CSC 411
Machine Learning & Data Mining
ASSIGNMENT # 3
Out: 19th November
Due: 3rd December

Please note that if you are asked to report/compute quantities these should be clearly displayed in your written report. It is not sufficient to simply print these as an output of your code. The same applies to plots and figures.

1 20 Newsgroups predictions (6%)

In this question you will work with the 20 newsgroups dataset. This dataset contains posts from 20 different newsgroups and the task is to classify each post to the correct newsgroup. Your goal will be to evaluate several models of your own choice and select those with the best test performance on this task.

We provide code that takes the original text and converts it into a bag-of-words counts representation. We also provide code that converts it to $tf - idf$ features, these features reweigh words according to frequency (so a common uninformative word like "the" would get a small weight) you can read about these features here `http://www.tfidf.com/`. You can work with either feature representation.

We also provide starter code to train and evaluate a very simple Bernoulli Naive-Bayes classifier as the baseline model. You should implement better models for this task and outperform the baseline.

- You may use any open-source code you want, as long as the algorithm was covered in class.
- You need to report the results of three different algorithms (you are encouraged to try out more and pick 3 out of them) and the baseline.
- Each result reported should include train and test loss.
- **You are to report evaluations on the test loss exactly 4 times. Once for each of your proposed model and a final time for the baseline.**
- Explain in your report how you picked the best hyperparameters.
- Explain why you picked these 3 methods, did they work as you thought? Why/Why not?
- For your best classifier compute and show the confusion matrix, which is a $k \times k$ matrix where C_{ij} is the number of test examples belonging to class j that were classified as i. This part you need to write yourselves.
- What were the two classes your classifier was most confused about?

2 Training SVM with SGD (5%)

In this question you will implement Stochastic Gradient Descent with momentum β and learning rate α. You will then use your implemented algorithm to approximately optimize the SVM objective.

2.1 SGD With Momentum

1. Implement SGD with momentum.
2. To verify your implementation find the minimum of $f(w) = 0.01w^2$ using gradient descent.

   Take $w_0 = 10.0$ and set your learning rate to $\alpha = 1.0$. Plot w_t for 200 time-steps using $\beta = 0.0$ and $\beta = 0.9$ on the same graph.

2.2 Training SVM

We denote the parameters of the SVM model by $\mathbf{w}$ and consider a dataset made up of N pairs $(\mathbf{x}^{(i)}, y^{(i)})$ where $\mathbf{x}$ are continuous features and $y \in \{\pm 1\}$ are the binary class labels.

The goal is to find

$$\mathbf{w}^*, b^* = \arg\min \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N}\sum_{i=1}^{N} \max(1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b), 0)$$

Write a function that takes as input:

- Data $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)} \in \mathbb{R}^d$ and $y^{(1)}, \ldots, y^{(N)} \in \{\pm 1\}$.
- Penalty parameter C of the error term
- Mini-batch size m, constant learning rate α and number of iterations T.

The function will perform T steps of SGD with batch size of m, learning rate α, and momentum β. The output will be the final weights $\mathbf{w}_T$.

A few notes:

- For gradient computation, you can take the gradient at the "kink" $y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)} = 1$ to be zero.
- You may include the bias implicitly by adding a column of ones to your data matrix. But remember not to regularize the bias weight!
- You should sample the mini-batches without replacement.
- Finally note that momentum may not have a significant effect on SVM training.

2.3 Apply on 4-vs-9 digits on MNIST

MNIST is a digit classification dataset consisting of 28×28 grayscale images of hand-drawn digits labelled from 0 to 9.

We turn it into a binary classification problem by trying to classify 4 vs 9 (the hardest 1-vs-1 pair) and discarding the rest of the classes. The remaining images are separated into 80% train and 20% test. We convert the images to vectors by flattening them to a vector of size 784.

Train two SVM models using gradient descent with a learning rate of $\alpha = 0.05$, a penalty of $C = 1.0$, minibatch sizes of $m = 100$, and $T = 500$ total iterations. For the first model use $\beta = 0$ and for the second use $\beta = 0.1$. For both of the trained models report the following:

1. The training loss.
2. The test loss.
3. The classification accuracy on the training set.
4. The classification accuracy on the test set.
5. Plot $\mathbf{w}$ as a 28×28 image.

We chose these hyperparameters as they seemed to work well ($\sim 90\%$ accuracy) - you might want to try cross validation to see how big of an improvement you can get!

3 Kernels (4%)

In this question you will prove some properties of kernel functions. The two main ways to show a function $k(\mathbf{x}, \mathbf{y})$ is a kernel function is to find an embedding $\phi(x)$ such that $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$ or to show the for all $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ the gram matrix $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is positive semi-definite (i.e. symmetric and no negative eigenvalues).

3.1 Positive semidefinite and quadratic form

1. Prove that a symmetric matrix $K \in \mathbb{R}^{d \times d}$ is positive semidefinite iff for all vectors $\mathbf{x} \in \mathbb{R}^d$ we have $\mathbf{x}^T K \mathbf{x} \geq 0$.

3.2 Kernel properties

Prove the following properties:

1. The function $k(\mathbf{x}, \mathbf{y}) = \alpha$ is a kernel for $\alpha > 0$.
2. $k(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) \cdot f(\mathbf{y})$ is a kernel for all $f : \mathbb{R}^d \to \mathbb{R}$.

3. If $k_1(\mathbf{x}, \mathbf{y})$ and $k_2(\mathbf{x}, \mathbf{y})$ are kernels then $k(\mathbf{x}, \mathbf{y}) = a \cdot k_1(\mathbf{x}, \mathbf{y}) + b \cdot k_2(\mathbf{x}, \mathbf{y})$ for $a, b > 0$ is a kernel.
4. If $k_1(\mathbf{x}, \mathbf{y})$is a kernel then $k(\mathbf{x}, \mathbf{y}) = \frac{k_1(\mathbf{x},\mathbf{y})}{\sqrt{k_1(\mathbf{x},\mathbf{x})}\sqrt{k_1(\mathbf{y},\mathbf{y})}}$ is a kernel (hint: use the features ϕ such that $k_1(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$).

Submission

Assignments must be submitted by **10:00 pm on December 3rd**; a late penalty of 10% per day will be assessed thereafter (up to 3 days, then submission is blocked). We highly recommend using Python 3.6 embedded in Anaconda 3.4 to solve the programming questions. However, you are welcome to use any programming language you like, given that your code solves the problem, and is functional and modular.

What to submit

- All work to be submitted via Markus.
- All mathematical work has to be clearly marked by the question it belongs to, we highly recommend using a math editor such as MathType, Word Equations or LaTex. However, if you prefer to use paper and pen(cil), you are welcome to do so as long as your hand writing is readable. Unreadable work will result in losing the full mark of the question.
- Your written report should be submitted in PDF format. Note that Markus has filesize limits and it is your responsibility to ensure that your work is uploaded on time.
- Your code must be clearly structured with an obvious entry point.
- All graphs and plots have to be clearly marked by their question. Use readable grids whenever applicable.