# CS 534: Machine Learning
# Homework 4

(Due Nov 3rd at 11:59 PM on Canvas)

**Submission Instructions:** The homework should be submitted electronically on Canvas. The code can be done in any language that `Euler` supports.

1. **(5 + 5 = 10 pts) Kernel Methods**

   (a) Assuming that $\mathbf{x} = [x_1, x_2], \mathbf{z} = [z_1, z_2]$ (i.e., both vectors are two-dimensional) and $\beta > 0$, show that the following is a kernel:

   $$k_\beta(\mathbf{x}, \mathbf{z}) = (1 + \beta \mathbf{x} \cdot \mathbf{z})^2 - 1$$

   Do so by demonstrating a feature mapping $\phi(\mathbf{x})$ such that $k_\beta(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$.

   (b) One way to construct kernels is to build them from simpler ones. Assuming $k_1(\mathbf{x}, \mathbf{z})$ and $k_2(\mathbf{x}, \mathbf{z})$ are kernels, then one can show that so are these:

   i. (scaling) $k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) f(\mathbf{z}) k_1(\mathbf{x}, \mathbf{z})$ for any function $f(\mathbf{x}) \in \mathbb{R}$
   ii. (sum) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$
   iii. (product) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z})\ k_2(\mathbf{x}, \mathbf{z})$

   Using the above rules and the fact that $k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}$ show that the following is also a kernel:

   $$\left(1 + (\frac{\mathbf{x}}{||\mathbf{x}||_2})^\top (\frac{\mathbf{z}}{||\mathbf{z}||_2})\right)^3$$

2. **(20 + 7 + 4 + 4 = 35 pts) Almost Random Forest to Detect Thyroid Disease**

   We will be using the hyperthyroid dataset `allhyper.data`, which is a subset of the Thyroid Disease Data Set found at `http://archive.ics.uci.edu/ml/datasets/Thyroid+Disease`. The problem is to determine whether a patient referred to the clinic is hypothyroid. The original problem is not a binary classification and contains negative (not hypothyroid), hyperthyroid, and some subnormal functioning. For the purpose of the homework, you will want use hyperthyroid as the positive class and the others as the negative class. Partition the data into 70%–30% train–test split that you will use for this problem. You will implement an adaptation of the random forest for detecting thyroid disease. Instead of subsetting the features for each node of each tree in your forest, you will choose a random subspace (i.e., limit the attributes to consider as the square root of the total number of features) that the tree will be created on. This allows you to use existing decision trees without having to build your own.

   (a) Build the adaptation of the random forest using your favorite language. You will want to make sure your forest supports the following parameters:

   - nest: the number of trees in the forest
   - criterion: the split criterion – either gini or entropy
   - maxDepth: the maximum depth of each tree
   - minSamplesLeaf: the minimum number of samples per leaf node

In addition, your forest should be able to provide the out-of-bag (OOB) sample accuracy, the importance of the features, in addition to prediction.

(b) For nest = [10, 25], find the best parameters for the split criterion, maxdepth, and minsamples leaf – justify your selection with a few plots.

(c) Using your optimal parameters, how well does your version of the random forest perform on the test data? How does this compare to the OOB sample accuracy?

(d) What are the most important features from your model?

3. **(2 + 3 + 10 + 3 + 10 + 5 + 3 + 2 = 38 pts) Spam Detection via Perceptron**

We have provided a new e-mail spam dataset `spamAssassin.data`, which is a subset of the SpamAssassin Public Corpus (see `https://spamassassin.apache.org/old/publiccorpus/`). Here is a sample email that contains a URL, an email address (at the end), numbers and dollar amounts.

```
> Anyone knows how much it costs to host a web portal ?
> Well, it depends on how many visitors youre expecting. This can be anywhere
from less than 10 bucks a month to a couple of $100. You should checkout
http://www.rackspace.com/ or perhaps Amazon EC2 if youre running something big...

To unsubscribe yourself from this mailing list,
send an email to: groupnameunsubscribe@egroups.com
```

