

Assignment 3

*Due By: 30th March 2016**Due By: 30th March 2016*

This assignment is worth total 25 points. Your assignment is due by 12:00pm, 30th March, 2016 either by email or in class.

Some Logistics

1. You are encouraged to discuss but you should write your own solutions in detail.
2. You will get 5 bonus points if you implement the kernel features computation (step 3) part on GPUs and show significant speedups.

1 Compare Kernel Features and Kernel SVMs (25 points)**Start Early, this may take some time**

Download LibSVM software package for running kernel SVM (<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>) and LIBLINEAR for running linear SVM <https://www.csie.ntu.edu.tw/~cjlin/liblinear/>. Install them (there are plenty of language choices, if you cannot find the one you prefer let the instructor know about it.) Play with them, make sure you can run some demos with them.

1. Download the MNIST dataset from the link ([\https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#mnist](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#mnist)). Use the train-test partition provided. It should be 60k for training and 10k for testing.
2. Run the Kernel SVM with radial basis kernel (RBF) $\exp(-\text{gamma} * |u - v|^2)$ (default option in kernel SVMs). Use default value of gamma default (default $1/\text{numfeatures} = 1/784$ for MNIST, you dont have to worry this will be default by the LIBSVM). Vary -c option of kernel SVM between 0.01 0.1 1 10 100 and report the best accuracy. Also measure the running time with the best value of c. (you can crossvalidate if you are familiar with it)
3. Run Linear SVM's (use LIBLINEAR (Not LIBSVM) default classifier) on the dataset. Vary -c option of kernel SVM between 0.01 0.1 1 10 100 and report the best accuracy. Also measure the running time with the best value of c. (you can instead crossvalidate if you like)
4. Run Linear SVM's with k kernel features for RBF kernels:
 - Generate k random and independent normal random vectors $w_i \in \mathbb{R}^D, i = \{1, 2, \dots, k\}$, of dimation D which is also equal to the dimension of the features (784 for MNIST) (note that we will be computing inner product $w_i^T x$ for every data point. (store them). Each component of this vector is drawn from $N(0,1)$ and multiplied by $\text{gamma} = 1/784$.
 - Compute the feature transformation for every data vector x (both train and test) as

$$\phi(x) = \frac{1}{\sqrt{k}}[\cos(w_1^T x), \sin(w_1^T x), \cos(w_2^T x), \sin(w_2^T x), \dots, \cos(w_k^T x), \sin(w_k^T x)],$$

Note we have $2k$ dimensional features.

- Vary k from 100 to 4000 with increments of 500.
- Run Linear SVMs on $\phi(x)$ as new features (use LIBLINEAR (Not LIBSVM) default classifier). Vary $-c$ option of kernel SVM between 0.01 0.1 1 10 100 and report the best accuracy. You have to find best c for all k . Also measure the running time with the best value of c . (you can crossvalidate if you need)

2 Outputs

- Plot of training time as function of k for random kernel features. Add a straight line (different color) for the training time taken by KERNEL SVM (using LibSVM) STEP 2). Add another straight line for the training time taken by Linear SVM (using Liblinear on original features) STEP 3). (only report the best accuracy from among the different values of SVM parameter c)
- Plot accuracy as function of k for random kernel features. Add a straight line (different color) for the accuracy of KERNEL SVM (using LibSVM) STEP 2). Add another straight line for the training time taken by Linear SVM (using Liblinear on original features) STEP 3). (only report the best accuracy from among the different values of SVM parameter c)
- A small (at max one-page), report summarizing your observations.