2

EECS 4404/5327

**The assignment is due on Thursday, March 22, before 10:30am (before class).**
Place your submission in the dropbox Lassonde building, ground floor.

1. **Gradient computation**
   Let $\mathbf{X} \in \mathbb{R}^{d \times d}$ be a symmetric matrix. Consider the function $f : \mathbb{R}^d \to \mathbb{R}$ defined by

   $$f(\mathbf{w}) = \mathbf{w}^T \mathbf{X} \mathbf{w}$$

   Derive its gradient with respect to $\mathbf{w}$.

   **(10 marks)**

2. **Stochastic Gradient Descent**
   Recall the logistic loss function, pointwise defined as

   $$\ell^{\text{logist}}(y_{\mathbf{w}}, (\mathbf{x}, t)) = \ln(1 + \exp(-t \langle \mathbf{w}, \mathbf{x} \rangle))$$

   We know that the empirical logistic loss over a dataset

   $$\mathcal{L}_D^{\text{logist}}(y_{\mathbf{w}}) = \frac{1}{N} \sum_{n=1}^{N} \ln(1 + \exp(-t_n \langle \mathbf{w}, \mathbf{x}_n \rangle))$$

   is a convex function in $\mathbf{w}$. We will now consider running stochastic gradient descent to minimize the logistic loss over a data set.

   (a) Compute the gradient with respect to $\mathbf{w}$ of the logistic loss $\ell^{\text{logist}}(y_{\mathbf{w}}, (\mathbf{x}, t))$ on a data point $(\mathbf{x}, t)$.

   (b) Describe Stochastic Gradient Descent with a fixed stepsize $\eta$ with respect to the logistic loss

   (c) Compare the derived algorithm to the SGD algorithm we derived for SVM. What is similar? What is different? On what type of data would you expect to see different behavior?

   **(5 + 8 + 7 marks)**

3. **SoftSVM optimization**

In this question you will implement the SGD optimization method for SoftSVM to find a linear with minimal empirical loss. The first dataset is stored in the file, `bg.txt`. The file contains only the feature vectors. There is one feature vector per line. The first 100 lines correspond to positive instances (label +1) and the next 100 lines are negative instances (label −1). The data can be loaded into a matrix with the load function in Matlab/Octave.

(a) Implement SGD for SoftSVM. You may skip the averaging over weight vectors for the output and instead, simply output the last iterate. During the optimization, keep track of both the empirical and the hinge loss of the current weight vector. Include a printout of your code.

**(2 marks )**

(b) Run the optimization method with various values for the regularization parameter $\lambda = \{100, 10, 1, .1, .01, .001\}$ on the data (remember to first add an additional feature with value 1 to each datapoint, so that you are actually training a general linear classifier). For each value plot the binary loss of the iterates and the hinge loss of the iterates (in separate plots). Include three plots of your choice where you observe distinct behavior (you may run the method several times for each parameter setting and choose).

**(4 marks )**

(c) Discuss the plots. Are the curves monotone? Are they approximately monotone? Why or why not? How does the choice of $\lambda$ affect the optimization? How would you go about finding a linear predictor of minimal binary loss?

**(4 marks )**

(d) Download the "seeds" data set from the UCI repository:
`https://archive.ics.uci.edu/ml/datasets/seeds`.
That data is also stored in text file and can be loaded the same way. It contains 210 instances, with three different label (the last column in the file corresponds to the label).

(e) Train three binary linear predictors. $w_1$ should separate the first class from the other two (ie the first 70 instances are labeled +1 and the next 140 instances −1). $w_2$ should separate the second class from the other two and $w_3$ should separate the third class from the first two classes (ie for training $w_2$ label the middle 70 instances positive and the rest negative and analogously for $w_3$). Report the binary loss that you achieve with $w_1$, $w_2$ and $w_3$ for each of these tasks.

**(5 marks )**

(f) Turn the three linear separators into a multi-class predictor for the three different classes in the seeds dataset using the following rule:

$$y(\mathbf{x}) = \text{argmax}_{i \in \{1,2,3\}} \langle \mathbf{w}_i, \mathbf{x} \rangle$$

(See also UML Chapter 17, page 190-191.) Compute and report the loss on the original labels using this rule.

Store the three weight vectors in a text file, $w_1$ on the first line, $w_2$ one the second and $w_3$ on the third. Make sure that the last entry for each of these corresponds to the bias. Submit the file by email to `ruth@eecs.yorku.ca`. For this last part, you may also choose a different method to find the three linear separators that give the best multi-class prediction. Report what you did and what the multi-class error of your submitted vectors is.

**(5 marks )**

IF ANYTHING IS UNCLEAR, COME TO MY OFFICE HOURS OR WRITE ME EMAILS AND ASK QUESTIONS :)