Assignment 2: Kernels and SVMs

Submission: Sunday April 28th
3 students per group

Prof. Fabio A. González

---

1. Train an SVM for detecting whether a word belongs to English or Spanish.

   (a) Build training and test data sets. You can use the most frequent words in `http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists`. Consider words at least 4 characters long and ignore accents.

   (b) Implement different string kernels:

      i. **Histogram cosine kernel**: calculate a bag of n-grams representation (use the `CountVectorizer` from scikit-learn) and apply the `cosine_similarity` from scikit-learn.

      ii. **Histogram intersection**: calculate a bag of n-grams representation, normalize it (the sum of the bins must be equal to 1 $\forall i, ||x_i||_1 = 1.$) and calculate the sum of the minimum for each bin of the histogram.

      iii. χ^2 **kernel**: calculate a bag of n-grams representation and apply the `chi2_kernel` from scikit-learn.

      iv. **SSK kernel**: use the code available at this repository `https://github.com/helq/python-ssk`.

   (c) Use scikit-learn to train different SVMs using precomputed kernels. Use cross validation to find appropriate regularization parameters plotting the training and validation error vs. the regularization parameter. Use a logarithmic scale for C, $\left\{2^{-15}, 2^{-14}, \ldots, 2^{10}\right\}$. Try different configurations of the parameters (in particular different n values for the n-grams).

   (d) Evaluate the performance of the SVMs in the test data set:

      i. Report the results in a table for the different evaluated configurations.

      ii. Illustrate examples of errors (English words mistaken as Spanish, Spanish words mistaken as English). Give a possible explanation for these mistakes.

      iii. Discuss the results.

2. Digit recognition model understanding.

   (a) Get the data for the MNIST data set: `https://scikit-learn.org/stable/auto_examples/linear_model/plot_sparse_logistic_regression_mnist.html`.

   (b) Normalize your features so that each one has mean 0 and standard deviation 1.

   (c) Choose two classes (e.g. 1 and 0, or 6 and 9) and train a linear SVM to discriminate between them. Find an optimal complexity parameter, C, plotting the training and test error vs. the regularization parameter. Use a logarithmic scale for C, $\left\{2^{-15}, 2^{-14}, \ldots, 2^{10}\right\}$. Discuss the results.

   (d) Extract the weights of the classification model found in (b).

   (e) Plot the discriminant function weights as follows:

i. Arrange the weights in a matrix with the same shape as the input image.

ii. Use a function such as `pcolor` `https://matplotlib.org/stable/gallery/images_contours_and_fields/pcolor_demo.html` to produce a color plot of the matrix.

iii. Use a diverging colormap that emphasizes negative and positive values `http://matplotlib.org/examples/color/colormaps_reference.html`.

iv. Discuss the results.

(f) Play with different pairs of digits and with different values for the C parameter (smaller values could produce smoother plots). Discuss the results.

3. Kernel logistic regression

We will implement a kernel version of logistic regression. The goal is to train a logistic regression model on a feature space F. Specifically, the discriminant function of the model is given by:

$$f(x) = P(C = 1|x) = \sigma(w\Phi(x)),$$

where $\Phi : X \to F$ is a mapping function associated with a kernel function $k : X \times X \to \mathbb{R}$ and σ is the logistic function.

Assume that the weight vector w is expressed as a linear combination of the training samples:

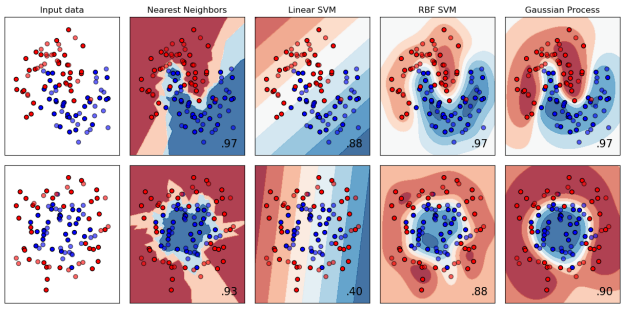
$$w = \sum_{i=1}^{\ell} \alpha_i \Phi(x_i)$$

(a) Write a expression of the discriminant function expressed in terms of the kernel and the coefficients α_i.

(b) Formulate the problem of learning the parameters of the model as an optimization problem that looks for the parameters α_i that minimize a cross entropy loss function.

(c) Write a function that receives a training data set and a kernel function and finds a vector α that minimizes the loss function using gradient descent.

(d) Test your algorithm using different kernels (linear, polynomial, Gaussian, etc.) on synthetic 2D datasets from sklearn (`https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`). Plot the decision regions and discuss the results:



The assignment must be submitted as a Jupyter notebook through the following Dropbox file request, before midnight of the deadline date. The file must be named as `ml-assign2-unalusername1-unalusername2-unalusername3.ipynb`, where `unalusername` is the user name assigned by the university (include the usernames of all the members of the group). Do not submit additional files. Make sure that the notebook renders correctly and is free of errors before submitting.