We have already implemented the following email preprocessing steps: lower-casing; removal of HTML tags; normalization of URLs, e-mail addresses, and numbers. In addition, words have been reduced to their stemmed form. For example, "discount", "discounts", "discounted" and "discounting" are all replaced with "discount". Finally, we removed all non-words and punctuation. The result of these preprocessing steps on the same email is shown below:

```
anyon know how much it cost to host a web portal well it depend on how mani visitor
your expect thi can be anywher from less than number buck a month to a coupl of
dollarnumb you should checkout httpaddr or perhap amazon ecnumb if your run someth
big to unsubscrib yourself from thi mail list send an email to emailaddr
```

(a) For this problem, you will implement the Perceptron algorithm and apply it to the problem of e-mail spam classification. You'll be comparing two different variants of the algorithm as well as the number of epochs through the data. What is your model assessment strategy? Justify your validation methodology. (You may want to read the rest of this problem before you proceed to understand what your tasks will be).

(b) Build a vocabulary list using only the e-mail training set by finding all words that occur across the training set. Ignore all words that appear in fewer than X = 30 e-mails of the e-mail training set – this is both a means of preventing overfitting and of improving scalability. For each email, transform it into a feature vector $\mathbf{x}$ where the ith entry, $x_i$, is 1 if the ith word in the vocabulary occurs in the email, and 0 otherwise.

(c) Implement the perceptron algorithm. You'll want at least two functions:

- Train the model that uses the examples provided to the function that returns the final classification vector, $\mathbf{w}$, with the number of updates (mistakes) performed, and the number of passes (epochs) through the data. Assume that the input data

provided to the function is linearly separable, so you can stop when all points are correctly classified. For the corner case of $\mathbf{w} \cdot \mathbf{x} = 0$, predict the $+1$ class.

- Test new data given the classification vector $\mathbf{w}$, and new data points. The function should return the test error.

(d) Train the perceptron using your training set. How many mistakes are made before the algorithm terminates? What is your estimated predictive error?

(e) Implement the averaged perceptron algorithm, which is the same as your current implementation but which, rather than returning the final weight vector, returns the average of all weight vectors considered during the algorithm (including examples where no mistake was made). Averaging reduces the variance between the different vectors, and is a powerful means of preventing the learning algorithm from overfitting (serving as a type of regularization).

(f) Tweak your two implementations to control the maximum number of epochs of the perceptron algorithm. Plot the training and estimated generalization error as a function of the maximum number of epochs. What is the optimal algorithm and parameter?

(g) What is your final or "optimal" algorithm? In other words, train the model with as much data as you can possibly with the optimal algorithm + hyperparameter (maximum number of epochs) values. What is the expected predictive performance of this model?

(h) Using the vocabulary list together with the parameters learned in the previous question, output the 15 words with the most positive weights. What are they? Which 15 words have the most negative weights?

4. **(2+2+3+4+4+2 = 17 points) Thyroid Prediction with Support Vector Machines**

We will be using the hyperthyroid dataset, and evaluating the model on misclassification rate, $F_1$ score, and $F_2$ score. Note that SVM with polynomial and RBF kernels can be quite expensive – you might want to use a beefier machine to do this.

(a) Why would you potentially be interested in optimizing for $F_2$ score compared to $F_1$ score in the context of detecting thyroid disease?

(b) It maybe helpful to preprocess the data for computation purposes. How do you plan to preprocess the data and why did you choose this?

(c) Build a linear SVM. How did you choose the optimal parameter(s) for your model?

(d) Build a SVM with polynomial kernel. How did you choose the optimal parameter(s) for your model?

(e) Build an RBF-kernel SVM. How did you choose the optimal parameter(s) for your model?

(f) Report the evaluation metrics for the training and test set for all of the above. How do they compare? What can you say about the models?